Universidade do Porto
Faculdade de Ciências
Departamento de Matemática Pura



O USO DA INTERNET NO ENSINO DA MATEMÁTICA:

O MathView e algumas alternativas

Luís Miguel dos Santos Ferreira

MAIO/2001

Universidade do Porto
Faculdade de Ciências
Departamento de Matemática Pura



O USO DA INTERNET NO ENSINO DA MATEMÁTICA:

O MathView e algumas alternativas

Tese submetida à Faculdade de Ciências da Universidade do Porto para obtenção do Grau de Mestre em Ensino da Matemática

Orientada por Fernando Jorge Soares Moreira

Realizada por Luís Miguel dos Santos Ferreira

MAIO/2001

ÍNDICE

CAPÍTULO I - INTRODUÇÃO

A SOCIEDADE DE INFORMAÇÃO 1
DO ÁBACO AO COMPUTADOR: UMA HISTÓRIA DE SUCESSO 3
OS EQUIPAMENTOS NAS ESCOLAS 6
OS PROFESSORES 8
O CASO PARTICULAR DA MATEMÁTICA 10
OS ALUNOS 12
ACERCA DESTE TRABALHO 14

CAPÍTULO II – HTML E JAVASCRIPT

INTRODUÇÃO 19
A LINGUAGEM HTML 20
O CÓDIGO HTML 21
PRINCIPAIS COMANDOS DO CÓDIGO HTML 23
CARACTERES ESPECIAIS 27
AS LISTAS 29
AS TABELAS 32
INSERIR IMAGENS 36
INTERLIGAÇÃO DE DOCUMENTOS 38
IMAGE MAPS 41
FRAMES 43
FORMS 47
A LINGUAGEM JAVASCRIPT 52
UM PRIMEIRO EXEMPLO 53
OPERADORES 54
CRIAÇÃO DE VARIÁVEIS 57
EVENTOS 59
FUNÇÕES 61
A MATEMÁTICA NO JAVASCRIPT 65
ABRIR E FECHAR JANELAS 66
CADEIAS DE CARATERES (*STRINGS*) 68
FRAMES 71
CONTROLO DAS MENSAGENS NA BARRA DE ESTADO 76
HTML DINÂMICO 78

A LINGUAGEM JAVA 81
INVOCAR *APPLETS* JAVA 81

CAPÍTULO III – O PROGRAMA MATHVIEW

DE MATHVIEW A LIVEMATH 84
AS POTENCIALIDADES DO MATHVIEW 85
AS LIMITAÇÕES DO MATHVIEW 86
OS FICHEIROS DO MATHVIEW 86
O FUNCIONAMENTO DO MATHVIEW 88
O DESEMPENHO DO MATHVIEW 92
CÁLCULOS SIMPLES 92
RESOLUÇÃO DE EQUAÇÕES DO 1º GRAU 95
RESOLUÇÃO DE SISTEMAS 96
DECOMPOSIÇÃO EM FACTORES 98
RESOLUÇÃO DE EQUAÇÕES DO 2º GRAU 100
DECLARAÇÕES 101
REGRAS DE TRANSFORMAÇÃO 104
ANÁLISE GRÁFICA 107
ESBOÇO DE GRÁFICOS 107
MAIS DO QUE UM GRÁFICO 111
ANIMAÇÃO DE GRÁFICOS 113
MARCAÇÃO DE PONTOS 115
MOVIMENTO DE UM PONTO 116
GRÁFICOS 3-D 117
TABELAS 120
MATRIZES 124
OPERAÇÕES COM MATRIZES 124
MATRIZ INVERSA E MATRIZ TRANSPOSTA 125
DETERMINANTE DE UMA MATRIZ 126
RESOLUÇÃO DE SISTEMAS UTILIZANDO MATRIZES 127
ÍNDICES DE MATRIZES 128
PRODUTO EXTERNO E PRODUTO INTERNO 128
CASE THEORIES 129
FORMAS DE DEFINIR UMA FUNÇÃO 132
AS FUNÇÕES NO ENSINO SECUNDÁRIO 137
FUNÇÕES LINEARES 137

FUNÇÕES QUADRÁTICAS 138
FUNÇÕES EXPONENCIAL E LOGARÍTMICA 142
FUNÇÕES TRIGONOMÉTRICAS 149
FUNÇÕES DEFINIDAS POR RAMOS 151
ALGUMAS CONVENÇÕES DE ESCRITA 153

CAPÍTULO IV – PÁGINAS INTERACTIVAS COM O MATHVIEW

PRELIMINARES 154
CÁLCULOS ALGÉBRICOS 159
OPERAÇÕES COM FRACÇÕES 159
OPERAÇÕES COM POLINÓMIOS 160
CÁLCULO MATRICIAL 161
RESOLUÇÃO DE EQUAÇÕES 165
EQUAÇÕES DO 1º GRAU 165
EQUAÇÕES DO 2º GRAU 165
EQUAÇÕES DO 3º GRAU 167
ESTUDO DAS FUNÇÕES 169
FUNÇÃO AFIM E AS RECTAS NÃO VERTICAIS 169
FUNÇÃO QUADRÁTICA 176
FUNÇÃO EXPONENCIAL 184
FUNÇÃO LOGARÍTMICA 188
FUNÇÕES TRIGONOMÉTRICAS 190
FUNÇÃO MÓDULO 199
FUNÇÕES DEFINIDAS POR RAMOS 204
PARIDADE DE FUNÇÕES 206
ASSÍMPTOTAS DE FUNÇÕES RACIONAIS 208
ESTUDO DE DERIVADAS 209
SOMBREAR ÁREAS 216
GEOMETRIA NO PLANO 217
COORDENADAS NO PLANO 217
DISTÂNCIA ENTRE DOIS PONTOS 218
CÓNICAS 218
PARÁBOLA 218
ELIPSE 223
HIPÉRBOLE 227
GRÁFICOS POLARES 231

EQUAÇÕES PARAMÉTRICAS NO PLANO 233
MOVIMENTO DE UM PONTO AO LONGO DE UM GRÁFICO 235
GEOMETRIA NO ESPAÇO 236
PLANOS DO TIPO X=a, Y=a e Z=a 236
SÓLIDOS DE REVOLUÇÃO 238
SUPERFÍCIES 240
EQUAÇÕES PARAMÉTRICAS NO ESPAÇO 242

CAPÍTULO V – ALTERNATIVAS AO MATHVIEW

PRELIMINARES 245
SIMBOLOGIA MATEMÁTICA 246
CARACTERES ESPECIAIS 246
VISUALIZADOR DE LATEX 246
CÁLCULOS ELEMENTARES 252
CALCULADORAS SIMPLES 252
CÁLCULO DE EXPRESSÕES 253
RAÍZES DE EQUAÇÕES DE 2º GRAU 254
ARITMÉTICA DE INTEIROS 255
CÁLCULOS COM FRACÇÕES 256
ÁLGEBRA SIMBÓLICA 258
HARTMATH 258
FORMEL MANIPULATOR 266
DERIVAÇÃO DE EXPRESSÕES 269
RESOLUÇÃO DE EQUAÇÕES DOS 1º E 2º GRAUS 271
REPRESENTAÇÕES GRÁFICAS 272
XFUNCTIONS 272
JAVAGRAPH PACKAGE 273
PLOTPANEL DO HARTMATH 274
LIVEGRAPHICS3D 276
JAVASKETCHPAD 279

BIBLIOGRAFIA 287
ENDEREÇOS ELECTRÓNICOS 290

Capítulo I – Introdução

A Sociedade de informação

É inegável que o aparecimento das novas tecnologias, em que os computadores assumem um papel de destaque, veio dar um novo rumo à sociedade em que vivemos. Sem o computador, muito do que hoje se faz não seria possível, pelo menos da mesma forma. Ou seja, o mundo em que hoje vivemos, sem o computador, seria clara e radicalmente diferente. O processo veloz de desenvolvimento e evolução das Tecnologias de Informação e Comunicação (TIC) trouxe consigo um novo período histórico da civilização humana: a sociedade de informação.

Em muitas áreas, o computador adquiriu, em poucos anos, um estatuto de ferramenta "imprescindível" e permitiu abrir outras portas ao conhecimento humano, tornando-se um importantíssimo instrumento de comunicação e de transmissão de informação. São raras as intervenções humanas que não tiram partido das enormes potencialidades que as novas tecnologias proporcionam. Os microprocessadores controlam a utilização de contas bancárias, das chamadas telefónicas, dos televisores, vídeos, automóveis, etc (Callixto, 1998) e, sem eles, áreas tão importantes para a Humanidade como é, por exemplo, a medicina, não assistiriam a avanços tão relevantes como aqueles a que têm assistido muito recentemente.

Não é, pois, mais possível voltarmos para trás. Os computadores estão aí, em força, e farão com que o mundo no futuro seja um mundo altamente tecnológico, em que as tecnologias da informação assumirão um papel de destaque.

A Internet cresce hoje a um ritmo vertiginoso. Estima-se que são já 370 milhões os seus utilizadores e que existem cerca de 1.000 milhões de páginas alojadas na rede que crescerão 10 vezes nos próximos 3 anos (in "Público",

21/07/2000), apesar de, como é evidente, informação não ser necessariamente sinónimo de conhecimento.

Assim, parece-nos claro que viveremos com computadores, não eternamente, mas pensamos poder dizer que viveremos com eles pelo menos até ao fim dos nossos dias. Mas, a acreditar na simpatia que estes meios tecnológicos e informativos gozam junto das crianças, podemos afirmar que a presença dos computadores prolongar-se-à pelas próximas gerações.

É por isso que, "quem não for capaz de utilizar e compreender minimamente os processos informáticos correrá o risco de estar tão desinserido na sociedade do futuro como um analfabeto o está na sociedade de hoje" (Ponte, 1997). A escola, já dotada de meios tecnológicos importantes, terá que se adaptar a esta nova realidade e deixar, uma vez por todas, de resistir a esta transformação como ainda, frequentemente, acontece sob pena de se desacreditar como instituição. Porque o que está em causa não é tanto o racio de número de computadores por aluno que devemos atingir mas, o mais importante é como as práticas pedagógicas, na sala de aula (ou fora dela) são influenciadas pela existência desses meios tecnológicos.

Este trabalho não pretende justificar a importância que as tecnologias de informação e comunicação assumem nos dias de hoje (apesar de se darem alguns exemplos), pois isso é reconhecido por todos. Este trabalho pretende ser uma forma de se apresentarem alguns meios que ajudem os intervenientes no processo ensino/aprendizagem (em particular os professores) a reflectirem sobre as vantagens de se alterarem algumas práticas pedagógicas.

Já em 1964, Sebastião e Silva defendia que a modernização do ensino da Matemática teria que ser feita não só quanto a programas, mas também quanto a métodos de ensino. O professor deve abandonar, tanto quanto possível, o método expositivo tradicional, em que o papel dos alunos é quase 100% passivo, e procurar diálogo com os alunos e estimulando a imaginação destes, de modo a conduzi-los, sempre que possível, à redescoberta (in Programas de Matemática – 10º, 11º e 12º anos). Ora, estas ideias mantêm-se, de facto,

actuais. O professor tem hoje à sua disposição um conjunto importante de alternativas ao método expositivo e, sem dúvida, que os meios audio-visuais e computacionais assumem um grande relevo. Aliás, as calculadoras simples no Ensino Básico e as gráficas no Ensino Secundário, assumem um papel de obrigatórias.

Assim, com esse objectivo, o uso da Internet e a elaboração das suas páginas, foram as vertentes privilegiadas ao longo deste trabalho. Na realidade, a Internet é hoje, em larga escala, utilizada pelos nossos alunos, muitas vezes até mais do que os próprios professores. Este aspecto poderá constituir uma importante forma de abordar determinados conteúdos programáticos muitas vezes, como no caso da Matemática, nada bem vistos pelos alunos.

Do Ábaco ao Computador: uma história de sucesso

Já é de longa data, a necessidade manifestada pelo Homem em automatizar e facilitar tarefas que se têm mostrado essenciais e decisivas para a sua evolução. O mais antigo instrumento de computação, conhecido e usado há mais de 2000 anos, é o ábaco, que permitia efectuar com facilidade adições e subtracções e, com alguma habilidade, multiplicações e divisões.

Desde essa altura que aparelhos ou métodos de cálculo foram ajudando o Homem a conhecer melhor o "seu" mundo. As réguas de cálculo e as primeiras calculadoras mecânicas surgidas no século XVII são exemplo disso mesmo.

Já com o vapor como fonte de energia privilegiada, por volta de 1833 o inglês Charles Babbage apresentava o seu "Analytical Engine", um computador mecânico automático e a vapor, totalmente controlado por um programa, que operava com números (ou palavras) de 50 dígitos decimais e cuja capacidade de armazenamento (memória) era já de 1000 números. Os cartões perfurados tinham entretanto nascido.

A exigência de levar por diante cálculos mais complexos em projectos de maior envergadura, como construção de pontes, vias férreas, navios a vapor, moinhos têxteis, entre outros, pressionaram o aparecimento de máquinas com maiores capacidades.

No final dos anos 30, Howard Aiken, em colaboração com engenheiros da IBM - International Business Machines, constróem um grande computador digital automático, fazendo aparecer os componentes electrónicos em substituição dos mecânicos. Tal máquina, chamada HARVARD MARK I, já permitia a elaboração de cálculos longos sem a intervenção humana, apesar de, por exemplo, cada multiplicação demorar entre 3 e 5 segundos.

A 2ª Guerra Mundial e a necessidade de cálculo do exército, motivam em 1945 o aparecimento de um computador electrónico 1000 vezes mais rápido do que o MARK I, o ENIAC (Electrical Numerical Integrator and Calculator). O avanço em termos de potencialidades não evitavam, contudo, o seu tamanho de 167,3 m^2 de espaço no solo e os 180.000 watts de energia consumida.

O aparecimento do transistor em 1948 foi decisivo para a diminuição do tamanho do computador, fazendo com que o UNIVAC I (Universal Automatic Computer) fosse o primeiro computador a entrar no circuito comercial apesar de, por ser ainda bastante caro, o seu uso se restringisse a grandes centros computacionais.

Vulgariza-se a utilização dos chips (circuitos integrados) em 1970 e a Intel apresenta o primeiro microprocessador, num final de década marcado pela disputa entre a Apple e a IBM no campo dos computadores pessoais.

Surge o sistema operativo MS-DOS lançado pela Microsoft de Bill Gates mas que, já em meados da década de 80, é substituído pelo WINDOWS em resposta ao sistema da Apple que utilizava os icons accionados com o rato no acesso à informação.

Estavam então criadas as condições para que os computadores entrassem nos lares das pessoas, tal era o reduzido tamanho da máquina, baixos custos de aquisição e manutenção, facilidade de manuseamento, aliado ao facto de, entretanto, ter já surgido a primeira linguagem de programação dirigida a não especialistas: o BASIC (Basic Instructions Code).

Mas, se dúvidas existiam quanto à penetração dos computadores nos lares das famílias da sociedade actual, deixaram de existir quando se dá o aparecimento da Internet, a partir de um projecto criado em 1969 pelo Departamento de Defesa do Governo dos EUA, designado por ARPANET (Advanced Research Projects Administration NETwork). Tal projecto "consistia no desenvolvimento duma rede segura para a transmissão de informação entre instituições Norte-Americanas que trabalhavam no desenvolvimento e fornecimento de recursos militares" (Lemos, 1998). Desde essa altura que a Internet não tem parado de crescer o que, na realidade, acabou por trazer ainda um novo fôlego à utilização dos computadores apesar de, actualmente, ser já possível o acesso à rede por meios que não apenas o computador (por exemplo, o telemóvel). A Internet e o seu crescimento tem constituído, também ela, um verdadeiro sucesso. Estas novas Tecnologias da Informação e Comunicação revolucionaram, de facto, o mundo actual de uma forma irreversível.

Em Portugal, a realidade é um pouco diferente da média europeia. Enquanto que na Europa, a média de número de PC's por 100 habitantes situa-se nos 18, no nosso País, temos uma taxa de 7 computadores por 100 habitantes. (Fonte: OCDE; Communications Outlook 1999). Por outro lado, segundo dados do barómetro Marktest, entre Dezembro de 1997 e Março de 1999, a percentagem da população portuguesa que tem computador em casa, subiu dos 25% para os 32%. Ainda no mesmo período, a taxa dos que têm acesso a computador subiu de 45,4% para os 50,9%. Para esta última subida, terá certamente contribuído o esforço que as empresas, a administração pública e as escolas desenvolveram (e continuam a desenvolver) no sentido de se dotarem de infra-estruturas e equipamentos nessas áreas.

Os Equipamentos nas Escolas

No caso particular das escolas, na verdade também elas passaram a usufruir, em larga escala, da utilização destes poderosíssimos equipamentos tecnológicos. Aliás, segundo Serge e Riché-Magnier (1998), a história das tecnologias educativas encontra-se num momento de mudança. A difusão dos computadores nos estabelecimentos escolares, induzida pela informatização da sociedade, é irreversível.

Esta entrada, por "razões pedagógicas ou pressões exteriores à escola (económicas/comerciais, sociais e políticas)" (Pais,1999), trouxe consigo muitas mudanças e alterações no seio da escola. No entanto, ainda hoje, "antes de ser um instrumento de ensino ou de aprendizagem, o computador é, em primeiro lugar, quer para os chefes dos estabelecimentos de ensino quer para os professores, uma ferramenta de gestão da vida escolar".

Na realidade, apesar de termos alunos motivadíssimos para a utilização das TIC no processo de ensino/aprendizagem em que estão envolvidos, apesar de termos orientações programáticas nesse sentido, apesar de termos escolas mais ou menos bem equipadas e os professores cada vez mais utilizadores de meios tecnológicos, ainda há um longo caminho a percorrer e grandes obstáculos a superar.

Ao longo deste processo em que as TIC "invadiram" a nossa sociedade, várias têm sido as medidas tomadas pelas mais variadas instâncias governamentais, para levar a cabo essa necessária e pacífica "computarização" dos estabelecimentos escolares.

O Despacho nº 206/ME/85, de 15 de Novembro, que instituiu em Portugal o Projecto MINERVA e que tinha como um dos objectivos a "inclusão do ensino das Tecnologias de Informação nos planos curriculares de ensino não superior", trouxe consigo uma forte campanha de apetrechamento das escolas de meios informáticos, bem como, a formação dos profissionais visando a sua plena utilização (nomeadamente os professores).

O desenvolvimento deste programa até 1994 levou a que, numa fase da sua avaliação, se concluísse e recomendasse a necessidade, "entendendo as tecnologias como meios facilitadores e potenciadores de processos de ensino e aprendizagem", de se concretizar uma estratégia integrada de introdução das TIC na Educação, com incidência científica e pedagógica. Estava, certamente a falar-se das escolas!! Outros projectos surgiram como o FORJA, IVA, EDUTIC e que, só em 1997/1998, envolveram investimentos na ordem dos 3 milhões de contos, segundo dados do Departamento de Avaliação, Prospectiva e Planeamento (DAPP), do Ministério de Educação.

Ao longo de todos estes anos, as escolas têm, portanto, arranjado formas de se equiparem e, sem dúvida que o aparecimento da Internet, veio ajudar nessa tarefa, através do surgimento de programas específicos nessa matéria. Por exemplo, em 1997 é lançado pelo Ministério da Ciência e Tecnologia (MCT) o programa "Internet na Escola" com a intenção de, já no ano seguinte, todas as escolas do 5º ao 12º ano terem acesso à Internet (Freitas, 1999). Desse modo, todas as bibliotecas das referidas escolas, públicas e privadas (cerca de 1600), foram ligadas em rede pelo MCT, criando-se, assim, a "Rede Ciências Tecnológicas e Sociedade" (RCTS). Para além da infra-estrutura de equipamento e suportes lógicos, foram instituídos sub-domínios de Internet através da disponibilização de contas e-mail, espaços de publicação de páginas World Wide Web (WWW), conversação Internet Relay Chat (IRC) e vídeo difusão.

Na realidade, segundo ainda dados do DAPP relativos às "conexões existentes nas escolas em 1997/1998 confirmam que mais de 90% das escolas (do 5º ao 12º ano) possuem pelo menos um acesso à Internet" e, como tal, qualquer aluno que se dirija à biblioteca tem a possibilidade de encontrar um equipamento com a possibilidade de acesso à Internet.

Porém, o alargamento desta rede também está pensado para as escolas do 1º ciclo. No discurso de abertura do debate sobre Política Geral, na Assembleia da República em 17 de Maio de 2000 pelo Chefe do Governo Português, Eng.

António Guterres, estava latente o objectivo de, em 2001, estar "concluída a ligação à Internet das escolas do 1º ciclo do Ensino Básico (EB) e, em 2003, a rede de postos públicos de acesso à Internet em todas as freguesias".

Em 1996 surge o Programa Nónio – Séc. XXI, que suscita o aparecimento de projectos na área das novas tecnologias, através da cooperação entre instituições do ensino superior e instituições de ensino não superior. No âmbito deste mesmo programa, desde 1999 que todos os professores têm acesso a caixa de correio electrónico, através do chamado PROFMAIL.

Por tudo isto, o acesso por parte de toda a comunidade escolar, em particular dos alunos, às novas tecnologias, tem vindo, cada vez mais a ser alargado. A maioria das escolas EB2/3 e Secundárias, a nível nacional possui um computador para um número de alunos que oscila entre 20 e 50 e a taxa de crescimento a acessos diários à Internet nas escolas ronda os 25% (DAPP, 1999).

Os Professores

"Sabemos que nenhuma vontade alheia, nenhuma legislação tem o poder, só por si, de inovar práticas educativas dos outros, e a mudança de prática exige, para além da vontade dos seus protagonistas, processos, por vezes bem difíceis e lentos, de construção" (Benavente, 1996, pág. 50). Na realidade, apesar dos números actuais confirmarem, como se viu, que também as escolas aderiram às TIC, não é contudo líquido, que os processos de ensino – aprendizagem tenham sofrido grandes alterações, apesar dos intervenientes neste processo já serem, por vezes, utilizadores assíduos desses meios tecnológicos.

Contudo, é importante ainda salientar que para o melhoramento do processo ensino – aprendizagem, para além da enorme contribuição das TIC, é necessário que os professores estejam abertos à mudança, e que lhes seja dada a formação que lhes permita utilizar as melhores estratégias na

integração dessas novas tecnologias nas actividades lectivas (Novais,1997). De facto, "a utilização de novas tecnologias ou materiais manipuláveis coloca igualmente ao professor novos desafios" (APM, 1997) e, ao mesmo tempo, novas oportunidades de exploração dos temas com os seus alunos.

Num trabalho de investigação levado a cabo por João Pedro da Ponte da Faculdade de Ciências de Lisboa, chegou-se à conclusão que, as razões que levam os professores a participar em programas de formação (como foi o Projecto Minerva) "são a curiosidade acerca dos possíveis usos educativos do computador e, em alguns casos, um desejo muito forte de o usar na sala de aula", em que o computador é visto como "podendo constituir um importante factor de motivação dos alunos, esperando que possa contribuir para uma nova dinâmica no processo de ensino-aprendizagem" (Ponte, 1994). De facto, já não existem dúvidas no reconhecimento que o "poder dos computadores para estimular, enriquecer e activar a aprendizagem parece infindável" (Kahn, 1991).

De facto, o uso dos computadores começa a ser relativamente frequente nas rotinas de trabalho dos professores, pois podem ser bastante úteis, ao nível da concepção e produção de material escrito (testes, fichas de trabalho, bem como todo o tipo de documentos gráficos) exigindo competências cada vez menos complexas (Ramos,1997). Para isto, para além da iniciativa pessoal de muitos professores, terá também certamente contribuído o plano de formação contínua dos professores, na área de informática, iniciado com o Projecto MINERVA, para além do facto de muitas das instituições do Ensino Superior que se dedicam à formação de professores, terem já incluído nos seus planos curriculares conteúdos na área das novas tecnologias. Daí que, segundo Cândido Freitas (1997), sejam os professores mais velhos os que se sentem menos confortáveis com o uso das novas tecnologias, precisamente porque não foram treinados para elas. De qualquer modo, a formação contínua dos professores, exigida para efeitos de progressão na carreira docente, tem passado, em diversos casos, pela formação na área da informática. Essa formação tem sido feita pelos professores na modalidade de Acções de Formação, Oficinas ou Círculos de Estudos e terão, certamente, continuidade, uma vez que está já assegurado o financiamento desse tipo de acções por

parte do PRODEP III (Programa de Desenvolvimento Educativo para Portugal), responsável pela gestão de verbas provenientes do III Quadro Comunitário de Apoio.

O caso particular da Matemática

Mas, mesmos os professores com alguma formação na área das TIC, mostram-se bastante renitentes à mudança das práticas lectivas e do processo de ensino – aprendizagem: factores como o número de alunos por turma, os programas curriculares, falta de meios em algumas escolas e a crença por parte dos professores que "o computador desestabiliza a sua autoridade" (Pais, 1999), fazem com que o dia-a-dia lectivo dos novos alunos se mantenha, na esmagadora maioria, inalterado. Segundo dados recolhidos do Diagnóstico e Recomendações para o Ensino e aprendizagem da Matemática – Matemática 2001, "a utilização de computadores (...) tem uma frequência muito pouco significativa que é quase uniforme nos diversos ciclos, onde a grande maioria dos professores (88%) declara nunca ou raramente utilizar" (APM, 1998).

A utilização da Internet na sala de aula, por parte dos professores no processo ensino-aprendizagem, também é ainda pouco significativa. No âmbito deste trabalho, foi realizado um inquérito a 313 professores de Matemática dos 2º e 3º ciclos do EB e ES de diversos pontos do país, ao qual 66,8% responderam que já tinham utilizado alguma vez a Internet mas nunca na sala de aula. Apenas 8,9% dos inquiridos declaram já terem utilizado a Internet na sala de aula, na abordagem de algum tema específico dos programas da disciplina de Matemática. De salientar ainda os 24,3% de professores de Matemática terem declarado que nunca tinham utilizado a Internet.

Em resumo, apesar dos intervenientes no processo de ensino aprendizagem (em particular, professores e alunos), na sua grande maioria já utilizarem (ou terem acesso) a computadores e já acederem à Internet, a realidade é que essas importantes ferramentas ainda não influenciaram, de uma forma significativa, a forma como se ensina e aprende nas nossas escolas. Esta

situação fica bem retratada na história contada por Seymour Papert (1996), uma das personalidades de maior relevância no mundo dos computadores na educação e inventor da linguagem Logo, no seu livro "A Família em Rede":

"Imagine um grupo de viajantes no tempo, entre os quais um grupo de médicos cirurgiões e um grupo de professores, que chegassem do século passado, para ver como as coisas se passam nos nossos dias. Imagine o espanto dos cirurgiões quando entrassem numa sala de operações de um hospital moderno! Os cirurgiões do século XIX não conseguiriam perceber o que aqueles fulanos, vestidos de maneira tão esquisita, estavam a fazer. Embora compreendendo que estava a decorrer uma operação cirúrgica qualquer, muito provavelmente seriam incapazes de identificá-la. Os rituais de anti-sepsia, a aplicação de anestésicos, os bips dos aparelhos electrónicos e até a intensa luminosidade ser-lhe-iam completamente desconhecidos. Certamente sentir-se-iam incapazes de dar uma ajuda.

Quão diferente seria, no entanto, a reacção dos professores viajantes no tempo ao entrarem numa moderna sala de aulas! Talvez se sentissem intrigados pela existência de alguns objectos mais estranhos, pelos estilos de vestuário e de corte de cabelo, mas perceberiam perfeitamente a maior parte do que se estava a passar e poderiam mesmo, num abrir e fechar de olhos, tomar conta da turma. Naturalmente discutiriam entre si se as mudanças observadas foram para melhor ou para pior".

Os alunos

Como já se referiu anteriormente, estamos perante um geração de jovens com uma grande predisposição para o uso das tecnologias no seu dia-a-dia, nas suas mais diversas actividades, tendo a explosão da Internet vindo reforçar essa realidade. A Internet, de facto, tornou-se uma forma importante dos jovens passarem os seus dias (e noites). Existem hoje os chamados cyber-cafés, essencialmente frequentados por jovens, onde "navegar na Net" é a actividade principal e, onde mesmo aqueles jovens que não possuem acesso à Internet em suas casas, poderão fazê-lo nesses espaço. Aliás, aceder à Web, por si só, já é uma experiência educativa, mesmo tratando-se, muitas vezes, de uma "aprendizagem acidental" (Bentes Paulo, 1999).

Estamos perante a geração a que os colaboradores da Microsoft chamam "e-generation". De facto, estão, neste momento, na escola os primeiros alunos que já nasceram nesta era da informática, o que, inevitavelmente, os põe numa situação de vantagem relativamente àqueles que viram, há poucos anos atrás, esta autêntica explosão tecnológica.

Por outro lado, os professores de Matemática, em particular, enfrentam uma dura realidade, de terem uma sociedade que encara essa disciplina como algo de muito difícil. Os alunos entram na escola a não gostarem da disciplina, com um preconceito que a matemática é para deixar de lado. De facto, estamos perante um fenómeno também de cariz social. Obviamente que a falta de motivação para o estudo da Matemática arrasta consigo o inevitável insucesso. Ora, a introdução das tecnologias e, em particular da Internet, no Ensino da Matemática, poderá ser uma forma interessante de motivar os alunos à participação. "A aversão ao trabalho escolar vem principalmente do facto de o considerarem [as crianças] aborrecido, exactamente o oposto de o acharem demasiado difícil" (Papert, 1996).

Ora, a utilização das tecnologias na sala de aula, na abordagem de determinados conteúdos programáticos, poderá funcionar como uma forte motivação dos alunos. Já por alturas das primeiras experiências da utilização

do computador no Ensino (através da linguagem BASIC e LOGO) verificou-se que "na maior parte dos casos, os alunos envolvidos nestas actividades melhoraram a sua relação com a disciplina de Matemática e criaram uma maior predisposição para a aprendizagem dentro da sala de aula, mesmo quando eram considerados problemáticos" (Ponte e Canavarro, 1997).

Numa tentativa de acompanharem esta realidade, também os programas de Matemática têm vindo a sofrer as necessárias (e desejáveis) adaptações, sugerindo-se em muitas situações a utilização das tecnologias. Aliás, refere-se mesmo nas Orientações Metodológicas dos Programas de Matemática do Ensino Secundário, que "a utilização obrigatória de tecnologias que, para além de ferramenta, é fonte de actividade, de investigação e de aprendizagem, pretende preparar os alunos para uma sociedade em que os meios informáticos terão um papel considerável na resolução de problemas de índole científica". Relativamente aos recursos a utilizar, é assumido no mesmo documento que é indispensável o uso de "um computador ligado a um *data-show* para demonstração, simulação ou trabalho na sala de aula com todos os alunos ao mesmo tempo".

Citado nos programas de Matemática do Ensino secundário, Sebastião e Silva já preconizava que o uso de tecnologias facilita ainda uma participação activa do aluno na sua aprendizagem, quando escrevia no "Guia para a utilização do Compêndio de Matemática" que "haveria muitíssimo a lucrar que o ensino fosse tanto quanto possível laboratorial, isto é, baseado no uso de computadores, existentes nas próprias escolas ou fora destas, em laboratórios de cálculo". Também no relatório Matemática 2001 (APM, 1998) se sugere que "a aprendizagem dos alunos pode reforçar-se com a utilização de materiais que proporcionem o seu forte envolvimento, nomeadamente, materiais manipuláveis, calculadoras e computadores".

Acerca deste Trabalho

Com este trabalho pretende-se, acima de tudo, que os professores do Grupo de Matemática obtenham informação suficiente para que possam, eles próprios, produzir os seus ambientes de trabalho.

A criação de determinados cenários de ensino/aprendizagem, através da utilização das TIC e, em particular da Internet, poderá ser, na realidade, uma forma interessante de motivar os alunos para o estudo da Matemática. Este trabalho visa a exploração desta metodologia: o aluno é conduzido pelos diversos tópicos em estudo através de um interface interactivo que lhe é familiar, fazendo da Internet uma arma poderosíssima à disposição dos professores para desenvolverem nos seus alunos, por um lado, o gosto pela Matemática e, por outro, o espírito crítico e a construção do seu próprio conhecimento. Isto com o rigor científico que se exige no ensino da Matemática, como não poderia deixar de ser. Segundo Seymour Papert (1996), "a familiarização com a Internet pode abrir imensas oportunidades novas para as crianças assumirem responsabilidade" (pág. 37).

É frequente ouvirmos os professores criticarem o software disponível no mercado, por razões parecidas às que são apontadas na crítica aos manuais escolares. Segundo ainda Seymour Papert (1996), "é estranho que o software educativo continue a seguir maioritariamente os velhos padrões, reforçando o engano em vez de o tornar desnecessário" (pág.100).

Assim, porque não o professor produzir o seu próprio software?

O professor pode criar e produzir materiais de suporte informático, sem que, para tal, necessite dominar linguagens de programação. Ora, na realidade, apesar da boa vontade, os especialistas em tecnologia informática não possuem uma visão real e concreta do curriculum, dos contextos, das características dos alunos, das dificuldades de aprendizagem e, muito pouco ou nada, podem fazer para resolver problemas de natureza educativa (Ramos,1997).

Como iremos ver, uma página desenhada para a Internet poderá funcionar perfeitamente como uma poderosíssima aplicação de apoio ao ensino de vários temas.

No **Capítulo II** é introduzida a Linguagem HTML (HyperText Markup Language), a linguagem que regula a apresentação das páginas que circulam na WWW. São exibidos alguns exemplos do resultado obtido depois de se utilizarem determinadas instruções específicas, apesar de, nesta versão do trabalho em papel, não seja possível, como é óbvio, a sua visualização interactiva. É certo que, com a existência de aplicações do tipo MS FrontPage, torna-se "quase" desnecessário conhecermos código HTML. Contudo, foi muitas vezes imprescindível efectuarem-se alguns pequenos (por vezes grandes) ajustes que, sem o conhecimento das instruções básicas de HTML, tornar-se-iam mais difíceis de concretizar. Aliás, para que o leitor produza as suas próprias páginas, poderá ter interesse em "visitar" o código inerente a estas páginas e fazer as suas próprias modificações e adaptações. Seria, pois, de extrema importância que o leitor se pudesse situar e orientar no meio do código HTML. Em suma, o Capítulo II ajudará a que se percebam algumas das acções realizadas nos capítulos seguintes.

Na versão "on-line" deste trabalho, também disponível em CD-ROM, as imagens que aparecem no Capítulo II, darão origem à abertura do respectivo documento, cujo código poderá ser visualizado através da selecção da função "Ver HTML" disponível no browser (nome que designa os programas que têm por finalidade interpretar ficheiros escritos em linguagem HTML). Ora, os dois browsers mais conhecidos são o Internet Explorer da Microsoft e o Netscape Navigator da Netscape, sendo também os mais usados pelos utilizadores que pertencem ao chamado "grande público". Aliás, têm sido estes dois fabricantes que têm impulsionado, nos últimos anos, as grandes evoluções do HTML. A "rivalidade" comercial a que hoje se assiste entre essas duas poderosas empresas, têm levado a que, ao contrário do que seria desejável, os dois produtos apresentem determinadas especificidades incompatíveis no produto concorrente, ou seja, algumas instruções só são correctamente interpretadas

num deles. Ora, este facto obriga a que os criadores de páginas Web optimizem os seus trabalhos para um dos browsers o que, só por si, constitui uma infeliz e indesejável limitação, que contraria, de certa forma, o cariz de universalidade que se deseja para a Internet. No entanto, as duas empresas, conscientes disso mesmo, quando lançam no mercado uma versão mais recente do respectivo produto, tentam fazê-lo compatibilizando, na medida dos possíveis, a sua nova versão com o que conhecem da versão concorrente. Por outro lado, para cativarem cada vez mais clientes, fazem-no apresentando também algumas inovações que o concorrente, obviamente, não conhece. Seria, pois, de toda a conveniência que, principalmente as duas empresas referidas (embora haja outras que também têm os seus próprios produtos) cheguem a pontos de entendimento sobre estas matérias uma vez que, não será por motivos meramente mercantilistas que se deixará de evoluir na era da Internet. Este trabalho foi optimizado para a utilização do Internet Explorer 5.0.

Ainda neste segundo capítulo, serão apresentadas algumas instruções básicas da Linguagem Javascript, cujo conteúdo pode ser mergulhado "internamente" no próprio código HTML. Além do JavaScript, também será explicado como podemos aproveitar as aplicações criadas com a Linguagem de programação JAVA para melhorar a interactividade dos documentos propostos neste trabalho.

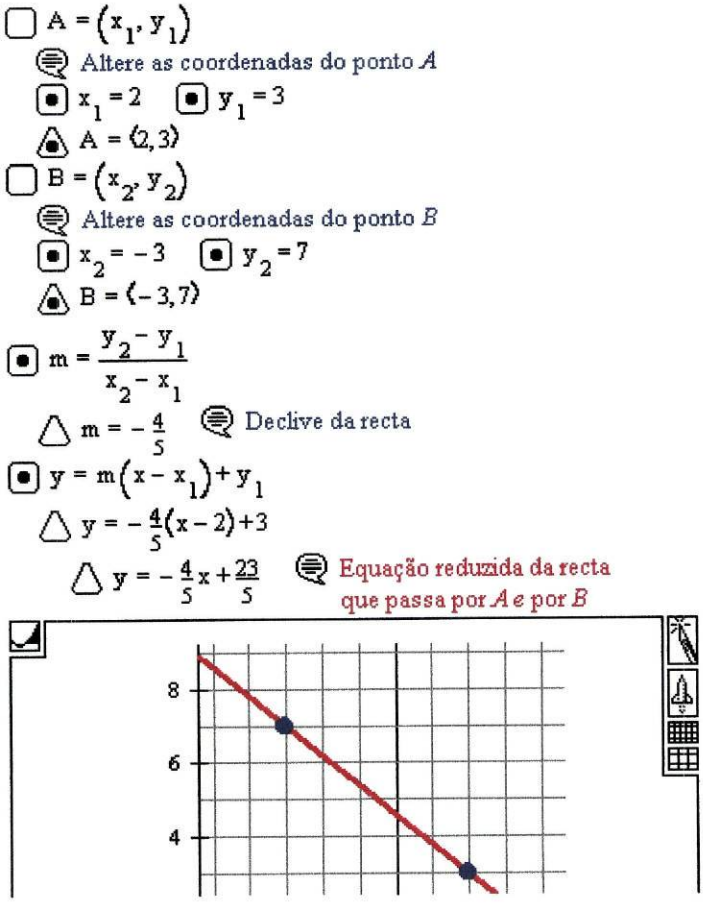
É no **Capítulo III** que é explicado o funcionamento do programa MathView, utilizado no Capítulo IV. Note-se que, para a visualização on-line do Capítulo IV e, nomeadamente, para se procederem às manipulações dos exemplos que são apresentados (como alteração dos valores que são carregados por defeito), não é necessário que se proceda à instalação do programa MathView no computador que está a ser utilizado. Será apenas necessária a instalação de um *plugin* (nome que se dá ao programa que permite que objectos mergulhados nas páginas Web sejam reproduzidos em linha - na própria página – sem a necessidade de abrir uma aplicação externa), para que o browser consiga ler os respectivos ficheiros oriundos do MathView. No entanto, para a criação de tais ficheiros (do tipo "*.thp") que são carregados em páginas HTML, torna-se necessário, de facto, o programa referido. Este terceiro

capítulo pretende dotar o utilizador do programa de conhecimentos básicos para a introdução e manipulação dos dados, bem como dotar os utilizadores das páginas onde constem ficheiros desse tipo, de conhecimentos que permitam alterar os dados que são apresentados. Aliás, este constitui um dos pontos fortes do programa. Um utilizador que tenha na sua máquina o *plugin* instalado e esteja a visualizar uma página que contém um destes ficheiros, poderá alterar os dados iniciais que são carregados por defeito e visualizar, de imediato, os novos resultados, quer sob o ponto de vista algébrico, quer sob o ponto de vista gráfico.

De salientar que, nas indicações metodológicas dos Programas de Matemática para o Ensino Secundário, mais propriamente do 10º ano de escolaridade, é dito que "os alunos devem sempre traçar um número apreciável de funções tanto manualmente em papel quadriculado ou em papel milimétrico, como usando calculadora gráfica ou computador". Para este fim, o recurso ao programa MathView, pode ser uma grande ajuda, nomeadamente quando se trata de esboçar famílias de funções. O programa poderá ser ainda rentabilizado, através da criação de modelos capazes de, por um lado, mostrar ao aluno como se chega a determinados resultados ou, por outro, facilitar, simplesmente, a elaboração de cálculos de certa forma rotineiros.

Refira-se ainda que o MathView neste momento já não se encontra disponível no mercado, uma vez que foi substituído pelo LiveMath desde meados de 1999, altura em que a empresa WebPrimitives LLC adquiriu a versão anterior do programa. No seu site http://www.livemath.com, é disponibilizado o download gratuito do *plugin* e, a título de shareware, uma licença de utilização do LiveMath por um período de 30 dias. Parece-nos que a nova política da empresa detentora dos direitos comerciais deste produto, por sobreviver apenas do sucesso do LiveMath (ao contrário do que acontecia no passado em que era pertença da mesma empresa que criou o *Maple*), poderá, de certa forma, prejudicar o fácil acesso dos utilizadores ao produto e o seu desejável desenvolvimento. Contudo, a nova versão do produto não trouxe alterações substanciais à versão inicial.

Os **Capítulos IV** e **V** contêm, de facto, o essencial deste trabalho. Nestes dois capítulos, são exemplificadas todas as potencialidades do MathView e da Linguagem HTML/JavaScript para o campo do Ensino. Os modelos tratados pretendem, prioritariamente, exibir conteúdos programáticos dos Ensino Básico e Secundário. No Capítulo IV usa-se, em concreto, o *plugin* do MathView na construção de páginas Web, interactivas e manipuláveis, como no exemplo que de seguida se apresenta.



O Capítulo V dedica-se à construção de software para o Ensino em que a interactividade permitida pelo MathView é obtida recorrendo a *scripts* escritos em JavaScript e a aplicações escritas em Java e disponíveis na Internet em regime de "freeware". Pretende-se, assim, sugerir alternativas ao uso do MathView para a construção do mesmo tipo de módulos obtidos no Capítulo IV.

Capítulo II – HTML e JavaScript

Introdução

Neste capítulo, apresentam-se os comandos mais básicos da linguagem HTML, que permitem a elaboração de páginas para a WEB. Com o intuito de se melhorarem as capacidades interactivas dessas mesmas páginas, faremos uma ligeira abordagem dos comandos essenciais da linguagem que constitui uma extensão do HTML - JavaScript. Finalmente, referem-se as instruções necessárias à invocação de aplicações criadas em Linguagem JAVA (as chamadas *applets*), uma vez que, no Capítulo V, foram amplamente utilizadas.

Utilizaram-se ao longo deste texto expressões como "clique no botão direito do rato" ou "clique sobre o link" por serem conhecidas pela generalidade dos utilizadores de computadores mas, como é óbvio, estamos perante um "abuso de linguagem".

Para a explicação de determinadas instruções não se fizeram as traduções de algumas palavras em virtude de serem bastante usadas nas suas versões originais. A título de exemplo, utilizar-se-ão termos como "frame", "form", "tag", entre outras.

Qualquer utilizador da WWW terá, em condições normais, acesso ao código HTML que está por trás de qualquer documento visualizado. Isto permite, inclusivamente, a reprodução desse mesmo código para a sua utilização na construção da nossa própria página e dos nossos próprios documentos. Este raciocínio pode também ser utilizado na obtenção de imagens, cores, partes de texto, sons, entre outros. Apesar do facto de haver protecção legal dos direitos de autor do material produzido e disponibilizado na Internet, a realidade é que existe pouca ou nenhuma possibilidade de fazer valer este tipo de direitos nos tribunais, porque, como é hábito, a tecnologia avança sempre depressa demais para a legislação produzida nesta área. Ora, isto obriga a que as referências

"Copyright" nas páginas que forem sendo produzidas, tenham resultados práticos duvidosos (Coelho, 1998).

A linguagem HTML

A sigla HTML representa a expressão ***HyperText Markup Language*** e constitui a linguagem utilizada na criação de homepages. Permite implementar ligações hipertexto, isto é, permite que variadíssimos documentos de texto, imagens, sons e vídeo, estejam interligados com outros documentos ou objectos do mesmo ou de diferentes tipos, através dos chamados "links". Esta característica de hipertexto, faz com que a consulta do documento que está a ser visualizado possa ser feita de uma forma não sequencial, ou seja, o utilizador pode escolher a ordem com que fará a consulta do documento, tal como faz com um livro em papel, embora neste caso, de uma forma menos eficaz. Para além da possibilidade destas interligações entre diferentes partes do mesmo documento, as interligações também poderão ocorrer para documentos diferentes. Aliás, a grande vantagem da Internet é essa (inter)ligação existente, directa ou indirectamente, entre os milhões de documentos disponibilizados na rede. E é precisamente por essa interligação que surgiu a expressão **World Wide Web** (**WWW**).

A Internet é, pois, um sistema cliente/servidor, em que os clientes são os computadores ligados à Internet e que utilizam um browser para requisitar páginas HTML armazenadas em servidores WEB e estes, por sua vez, são computadores com software especializado para a gestão de documentos HTML e que enviam, para o computador requisitante, os documentos no formato HTML, aos quais o cliente pretende aceder.

Apesar de todas as páginas disponíveis na WWW conterem instruções escritas em HTML e de ser esta a linguagem de publicação das mesmas, há também transporte de dados entre cliente e servidor, em formatos diferentes do HTML. A título de exemplo, temos os formatos correspondentes a imagens (conhecidos pelas siglas GIF, TIF, JPEG, PNG e correspondentes a diferentes

tipos de compressão dos dados da imagem), vídeos (MPEG, MOV, AVI), sons (WAV, AU), texto (DOC, PS, TXT), entre outros.

O aumento significativo da capacidade interactiva e a riqueza do conteúdo das páginas de informação, pode ser conseguido através da utilização da linguagem de programação JAVA. As applets JAVA são, pois, programas que são referenciados nas respectivas páginas que os contêm, o que faz com que apenas sejam carregadas após o programa de navegação ter acedido à página que o utiliza. Existe também outro processo de melhorar as potencialidades interactivas das páginas Web, minimizando a necessidade de contacto entre o computador do utilizador e computador do servidor, diminuindo, portanto, o tempo de acesso à informação. Esse processo é a utilização de códigos em JavaScript, que permitem injectar lógica nas páginas criadas em HTML. Esses códigos estão "embebidos" no documento em HTML da própria página e aparecem sob a forma de texto, ao contrário das *applets* Java que são obtidos através do carregamento de um ficheiro específico. Este aspecto também será abordado neste capítulo.

O código HTML

Os documentos HTML têm um formato de texto (ASCII – American Standard Code for Interchange Information) e podem ser criados usando qualquer editor ou processador de texto. Para especificar a aparência da informação que se deseja mostrar a quem acede a determinada página através de um browser, parte-se da base de um documento ASCII com os conteúdos da informação a mostrar, que, posteriormente, dá formato aos mesmos usando códigos ou marcas especiais que regulam a sua aparência e a forma como irão aparecer no ecrã de quem os visita. Algumas dessas marcas ou instruções são comuns a todos os documentos e outras que, na sua generalidade, são reconhecidas pelos diferentes browsers, são utilizadas apenas quando se pretende obter determinado resultado ou formatação. De salientar que uma das características que é mais importante no HTML é a total independência das páginas, geradas em relação ao ambiente em que irão ser visualizadas (Coelho, 1998). Neste

capítulo, irá fazer-se referência a essas principais marcas, instruções e formatações, uma vez que foi importantíssimo para nós, no decorrer deste trabalho, perceber qual o efeito no ecrã após a utilização de cada uma delas, nomeadamente pelos pequenos (e, por vezes, grandes) acertos que foram necessários levar a cabo. Aliás, para o que será feito nos Capítulos IV e V, será necessário, muitas vezes, manipular o próprio código, por forma a que o resultado final fosse apresentado da forma que ocorreu.

É evidente que, um utilizador da Internet, para elaborar a sua própria página, não necessita de conhecer em profundidade a linguagem HTML. Isto porque, hoje em dia, existem já programas que permitem a realização dos documentos de uma forma directa, isto é, o que introduzimos e visualizamos no ecrã, é já o resultado final. Tal é possível, uma vez que o programa que está a ser utilizado permite, de uma forma automática, a conversão daquilo que introduzimos no código HTML. Tais editores de HTML seguem o princípio conhecido pela sigla WYSIWYG (What you see is what you get). A título de exemplo desse tipo de software, refira-se o Microsoft Frontpage da Microsoft ou o Netscape Composer da Netscape. No entanto, o Microsoft Word e Microsoft Excel, processador de texto e folha de cálculo mais conhecidos pela maioria dos utilizadores informáticos, também já permitem a conversão automática para código HTML de ficheiros "*.doc" e "*.xls", respectivamente.

Queria ainda referir que, este trabalho, por ser também dirigido a professores de Matemática, poderá ser uma forma de motivação para a elaboração de materiais a serem usados na sala de aula, diferentes daqueles que, tradicionalmente, são utilizados. Em vez das habituais fichas de trabalho que são fornecidas, em papel, aos alunos, porque não utilizar-se uma ficha de trabalho interactiva, com exercícios e propostas de trabalho diferentes, do género daqueles que são apresentados nos capítulos IV ou V. De facto, para a elaboração desse tipo de tarefas é aconselhável que se conheçam as principais instruções do código HTML, e é isso que será feito neste capítulo.

Principais comandos do código HTML

Num documento HTML, as marcas de formatação são sempre delimitadas pelos caracteres "<" e ">" e para finalizar o efeito causado por uma determinada marca de formatação, deverá ser usado o carácter "/" a preceder o carácter correspondente à marca que está a ser utilizada. Por exemplo, todo o texto que aparecer entre as marcas (ou "tags") **** e **** irá aparecer no ecrã do cliente em itálico. À marca **** chamamos tag inicial e à marca ****[1] chamamos tag final.

De salientar que os caracteres que identificam as marcas utilizadas na linguagem HTML, constituem abreviaturas das palavras inglesas que produzem os respectivos efeitos. A título de exemplo referem-se as seguintes: **<tr>** - table row; **
** - break; **<b>** - bold; **** - italic; **<u>** - underlined; **<li>** - list; **<ul>** - unumerated list; **<ol>** - ordered list.

A figura que a seguir se apresenta, mostra algumas instruções inerentes a um documento criado em Notepad e que constituem um exemplo de um código HTML que irá dar origem a um documento a ser visualizado num browser. Relativamente à estrutura básica do documento HTML, destacam-se as seguintes situações:

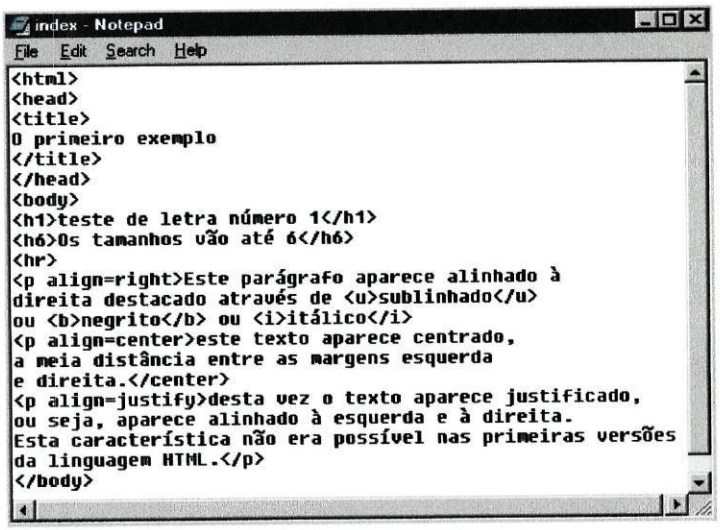
1. As tags **<html>**[2] e **</html>** representam o início e o fim do documento.
2. As tags **<head>** e **</head>** delimitam o cabeçalho que pode conter várias informações relativas ao documento, informações essas que poderão não influenciar directamente o conteúdo do documento tal como este é apresentado na janela do browser.
3. As tags **<title>** e **</title>** delimitam o título que aparecerá no topo da janela referente do browser que apresenta o documento visualizado. Estão contidas entre as tags **<head>** e **</head>**.

---

[1] Como veremos mais adiante, há excepções a este funcionamento das tags. Por exemplo, a tag ** **, que indica um parágrafo, não necessita da ** **; a tag que indica quebra de linha, **
** também não necessita da **</br>**. O mesmo acontece para outras marcas como a **<hr>** (define linha horizontal), **<li>** (define uma lista), entre outras.

[2] Os caracteres podem ser maiúsculos ou minúsculos.

4. As tags **<body>** e **</body>** marcam os limites do corpo do documento. Os vários elementos constituintes do conteúdo do documento são apresentados dentro destas marcas.



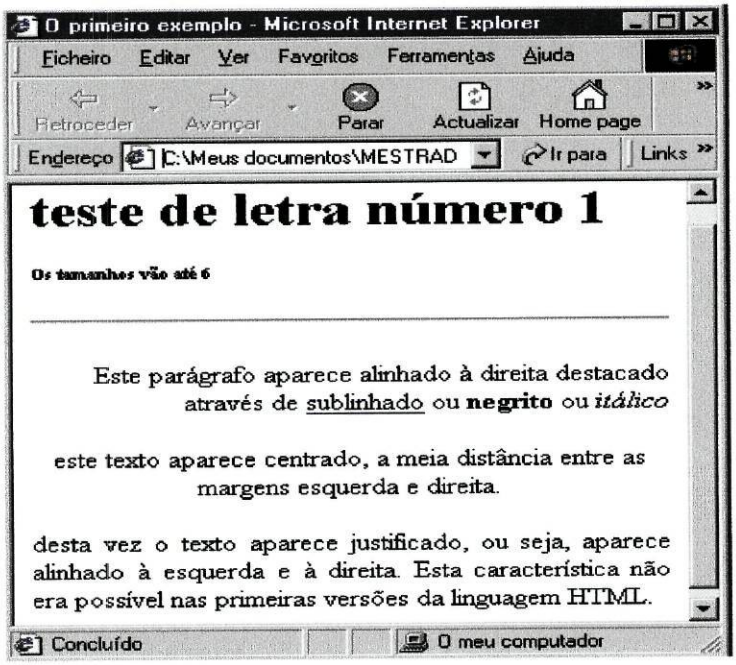
O resultado obtido e visualizado após a interpretação, pelo browser, do código anterior, é o que aparece retratado na imagem seguinte. De notar que, para além das marcas já referidas anteriormente, aparecem ainda outras marcas, cujos resultados são específicos, nomeadamente:

1. Na linguagem HTML foram definidos seis níveis de cabeçalhos, numerados de 1 ao 6, conforme os seus tamanhos decrescentes. Os cabeçalhos são especificados, portanto, usando no início e no fim, respectivamente, as tags **<hn>** e **</hn>**, sendo *n* um número (do 1 ao 6) que especifica o nível do cabeçalho. De referir ainda que, muitas pessoas que utilizam o HTML, pelo menos duma forma menos profissional, usam estes comandos para definir o tamanho da letra do texto a ser apresentado na documento HTML.

2. As tags **<u>** e **</u>** delimitam texto sublinhado, as tags **** e **** delimitam texto em itálico e, por sua vez, as tags **** e **** delimitam texto a negrito.

3. O atributo **<hr>** define uma linha horizontal no ecrã, muitas vezes usada para dividir fisicamente as zonas de texto em HTML.

4. A marca **<p>** é utilizada para definir o início de um novo parágrafo, deixando uma linha em branco entre cada parágrafo. Essa marca pode receber a atributo ALIGN que, por sua vez, pode conter os atributos de LEFT, CENTER, RIGHT e JUSTIFY. Apesar de, por defeito, o texto aparecer alinhado à esquerda, se pretendermos podemos "levar" o texto totalmente à direita usando o atributo RIGHT. O atributo CENTER, faz com que o conteúdo de cada linha do texto do parágrafo seja alinhado ao centro, tendo em conta a dimensão da janela do browser. Finalmente, com o atributo JUSTIFY, o texto aparece alinhado à esquerda e à direita, tal como, hoje em dia, aparece num livro qualquer.



De seguida apresentam-se ainda mais algumas marcas que nos permitem obter resultados específicos e importantes em termos da apresentação da informação. A figura seguinte representa o código HTML de um documento e, para além das marcas referentes à estrutura básica, há a destacar:

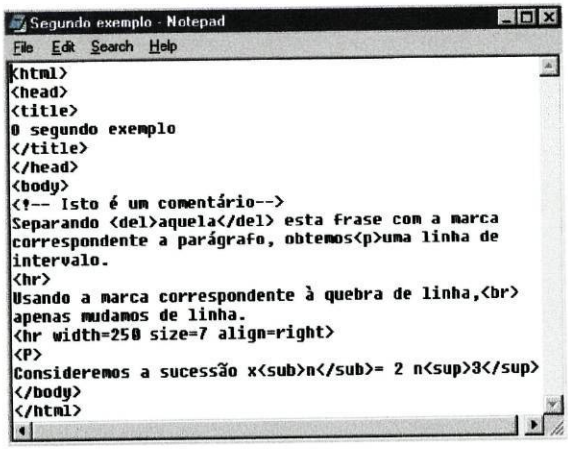
1. O texto que aparece entre as marcas **<!--** e **-->**, constitui um comentário inserido num documento HTML. Embora seja visível no código fonte, fica invisível na página que aparece nos browsers.

2. O texto que aparece entre as tags **~~** e **~~** aparece rasurado.

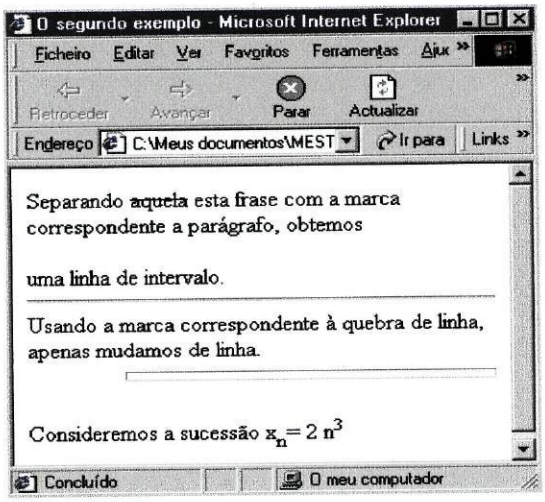
3. Enquanto que a marca **
** define uma quebra de linha precisamente no local em que aparece no código, a marca **<p>**, como se disse anteriormente, define um novo parágrafo deixando uma linha de intervalo.

4. A tag **<hr>**, que introduz uma linha horizontal, pode ainda ser manipulada. Através das instruções *width* e *size*, podemos controlar, respectivamente, a largura e a altura da linha. Os valores atribuídos a estes comandos podem ser absolutos: pixels (px), points (pt), centímetros (cm) e polegadas (in) e, no caso do primeiro (width), pode ainda tomar o valor relativo de percentagem (%), relativamente à largura do elemento onde se insere. Podemos ainda alinhar a linha criada à esquerda, ao centro ou à direita, se bem que, por defeito, a linha é colocada ao centro.

5. O texto que aparece entre as tags **_{** e **}** aparece visualizado em *subscript* (abaixo da posição normal do texto) e, por sua vez, o texto que aparece entre as tags **^{** e **}** aparece em *superscript* (acima da posição normal do texto).



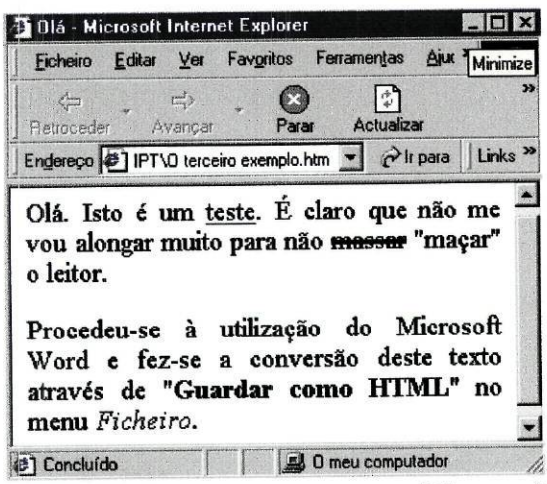
A figura que se segue mostra o efeito que pode ser observado no browser, com um documento que tem o código HTML que aparece na figura anterior.



Existe ainda uma *tag* que poderá ser utilizada para definir, para um determinado texto, um tipo de letra, de tamanho fixo, preservando as quebras de linha e os espaços entre as palavras. Essa *tag* é **<PRE>** e termina com a respectiva **</PRE>**.

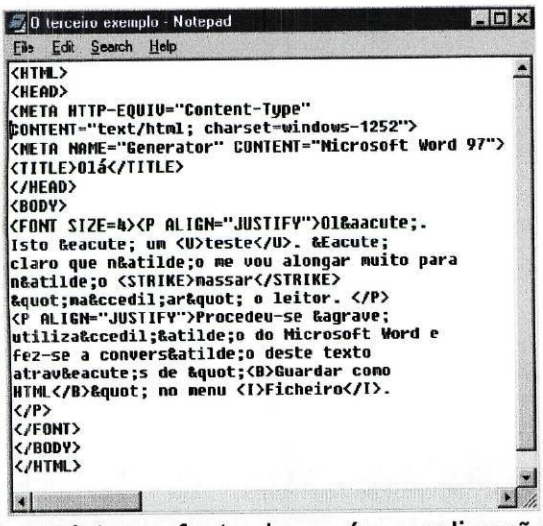
Caracteres especiais

A única forma de termos a garantia de que um determinado carácter é correctamente visualizado por qualquer browser, é usar a codificação HTML adequada. Apesar de, como já se disse anteriormente, existirem hoje no mercado, programas que nos permitem digitar directamente aquilo que se vai visualizar no browser, programas esses que procedem à codificação automática daquilo que é introduzido pelo utilizador, este facto pode trazer problemas com a visualização das páginas em clientes WWW distintos.



A imagem ao lado, representa um ficheiro HTML, criado com o Microsoft Word. Para o gerar, o utilizador escreve o texto que pretende visualizar no Browser e escolhe a função "Guardar como HTML..." no menu "Ficheiro".

Depois de se abrir esse mesmo ficheiro no respectivo browser, o código que se obtém é o que consta da figura seguinte:



Repare-se que, por exemplo, as palavras "Olá" e "à" aparecem codificadas como, respectivamente, "Olá" e "à". Neste dois casos, os caracteres a acentuar, aparecem precedidos do símbolo "&" e seguidos do tipo de acento a utilizar. Uma regra importante acerca da utilização destes caracteres especiais, consiste no facto de, após a aplicação do código, dever ser colocado um ponto e vírgula (";") a separar o código do resto do texto.

De seguida apresenta-se uma tabela com algumas instruções necessárias à introdução de determinado símbolo ou expressão. Destacam-se os símbolos relacionados com a linguagem matemática.

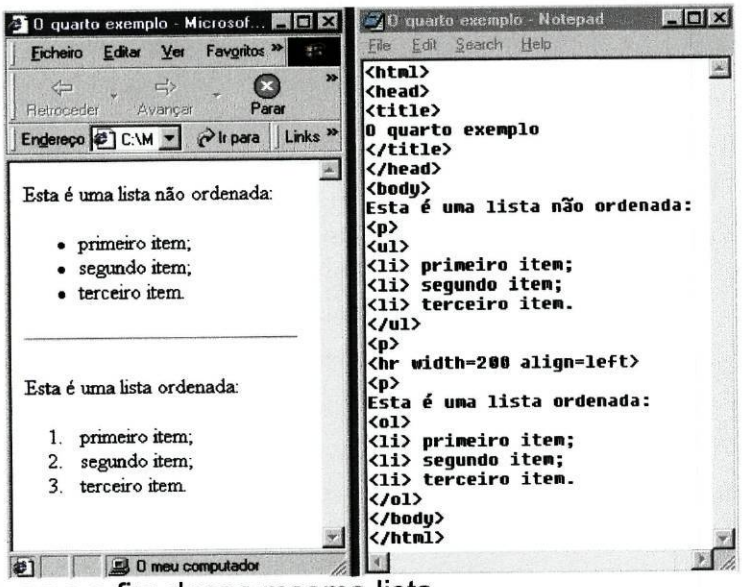
←	¬	⊗	Ä
→	®	∑	å
⇐	Ü	∏	Õ
⇒	Þ	∫	ò
⇔	Û	∀	"
≤	£	∃	&#$;
≥	³	∞	¥
∈	Î	π	p
∉	Ï	o	°

≅	@	÷	÷
≠	¹	©	©
±	&#plusmn;	1/2	½
⊕	Å		

As listas

As listas em HTML, são blocos constituídos por um conjunto de itens. Existem comandos específicos a ter em conta na elaboração de listas e, a imagem seguinte, mostra já dois tipos de listas: as ordenadas e as não ordenadas. A figura da esquerda mostra o que é visualizado no browser após a abertura de um documento cujo código aparece representado na figura da direita. De destacar os seguintes comandos:

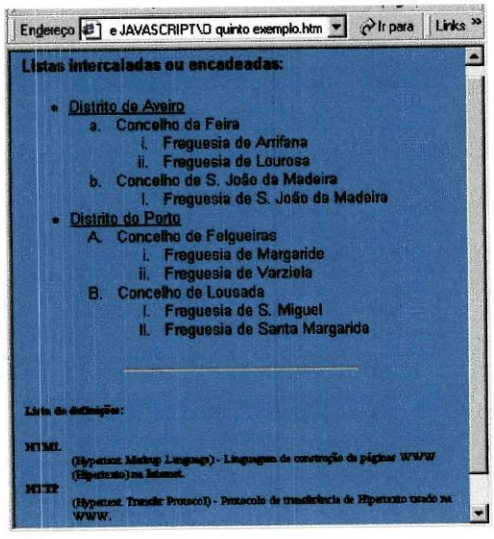
1. A tag **
** marca o início de uma lista não numerada e a tag **
** marca o fim dessa mesma lista.
2. A tag **
** deve ser inserida antes de cada item da lista, não sendo necessária a tag final **
**.
3. A tag **
** marca o início de uma lista numerada e a tag **
** marca o fim dessa mesma lista.



A figura seguinte mostra ainda dois tipos de listas. Na primeira parte, temos listas dentro de outras listas, as chamadas intercalares ou encadeadas. Na segunda parte da imagem, temos uma lista de definições. Aproveitamos a

imagem para dar conta de mais algumas instruções que nos permitem formatar partes do texto. De seguida, é apresentado o código HTML da imagem a que se fez referência.

```
<html>
<head>
<title>O quinto exemplo</title>
<basefont face="arial" size=2>
</head>
<body bgcolor="#00FFFF">
<b>Listas intercaladas ou encadeadas:</b>
<ul>
<li><u>Distrito de Aveiro</u>
        <ol type=a>
        <li>Concelho da  Feira
                <ol type =i>
                <li>Freguesia de Arrifana
                <li>Freguesia de Lourosa
                </ol>
        <li>Concelho de S. João da Madeira
                <ol type=I>
                <li>Freguesia de S. João da Madeira
                </ol>
        </ol>
<li><u>Distrito do Porto</u>
        <ol type=A>
        <li>Concelho de Felgueiras
                <ol type=i>
                <li>Freguesia de Margaride
                <li>Freguesia de Varziela
                </ol>
        <li>Concelho de Lousada
                <ol type=I>
                <li>Freguesia de S. Miguel
                <li>Freguesia de Santa Margarida
                </ol>
  </ol>
</ul>
<hr width=200 align=center>
<p><font face="times" size=1>
<b>Lista de definições:</b>
<dl>
<dt><b>HTML</b>
<dd>(Hypertext Markup Language) – Linguagem de construção de páginas WWW (Hipertexto) na Internet.
</dd>
<dt><b>HTTP</b>
<dd>(Hypertext Transfer Protocol) – Protocolo de transferência de Hipertexto usado na WWW.
</dd>
</dl>
</font></p>
</body>
</html>
```



Relativamente às tags apresentadas, há a destacar as seguintes:

1. O atributo TYPE é utilizado para escolher o tipo de ordenação que se pretende dar às listas ordenadas. As opções são as que se apresentam na lista seguinte:

   a) TYPE=1: a numeração aparece sequencial (1, 2, 3, ...);
   b) TYPE=i: a numeração utilizada é a romana minúscula (i, ii, iii, ...);
   c) TYPE=I: a numeração utilizada é a romana maiúscula (I, II, III, ...);
   d) TYPE=a: a ordenação aparece em letras minúsculas (a, b, c, ...);
   e) TYPE=A: a ordenação aparece em letras maiúsculas (A, B, C, ...).

2. Numa lista de definição, cada item é constituído por dois elementos: o elemento que apresenta o termo e o elemento que contém a definição. A lista é criada através do elemento DL, definido através das tags **<dl>** e **</dl>**. Por sua vez, o termo é definido pelas tags **<dt>** e **</dt>** e a definição do termo é representada pelas tags **<dd>** e **</dd>**.

3. O elemento BASEFONT, incluído no **<head>**, pode ser utilizado para definir o tipo de letra e o seu tamanho que, por defeito, queremos utilizar ao longo do nosso documento, através dos atributos FACE e SIZE. Se pretendermos no **<body>** do documento utilizar outro tipo de letra, usamos o elemento FONT e definimos novos atributos FACE e SIZE.

4. O atributo BGCOLOR pode ser utilizado para definirmos a cor de fundo de todo o documento. Existem duas formas de identificar a cor de fundo pretendida: uma pela designação da própria cor (por exemplo, black, silver, yellow, red, entre 16 cores disponíveis) e outra pela introdução de um código[3] como aquele que foi utilizado no nosso exemplo. Do mesmo modo, pode ser definida a cor do texto no corpo do documento, através do atributo

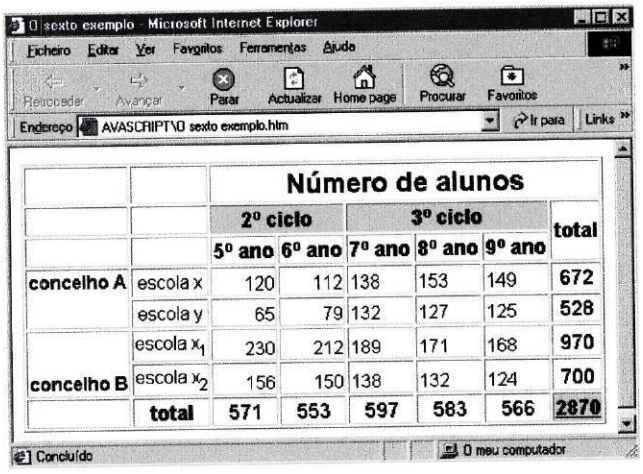
---

[3] As cores do ecrã são formatadas através de uma combinação das cores vermelho, verde e azul (RGB). Cada uma dessas três cores básicas pode assumir uma intensidade que vai de 0 a 255, valores indicados em sistema hexadecimal. Assim, uma cor é uma combinação de seis dígitos (precedida pelo símbolo #), em que dois a dois, a contar da esquerda, dizem respeito, respectivamente, ao vermelho, ao verde e ao azul.

TEXT, seguido do código da cor idêntico ao que foi apresentado para o atributo BGCOLOR Pode ser visualizada uma lista de cores bastante completa no endereço: http://www.infi.net/wwwimages/colorindex.html.

As tabelas

A criação de tabelas é uma das tarefas mais usadas por quem constrói uma página Web. Através do exemplo que é apresentado de seguida, são mostradas as tags essenciais usadas na construção de tabelas. A figura seguinte mostra uma tabela em que estão contempladas as principais formatações que são possíveis obter.



		Número de alunos					
		2º ciclo		3º ciclo			total
		5º ano	6º ano	7º ano	8º ano	9º ano	
concelho A	escola x	120	112	138	153	149	672
	escola y	65	79	132	127	125	528
concelho B	escola x_1	230	212	189	171	168	970
	escola x_2	156	150	138	132	124	700
	total	571	553	597	583	566	2870

O código HTML inerente à tabela anterior, é o que de seguida se apresenta.

```
<HTML>
<HEAD>
<TITLE>O sexto exemplo</TITLE>
</HEAD>
<BODY>
<Table border>
<TR VALIGN="bottom">
<TD ALIGN="right"><FONT FACE="Arial"> </FONT></TD>
<TD ALIGN="right"><FONT FACE="Arial"> </FONT></TD>
```

```
<TD COLSPAN=6 ALIGN="center"><FONT FACE="Arial" SIZE=+2><B>N&uacute;mero de
alunos</B></FONT></TD>
</TR>
<TR>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial"> </FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial"> </FONT></TD>
<TD BGCOLOR=#C0C0C0 COLSPAN=2 ALIGN="center" VALIGN="bottom"><FONT
FACE="Arial" SIZE=+1><B>2º ciclo</B></FONT></TD>
<TD BGCOLOR=#C0C0C0 COLSPAN=3 ALIGN="center" VALIGN="bottom"><FONT
FACE="Arial" SIZE=+1><B>3º ciclo</B></FONT></TD>
<TD ROWSPAN=2 ALIGN="center" VALIGN="middle"><FONT FACE="Arial"
SIZE=+1><B>total </B></FONT></TD>
</TR>
<TR>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial"> </FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial"> </FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1><B>5º
ano</B></FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1><B>6º
ano</B></FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1><B>7º
ano</B></FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1><B>8º
ano</B></FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1><B>9º
ano</B></FONT></TD>
</TR>
<TR>
<TD ROWSPAN=2 ALIGN="center" VALIGN="top"><FONT FACE="Arial"><B>concelho
A</B></FONT></TD>
<TD ALIGN="center" VALIGN="middle"><FONT FACE="Arial">escola x</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">120</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">112</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">138</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">153</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">149</FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1>672</FONT></TD>
</TR>
<TR>
<TD ALIGN="center" VALIGN="middle"><FONT FACE="Arial">escola y</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">65</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">79</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">132</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">127</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">125</FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1>528</FONT></TD>
</TR>
<TR>
<TD ROWSPAN=2 ALIGN="center" VALIGN="bottom"><FONT FACE="Arial"><B>concelho
B</B></FONT></TD>
<TD ALIGN="center" VALIGN="middle"><FONT FACE="Arial">escola x<SUB>1</SUB>
</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">230</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">212</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">189</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">171</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">168</FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1>970</FONT></TD>
</TR>
<TR>
<TD ALIGN="center" VALIGN="middle"><FONT FACE="Arial">escola x<SUB>2</SUB>
```

```
</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">156</FONT></TD>
<TD ALIGN="right" VALIGN="bottom"><FONT FACE="Arial">150</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">138</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">132</FONT></TD>
<TD ALIGN="left" VALIGN="bottom"><FONT FACE="Arial">124</FONT></TD>
<TD ALIGN="center" VALIGN="bottom"><FONT FACE="Arial" SIZE=+1>700</FONT></TD>
</TR>
<TR VALIGN="bottom">
<TD ALIGN="right"><FONT FACE="Arial"> </FONT></TD>
<TD ALIGN="center"><FONT FACE="Arial" SIZE=+1><B>total</B></FONT></TD>
<TD ALIGN="center"><FONT FACE="Arial" SIZE=+1>571</FONT></TD>
<TD ALIGN="center"><FONT FACE="Arial" SIZE=+1>553</FONT></TD>
<TD ALIGN="center"><FONT FACE="Arial" SIZE=+1>597</FONT></TD>
<TD ALIGN="center"><FONT FACE="Arial" SIZE=+1>583</FONT></TD>
<TD ALIGN="center"><FONT FACE="Arial" SIZE=+1>566</FONT></TD>
<TD BGCOLOR=#C0C0C0 ALIGN="center"><FONT FACE="Arial"
SIZE=+1><B><U>2870</U></B></FONT></TD>
</TR>
</Table>
</BODY>
</HTML>
```

Pela análise deste código, podemos realçar os seguintes aspectos:

1. Toda a tabela deve ser iniciada com a tag **<table>** e encerrada com a tag **</table>**. De notar que, por defeito, a tabela aparece sem limites, apesar de fazer as divisões em linhas e colunas. Para que a tabela defina bordas simples, tal como no nosso exemplo, deve ser iniciada com **<table border>**[4] e encerrada com **</table>**.

2. As linhas começam com a tag **<TR>** e terminam com a correspondente **</TR>**. Dentro de cada linha (antes de se utilizar a respectiva tag de encerramento **</TR>**), definem-se todas as células que irão existir, ou seja, definem-se as colunas. As tags **<TD>** e **</TD>** definem cada célula de uma tabela.

3. Os browsers WWW determinam automaticamente a largura da tabela. No entanto, este parâmetro pode ser fixado usando a opção WIDTH, que permite definir o tamanho da tabela como percentagem do tamanho da janela de visualização. Por exemplo, se na nossa tabela substituíssemos a

---

[4] Acrescentando um valor ao atributo BORDER é possível aumentar a espessura dos limites da tabela. Por exemplo **<table border=n>**, em que n é a espessura do traço em pixels.

linha de definição da tabela por **<TABLE BORDER WIDTH=60%>**, obtínhamos uma tabela que ocuparia 60% do espaço disponível na janela de visualização que o cliente estiver a utilizar[5].

4. Os qualificadores COLSPAN e ROWSPAN são aplicados às marcas **<TD>** e permitem definir o tamanho de cada célula em linhas ou colunas. Por defeito, o valor de cada um desses qualificadores é 1 (um). Por exemplo, a linha de código **<TD COLSPAN=6 ALIGN="center"><FONT FACE="Arial" SIZE=+2><B>Número de alunos</B></FONT></TD>**, coloca a expressão "Número de alunos" ao longo de seis colunas, alinhada ao centro. Por seu turno, a linha de código **<TD ROWSPAN=2 ALIGN="center" VALIGN="top"><FONT FACE="Arial"><B>concelho A</B></FONT></TD>**, coloca a expressão "concelho A" ao longo de duas linhas, alinhada ao topo da célula.

5. O atributo ALIGN pode ser utilizado nas tags **<TR>** e **<TD>** e permite o controlo do alinhamento do texto dentro de uma célula, em relação aos limites laterais. Pode tomar os valores left, right ou center e os exemplos vistos no ponto anterior retratam também isso.

6. O atributo VALIGN, que pode ser utilizado também nas mesmas tags, permite o controlo do alinhamento do texto dentro de uma célula, em relação aos limites superior e inferior. Pode tomar os valores top, middle e bottom.

7. Os atributos **CELLSPACING=value** e **CELLPADDING=value**, aplicáveis à tag **<TABLE>**, definem, respectivamente, o espaço entre cada célula na tabela e a distância entre o texto e o limite de cada célula.

---

[5] O valor que este atributo pode tomar também pode ser referido em "pixels" (ponto em tela gráfica)

Inserir Imagens

Nas páginas Web é comum o uso de imagens, inclusivamente, imagens animadas (ou seja, que têm movimento). Neste aspecto, há que ter em atenção o tamanho do ficheiro que diz respeito a uma determinada imagem, uma vez que isso se vai reflectir na velocidade com que a página que está a ser criada irá ser carregada no computador do utilizador, principalmente aquele que acede à rede via Modem. De facto, os ficheiros de imagem (bem como os de som e vídeo) podem facilmente atingir tamanhos consideráveis, pelo que, há que ter, antes de mais, cuidado com a forma como são criados, com a extensão com que são armazenados. Os formatos mais populares de ficheiros de imagem que existem nos servidores WWW, são o GIF e JPG, formatos estes que, geralmente, não trazem problemas de visualização. Existem também outros diferentes tipos de compactação de ficheiros (como por exemplo BMP, TIFF, PCX, entre outros) e que são enviados muitas vezes para aplicações externas ("Helper Applications") ao próprio browser que está a ser usado.

A notação utilizada para incluir uma imagem num documento é a seguinte:

<img src="xxx.jpg">

em que "xxx.jpg" identifica o ficheiro da imagem a incluir. A figura que se segue representa um documento no qual foi introduzida uma imagem. Na parte lateral da janela representativa do que é visualizado no browser, aparece o código HTML respectivo.




```
<HTML>
<HEAD>
<TITLE>O sétimo exemplo</TITLE>
</HEAD>
<BODY>
<img align="top" src="brasaosjm.jpg"
title="Brasão da Cidade de S. João da Madeira">
texto no topo
<img align="middle" border=1 height=120 widht=90
src="brasaosjm.jpg">
texto no meio
</BODY>
</HTML>
```

Pela análise da imagem anterior e do respectivo código, podemos destacar os seguintes aspectos:

1. O atributo ALIGN é utilizado para definir o alinhamento da imagem face aos restantes elementos da página que está a ser visualizada. Pode tomar os valores: TOP, MIDDLE e BOTTOM[6]. Se nada se disser para o atributo ALIGN, a imagem aparecerá alinhada à esquerda da página e o texto alinhado pela parte inferior da imagem.

2. O atributo TITLE pode ser utilizado para fazer aparecer uma legenda da imagem respectiva, sempre que posicionarmos o ponteiro do rato sobre a imagem[7].

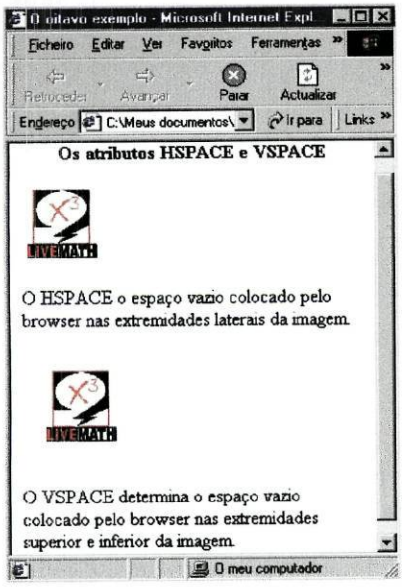
3. Os atributos HEIGHT e WIDTH determinam o tamanho da imagem. O primeiro define a altura e o segundo a largura. A medida pode ser expressa em pixels ou em percentagem e, caso estes atributos não sejam referidos, o

---

[6] O atributo ALIGN também pode tomar os valores LEFT e RIGHT, aparecendo a imagem posicionada, respectivamente, à esquerda e à direita do texto que se encontra no mesmo parágrafo.

[7] O atributo ALT faz também aparecer uma legenda da imagem, mesmo nos casos em que o browser não suporta imagens, ou no caso da apresentação de imagens ter sido desactivada pelo utilizador.

browser apresenta a imagem no seu tamanho original. Ao manipular estes atributos, o utilizador deve ter em atenção o facto de poder estar a distorcer a própria imagem.



Na figura ao lado, são apresentadas outras formatações que podemos utilizar com imagens.

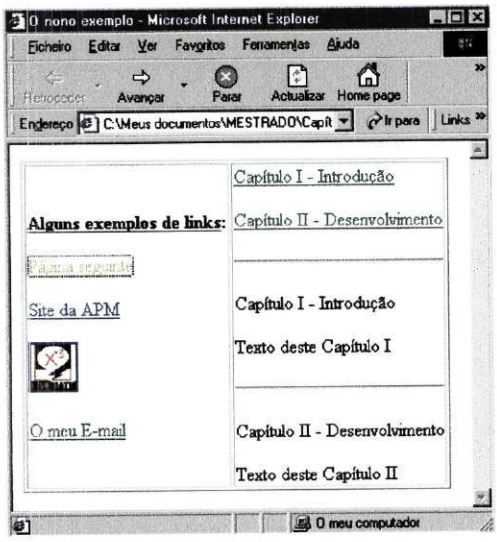
Do código que é apresentado de seguida, podemos verificar que o atributo HSPACE pode ser utilizado para definir o espaço lateral à imagem deixado em branco e, por sua vez, o atributo VSPACE define as margens superior e inferior da imagem.

```
<HTML>
<HEAD>
<TITLE>O oitavo exemplo</TITLE>
</HEAD>
<BODY>
<b><center>Os atributos HSPACE e VSPACE</center></b>
<p>
<img height=60 widht=50 hspace=5 vspace=3 src="livemath.gif">
<p>
O HSPACE o espaço vazio colocado pelo browser nas extremidades
laterais da imagem.
<p>
<img height=60 widht=50 hspace=20 vspace=15 src="livemath.gif">
<p>
O VSPACE determina o espaço vazio colocado pelo browser nas
extremidades superior e inferior da imagem.
</BODY>
</HTML>
```

Interligação de documentos

Como já se disse anteriormente, o que dá a um documento HTML a característica de hipertexto é a existência de links, que estabelecem ligações entre elementos de uma página com outros elementos dessa mesma página ou de outras páginas, situadas no mesmo servidor ou em servidores distintos. Por

outro lado, quer a origem, quer mesmo o destino de um hiperlink não têm necessariamente que ser texto. De facto, a origem do link pode ser também um elemento gráfico e o destino pode igualmente ser uma imagem, um programa ou qualquer outro recurso multimédia. Por isso é que, cada vez mais, se usa a expressão hipermédia em vez da expressão hipertexto.



O início e o fim de um link são assinalados, respectivamente, pelas marcas **<A>**[8] e **</A>**, estando a especificação do destino desse link a cargo do atributo **HREF**. A figura ao lado, representa alguns exemplos de links, tanto ao nível da sua origem, como do seu destino.

O código HTML referente à figura anterior é o que de seguida se apresenta:

```
<HTML>
<HEAD>
<TITLE>O nono exemplo</TITLE>
</HEAD>
<BODY link=blue alink=orange vlink=green>
<table border=1>
<tr>
<td>
<b><u>Alguns exemplos de links</u>:</b><p>
<a href="O oitavo exemplo.htm">Página seguinte</a><p><!-- link interno -->
<a href="http://www.apm.pt">Site da APM</a><p><!-- link externo com origem em palavras -->
<a href="http://www.livemath.com"><img height=40 width=30 src="livemath.gif"></a><!-- link
externo com origem em imagens -->
<p>
<a href="mailto:luismiguelferreira@mail.telepac.pt">O meu E-mail</a>
</td>
<td>
<a href="#A">Capítulo I - Introdução</a><p>
<a href="#B">Capítulo II - Desenvolvimento</a><p>
<p><hr><p>
<a name="A">Capítulo I - Introdução</a><p>Texto deste Capítulo I<p><hr><p>
<a name="B">Capítulo II - Desenvolvimento</a><p>Texto deste Capítulo II
</td>
</tr>
</BODY></HTML>
```

---

[8] A letra "A" deriva da palavra anchor (em português âncora).

Relativamente às instruções mais usadas na obtenção de links, destacam-se as seguintes:

1. Na linha de comandos **<BODY link=blue alink=orange vlink=green>** aparecem os atributos link=blue, alink=orange e vlink=green. O primeiro determina a cor do link que ainda não foi visitado, o segundo determina a cor do link que está a ser visitado e, por sua vez, o terceiro determina a cor do link que já foi visitado.

2. O link "Página seguinte" dá acesso a uma nova página dentro do mesmo documento. A instrução terá que fazer referência ao ficheiro que será carregado. Neste caso, ao clicarmos em cima da expressão "Página seguinte", acederíamos ao ficheiro "O oitavo exemplo.htm". A instrução respectiva é **Página seguinte**.

3. O link "Site da APM" dá acesso à página WEB da Associação de Professores de Matemática, um site externo ao documento que está a ser visualizado. Obviamente que o endereço deve ser mencionado. A instrução referente a este link é**: Site da APM**.

4. A figura que aparece no nosso exemplo, constitui um link para a página WEB cujo endereço é o seguinte: www.livemath.com. É idêntico ao caso anterior, diferindo apenas na origem uma vez que, neste caso, em vez de texto, é uma imagem. A instrução que diz respeito a este link é: **<img height=40 width=30 src="livemath.gif">**

5. Podemos também criar links cujo destino seja o Endereço de Correio Electrónico de alguém. No exemplo em análise, o link cuja origem é a frase "O meu E-mail" permite enviar uma mensagem para a minha caixa de correio, cujo endereço é luismiguelferreira@mail.telepac.pt. A instrução que o permite fazer no exemplo em análise é: **O meu E-mail**

6. A parte direita da figura que está em análise, ilustra o processo de utilização da técnica que consiste em manter, num mesmo documento, vários links que dão acesso a secções distintas dentro do próprio documento. A instrução **Capítulo I - Introdução** cria o link, através do qual se poderá chegar com mais facilidade a essa determinada sessão, neste caso, cujo início aparecerá referenciado com a instrução **Capítulo I - Introdução**.

Em suma, cada recurso existente na Internet é referenciado por um URL – Uniform Resource Locators, que constitui o endereço desse recurso. Além disso, o URL pode ser um ficheiro ou uma determinada secção dentro de um outro ficheiro, acessível na própria página que contém o documento que está a ser visualizado ou num site exterior ao próprio documento.

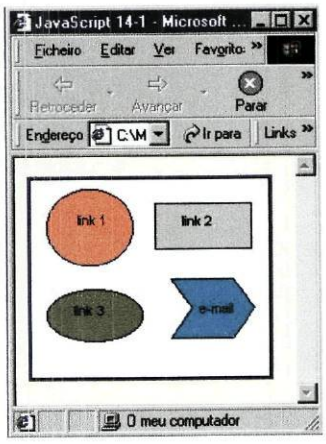
Image maps

Os mapas de imagens interactivos (image maps), aparecem com frequência nas páginas Web pelas importantes potencialidades que apresentam e constituem uma extensão dos links que têm por origem uma imagem. No entanto, enquanto que quando uma imagem constitui a origem de um único link, esse link é activado através do clique sobre qualquer ponto da imagem respectiva, já nos "image maps", a imagem inerente poderá constituir dois ou mais links definidos em zonas diferentes[9] da mesma imagem.

---

[9] Essas zonas poderão tomar variadíssimas formas, designadamente, rectangulares, circulares, poligonais, ovais, entre outras, mais ou menos regulares.

A imagem seguinte, constitui um exemplo de um "Image map", em que cada



uma das figuras geométricas visíveis na mesma figura, constitui um link para uma página diferente. Uma nota prévia para referir que a identificação das zonas que constituem as origens dos links, é feita através de uma sistema de coordenadas. As coordenadas dos pontos a que se fará referência, poderão ser obtidas através da utilização de um qualquer programa de desenho, como por exemplo, o MS Paint.

De seguida apresenta-se o código da página associada à figura anterior:

```
<html>
<head>
<title>JavaScript 14-1</title>
</head>
<body bgcolor="cornsilk">
<map name="map4">
<area shape="rect" coords="100,19,178,55" href="jscript14-2.html">
<area shape="circle" coords="47,38,30" href="jscript14-3.html">
<area shape="poly" coords="112,80,159,80,178,103,161,126,112,126,130,103"
href="mailto:luismiguelferreira@mail.telepac.pt" alt="Enviar mensagem">
<area shape="oval" coords="11,88,90,126" href="jscript14-4.html">
</map>
<img src="map4.jpg" usemap="#map4">
</body>
</html>
```

Pela análise do código anterior, há a destacar os seguintes aspectos:

1. As tags **<map>** e **</map>** indicam o início e o fim de uma descrição de um "map". O atributo NAME associado à tag map, identifica o "image map" respectivo.

2. A instrução **<img src="map4.jpg" usemap="#map4">** introduz na nossa página a figura onde serão definidas as zonas que constituirão os links (no nosso exemplo, essa figura está associada ao ficheiro "map4.jpg".). O atributo usemap contém o valor que figura no atributo NAME da tag **<map>**, referido no ponto anterior, precedido do símbolo #.

3. No interior da tag **<map>**, cada zona de origem de link é criada através do elemento **AREA**. O tipo de zona que constituirá o próprio link, é definido pelo atributo **SHAPE**, que poderá tomar os seguintes valores: **RECT** (identifica rectângulos), **CIRCLE** (identifica círculos), **POLY** (identifica polígonos), **POINT** (identifica pontos) e **OVAL** (identifica formas ovais). Por exemplo, a instrução **<area shape="circle" coords="47,38,30" href="jscript14-3.html">** cria uma área de link em círculo, cujo destino é o documento com o nome "jscript14-3.html".

4. O atributo **COORDS**, visível no ponto anterior, especifica as coordenadas da área que constitui a origem de um link. No caso dessa região ser um **CIRCLE**, as coordenadas a usar são um ponto (centro) e um raio; se a região for **OVAL**, deverá imaginar-se essa região inscrita num rectângulo, sendo usadas as coordenadas dos vértices dos cantos superior direito e inferior esquerdo do rectângulo; se a região for **RECT**, as coordenadas usadas são os vértices do rectângulo, dos cantos superior direito e inferior esquerdo; se a região for **POLY**, as coordenadas a usar são as coordenadas de tantos pontos quantos se queira, num máximo de 100; finalmente, se a região for um **POINT**, as coordenadas a usar são as desse mesmo ponto.

5. À semelhança do que acontece com o que já foi dito anteriormente acerca dos links, aqui também o destino dos links definidos poderão ser uma nova página (associada a um novo ficheiro HTML, conforme se pode verificar nos links 1, 2 e 3) ou um endereço e-mail (sendo associado o endereço respectivo, conforme se pode verificar na zona identificada com "e-mail").

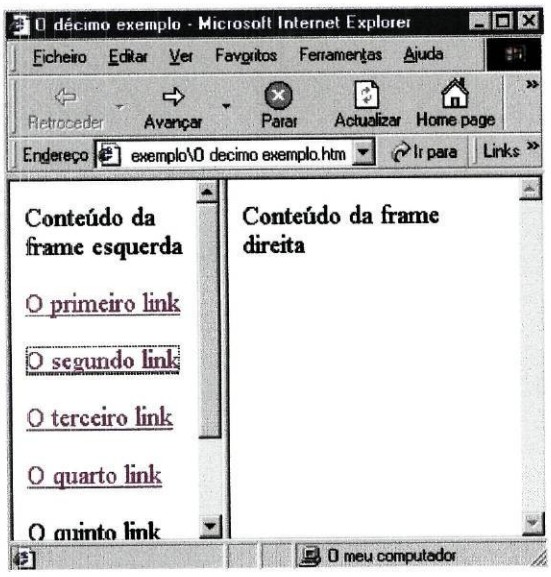
Frames

As frames são divisões lógicas do ecrã visualizado pelo utilizador, permitindo ver mais do que uma página ao mesmo tempo, apesar de hoje em dia, haver determinados browsers (usados por menos de 10% da totalidade dos utilizadores da rede) que não conseguem ver frames. No entanto, o Netscape

Navigator e o Internet Explorer permitem-no. Aliás, trata-se de uma inovação introduzida desde o Netscape 2.0 e que a Microsoft, no seu IE 3.0, acompanhou.

Ora, para que se perceba melhor o que é uma frame, refira-se que cada uma das zonas com que o ecrã aparece dividido, corresponde a um documento independente, associado a um ficheiro HTML distinto, isto é, dependendo do número de frames criadas, vários documentos poderão ser visualizados ao mesmo tempo.

A imagem que se segue, representa um documento, cujo código respectivo pode ser visualizado na parte lateral, e que contém duas frames.




```
<html>
<head>
<title>O décimo exemplo</title>
</head>
<frameset cols=40%,60%>
<frame src="ex10esq.htm"
name=frameesquerda>
<frame src="ex10dir.htm" name=framedireita
scrolling=yes>
</frameset>
</html>
```

Da análise do exemplo anterior, destacam-se os seguintes aspectos:

1. O elemento **<FRAMESET>**[10] divide a janela do browser em várias zonas, cada uma delas constituindo uma área independente, na qual poderá ser apresentado um documento. Este elemento deverá também ser seguido da respectiva tag final **</FRAMESET>**, depois de definidas as frames a inserir no documento.

[10] Este elemento é usado em substituição do elemento <BODY>.

2. No nosso exemplo, através da instrução **<frameset cols=40%,60%>**, foi usado o atributo **COLS**, que determina que as frames são inseridas em coluna. No caso de pretendermos criar frames em linha, deveria usar-se o atributo **ROWS**. Este atributo aparece seguido do tamanho que pretendemos para cada frame, tamanho esse que pode ser expresso em percentagem (usando-se o símbolo %), ou em pixels. No exemplo que é apresentado, a frame esquerda ocupa 40% do ecrã, enquanto que a frame direita ocupa os restantes 60% do ecrã.

3. O elemento **<FRAME>** é usado para se definir as características de cada frame. O atributo NAME define o nome que é dado à respectiva frame. Além disso, temos o atributo SRC que é utilizado para definir o URL do ficheiro associado à frame. No nosso exemplo, o ficheiro "ex10esq.htm" é apresentado na parte esquerda do ecrã, enquanto que o ficheiro "ex10dir.htm" é apresentado na parte direita do ecrã.

4. O atributo **SCROLLING** indica se vai aparecer, ou não, a barra de deslocação associada à frame. O utilizador poderá escolher "YES" (se pretender que a barra de deslocação seja sempre visualizada), "NO" (se pretender que a barra de deslocação nunca seja visualizada) ou "AUTO"[11] (se pretender que a barra de deslocação apareça automaticamente, quando tal for necessário).

Podemos ainda negar a possibilidade de alteração das dimensões das frames a quem está a visualizar a nossa página, usando o atributo **NORESIZE** depois de definirmos o URL da frame respectiva. Há também a possibilidade de definirmos o "border" da frame, usando para tal o atributo **FRAMEBORDER**=0 ou **FRAMEBORDER**=1 se pretendermos, ou não, ausência do mesmo. Em cada frame é também possível definir o espaço, em pixels, que separa o conteúdo do documento dos limites da frame. Assim, o atributo **MARGINWIDTH** define o espaço a introduzir à esquerda e à direita, enquanto

---

[11] Valor assumido por defeito, quando nada é mencionado.

que o atributo **MARGINHEIGHT** define o espaço a introduzir na parte superior e inferior.

Uma questão que também se coloca na criação de frames, prende-se com o destino dos links situados numa determinada frame, incluída num documento que contém várias frames. O atributo **TARGET**, colocado em cada elemento que activa os hiperlinks ou definido no **HEADER** do documento, especifica a janela ou a frame em que o documento que constitui o destino do hiperlink será apresentado. O código que de seguida se apresenta, refere-se ao ficheiro que constitui a frame esquerda do nosso exemplo.

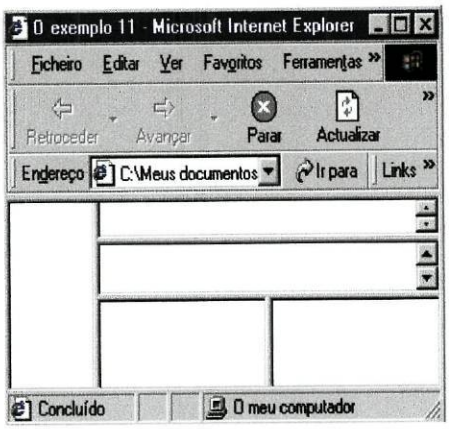
```
<html>
<head>
<title>Frame esquerda</title>
</head>
<body>
<font size=4>
Conteúdo da frame esquerda
<p>
<a target=framedireita href="ex10result1.htm">O primeiro link</a>
<p>
<a target=_blank href="ex10result2.htm">O segundo link</a>
<p>
<a target=_self href="ex10result3.htm">O terceiro link</a>
<P>
<a target=_top href="ex10result4.htm">O quarto link</a>
<p>
O quinto link</a>
<p>
O sexto link<p>
</body>
</html>
```

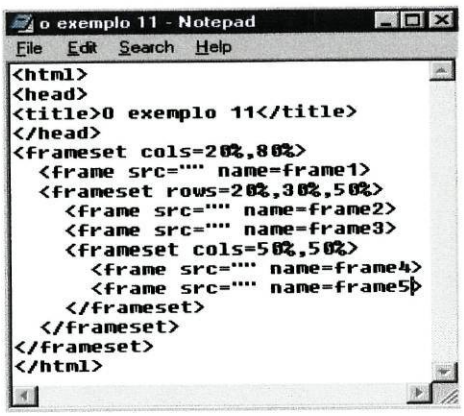
Da sua análise, retiram-se alguns exemplos de valores que o atributo **TARGET** pode tomar. Assim:

1. O destino d' O primeiro link será a frame do lado direito, através da instrução **O primeiro link**, uma vez que à frame direita foi dado o nome "framedireita", através da instrução disponível no documento principal **<frame src="ex10dir.htm" name=framedireita scrolling=yes>**;

2. O destino d' O segundo link será uma nova janela que se abrirá, efeito conseguido através da instrução **O segundo link**;
3. O destino d´ O terceiro link será a própria frame em que o link foi activado, através da instrução **O terceiro link**;
4. O destino d' O quarto link será a janela que contém as duas frames, desta vez não tendo qualquer divisão.[12] Este efeito foi conseguido através da utilização da instrução: **O quarto link**.

Para finalizar esta questão das frames, refira-se que é possível criar ainda uma hierarquia de frames, como a que de seguida se apresenta.





Forms

Em termos de capacidades interactivas, as "forms" trazem um grande valor acrescentado à WWW. O uso de formulários é frequente quando se pretende o preenchimento e envio de dados, por parte do utilizador, sendo a informação comunicada a um servidor ou a um processador de modo a que se obtenha determinada acção. Um exemplo frequente na Web verifica-se quando uma

[12] Este valor _TOP, é o valor assumido por defeito quando o argumento TARGET não é especificado e o destino do hiperlink é exterior ao site actual.

form é utilizada para recolher a identificação de um utilizador. Neste caso, o utilizador digita na form a sua identificação e a sua password, informação essa que terá de ser enviada ao servidor para processamento e respectiva validação. Para a introdução dessas informações, podem ser usados elementos como caixas de texto, botões, caixas de listagem, de verificação ou de opção, entre outros. O início e o fim da definição de uma "form", são referenciados, respectivamente, pelas tags **<FORM>** e **</FORM>**.

O elemento **FORM**, tem vários atributos, dos quais se começa por destacar dois, necessários à definição da entidade de recepção dos dados que estão associados à form: por um lado temos o **ACTION** (que contém o endereço do programa utilizado para processar os dados da form) e **METHOD** (que define o formato da mensagem enviada com os dados da form, para o servidor, através dos valores **GET** e **POST**).



A figura ao lado representa um exemplo de um documento que utiliza alguns tipos de *forms*. Passemos à análise do código respectivo, que aparece imediatamente a seguir à imagem.

```
<html>
<head>
<title>O exemplo 12</title>
</head>
<body>
<p align=center>
<font size=5>
<B>Exemplo de alguns formulários</b>
</font>
</p>
<table border=1>
<tr>
<td colspan="2">
```

```
<!-- dados pessoais-->
<form>
<b>Dados Pessoais</b><br>
Primeiro nome:
<input type=text name="text1" maxlength=20 size=15><br>
Último nome:
<input type=text name="text2" maxlength=20 size=16><br>
<label>Morada:<input type=text name="text3" maxlength=60 size=21>
</form>
</td>
<td valign="top">

<!-- sexo -->
<form>
<b>Sexo</b><br>
<input type="radio" name="radio1" value="escolha1">Masculino<br>
<input type="radio" name="radio2" value="escolha2">Feminino<br>
<input type=reset name"restaurar" value="reset">
</form>
</td>
<tr>
<td valign="top">

<!-- disciplinas preferidas -->
<form>
<b>Disciplinas preferidas</b><br>
<input type=checkbox name="check1" checked>Matemática<br>
<input type=checkbox name="check2">Geografia<br>
<input type=checkbox name="check3">História
</form>
</td>
<td valign="top">

<!-- nacionalidade -->
<form>
<b>Nacionalidade</b><br>
<select name="lista" size=1>
<option>portuguesa</option>
<option>brasileira</option>
<option>espanhola</option>
<option>outra</option>
</select>
</form>
</td>
<td>

<!-- área de texto -->
<form>
<textarea cols=15 rows=5>
isto é uma "textarea"
</textarea>
</form>
</td>
</tr>
</table>
</body>
</html>
```

Da análise do código anterior, há a destacar os seguintes aspectos:

1. O elemento "**INPUT**", é o que contém uma maior diversidade de opções: **NAME** (que define o nome do elemento), **SIZE** (que define o tamanho visível do campo em número de caracteres), **MAXLENGTH** (que define o número máximo de caracteres admissível no campo), **VALUE** (que define o texto inicial a ser mostrado ou a posição inicial do botão de selecção), **CHECKED** (que coloca o valor do campo como "Activo" - seleccionado - no caso dos botões de selecção ou das caixas de selecção) e **TYPE** (que define o tipo de elemento em causa);

2. O atributo **TYPE** pode tomar os seguintes valores: Button (botão de comando), Checkbox (caixa de verificação), Radio (caixa de opção), Text (caixa de texto), Submit (botão Submit), Reset (botão Reset), Password (caixa de password), Image (imagem), File (ficheiro externo) e Hidden (elemento escondido). As diferentes forms que foram criadas mostram isso mesmo.

3. Na *form* que diz respeito à Nacionalidade, os elementos **SELECT** e **OPTION** foram utilizados conjuntamente, o que permitiu que se criasse uma caixa de listagem (list box). As tags **<option>** e **</option>** envolvem cada uma das opções que queremos fazer constar da lista. De referir que no ecrã é visível a lista completa. Para o conseguirmos, deveremos pressionar a seta lateral ao valor que aparece por defeito (se nada se disser é o primeiro valor da lista ou, no caso de pretendermos outro valor por defeito, devemos escrever a seguir a *option* do valor pretendido a palavra *selected*). No nosso exemplo, a lista aparece e, nessa altura, é que se escolheria a opção pretendida. O atributo size=1 (valor que é atribuído se nada se disser) no elemento **SELECT**, diz respeito ao número de opções que aparecem visíveis. Por exemplo, se pretendêssemos que aparecessem visíveis duas opções, deveríamos atribuir o valor de "2" ao atributo size. Utilizando, por sua vez, o atributo **MULTIPLE** no elemento **SELECT**, todas as entradas pertencentes à lista apareceriam visíveis.

4. Na form referente aos dados pessoais, por exemplo na primeira caixa, não são visíveis todos os caracteres que foram introduzidos. Isto tem a ver com as opções **size** e **maxlength** do código (ver 1.). Ainda nesta form, para o campo "Morada" foi usado o elemento **LABEL**, que permite criar uma legenda associada a um controlo, neste caso a uma caixa de texto.

5. Na form "Sexo", foi usado o valor **RESET** que cria um botão cuja funcionalidade é a de restaurar os valores dos vários elementos da form.

6. A última form, **TEXTAREA**, funciona de forma idêntica à caixa de texto, havendo, neste caso, a possibilidade de controlarmos a dimensão da caixa. Os atributos **COLS** e **ROWS** fixam o número de colunas e de linhas da caixa, havendo também a possibilidade de fazer aparecer um valor por defeito, susceptível de ser alterado.

A Linguagem JavaScript

O JavaScript é uma linguagem que é usada dentro de um documento HTML e que permite melhorar a interactividade das páginas Web. Originalmente criada pela Netscape, trata-se de uma linguagem de *script*[13], baseada em objectos e dirigida por eventos, podendo responder às selecções e acções específicas levadas a cabo por cada utilizador. É possível responder com facilidade a eventos iniciados com o utilizador, sem que seja necessário o recurso à informação existente no servidor, tornando, portanto, esse "diálogo" utilizador-servidor mais dinâmico.

Ao longo dos Capítulos IV e V, que deverão ser visualizados on-line para que se tenha acesso à interactividade desejada, foram usados diversos *scripts*, criados por mim próprio ou retirados simplesmente de uma página algures na WWW. Por exemplo, se desejar obter um simples relógio, há vários sites onde pode adquirir o respectivo código, e "ancorá-lo" na sua página. Por exemplo, no Capítulo V deste trabalho, é apresentada, entre outras coisas, uma calculadora simples criada usando simplesmente HTML e o respectivo script de JavaScript. Ou seja, nesta situação, para realizarmos pequenos cálculos, nem necessitamos de instalar o *plugin* do MathView (ou de outro programa qualquer), pois há scripts que fazem o mesmo. Isto quer dizer que não necessitamos de recorrer a software específico para a realização de determinadas tarefas.

Para a inserção de parágrafos de programação dentro da linguagem HTML (aos quais estamos a chamar *scripts*), devemos identificar esse facto com as marcas inicial **<script>** e final **</script>**. Tais parágrafos poderão, por um lado, estar soltos ao longo do código HTML (sendo executados, neste caso, na sequência em que aparecem no documento) ou, por outro lado, aparecer acoplados a eventos (sendo, neste caso, executados quando esse evento ocorre). Para uma melhor visualização e facilidade de manutenção, é

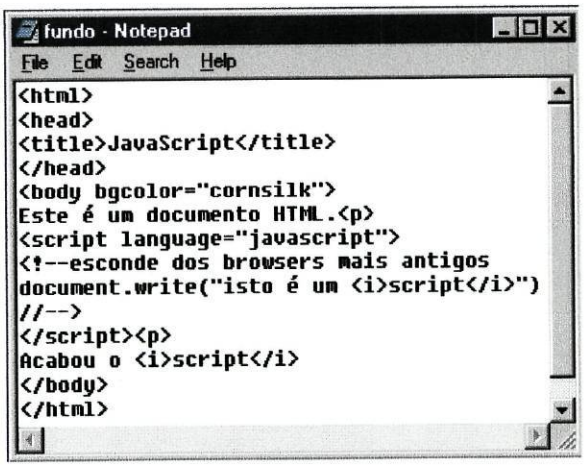
---

[13] "Pedaços de Programa", não compilados mas interpretados linha a linha, destinados à execução de determinadas tarefas (F. Stephen, 1997).

recomendável, que todo o *script* seja escrito no início do documento, através da criação de funções a serem invocadas quando tal for necessário.

Um primeiro exemplo





É evidente que este *script* não acrescentou nada ao documento HTML. De qualquer modo, aparecem retratadas as tags **<script>** e **</script>**, bem como um dos comandos mais usados na linguagem JavaScript – ***document.write()***. De referir ainda o facto dos *scripts* em JavaScript não serem interpretados pela totalidade dos browsers existentes, uma vez que apenas a partir do Netscape Navigator 2.0 e do Microsoft Internet Explorer 3.0 é que tal passou a ser possível. Contudo, tais versões desses dois browsers não são praticamente usadas por ninguém, o que faz com que não haja quaisquer preocupações sobre essa matéria.

Comentários no código JavaScript

Para que os *scripts* não interfiram nos browsers anteriores, utilizaram-se no nosso exemplo, frases incluídas entre as marcas **<!--** e **-->**, que fazem alusão a uma mensagem que justifique o aspecto menos feliz do documento carregado. Um último comentário para referir que nesses browsers mais antigos, o output não contemplaria a linha "isto é um *script*".

É possível também a introdução de comentários, que não são lidos pelo browser em circunstância alguma mas que, muitas vezes, tornam mais fácil e lógica a leitura do código que estamos a elaborar. Para tal, deverá utilizar-se a seguinte sintaxe: **//isto é um comentário**.

Operadores lógicos

Os operadores lógicos são utilizados em comandos condicionais do tipo IF, FOR e WHILE. Os mais importantes são:

= = Igual
! = Diferente
> Maior
< Menor
> = Maior ou igual
< = Menor ou igual
& & E
| | Ou

Operadores matemáticos

Os operadores matemáticos, são utilizados em cálculos, referências de indexadores e manuseio de strings. Os mais importantes são:

\+ adição de valor e concatenação de strings
\- subtracção de valores
* multiplicação de valores
/ divisão de valores
% resto de uma divisão (ex: 15 % 4 retorna 3)
+ = soma o operando da esquerda ao da direita e atribui o resultado ao operando da direita (ex: x+=y é o mesmo que x=x+y)
- = subtrai ao operador da esquerda o da direita e atribui o resultado ao operador da direita (ex: x-=Y é o mesmo que x=x-y)

* = multiplica o operando da direita pelo da esquerda e atribui o resultado ao operando da direita (ex: x*=y é o mesmo que x=x*y)

/= divide o operando da esquerda pelo da direita e atribui o resultado ao operando da direita (ex: x/=y é o mesmo que x=x/y)

x + + devolve o valor de x adicionado de uma unidade

x - - devolve o valor de x subtraído de uma unidade

- x devolve o simétrico de x

Operadores condicionais

Os operadores condicionais são comandos cuja veracidade (ou não) condicionam a execução de determinada tarefa. Destacam-se os seguintes operadores condicionais:

1. **IF**

   if (condição)
   {acção para a condição satisfeita}
   else
   {acção da condição não satisfeita}

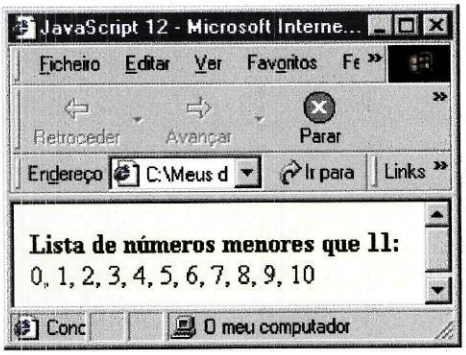
2. **FOR**

   for(valor inicial;criação da variável de controle;)
   [condição;]
   [incremento da variável de controle]
   {acção}

3. **WHILE**

   While(condição)
   {acção}

No exemplo seguinte, utiliza-se o operador condicional **FOR** para "exportar" uma lista de números menores que um determinado número (neste caso de 10).




```
<html>
<head><title>JavaScript 12</title></head>
<body bgcolor="cornsilk">
<b>Lista de números menores que 11:</b><br>
<script language="JavaScript">
for(x=0;x<10;x++)
document.write(x+", ")
</script>10</body></html>
```

Do código respectivo, de destacar apenas o facto do 10 ser digitado fora da rotina, para que se possa parar com a exportação da vírgula, escrita após cada número.



No exemplo do lado, é utilizado o operador condicional **IF**. O que faz esse script é um teste àquilo que o utilizador escreveu na caixa de texto, através da função "teste(nome)". Pela análise do código respectivo, que se apresenta de seguida, há a destacar os seguintes aspectos:

```
<html>
<head>
<title>JavaScript 2</title>
<script language="JavaScript">
function teste(nome)
{
  if (nome.caixa.value =="" ||
     nome.caixa.value.indexOf('@',0) ==-1)
  alert("E-mail escrito erradamente!");
  else alert ("E-mail escrito correctamente!");
}
</script>
</head>
<body bgcolor="cornsilk">
Escreva o seu e-mail na caixa:
<form>
<input type="text" name="caixa">
<br>
<input type="button" name="botão" value="clique para testar" onclick="teste(this.form)">
</form>
</body>
</html>
```

1. A função **teste(nome)** testa se o que aparece na caixa é uma sequência vazia (através da expressão **nome.caixa.value=""**) ou (operador lógico "| |") se não foi digitado o símbolo @ (através da expressão **nome.caixa.value.indexOf('@',0) = = - 1**).

2. Caso a primeira das situações descritas no ponto anterior seja verdadeira, uma instrução do tipo ***alert(...)*** permite abrir uma caixa de diálogo que contém a mensagem ("E-mail escrito erradamente") ou, caso contrário, a mensagem ("E-mail escrito correctamente").

3. Apesar de se ter apresentado já o exemplo de uma função "teste(nome)" e de um evento "onclick", estas questões serão abordadas de uma forma mais pormenorizada na sessão que diz respeito aos eventos.

Nota:
A principal diferença entre as instruções **IF** e **WHILE**, reside no facto da primeira ser executada apenas uma vez, enquanto que a segunda poderá ser executada mais do que uma vez, até que termine o ciclo respectivo ou se utilize a instrução break. Esta instrução é geralmente utilizada quando estamos perante um ciclo infinito. No ciclo **WHILE**, poderá ser também utilizada a expressão *continue*, forçando a ida do controlo do fluxo para a instrução de condição do ciclo, não executando as restantes instruções.

Criação de variáveis

Uma variável é criada automaticamente, pela simples associação de valores da mesma. De facto, não existe forma de especificar que determinada variável representa uma sequência de caracteres, um número inteiro ou um número decimal. O tipo de variável é especificado pelo valor inicial que lhe é atribuído.

As variáveis podem ser locais ou globais, conforme são criadas, respectivamente, dentro ou fora de uma função. As variáveis locais são, portanto, referenciáveis apenas dentro da função a que dizem respeito,

enquanto que as variáveis globais podem ser referenciadas em qualquer parte do documento. A sintaxe para a declaração de uma variável é dada por:

var nomedavariável=valorinicial.

No exemplo seguinte, aparece retratado um exemplo de uma variável (**var cor = document.teste.caixa.value**) que representa o conteúdo da caixa de texto que aparece na imagem: ***document*** representa que se está a referir a um objecto que está no próprio documento, ***teste*** é o nome dado à form, ***caixa*** é o nome dado ao objecto da form (neste caso a caixa de texto) e ***value*** significa o valor do objecto que se irá escrever dentro do objecto.






```
<html>
<head>
<title>JavaScript 3</title>
<script language="JavaScript">
function cores()
{
var cor=document.teste.caixa.value
  if(cor=='black')
{
  alert("Esta cor de fundo faz desaparecer o seu texto! Escolha outra cor!!!");
  return;
}
document.bgColor=cor
}
</script>
</head>
<body bgcolor="cornsilk">
Este é o seu texto!!<br>
Digite na caixa seguinte uma cor!!
<form name="teste">
<input type="text" value="" name="caixa" size="15" onfocus="alert('digite as cores em Inglês')">
<br>
<input type="button" value="clique para trocar de cor" onclick="cores()">
</form></body></html>
```

Pela análise do código referente às imagens anteriores, podemos verificar a existência de uma janela que se abre quando se tenta posicionar o cursor dentro da caixa de texto, efeito conseguido através da instrução **<input type="text" value="" name="caixa" size="15" onfocus="alert('digite as cores em Inglês')">**. Essa caixa de texto tem como finalidade avisar o utilizador de que as cores terão que ser introduzidas em inglês. Por outro lado, quando o utilizador introduz a cor preta (Black), abre-se uma outra caixa de texto que refere a impossibilidade de opção dessa cor, na medida em que iria fazer desaparecer o texto, escrito também a preto.

Eventos

Por vezes, efectuar determinada acção por parte do utilizador provoca aquilo a se chamam os eventos. Por exemplo, o pressionar um botão ou colocar o ponteiro do rato sobre um link, poderá dar origem a determinada acção. Já foi visto quando se falou da linguagem HTML que já é possível, através da utilização de formulários, por exemplo, aceitar dados (com caixas de texto, botões e selecções de múltipla escolha), tornando a página um autêntico impresso electrónico. O utilizador preenche as diversas respostas e submete a página.

Usando apenas código HTML, poderemos estar perante um processo lento, uma vez que a submissão dos dados implica enviar essa página para um URL algures na Rede, ser lá processada e depois ser enviada a resposta no sentido inverso. Ora, através da utilização do JavaScript, é possível localizar na máquina do utilizador a maior parte de todo este processo, executando a validação da página no próprio browser do utilizador. É possível, por exemplo, obrigar o utilizador a preencher apenas campos que estejam errados, evitando o preenchimento, de novo, da totalidade da página. Mas vejamos alguns exemplos relacionados com a utilização de eventos.

No exemplo seguinte, cuja acção é idêntica à que foi apresentada no exemplo



anterior, pressionar o botão, dá origem ao aparecimento da janela. Chama-se a esse evento um Onclick (válido para os objectos **BUTTON**, **CHECKBOX**, **RADIO**, **LINK**, **RESET** e **SUBMIT**). O código para conseguir tal acção é o seguinte:

```
<form>
<input type="button" value="clique aqui" onClick="alert('clicou no botão')">
</form>
```

Na realidade, já se falou de forms anteriormente mas, a novidade está no *OnClick="alert('clicou no botão')"* na marca **<input>**. O atributo "value" diz respeito ao que aparecerá inscrito no próprio botão e o que aparece entre parêntesis será o que constará na janela que se abrirá com o pressionar do botão. De notar que, neste caso, não foi preciso utilizar a marca **<script>**, mas antes foi aproveitado o elemento "form" que suporta scripts para inserir novos atributos de scripts.

Outros eventos, para além do Onclick poderão ocorrer durante a execução do sistema. De seguida apresenta-se uma lista de alguns desses eventos, bem como os momentos em que os mesmos podem ocorrer.

1. onload – Ocorre no carregamento do documento (no BODY);
2. onunload – Ocorre na saída do documento (no BODY);
3. onchange – Ocorre quando o objecto perde o focus e houve mudança de conteúdo (válido para os objectos TEXT, SELECT e TEXTAREA);
4. onblur – Ocorre quando o objecto perde o focus, independentemente de ter havido mudança (válido para os objectos TEXT, SELECT e TEXTAREA);
5. onfocus – Ocorre quando o objecto recebe o focus[14] (válido para os objectos TEXT, SELECT e TEXTAREA);

[14] Por exemplo, se o utilizador estiver a inserir dados numa caixa de texto, diz-se que esta ganha o focus.

6. onmouseover – Ocorre quando o ponteiro do rato passa sobre o objecto (válido apenas para links);
7. onselect – Ocorre quando o objecto é seleccionado (válido para os objectos TEXT e TEXTAREA);
8. onsubmit – Ocorre quando um botão tipo SUBMIT recebe um clique (válido apenas para o form).

Vejamos uma situação em que se utiliza um dos eventos anteriores, por exemplo, o Onload. Quando se carrega uma determinada página em que, no BODY, é definido o evento Onload, poderemos associar a esse evento a execução de determinada função, no nosso caso a função **abertura( )**. Esta função abre uma caixa de texto com a inscrição de uma mensagem, definida previamente. A figura ao lado, retrata essa situação, em que é visível, por um lado, a página totalmente carregada e, por outro, a caixa de diálogo a que se fez referência. O código necessário à realização deste exemplo é o que de seguida se apresenta.




```
<html>
<head>
<title>JavaScript_14</title>
<script language="javascript">
function abertura()
{
window.alert("abre-se esta caixa de diálogo");
}
</script>
</head>
<body Onload="abertura()" bgcolor="cornsilk" >
Ao ser totalmente carregada esta página, aparecerá uma caixa de diálogo.
</body>
</html>
```

Funções

Geralmente as funções aparecem definidas entre as marcas **<head>** e **</head>**, sendo, por isso, carregadas antes que o utilizador possa desenvolver

qualquer acção relacionada com essa função. Uma função é um bloco de instruções ao qual é atribuído um nome que, quando é invocado ao longo do código, essa função é chamada e o seu bloco de instruções executado.

A utilização de funções poderá trazer algumas vantagens. Uma das mais importantes é a de termos um bloco de texto disponível a ser invocado, uma ou várias vezes no mesmo documento, sem que tenhamos que escrever a mesma função mais do que uma vez.

No exemplo seguinte, cujo código também se apresenta, podemos fazer accionar um botão, por exemplo, do mesmo género do apresentado no exemplo anterior, utilizando a definição de uma função. Neste exemplo, pressionar o botão apresentado na tela, acciona uma função que abrirá uma caixa de *alert* com uma mensagem definida pelo programador.




```
<html>
<head><title>JavaScript</title>
<script language="JavaScript">
function clicar()
{alert("Olá");}
</script>
</head>
<body bgcolor="cornsilk">
Este é um documento HTML.<p>
<form><input type="button" name="botão 2"
value="agora aqui" onclick="clicar()"></form>
</body></html>
```

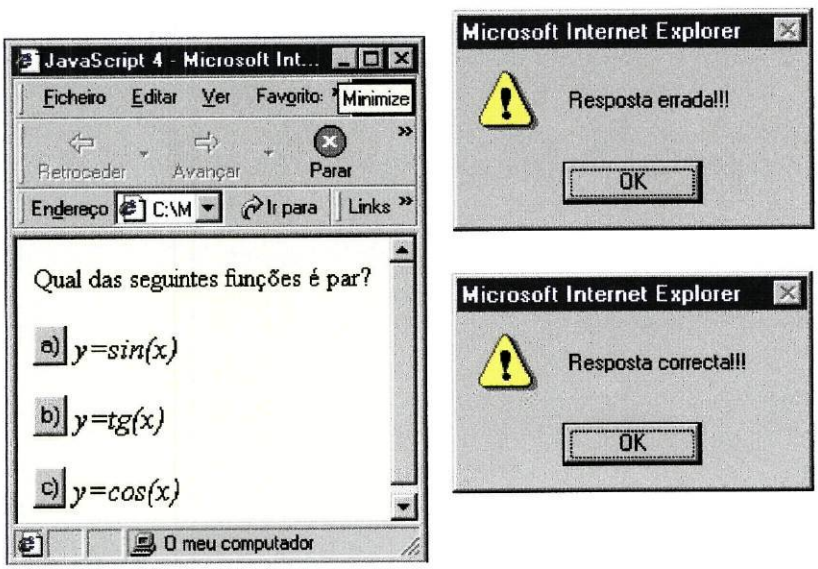
De seguida, apresenta-se um exemplo de um documento HTML que apresenta um formulário em que o utilizador poderá introduzir o seu nome. Depois de o fazer, ao premir a tecla "Enter" (ou simplesmente quando se sai do formulário) chamará a função que foi definida, aparecendo uma janela com aquilo que foi introduzido na caixa de texto, mais alguma coisa pré-definida.



```
<html>
<head>
<title>JavaScript</title>
<script language="JavaScript">
function nome(utiliz){
alert("Olá, "+ utiliz+"!!");}
</script>
</head>
<body bgcolor="cornsilk">
Este é um documento HTML.<br>
Escreva o seu nome na caixa:
<form>
<input type="text" name="cx"  onBlur="nome(this.value)" value="">
</form></body></html>
```

O "onBlur" informa o utilizador sobre qual a função que será chamada quando se coloca alguma coisa no formulário. A expressão "**this.value**" no comando "nome(this.value)" representa o valor colocado pelo utilizador no formulário.

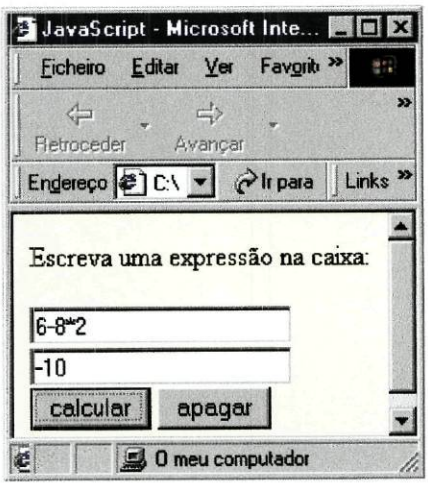
O exemplo seguinte retrata duas funções que serão activadas conforme o botão pressionado pelo utilizador. Se a resposta dada à questão apresentada for errada, é activada uma função que abre uma janela que refere "Resposta errada!!!". Caso contrário, se a resposta à questão formulada for correcta, é activada uma outra função, dando origem à abertura de outra janela que refere "Resposta correcta!!!". Poderá ser uma boa forma de apresentar aos alunos uma ficha de trabalho de escolha múltipla.



O código HTML referente ao exemplo anterior é que de seguida se apresenta.

```
<html>
<head>
<title>JavaScript 4</title>
<script language="JavaScript">
function escolhaerrada()
{
 alert("Resposta errada!!!");
}
function escolhacerta()
{
 alert("Resposta correcta!!!");
}
</script>
</head>
<body bgcolor="cornsilk">
Qual das seguintes funções é par?<br>
<form>
<input type="button" name="botao1" value="a)" onclick="escolhaerrada()"><font size=4>
<i>y=sin(x)</i></font><p>
<input type="button" name="botao2" value="b)" onclick="escolhaerrada()"><font size=4>
<i>y=tg(x)</i></font><p>
<input type="button" name="botao3" value="c)" onclick="escolhacerta()"><font size=4>
<i>y=cos(x)</i></font>
</form>
</body>
</html>
```

Um dos aspectos em que a linguagem JavaScript se mostra de maior utilidade é, sem dúvida, o facto de serem possíveis algumas manipulações matemáticas com grande facilidade. A função **eval()** que calcula o conteúdo de uma string, é um exemplo dessas funções. Vejamos um exemplo em que é utilizada essa função.




```
<html>
<head>
<title>JavaScript</title>
<script language="JavaScript">
function calc(obj)
{obj.resultado.value=eval(obj.caixa1.value)}
</script>
</head>
<body bgcolor="cornsilk">
Escreva uma expressão na caixa:
<form>
<input type="text" name="caixa1"><br>
<input type="text" name="resultado"><br>
<input type="button" name="botao" value="calcular"
onclick="calc(this.form)">
<input type="button" name="botao2" value="apagar"
 onclick="reset(this.form)">
 </form></body></html>
```

Neste exemplo, uma caixa de texto diz respeito à introdução dos dados ("caixa1") e outra caixa de texto diz respeito aos resultados ("resultado"). Quando se pressiona sobre o botão "calcular", através da instrução **<input type="button" name="botao" value="calcular" onclick="calc(this.form)">** é activada a função **calc(obj)**. No código que é visível na imagem anterior, a linha ***"obj.resultado.value=eval(obj.caixa1.value)"*** quer dizer que o valor (**"value"**) do (**"obj"**) que aparecerá na caixa de texto **resultado** será igual ao desenvolvimento numérico, através da função **eval()**, do valor (**"value"**) do (**"obj"**) que foi introduzido na caixa de texto **caixa1**. De destacar ainda a utilização da função **reset()**, cuja acção consiste em apagar o conteúdo das duas caixas de texto existentes na form, pressionando no botão respectivo ("apagar").

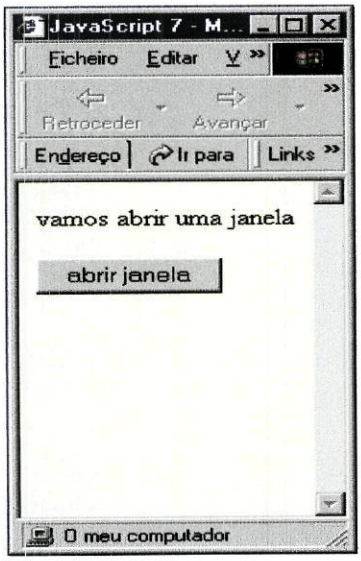
A Matemática no JavaScript

É possível efectuar determinados cálculos numéricos, sem que se tenha de entrar em contacto com o servidor, havendo ganhos de tempo. As funções seguintes, constituem alguns exemplos do que se poderá fazer neste domínio, usando JavaScript.

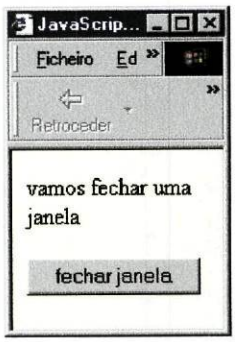
- math.abs(número) – retorna o valor absoluto do número;
- math.ceil(número) – retorna o próximo valor inteiro maior que o número;
- math.floor(número) – retorna o valor inteiro menor ou igual ao número;
- math.round(número) – retorna o valor arredondado às unidades do número;
- math.pow(base,expoente) – retorna o cálculo da potência respectiva;
- math.random() – gera números aleatórios;
- math.max(número1,número2) – retorna o maior de dois números;
- math.min(número1,número2) – retorna o menor de dois números;
- math.sqrt(número) – retorna a raíz quadrada do número;
- math.sin(número) – retorna o seno do ângulo número (em radianos);
- math.cos(número) – retorna o cosseno do ângulo número (em radianos);
- math.tan(número) – retorna a tangente do ângulo número (em radianos);
- math.asin(número) – retorna o arco seno do ângulo número (em radianos);
- math.acos(número) – retorna o arco cosseno do ângulo número (em radianos);

- math.atan(número) – retorna o arco tangente do ângulo de um número (em radianos);
- math.pi – retorna o valor de π;
- math.log(número) – retorna o logaritmo natural de um número;
- math.exp(número) – retorna a exponencial, base E, de um número;
- math.E – retorna a base dos logaritmos naturais;
- math.LN2 – retorna o valor do logaritmo de 2 (base E);
- math.LOG2E – retorna logaritmo base 2 de E;
- math.LN10 – retorna o valor do logaritmo de 10;
- math.LOG10E – retorna logaritmo base 10 de E.

Abrir e fechar janelas



Criar janelas é uma das principais características do JavaScript. Isto porque, através de um clique sobre um botão, poderemos accionar um evento que consiste na abertura de uma janela com algumas propriedades definidas pelo utilizador. No exemplo seguinte, a imagem ao lado representa o documento principal. Ao pressionarmos o botão "abrir janela" damos origem à figura seguinte. Por sua vez, ao clicarmos no botão "fechar janela", fechamos a respectiva janela.



De notar que a janela que se abre, também carrega um documento HTML, independente do anterior, conforme poderá ser possível visualizar no código que de seguida se apresenta:

```
<html>
<head>
<title>JavaScript 7</title>
</head>
<body bgcolor="cornsilk">
vamos abrir uma janela
<form>
<input type="button" value="abrir janela"
onclick="window.open('jscript8.html','','toolbar=yes,
```

```
location=no, directories=no, status=no, menubar=yes,
scrollbars=no,resizable=no,width=140,height=100')">
</form>
</body>
</html>
```

Pela análise do código anterior, poderemos referir o seguinte:

1. O evento **Onclick()**, neste caso, dá origem à abertura de uma janela, cujo conteúdo corresponde ao ficheiro "jscript8.html" (apresentaremos o respectivo código mais à frente).

2. A função **window.open()**, faz com que se abra uma janela, neste caso através do clique sobre um botão, havendo a possibilidade de se definirem algumas características da janela a abrir. A instrução necessária a tal é : **<input type="button" value="abrir janela" onclick="window.open('jscript8.html', ','toolbar=yes, location=no, directories=no, status=no, menubar=yes, scrollbars=no,resizable=no,width=140, height=100')">**.

3. As características da janela, referidas no ponto anterior, são definidas por opções que podem tomar os valores de "***yes***" ou "***no***". Tais opções, caso estejam na posição "***yes***" permitem o seguinte: **toolbar** (cria uma barra de ferramentas tipo "back", "forward", etc); **location** (abre a barra de location do browser); **directories** (abre a barra de ferramentas tipo "Links", "Best of the Web", Channel Guide, etc); **status** (abre uma barra de status no rodapé da janela); **menubar** (cria uma barra de menu do tipo "File", "Edit", etc); **scrollbars** (abre barras de rolamento vertical e horizontal mesmo que tal não seja necessário); **resizable** (permite ao utilizador redimencionar a janela). As opções **Width** e **Height** permitem especificar, respectivamente, a largura e a altura da janela, em pixels.

O código do documento contido na janela que se abriu, é o seguinte:

```
<html>
<head>
<title>JavaScript 8</title>
</head>
<body bgcolor="cornsilk">
```

```
vamos fechar uma janela
<form>
<input type="button" value="fechar janela" onclick="window.close()">
</form>
</body>
</html>
```

De referir ainda que a função **window.close()** é activada apertando o botão ("fechar janela") que aparece na imagem em baixo, fechando a janela. A instrução necessária a tal operação é: **<input type="button" value="fechar janela" onclick="window.close()">**.



A figura ao lado, diz respeito à janela que se abre depois de se pressionar o mesmo botão "abrir janela" do exemplo anterior, desta vez com o valor "***no***" nas opções "**toolbar**" e "**menubar**".

Cadeias de caracteres (strings)

Os objectos de *string* e os seus métodos permitem mostrar os resultados dos cálculos efectuados, por exemplo, através de funções matemáticas. As *strings* têm apenas uma propriedade (lenght) e que diz respeito ao número de caracteres nela contidos. A sintaxe necessária à obtenção de tal valor é a seguinte: **nomedastring.lenght**.

De seguida apresenta-se um exemplo em que é utilizada a propriedade que permite a obtenção do comprimento de uma dada *string*.



Na figura ao lado, está contemplada uma página em que na primeira linha aparece uma *string*, neste caso "o meu nome". Na segunda linha, determina-se o comprimento dessa mesma *string*, contando com os espaços que aparecem entre as diferentes palavras. O código necessário para levar por diante essa situação é o que de seguida se apresenta.

```
<html>
<head>
<title>JavaScript_15</title>
<script language="javascript">
var meunome="Luís Miguel dos Santos Ferreira"
var medida=meunome.length;
document.write(meunome+"<br>");
document.write("O meu nome tem "+medida+" caracteres");
</script>
</head>
<body bgcolor="cornsilk" >
</body>
</html>
```

Existem métodos para a utilização de *strings* e que proporcionam o controlo sobre como as informações são apresentadas, dos quais se destacam os de atributo, minúsculas/maiúsculas e manipulação.

Quanto ao método de atributo, especifica a forma em que o texto normal aparecerá na página. As funções mais importantes que poderão ser utilizadas são:

1. blink() – faz com que o texto pisque (sintaxe: nomedastring.blink();)
2. bold() – faz com que o texto apareça a negrito (sintaxe: nomedastring.bold();)
3. fontcolor() – permite mudar a cor de uma *string* (sintaxe: nomedastring.fontcolor(corRGB[15]);)
4. italics() – faz com que o texto apareça a itálico (sintaxe: nomedastring.italics();)
5. strike() – faz com que o texto apareça rasurado (sintaxe: nomedastring.strike();)

Relativamente ao método Maiúsculas/Minúsculas, permite alterar texto de maiúsculas para minúsculas e vice-versa. Destacam-se as seguintes funções:

1. toLowerCase() – traduz todos os caracteres de uma *string* em minúsculas (sintaxe: nomedastring.toLowercas();)
2. toUpperCase() - traduz todos os caracteres de uma *string* em maiúsculas (sintaxe: nomedastring.toUppercas();)

---

[15] A representação desta "corRGB" é qualquer uma das formas válidas para a representação de cores. Pode ser através do seu nome (red, blue, brown, etc) ou através de uma conjunto de três pares de dígitos hexadecimais que representam, por esta ordem, os valores vermelho, azul e verde.

No que concerne ao método de manipulação, fornece diversas maneiras de procurar caracteres e extraí-los de uma *string*. Das funções mais importantes destacamos as seguintes:

1. charAt() – retorna um carácter específico de uma *string* (sintaxe: nomedastring.charAt(índice); em que índice é um número entre 0 e N-1, onde N é o comprimento da *string*)
2. substring() – retorna uma sequência de caracteres de uma *string* (sintaxe: nomedastring.substring(índice1,índice2); em que índice1 e índice2 são números entre 0 e N-1, onde N é o comprimento da *string*)
3. indexOf() – retorna um valor numérico referente à localização da primeira ocorrência (da esquerda para a direita) de um determinado valor (sintaxe: nomedastring.indexOf(valoraprocurar,apartirde);)
4. lastindexOf() – faz o mesmo que o anterior mas da direita para a esquerda (sintaxe: nomedastring.lastindexOf(valoraprocurar,apartirde);)

Existem ainda outros métodos de utilização das strings, cujas funções, produzem alguns efeitos interessantes. Com a aplicação de tais funções pode ser possível referenciar determinadas zonas da página, através das funções:

- anchor() – define uma âncora para o destino de um link (sintaxe: nomedastring.anchor(nomedaancora);)
- link() – poderá ser utilizado para substituir uma href de que já foi feita referência aquando da abordagem do HTML (sintaxe: textodolink.link(href);)

podemos colocar o texto em subscrito ou sobrescrito (por exemplo expoentes ou índices), através das funções:

- sub() – define um texto em subscrito (sintaxe: texto.sub();)
- sup() – define um texto em sobrescrito (sintaxe: texto.suo();)

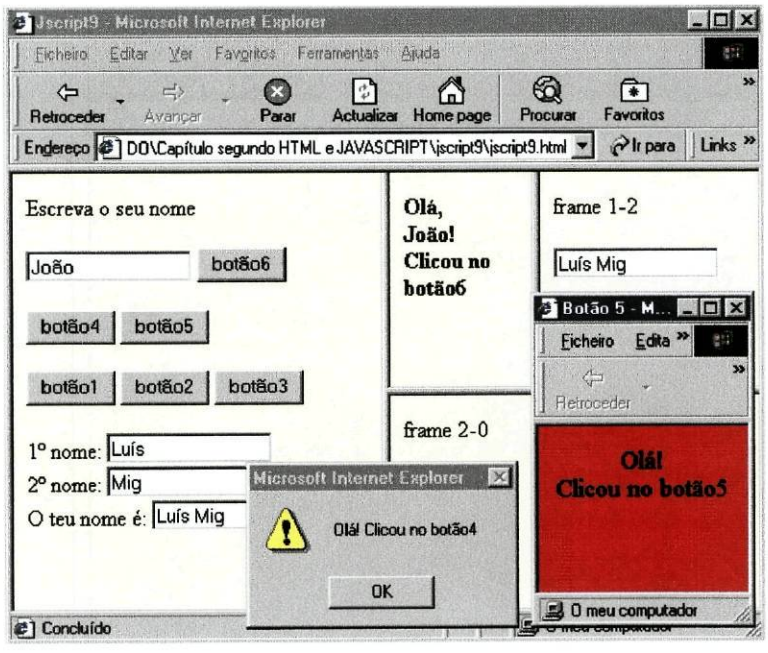
ou poderemos ainda controlar o tamanho do texto a visualizar na página, através das funções:

- big() – define um texto num tamanho superior ao tamanho normal (sintaxe: nomedastring.big();)
- fixed() – define a fonte de espaçamento do texto (sintaxe: nomedastring.fixed();)
- fontsize() – faz variar o tamanho da letra a apresentar na página (sintaxe: nomedastring.fontsize(tamanho);)
- small() – define um texto num tamanho inferior ao tamanho normal (sintaxe: nomedastring.small();)

Frames

Com a linguagem JavaScript, poderemos conseguir bons resultados ao trabalharmos com frames, uma forma que se utiliza frequentemente para dividir o ecrã num determinado número de partes, independentes umas das outras (com a possibilidade de se carregarem páginas diferentes em frames diferentes) mas havendo comunicação dinâmica entre elas.

O exemplo que se apresenta de seguida, mostra, por um lado, como se efectuam tais divisões do ecrã e, por outro, diferentes formas de comunicação entre as diferentes frames que foram criadas.



O documento principal tem o código que de seguida se apresenta e, sobre ele, não nos iremos agora debruçar uma vez que já o fizemos anteriormente, quando falámos de frames na parte de HTML (pág. 43).

```
<html>
<head>
<title>Jscript9</title>
</head>
<frameset cols="50%,50%">
 <frame src="jscript9_esq.html" name="pagesq">
  <frameset rows="50%,50%">
   <frameset cols="40%,60%">
    <frame src="jscript9_dir1.html" name="pagdir">
    <frame src="jscript9_dir3.html" name="pagdir3">
   </frameset>
  <frame src="jscript9_dir2.html" name="pagdir2">
 </frameset>
</frameset>
</html>
```

Dos resultados obtidos através das funções definidas no nosso documento, há a destacar os seguintes:

1. Depois de escrevermos alguma coisa na primeira caixa, do clique sobre o "botão6" resulta o aparecimento na frame imediatamente à direita "frame1-1" uma mensagem que faz alusão ao que foi digitado na própria caixa de texto. A instrução capaz de produzir tal efeito foi a seguinte: **<input type="button" name= "botão6" value= "botão6"onClick= "mostrartexto(document.form1.cx1.value)">**, o que provocou que se accionasse a função **mostrartexto( )**. Esta função, escreve através de instruções do tipo **caixa.write("qualquercoisa")** na frame cujo nome é "**pagdir**".

2. Quando se clica no "botão4", através da instrução **<input type="button" value="botão4" onclick="alert('Olá! Clicou no botão4')">**, consegue-se a abertura de uma caixa de diálogo, com uma mensagem pré-definida. Neste caso, a mensagem é "Olá! Clicou no botão4".

3. O "botão5" abre uma janela com um tamanho e propriedades definidas pelo utilizador, através da instrução **<input type=button value="botão5"**

onClick="abrejanela()">. De salientar apenas que a função **abrejanela()** está definida no HEAD do documento.

4. Os botões "1" e "2" colocam as mensagens "Clicou no botão1" e "Clicou no botão2" nas frames 1-1 e 2-0, respectivamente. A instrução necessária a tal operação é, no caso do "botão2", **<input type="button" value="botão2" onclick="parent.pagdir2.ola2()">**, fazendo accionar a função **ola2()** definida no HEAD do documento que constitui a respectiva frame destino.

5. O "botão3" coloca a mensagem "Clicou no botão3" na frame 1-1, através da instrução **<input type="button" value="botão3" onclick="ola3(document.coisas)">**. A diferença entre este botão e os botões 1 e 2 prende-se com a forma de definir as funções. Nos botões 1 e 2 as funções accionadas pelo clique no botão respectivo, estão definidas na frame de destino, enquanto que no botão 3, a função está definida na própria frame em que se encontra o botão. Em termos de resultados práticos, há a referir o seguinte: depois de clicarmos no botão 3, na frame 1-1 é escrita a mensagem "Clicou no botão 3". Desta forma, se agora clicarmos no botão 1, nada acontece uma vez que a frame de destino (1-1) se encontra modificada. Para termos acesso ao botão 1, teríamos que retroceder na "toolbar" do browser e voltar a clicar no botão1. Por outro lado, o mesmo não acontece se clicarmos primeiro no botão 1 e só depois no botão 3, uma vez que a função inerente ao botão 3 não depende da frame destino.

6. Ao digitarmos o 1º e 2º nomes nas caixas respectivas, aparecem escritos esses dois nomes na caixa do "nome completo" na mesma frame e na caixa existente na frame 1-2. Este resultado consegue-se através da função **"adicionar()"** definida no HEAD do documento, conforme se pode verificar no código da *form* que foi criada para o efeito e que, de seguida, se apresenta:

```
<form name="form2" method="POST">
1º nome: <input name="first" type="text" size="15" Onfocus="adicionar()"><br>
2º nome: <input name="last" type="text" size="15" Onfocus="adicionar()"><br>
O teu nome é: <input name="full" type="text" size="15">
</form>
```

Do código da frame mais à esquerda (que aparece representado de seguida) e que contém todos os botões que aparecem na imagem anterior, há alguns aspectos importantes que gostaríamos de salientar.

```
<html>
<head>
<title>página esquerda</title>
<script language="JavaScript">

function mostrartexto(textodacaixa)
{
 var caixa = parent.frames.pagdir.document;
 caixa.open();
 caixa.write("<html><head></head><body bgcolor='cornsilk'>");
 caixa.write("<h4>");
 caixa.write("Olá, "+textodacaixa+'!');
 caixa.write("<br>Clicou no botão6");
 caixa.write("</h4></body></html>");
 caixa.close();
}

function abrejanela()
{
myWin= open("", "displayWindow", "width=150, height=50, status=yes, toolbar=yes,
menubar=yes");
myWin.document.open();
myWin.document.write("<html><head><title>Botão 5");
myWin.document.write("</title></head><body bgcolor='red'>");
myWin.document.write("<center><font size=+1>");
myWin.document.write("Olá!");
myWin.document.write("<br>Clicou no botão5");
myWin.document.write("</font></center>");
myWin.document.write("</body></html>");
myWin.document.close();
}

function ola3(coisas)
{
var bt=parent.frames.pagdir.document;
bt.open();
bt.write("<html><head></head><body bgcolor='cornsilk'>");
bt.write("<h4>");
bt.write("Clicou no botão3");
bt.write("</h4></body></html>");
bt.close();
}

function adicionar()
{
document.form2.full.value = document.form2.first.value + " "
+ document.form2.last.value;
```

```
 parent.pagdir3.cx.full2.value = document.form2.first.value + " "
+ document.form2.last.value;
setTimeout("adicionar()",100)
}
</script>
</head>
<body bgcolor="cornsilk">

Escreva o seu nome
<form name="form1">
<input type="text" size="15" name="cx1" value="">
<input type="button" name="botão6" value="botão6"
 onClick="mostrartexto(document.form1.cx1.value)">
</p>
<p>

<form name="botoes2">
<input type="button" value="botão4" onclick="alert('Olá! Clicou no botão4')">
<input type=button value="botão5" onClick="abrejanela()">
</form>
<p>

<form name="botoes">
<input type="button" value="botão1" onclick="parent.pagdir.ola()">
<input type="button" value="botão2" onclick="parent.pagdir2.ola2()">
<input type="button" value="botão3" onclick="ola3(document.coisas)">
</form>
<p>

<form name="form2" method="POST">
1º nome: <input name="first" type="text" size="15" Onfocus="adicionar()"><br>
2º nome: <input name="last" type="text" size="15" Onfocus="adicionar()"><br>
O teu nome é: <input name="full" type="text" size="15">
</form>

</body>
</html>
```

De entre eles destacamos os seguintes:

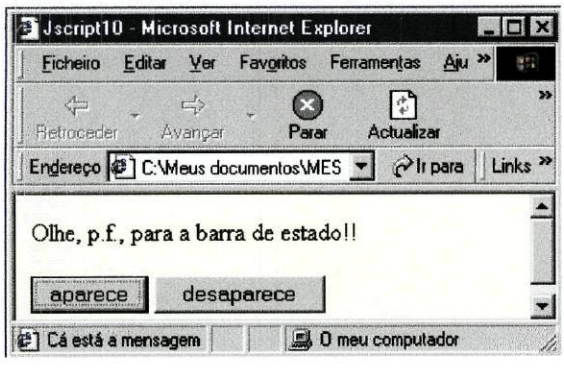
1. A função "**adicionar()**" faz referência a duas caixas de texto de destino: a que se situa na mesma frame e a que se situa na frame1-2. As instruções que aparecem à esquerda do sinal "=" (**document.form2.full.value** e **parente.pagdir3.cx.full2.value**) dizem respeito a essas mesmas caixas de texto. No primeiro caso "document" quer dizer que o resultado irá acontecer no mesmo documento, "form2" é o nome da form que contém a caixa de texto de destino, "full" é o nome da caixa de texto e "value" diz respeito ao conteúdo da caixa.

2. Aparece definida a função **setTimeout("adicionar()",100)** na instrução que definiu, no HEAD, a função "**adicionar()**" e que acciona esta mesma função ao fim de 100 milisegundos. Por conseguinte, a função "**adicionar()**" é executada de 100 em 100 milisegundos.

3. Nas form's destino de nomes "full" no caso da frame da esquerda e "full2" no caso da frame1-2, são preenchidas automaticamente, logo que se escreve alguma coisa numa das caixas referentes à introdução dos nomes. Isto consegue-se através da utilização da função "**Onfocus=adicionar()**" definida na caixa referente aos input's, isto é, a função "**adicionar()**" é accionada logo que o cursor entra na caixa de texto referente à introdução do 1º nome.

Controlo das mensagens na barra de estado

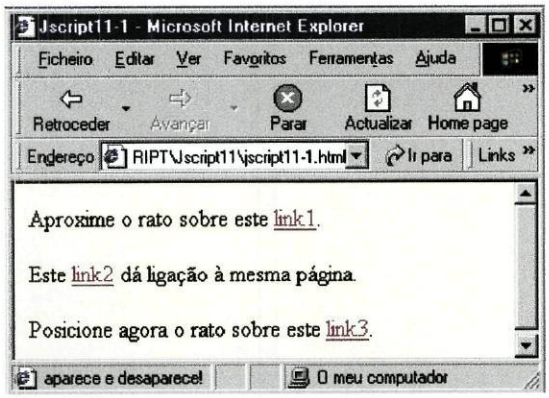
Para fazermos aparecer algumas mensagens na barra de estado que aparece na parte inferior (à esquerda) do browser, podemos também usar o JavaScript.



O exemplo ao lado, mostra dois botões que activam e desactivam essas mensagens. Ao clicarmos no botão "aparece" fazemos aparecer o texto que é visível na barra de estado ("Cá está a mensagem"). Por sua vez, ao clicarmos no botão "desaparece", fazemos desaparecer a mesma mensagem. Isto porque se definiu no **HEAD** do documento uma função "**barra(texto)**". O código para que tal aconteça, é apresentado de seguida:

```
<html>
<head>
<title>Jscript10</title>
<script language="JavaScript">
function barra(texto)
{
window.status=texto;
}
</script>
</head>
```

```
<body bgcolor="cornsilk">
Olhe, p.f., para a barra de estado!!
<form>
<input type="button" name="sim" value="aparece" onclick="barra('Cá está a mensagem');">
<input type="button" name="nao" value="desaparece" onclick="barra('');">
</form>
</body>
</html>
```



Como se pode verificar no código anterior, este controlo da barra de estado é conseguido através da utilização da função "**barra(texto)**" que, com um clique sobre o botão respectivo, coloca a mensagem pretendida na barra de estado. De referir que, o facto de se ter usado neste exemplo uma variável ("**texto**") evita que se definam duas funções diferentes.

O exemplo retratado na imagem anterior, mostra mais algumas coisas que se podem fazer aproveitando a barra de estado. São visíveis na figura três links (link1, link2 e link3), cada um dos quais produzindo situações diferentes no momento em que se posiciona o ponteiro do rato sobre eles.

Quanto ao link1, quando o ponteiro do rato se posiciona sobre ele, na barra de estado aparece uma mensagem ("está sobre o link1") que permanece visível até que se faça o respectivo clique (ou se volte a afastar o ponteiro). Já o link2 produz o que geralmente é produzido, ou seja, é visível na barra de estado durante os momento em que o ponteiro do rato permanece sobre ele, o URL da página destino. Finalmente, quanto ao link3, quando se posiciona o ponteiro do rato sobre ele, aparece na barra de estado uma mensagem que permanece visível algum tempo, tempo esse definido pelo utilizador. Mas vejamos o respectivo código para que se percebam melhor estas situações.

```
<html>
<head>
<title>Jscript11-1</title>
<script language="JavaScript">
function desaparecer(texto)
{
```

```
window.status=texto;
setTimeout("erase()",1000);
}
function erase()
{
window.status="";
}
</script>
</head>
<body bgcolor="cornsilk">
Aproxime o rato sobre este <a href="jscript11-2.html" onmouseover="window.status='está sobre
o link1'; return true">
link1</a>.<p>
Este <a href="jscript11-2.html">link2</a> dá ligação à mesma página.<p>
Posicione agora o rato sobre este <a href="jscript11-2.html"
onMouseOver="desaparecer('aparece e desaparece!');return true;">link3</a>.
</body>
</html>
```

Da análise deste código, destacamos os seguintes aspectos:

1. Por defeito, quando não se dá qualquer instrução especial (situação referente ao link2), a mensagem que é visível na barra de estado é a do URL da página destino;

2. Se pretendermos alterar essa mensagem, basta usarmos, ao definirmos o link, a instrução **onmouseover="window.status='mensagempretendida';return true".**

3. Para conseguir o efeito do link3, temos que definir duas funções. Uma que dá origem à mensagem propriamente dita ("**desaparecer(texto)**") e que, recorrendo a uma outra função ("**erase()**") através da função **settimeout("nomedafunçãooutarefa",tempo)** permite que essa mesma mensagem deixe de ser visível ao fim de certo tempo. Esta última função, no nosso exemplo, define 1000 milisegundos (1 segundo) para que a função **erase()** seja invocada.

HTML dinâmico

Mais recentemente surgiu uma extensão do HTML (com um conjunto de novas instruções) que vieram aumentar ainda mais as possibilidades de

representação da informação na tela. O HTML dinâmico não é, portanto, mais do que a associação de folhas de estilo e de programas de scripting como o JavaScript. Poderão associar-se a páginas WWW estilos para definir a formatação de determinado bloco de texto ou ainda, certos eventos que são ocorrências diversas que se podem verificar durante a visualização dessa página por parte do cliente (por exemplo, cliques de rato ou a passagem do cursor sobre determinadas zonas da tela). Dessas novas instruções surgidas com o HTML dinâmico, destacaríamos as seguintes: DIV, SPAN, STYLE e a LINK.

A título de exemplo, no Capítulo V deste trabalho, em que são apresentados alguns exemplos de *applets* existentes para o desenvolvimento de determinados procedimentos matemáticos, no **HEAD** das várias páginas, usou-se uma instrução do tipo:

```
<style><!--a:{color:#008241}a:vlink{color:#005128} a:hover{color:#008241} --></style>
```

e que permitiu que se definissem as cores envolvidas nos links, ou seja, a cor do link, a cor do link quando este foi visitado ou quando está a ser visitado.

Também uma novidade mais recente diz respeito ao elemento <a href="xxx"> utilizado tradicionalmente quando se pretende introduzir um link para um URL ("xxx") interno ou externo à página que está a ser lida. Agora, em vez de definirmos esse URL ("xxx") poderemos definir uma instrução JavaScript. A título de exemplo, podemos referir aquilo que foi feito também, por exemplo, no Capítulo V. Nesse capítulo, assim como no Capítulo IV, optou-se pela definição de uma função para permitir a abertura de janelas. Assim, no **HEAD** das diferentes páginas, foi inserida a seguinte instrução:

```
<script language="javascript">
function abrejanela(url,name,w,h)
{
Win=window.open(url, name, 'width=' + w + ',height=' + h);
}
</script>
```

Já no **BODY**, ao longo da página, quando se pretendeu abrir uma janela para visualizar um exemplo, criou-se um (ou mais) link(s), através da seguinte instrução:

link.

Do exemplo considerado, há a destacar a função **abrejanela()** onde são definidas, genericamente, as propriedades da mesma. Neste caso, URL diz respeito à página a carregar na janela a abrir, NAME diz respeito ao nome da janela, **w** e **h** à largura e à altura da janela a abrir, respectivamente.

A Linguagem JAVA

Falar de Java não é a mesma coisa que falar em JavaScript. Enquanto que a segunda é, como já se viu, uma linguagem de *scripting* e que funciona como que uma extensão do HTML, já a primeira é uma linguagem de programação, com tudo o que isso implica em termos de liberdade de criação (Coelho P., 1998), cujos programas ("*applets*") podem ser introduzidos no interior de código HTML de páginas WWW. Quando um utilizador acede a uma página que contém uma *applet* JAVA, esta é carregada e executada naturalmente, no computador do cliente que acede a essa página. Isto faz com que a comunicação, por vezes lenta entre o cliente e o servidor WWW, se processe de uma forma mais rápida. De facto, esta linguagem de programação desenvolvida nos laboratórios da Sun Microsystems, permite muitas possibilidades interactivas e de animação.

Invocar *applets* JAVA

Não vamos desenvolver aqui, neste trabalho, a forma de construir esses pequenos programas ("*applets*"). Vamos simplesmente, falar da forma de os introduzir no código HTML, uma vez que foi isso que se fez no Capítulo V.

Nesse capítulo, não se construíram as *applets* utilizadas mas utilizaram-se algumas que existem na Net prontas a serem usadas, de uma forma gratuita, por qualquer utilizador. De referir apenas que, à semelhança do que acontece com qualquer linguagem de programação, as *applets* JAVA são ficheiros binários, escritos em código-fonte (texto), sendo necessário o uso de um compilador para gerar um programa executável. Não se trata de ficheiros propriamente executáveis mas antes de ficheiros que são executados por uma aplicação residente na máquina do cliente. As *applets* são portanto residentes no servidor WWW e podem ser importadas e executadas em qualquer computador, independentemente da sua arquitectura.

Vejamos o seguinte exemplo:




```
<HTML>
<HEAD>
<TITLE>applet's</TITLE>
</HEAD>
<BODY bgcolor=cornsilk>
<font size=4>
Introdução de <i>applet's</i> em documentos
<p>
<center>
<applet
   code    = CenterExpand.class
   width   = 200
   height  = 35>
<param name=TextCount  value=5>
<param name=text1      value="Tese de Mestrado">
<param name=text2      value="Faculdade de Ciências">
<param name=text3      value="Universidade do Porto">
<param name=text4      value="Luís Miguel Ferreira">
<param name=text5      value="O Uso da Internet no Ensino da Matemática">
<param name=HorizCenter value=true>
<param name=VertCenter  value=true>
<param name=style      value=bold>
<param name=pointsize  value=20>
<param name=AppBGColor      value="blue">
<param name=TextColor       value="#ffffff">
</applet>
</center>
Fim da <i>applet</i>
</BODY>
</HTML>
```

A imagem apresentada, retrata um documento que contém uma "*applet*" no interior do seu código, código esse também apresentado em cima. Pela sua análise, há a destacar os seguintes aspectos:

1. A utilização de uma *applet* é conseguida inserindo a tag **<applet>**, assim como um conjunto de parâmetros e atributos a ela associados. O final da *applet* é identificado pela tag **</applet>**.

2. O ficheiro "**CenterExpand.class**" (de extensão ".class"), referente ao atributo **CODE**, é o ficheiro binário resultante da compilação do programa JAVA (escrito inicialmente com extensão ".java").

3. Os atributos **WIDTH** e **HEIGHT** definem a largura e a altura iniciais da "applet", em pixels.

4. As linhas do código iniciadas pela expressão **<param name ...**, representam as definições de parâmetros adicionais da *applet*. A declaração **PARAM** pode tomar dois valores: **NAME** (que especifica um nome de um determinado atributo) e **VALUE** (que especifica o valor correspondente para o atributo respectivo). Por exemplo a instrução **<param name=TextColor value="#ffffff">** diz respeito à cor do texto que aparece na nossa *applet*.

Em resumo, os parâmetros mais importantes para que possamos introduzir uma *applet* em linguagem Java numa das nossa páginas WWW são os seguintes:

PARÂMETRO	DESCRIÇÃO
CODE	Nome do URL que identifica o ficheiro, de extensão "*.class" que contém a compilação do programa Java e que permite a sua inicialização.
CODEBASE	Nome do directório que contém o ficheiro referenciado em CODE.
ARCHIVE	Nome do arquivo compactado, com extensão "*.zip" ou "*.jar" que contém o ficheiro referenciado em CODE.
WIDTH	Define a largura da caixa da tela onde aparecerá a *applet*
HEIGHT	Define a altura da caixa da tela onde aparecerá a *applet*
HSPACE	Define os espaços livres deixados nos lados esquerdo e direito da *applet*
VSPADE	Define os espaços livres deixados nos lados superior e inferior da *applet*

Capítulo III – O Programa MATHVIEW

Este capítulo pretende dotar o utilizador do programa MathView de informação necessária ao seu manuseamento, para que se possam aproveitar, o mais possível, as suas potencialidades. Por outro lado, são dadas também algumas instruções que foram necessárias para a elaboração de alguns exemplos expostos no Capítulo IV. Assim, dão-se a conhecer os comandos mais importantes do programa e as funções mais relevantes. Aqui também são esclarecidos alguns aspectos importantes para quem, não tendo o programa disponível, pretenda usar o *plugin*[1] do MathView para a manipulação dos ficheiros criados com o programa e que estão inseridos em páginas WWW.

De MathView a LiveMath



O programa MathView foi criado em 1993 pela empresa Waterloo Maple, Inc., uma empresa que também criou o conhecido programa de álgebra computacional *Maple*. Em Junho de 1999, o produto foi vendido à empresa WebPrimitives, LLC, uma empresa pioneira na criação de Matemática interactiva e em projectos de livros interactivos de ciência. Aliás, esta venda originou a mudança de nome do produto, passando agora a nova versão do programa a chamar-se LiveMath. Para obter mais informações acerca deste novo produto, poderá fazê-lo no endereço http://www.livemath.com. Neste nosso trabalho, pelo facto de ter sido iniciado antes mesmo da saída do LiveMath, usamos o produto anterior que, de uma forma geral, se manteve inalterado em termos de conteúdo. O manuseamento do programa, a todos os níveis, inclusivamente no que toca ao uso do *Plugin*, mantém-se idêntico, pelo que, o que de seguida se expõe, mantém-se actual e facilmente adaptável ao novo produto.

[1] Programa que permite que objectos mergulhados nas páginas Web sejam reproduzidos em linha (na própria página), sem a necessidade de abrir uma aplicação externa.

As potencialidades do MathView

Ao longo deste trabalho, foram utilizadas por diversas vezes e em variadas situações, algumas das potencialidades deste programa, quer ao nível de representação gráfica e estudo analítico de funções, bem como na simples manipulação de expressões matemáticas.

Trata-se, de facto, de um programa para se trabalhar em Matemática, com grande capacidade de interface gráfica. A vantagem deste programa relativamente a outros, é a sua notação. Quer as entradas das expressões quer mesmo os resultados depois daquelas terem sido manipuladas, aparecem em notação matemática. Depois de introduzida uma expressão, podemos manipulá-la algebricamente, desenvolvê-la numericamente ou fazer aparecer o seu gráfico com uma simples operação, não sendo necessário aprender qualquer linguagem específica de programação. Por outro lado, o seu uso bastante simples, faz com que o MathView seja "uma ferramenta de exploração extremamente eficaz, para estimular os alunos a descobrirem relações matemáticas, fazerem conjecturas e verificar a sua validade" (in Brophy, 1997). O MathView poderá ser, de facto, um programa com excelentes resultados para a investigação interactiva de relações matemáticas.

A possibilidade de instalação de um simples *plugin*, permite ao utilizador a manipulação algébrica, numérica e gráfica, sem haver a necessidade de se adquirir o software. Tal *plugin* poderá ser carregado gratuitamente no endereço http://www.livemath.com e, apenas com esse ficheiro, o código inerente ao programa torna-se perceptível pelo browser fazendo com que sejam possíveis pequenas manipulações, quer algébricas quer gráficas.

Este capítulo pretende ser um pequeno guião do programa, sendo apresentadas as instruções básicas quer em termos de introdução de expressões, esboço de gráficos 2-D ou 3-D, quer mesmo as principais potencialidades do programa. Serão também apresentados "cenários" do programa para que o leitor tenha uma percepção mais próxima quanto possível do ambiente do próprio programa. Na versão electrónica deste trabalho, que

está disponível em CD-ROM ou, futuramente, no endereço http://7mares.terravista.pt/luismigferreira, a manipulação das expressões já é possível por parte do utilizador, dando, portanto, uma ideia mais real do que se pode conseguir com o MathView. Obviamente que, nesta versão em papel, tal não é possível e, portanto, torna-se menos atractiva.

As limitações do MathView

Relativamente às limitações deste programa, há a referir o facto de não ser permitida programação com operadores condicionais (do tipo "if...then...", "while") como no Javascript ou programas de Matemática como o *Maple* ou o *Mathematica*. Assim, poderão surgir, em alguns cálculos, determinados problemas de contornos difíceis. Por exemplo, quando se aborda a construção de uma recta dados dois pontos, na secção "Dois pontos definem uma recta" do Capítulo IV, quando as abcissas dos dois pontos são coincidentes, os resultados não são aqueles que se poderiam esperar. O professor terá, neste caso, que estar atento a estas limitações.

Outra limitação do MathView reside no facto de não ser possível, por exemplo, restringir a resolução de equações ao corpo dos números reais. Na abordagem desta situação, quando os coeficientes de uma equação polinomial do 2º grau dão origem a soluções complexas, o resultado aparece dessa forma. Ora, na maior parte do Ensino Básico e Secundário, em que não constam do programa os números complexos (excepto no 12º ano), isto constitui, de facto, um inconveniente para o qual o professor deverá estar alertado.

Os ficheiros do MathView

Quando procedemos à abertura do programa, um novo Notebook é apresentado, cujo nome é "**UNTITL1.THE**" (ou **UNTITLED1** no caso dos Macintosh). Estes novos documentos abrem, por defeito, um Notebook chamado **Newnote.the** (ou **New Notebook** nos Macintosh), que já contém

algumas configurações pré-definidas, bem como algumas funções, variáveis, constantes, regras de transformação de determinadas expressões, entre outros elementos. É possível alterar estas configurações e acrescentar, eventualmente, outras regras, de modo a que, quando abrimos um novo documento, este surja com as configurações que pretendemos.

Será também importante salientar que, por defeito, quando gravamos um documento criado por nós, este será gravado com a extensão "*.the". No entanto, quando pretendemos incluir esse documento numa página WWW, a extensão com que deverá ser guardado o documento deverá ser "*.thp". A este nível, será conveniente referir que, por exemplo, se pretendermos incluir numa página WWW o ficheiro "exemplo.THP", a invocação do ficheiro deverá ser feita nos seguintes termos:

```
<embed src="exemplo.THP" align="baseline" border="0" width="330" height="400">
```

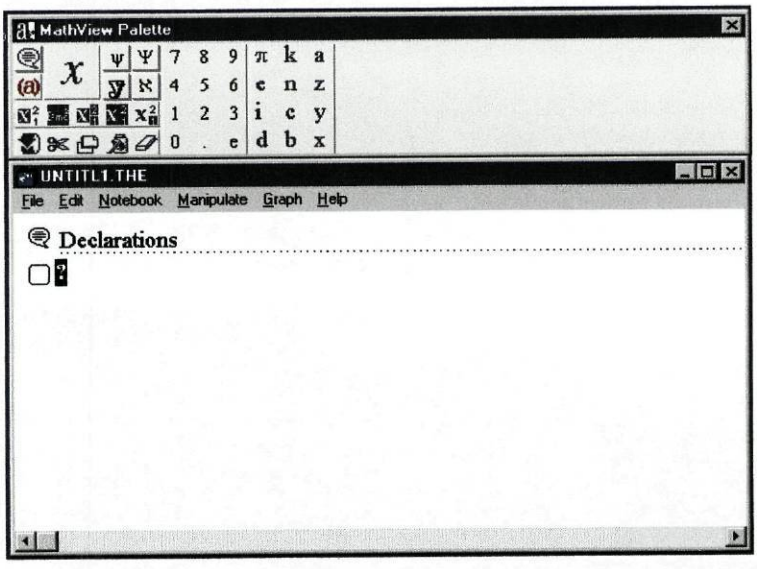
em que width e height controlam a largura e a altura do espaço da tela ocupado pela imagem referente ao ficheiro introduzido.

O funcionamento do MathView

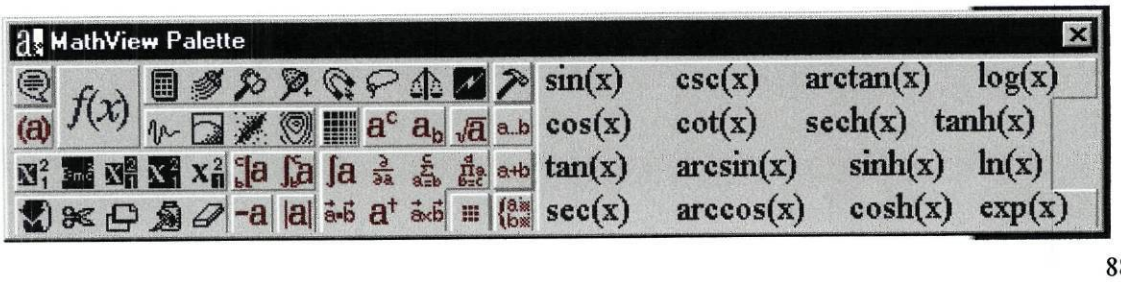


Quando o icon do programa é accionado, aparece no ecrã uma janela como a que de seguida é apresentada.

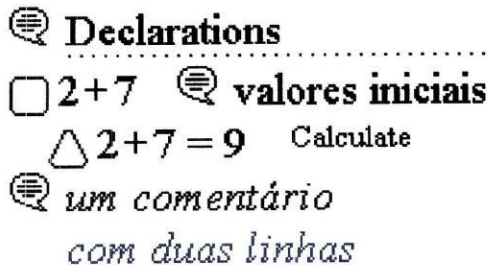
A barra de comandos que aparece na parte superior, chama-se palete das variáveis.



Para se proceder à introdução de dados, poderá recorrer-se ao teclado ou directamente à palete das variáveis. De referir ainda que, tal palete poderá tomar ainda a forma de palete de funções, conforme figura em baixo, bastando para tal, clicar-se no icon x, ou então, depois de se introduzirem quaisquer valores, ela será visualizada automaticamente. Por outro lado, quando posicionamos o ponteiro do rato sobre qualquer um dos botões disponíveis quer na palete das variáveis quer na palete das funções, o programa disponibiliza uma pequena legenda da sua função. O importante a reter é o princípio básico de que a introdução dos dados pode ser feita através do uso das funções disponíveis nos vários menus ou através da palete. De notar que a palete poderá ser tornada invisível, bastando para tal clicar em ☒ situado no canto superior direito da palete. Para tornar novamente visível a palete, basta escolher a função ***Windows/Palette*** no menu ***Notebook***.



Há algumas noções prévias a ter em conta antes de entrarmos propriamente na exploração do MathView e que se prendem com notações e simbologias inerentes ao seu próprio funcionamento. Na imagem do lado aparece o símbolo

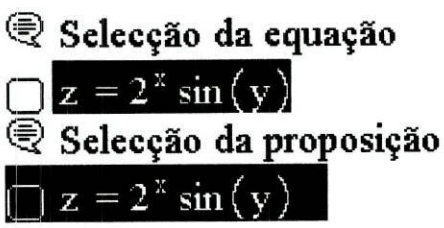


seguido da palavra **Declarations** e, se fizermos um duplo clique sobre o mesmo, aparecem algumas definições e regras do programa (ver Declarações e Regras de Transformação). Podemos substituir essa palavra por outra(s), bastando para tal seleccionar com o ponteiro do rato e escrever o que se pretende. O mesmo balão pode ser obtido para tecer um comentário no decorrer de um Notebook. Para tal, devemos clicar na tecla **ENTER** da parte numérica do teclado (ou então no menu da palete) e escrever o que pretendermos. No nosso exemplo aparece "**valores iniciais**". De notar que, quando fazemos aparecer um novo balão, ele fica posicionado no final de todas as expressões contidas no nosso Notebook. Para o colocarmos no local que mais nos convém, devemos seleccionar essa linha (com um clique sobre o balão respectivo), premir a tecla **CTRL** até que o ponteiro do rato adquira o formato de e depois movê-lo para o local pretendido. No nosso exemplo, à direita da linha "2+7".

Quanto aos símbolos ▢ e △ (quadrado e triângulo com os cantos arredondados) que aparecem à esquerda de cada uma das linhas, o primeiro diz respeito a uma nova proposição (**Prop icon**) a ser introduzida e que se obtém premindo a tecla **RETURN** (**ENTER** na parte não numérica do teclado), enquanto que o segundo aparece quando se leva a efeito determinada operação, mostrando o resultado obtido. Aparece ainda à direita da linha que contém o resultado a informação de "***Calculate***". Essa informação diz respeito à operação que foi levada a efeito para se obter o resultado conseguido. De referir que essa informação pode ser desactivada através da função ***Notebook Prefs/Show Steps*** no menu ***Notebook***.

Relativamente ao comentário que aparece no final "***um comentário com duas linhas***", há a destacar o facto de terem sido usadas duas linhas para o escrever o que, por vezes, traz algumas vantagens. Para conseguir tal

situação, deverá premir a tecla **ENTER** na parte numérica do teclado, escrever, por exemplo "*um comentário*" e, de seguida, para mudar de linha, basta premir as teclas **SHIFT** e **RETURN**. Depois escreva "*com duas linhas*". Por outro lado, há a destacar o facto desse último comentário estar em itálico e, em particular, a linha de baixo estar escrita a azul. Ou seja, o programa permite que se mudem as fontes não só dos comentários mas também de todo o Notebook. No primeiro caso, devemos seleccionar com o rato o comentário que desejamos alterar as fontes, e escolher as funções ***Comment Style*** (para mudar para itálico) e ***Comment Color*** (para mudar para azul), ambas as funções no menu ***Edit***. No segundo caso, deverá escolher-se a função ***Notebook Prefs*** no menu ***Notebook***, havendo, desse modo, a possibilidade de alterar o tipo de letra a usar nos Notebooks (por defeito, Times New Roman), bem como o seu tamanho (por defeito, 14).



A figura ao lado mostra dois tipos de selecção: no primeiro caso, mostra-se a selecção da equação (consegue-se com um clique sobre o sinal igual) e, no segundo caso, mostra-se a selecção da proposição na sua totalidade (consegue-se com um clique sobre o ▢). De salientar ainda que, não é possível, por exemplo, escrever a palavra "selecção" num comentário do programa, directamente. Isto acontece porque o programa não reconhece os ~. Para que isso seja possível, recomenda-se que o utilizador utilize o programa "Notepad", escreva a palavra desejada e depois, com as operações ***Copy/Paste***, a transfira para o comentário do MathView, ao qual se pretendia escrever essa mesma palavra.

Existem ainda questões que se prendem com a introdução de dados que, sem o conhecimento das mesmas, poderá fazer com se perca muito mais tempo do que aquele que é necessário para se levar a cabo uma pequena alteração numa expressão já introduzida. De seguida apresentam-se as instruções necessárias a algumas dessas alterações.

Expressão inicial com o valor seleccionado	Introduza a expressão seguinte	Expressão final
☐$3x^2$	+ a	☐$(3+a)x^2$
☐$3x^2$	+ a	☐$3(x+a)^2$
☐$3x^2$	+ a	☐$3x^{2+a}$

O desempenho do MathView

Com este programa, podemos fazer todo o tipo de cálculos, dos mais simples aos mais complexos, duma forma mais ou menos directa. O próprio programa contém regras de simplificação de cálculos, regras de transformação de expressões conhecidas (por exemplo trigonométricas, exponenciais, logarítmicas, entre outras) e permite a introdução de outras regras definidas pelo utilizador conforme as suas necessidades. A capacidade para o esboço de gráficos por parte deste programa é notável e bastante eficaz, inclusivamente no que se refere à sua animação.

À medida que os exemplos vão sendo ilustrados, serão apresentadas as instruções relativas à forma como é que poderão ser levadas a cabo as operações, mas para facilitar quem procura tal informação, de seguida apresentam-se alguns aspectos preliminares.

Cálculos simples

A figura da esquerda das imagens seguintes, mostra uma operação feita de uma forma directa, ou seja, depois de se introduzirem os dados iniciais, $2+5$, e de se seleccionar a expressão através de um duplo clique sobre o sinal "+", basta no menu ***Manipulate*** escolher a função ***Calculate***. Depois de efectuada a operação, se seleccionarmos (com duplo clique) e apagarmos o 5 (com a tecla **BACKSPACE**), aparece uma expressão semelhante à da figura da direita, em que o rasurado aguarda a introdução do novo valor pretendido.

Depois de introduzidos os novos valores, a expressão adaptar-se-à de imediato a eles. De referir apenas que, para que tal seja possível, a função ***Always ReManipulate*** do menu ***Manipulate*** tem que estar activada, o que acontece por defeito. Se desactivarmos essa função, os cálculos não se adaptarão automaticamente aos novos valores entretanto introduzidos. Para isso teríamos que escolher a função ***ReManipulate Now*** no menu ***Manipulate***.

$\square 2+5$
$\triangle 2+5=7$ Calculate

$\square 2+?$
$\triangle 2+5=7$ Calculate

Relativamente ao exemplo que se segue, há a destacar o seguinte: enquanto que no caso da soma "$3+7$" as funções ***Calculate*** e ***Simplify*** no menu ***Manipulate*** dão origem ao mesmo resultado, já no caso da soma "$\sqrt{3}+7$" a situação altera-se. A função ***Calculate*** dá-nos um valor aproximado, com uma precisão determinada pelo utilizador. Por defeito, a precisão são cinco dígitos mas, através da função ***Display Precision*** no menu ***Notebook***, podemos obter uma precisão de quinze dígitos.

$\square 3+7$
$\triangle 3+7=10$ Calculate
$\triangle 3+7=10$ Simplify
$\square \sqrt{3}+7$
$\triangle \sqrt{3}+7=8.7321$ Calculate
$\triangle \sqrt{3}+7=\sqrt{3}+7$ Simplify

A função ***Auto Simplify*** (disponível em ***Manipulation Prefs*** no menu ***Manipulate***) aparece, por defeito, activa. Se a desactivarmos, o programa processa as simplificações, passo a passo, à medida que vamos dando as instruções. De salientar ainda que, simplificar no MathView significa, de facto, usar as regras definidas na categoria "***Simplify***", disponíveis no Notebook que é aberto por defeito.

Nos exemplos seguintes, podemos observar que, enquanto no primeiro caso (***Auto Simplify*** activada), a simplificação é imediata, já no segundo caso (***Auto Simplify*** desactivada), a simplificação é conseguida em dois passos.

$$\square\ x^{-4}y^{-5}a^2b^3 \quad \text{Auto Simplify on}$$

$$\triangle\ x^{-4}y^{-5}a^2b^3 = \frac{a^2b^3}{x^4y^5} \quad \text{Simplify}$$

$$\square\ x^{-4}y^{-5}a^2b^3 \quad \text{Auto Simplify off}$$

$$\triangle\ x^{-4}y^{-5}a^2b^3 = a^2b^3\frac{1}{x^4}\frac{1}{y^5} \quad \text{Simplify}$$

$$\triangle\ x^{-4}y^{-5}a^2b^3 = \frac{a^2b^3}{x^4y^5} \quad \text{Simplify}$$

Vejamos ainda outro exemplo relativo a cálculos simples. Na figura que se segue, a operação é efectuada de uma forma directa, isto é, depois de se introduzir e seleccionar (com um duplo clique sobre o sinal "+") a expressão $\frac{2}{3}+\frac{10}{4}$, basta escolher a função ***Calculate*** no menu ***Manipulate***, aparecendo de imediato o resultado pretendido, conforme aparece retratado na imagem da direita (em baixo). Por outro lado, se escolhermos a função ***Simplify*** no menu ***Manipulate***, o resultado aparecerá sob a forma de fracção, o que é mais interessante para o Ensino, uma vez que nos dá o valor exacto e não arredondado para um número fixo de casas decimais (imagem da esquerda, em baixo).

$$\square\ \frac{2}{3}+\frac{10}{4}$$

$$\triangle\ \frac{2}{3}+\frac{10}{4} = \frac{19}{6} \quad \text{Simplify}$$

$$\square\ \frac{2}{3}+\frac{10}{4}$$

$$\triangle\ \frac{2}{3}+\frac{10}{4} = 3.1667 \quad \text{Calculate}$$

Podemos também fazer a mesma operação de uma outra forma menos directa mas, eventualmente, mais didáctica que a anterior, permitindo que o aluno visualize, o que está por trás de cada passo. Esta possibilidade, para certos conteúdos programáticos será, com toda a certeza, uma grande vantagem na medida em que a aluno visualiza o que foi feito em cada momento.

A figura seguinte, mostra uma sequência de operações que foram efectuadas antes de se chegar ao resultado final. De referir que as funções ***Apply***, ***Simplify***, ***Collect*** e ***Calculate***, foram obtidas no menu ***Manipulate*** mas poderiam ter sido também obtidas na palete das funções. A título de exemplo, a

operação ***Calculate*** poderia ser conseguida clicando no icon  , disponível na palete.

$$\square\ \frac{2}{5}+\frac{10}{4}$$
$$\triangle\ \frac{2}{5}+\frac{10}{4}=\frac{2}{5}+\frac{5\ 10}{5\ 4} \quad \text{Apply}$$
$$\triangle\ \frac{2}{5}+\frac{10}{4}=\frac{4\ 2}{4\ 5}+\frac{5\ 10}{5\ 4} \quad \text{Apply}$$
$$\triangle\ \frac{2}{5}+\frac{10}{4}=\frac{4\ 2}{20}+\frac{5\ 10}{5\ 4} \quad \text{Simplify}$$
$$\triangle\ \frac{2}{5}+\frac{10}{4}=\frac{4\ 2}{20}+\frac{5\ 10}{20} \quad \text{Simplify}$$
$$\triangle\ \frac{2}{5}+\frac{10}{4}=\frac{29}{10} \quad \text{Collect}$$
$$\triangle\ \frac{2}{5}+\frac{10}{4}=2.9 \quad \text{Calculate}$$

De salientar que, se substituirmos, por exemplo, na expressão inicial o 4 por um 7 (fazendo duplo clique sobre o 4 e escrevendo 7), todos os cálculos seguintes se adaptarão, de imediato, à nova expressão.

Resolução de equações do 1º grau.

Quanto à resolução de equações, de facto o programa permite fazê-lo de uma forma bastante simples. A figura seguinte mostra uma sequência possível de passos necessários à resolução de uma equação, neste caso do 1º grau.

$$\square\ 2(3x-5)=4(x-6)$$
$$\triangle\ 6x-10=4x-24 \quad \text{1° passo}$$
$$\triangle\ 2x-10=-24 \quad \text{2° passo}$$
$$\triangle\ 2x=-14 \quad \text{3° passo}$$
$$\triangle\ x=-7 \quad \text{4° passo}$$

No menu ***Manipulate***, poderemos encontrar a função ***Expand***, necessária ao desenvolvimento da propriedade distributiva. Depois de se seleccionar a expressão com um duplo clique sobre o sinal "=", escolhemos então ***Expand*** no menu ***Manipulate***. Para o segundo passo que consistiu em mudar de membro o "4x", basta seleccionar essa parcela e, sobre ela, posicionar o ponteiro do rato ao mesmo tempo que se pressiona a tecla **CTRL** até aparecer o símbolo . Basta depois arrastar, com o rato, a parcela seleccionada para o primeiro membro, com a tecla **CTRL** pressionada. No 3º passo, faz-se o mesmo que se fez no 2º passo, agora para o "10". Finalmente, para ocorrer o 4º passo, bastará seleccionar "x" com duplo clique sobre ele e, dentro do menu ***Manipulate*** seleccionar-se a função ***Isolate***. De notar que esta última função poderá também ser obtida na palete das funções, clicando no símbolo . Poderia ainda conseguir-se isolar o "x", seleccionando-o e, sobre ele com a tecla **CTRL** premida, arrastá-lo para cima do triângulo que aparece à esquerda da expressão "$2x = -14$".

Resolução de sistemas de duas equações e de duas incógnitas.

A resolução de sistemas de duas equações e duas incógnitas (ou mais), também se torna bastante simples. De seguida, vejamos como é que se pode utilizar as potencialidades deste programa, para levar a cabo a resolução de um sistema de duas equações e de duas incógnitas, pela mesma forma como a que se resolve com papel e lápis. Consideremos, por exemplo, o seguinte sistema:

$$\begin{cases} 3x + 4y = 18 \\ 2x + y = 7 \end{cases}$$

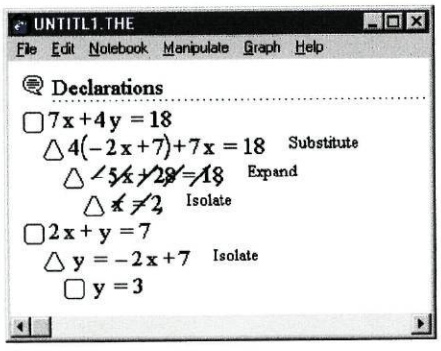
A primeira operação consiste em introduzir as duas equações, ficando algo com o seguinte aspecto:

$3x+4y=18$ Equação A
$2x+y=7$ Equação B

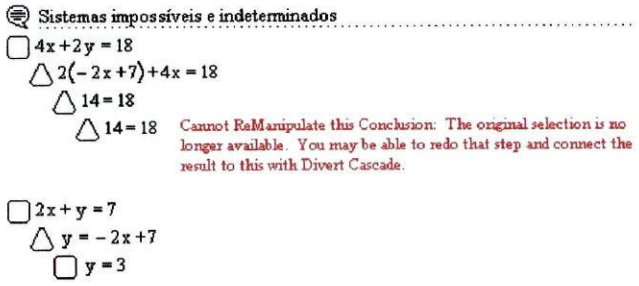
De seguida, apenas utilizando funções do menu ***Manipulate*** (ou as equivalentes existentes na palete) poderemos obter a solução final. Basta, para isso, por exemplo, seleccionar o *y* na equação "2x+y=7" e fazer ***Manipulate/Isolate***. De seguida, seleccionar a equação obtida na operação anterior e com o cursor sobre ela, premir a tecla **CTRL** até aparecer o símbolo , para assim arrastá-la para cima da equação "3x+4y=18". Fazemos então duplo clique sobre o sinal "=" para seleccionarmos a equação obtida e escolhemos ***Manipulate/Expand***. Na equação obtida, seleccionamos x (com duplo clique sobre ele) e escolhemos ***Manipulate/Isolate***, obtendo-se assim o valor de x = 2. Finalmente, devemos seleccionar esta última equação e arrastá-la para cima da equação "y=-2x+7", obtendo-se deste modo o valor de y = 3. A figura que se segue mostra a sequência das expressões obtidas ao longo das operações que atrás foram descritas.

$\square\ 3x+4y=18$ Passo (A)
$\quad\triangle\ 4(-2x+7)+3x=18$ Passo (C)
$\quad\quad\triangle\ -5x+28=18$
$\quad\quad\quad\triangle\ x=2$
$\square\ 2x+y=7$ Passo (B)
$\quad\triangle\ y=-2x+7$
$\quad\quad\triangle\ y=3$ Passo (D)

Tal como já foi ressaltado anteriormente, uma das vantagens deste programa é a facilidade com que se obtêm novas soluções fazendo simplesmente alterações aos dados iniciais. Se seleccionarmos, por exemplo, o "3" na equação escrita no passo (A) e alterarmos para "7", as novas soluções aparecem de imediato. Na figura abaixo, os traços que aparecem a rasurar algumas das expressões, são visíveis apenas durante o tempo em que as novas soluções não são apresentadas.



Importa ressaltar o facto de, no caso de estarmos perante sistemas impossíveis no corpo dos números reais, é apresentada ao utilizador uma mensagem como a que aparece retratada na imagem seguinte, o mesmo acontecendo no caso de sistemas indeterminados.



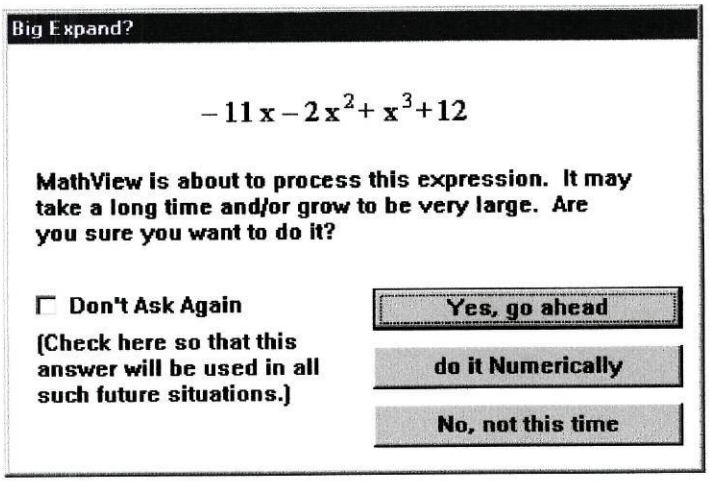
Decomposição em factores:

A decomposição em factores de polinómios, com coeficientes no corpo dos números inteiros, é uma tarefa que, com o MathView, apresenta bons resultados. Com apenas uma operação poderemos decompor, de uma forma rápida, polinómios de grau bastante grande. Vejamos por exemplo, a decomposição em factores do polinómio $y = 2x^6 + 22x^5 - 78x^4 - 982x^3 + 1660x^2 + 9456x - 17280$. Para tal, devemos introduzir a expressão e, de seguida seleccionar apenas o segundo membro (com um duplo clique sobre um dos sinais "+" ou "-"). Depois disso, vamos ao menu ***Manipulate*** e escolhemos a função ***Factor***. A imagem seguinte, mostra o que é visualizado depois dessa operação executada.

$$\square\ y = 2x^6 + 22x^5 - 78x^4 - 982x^3 + 1660x^2 + 9456x - 17280$$
$$\triangle\ y = 2(x+9)(x+8)(x-5)(x+4)(x-3)(x-2) \quad \text{Factor}$$

De notar ainda que, a mesma operação ***Factor*** poderia ter sido executada com um clique sobre o icon  , na palete das funções.

Em alguns casos, nomeadamente naqueles em que a expressão é mais complicada, quando se escolhe a função "***Factor***", poderá aparecer uma janela como a que de seguida se mostra. Nesses casos, para continuar, deve clicar-se no botão "***Yes, go ahead***" (se for esse o seu desejo).



Por seu turno, se pretendermos simplificar expressões do tipo daquelas que de seguida se apresenta, há a destacar duas funções: A função ***Expand*** (obtida no menu ***Manipulate***) e a função ***MiniExpand*** (obtida em ***Other*** no menu ***Manipulate***). As diferenças são visíveis na figura seguinte.

$\square\ (2x+3)^4$

$\triangle\ (2x+3)^4 = 16x^4+96x^3+216x^2+216x+81$ Expand

$\square\ \left([x+2]^4+5\right)^2$

$\triangle\ \left([x+2]^4+5\right)^2 = \left(x^4+8x^3+24x^2+32x+16\right)^2+10\left(x^4+8x^3+24x^2+32x+16\right)+25$ Expand

$\square\ \left([x+2]^4+5\right)^2$

$\triangle\ \left([x+2]^4+5\right)^2 = (x+2)^8+10(x+2)^4+25$ MiniExpand

Para colocarmos factores em evidência, podemos usar a função ***Collect***, obtida no menu ***Manipulate***. Os exemplos que se apresentam de seguida são demonstrativos das capacidades dessa função. De salientar que no primeiro exemplo foi usada também a função ***Commute*** obtida escolhendo ***Other*** no menu ***Manipulate***. Esta função pode ser conseguida de uma forma bem mais rápida. Basta seleccionarmos com o rato a/o parcela/factor que queremos comutar (no nosso exemplo a expressão x^2), premir a tecla **CTRL** para que o

ponteiro do rato fique como e arrastá-la para a posição em que a queremos situar.

$$\square\ 5x^3+15x^2$$
$$\triangle\ 5x^3+15x^2=5(x+3)x^2 \quad \text{Collect}$$
$$\triangle\ 5x^3+15x^2=5x^2(x+3) \quad \text{Commute}$$
$$\square\ 4x^5y^2z^3-6x^2y^7$$
$$\triangle\ 4x^5y^2z^3-6x^2y^7=2(-3y^5+2x^3z^3)x^2y^2 \quad \text{Collect}$$

Resolução de equações do 2º grau:

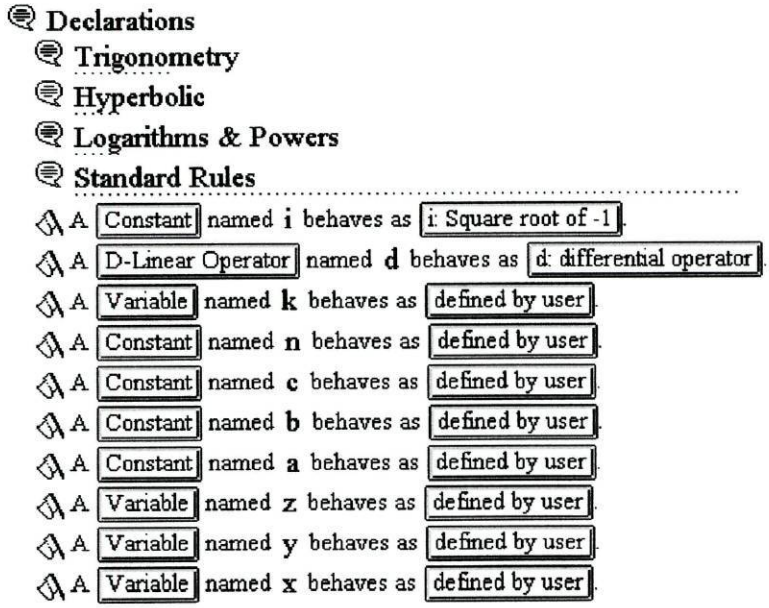
A resolução de equações do 2º grau, também é uma das potencialidades do MathView, como se pode verificar na figura seguinte. Dada uma equação, por exemplo, $2x^2-x-6=0$, podemos encontrar as duas soluções accionando a função ***Manipulation Prefs/Auto Casing***, no menu ***Manipulate***, seleccionar os dois "xx's" que aparecem na equação com duplo clique (mais **ALT** para seleccionar o segundo x) e depois escolher a função ***Isolate*** no menu ***Manipulate***. Como se disse anteriormente, caso a função ***Auto Casing*** não esteja accionada, o programa, depois de escolhida a opção ***Isolate*** só apresenta uma das soluções.

$$\square\ 2x^2-x-6=0$$
$$\triangle\ x=2 \quad \text{Isolate} \qquad \triangle\ x=-\frac{3}{2} \quad \text{Isolate}$$

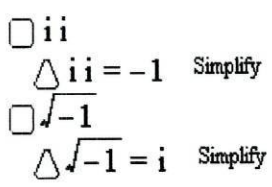
Depois de resolvida a equação, podemos alterar qualquer um dos três coeficientes (seleccionando-o com duplo clique e substituindo-o pelo novo valor) e, automaticamente, obter as respectivas soluções. De referir que o aluno deverá ser alertado para a questão, já referida anteriormente, das equações cujas soluções são complexas.

Declarações

Tal como se fez referência na parte inicial deste capítulo, é possível visualizar o que aparece, por defeito, nas ***Declarations***. Ao fazermos um duplo clique no balão 🗨, surgem algumas informações como aquelas que aparecem retratadas na figura que de seguida se apresenta. Trata-se de uma lista das constantes, das variáveis, funções e regras que, por defeito, o programa já traz declaradas/definidas.

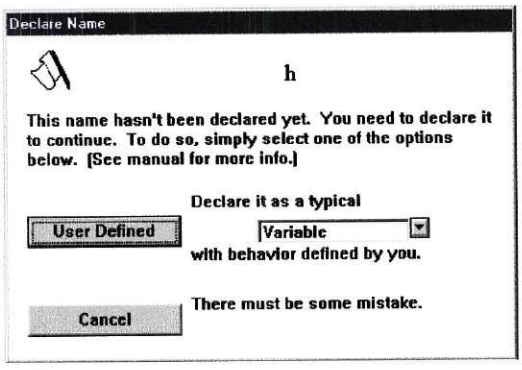


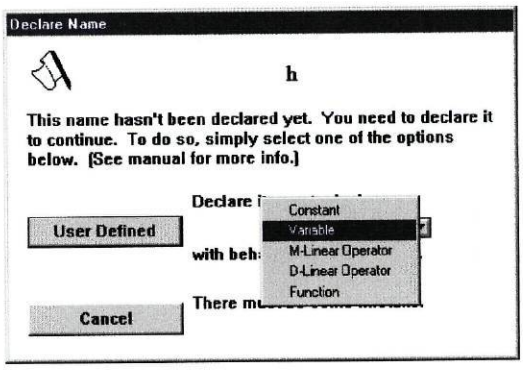
Por exemplo, o parâmetro ***i***, pertence à classe das constantes e, está pré-definido pelo próprio programa como sendo $\sqrt{-1}$. Deste modo, quando se simplificam as expressões $\sqrt{-1}$ e ***i*i*** (através da função ***Simplify*** no menu ***Manipulate***), obtém-se aquilo que se visualiza na imagem ao lado.



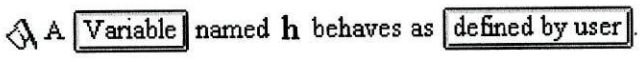
Cada um dos parâmetros que aparece antecedido do símbolo 🜂, poderá ser uma **constante**, **variável**, **M-Operador linear** (operador de matrizes), **D-Operador linear** (operador diferencial) ou **função**, havendo também a possibilidade de controlarmos, nós mesmos, esses aspectos.

Existem dois métodos à disposição do utilizador para a definição de novos parâmetros. Por um lado, o parâmetro pode ser definido através da sua mera utilização numa manipulação, ou através da função ***Clarify***. Sempre que o programa se depara com a utilização de um novo parâmetro, abre-se uma caixa de diálogo para que este seja definido. Por defeito, é atribuída a qualidade de "variável definida pelo utilizador" ao novo parâmetro. No exemplo que de seguida se apresenta, foi utilizado o parâmetro ***h*** que, note-se, não consta daquela lista inicial. Se escolhermos a função ***Clarify*** no menu ***Notebook***, de imediato surge a caixa de diálogo seguinte (à esquerda) com o valor por defeito a que já se fez referência. Clicando na seta à direita de "***Variable***", surgem 5 possibilidades à disposição do utilizador.





Depois de aceitarmos uma das possibilidades, se abrirmos novamente aquela lista inicial com um duplo clique sobre ***Declarations***, verifica-se que a linha seguinte foi introduzida.

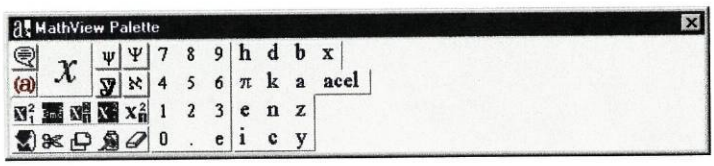


Por outro lado, o segundo método de definição de parâmetros, consiste na utilização da função ***Insert/Name Decl.*** no menu ***Notebook***. Aparece de imediato uma linha em que o parâmetro aparece substituído por um "***?***", para que o utilizador escolha o parâmetro a usar. De salientar que este método pode ser útil quando pretendermos, por exemplo definir o parâmetro "aceleração" usando uma expressão sugestiva, como por exemplo as letras "***acel***".

A Variable named ? behaves as defined by user.

A Variable named **acel** behaves as defined by user

Uma última nota para dar conta do que vai acontecendo com a palete das variáveis. Por exemplo, depois de termos definido o parâmetro "**acel**", ela já consta da palete, como mostra a figura seguinte.

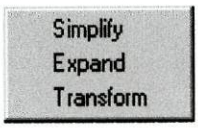


Regras de transformação

É possível aumentar ainda mais as potencialidades da MathView, através das regras de transformação. São utilizadas para "transformar" uma expressão de um determinado tipo numa expressão de outro tipo mas, logicamente, equivalente à anterior. Existem três tipos de manipulação disponíveis no menu ***Manipulation***: ***Transform***, ***Simplify*** e ***Expand***, ficando ao critério do utilizador a operação a levar a efeito. Há que ter, no entanto, atenção na forma como está, à partida, definida cada uma dessa regras. Por exemplo, podemos aceder à identidade $\tan(x) = \frac{sin(x)}{\cos(x)}$ que está pré-definida na programa através de duplos cliques sucessivos nos balões de ***Declarations/Standard Rules/Trigonometry/Transform to basic types***, até chegarmos a um conjunto de expressões que incluem a que de seguida se mostra.

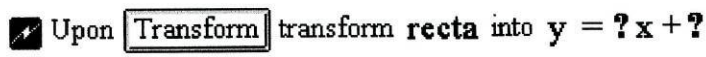
Upon [Transform] transform $\tan(\square)$ into $\frac{\sin(\square)}{\cos(\square)}$.

Ou seja, se no Notebook precisarmos de fazer tal transformação, bastaria escolhermos a função ***Transform*** no menu ***Manipulate***. De notar que, poderíamos mudar essa operação. Se clicarmos no "botão" [Transform], aparece de imediato a janela de opção que de seguida se apresenta, permitindo que o utilizador escolha a função que mais lhe agrade.



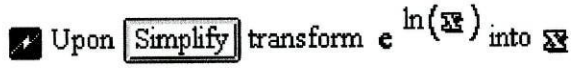
O Notebook que é aberto, por defeito, quando abrimos o MathView, já contém muitas regras. Poderão ainda encontrar-se outras regras nos Notebooks que são fornecidos com o próprio MathView, havendo a possibilidade de, utilizando as tradicionais operações de ***Copy/Paste***, os transferir para o Notebook que estamos a utilizar.

Podemos ainda criar algumas regras de Transformação que nos facilitem o trabalho. Por exemplo, na seguinte regra, a palavra "recta" é transformada, depois de declararmos "recta" como a "***User Defined Variable***", na equação reduzida da recta, ficando o programa a aguardar que se introduzam o declive e a ordenada na origem respectivas.



Para a criação de tal recta, devemos escolher as funções ***Insert/Transform rule*** no menu ***Notebook***. Depois de isso feito, escreva numa nova proposição a palavra "recta", selecciona-a e faça ***Manipulate/Transform***. Apenas uma última nota para referir que a regra que se acabou de definir pode ficar visível no nosso Notebook, ou podemos colocá-la junto da lista inicial das Declarações, mencionadas anteriormente.

Para a definição de tais regras, podemos usar as variáveis usuais ou, se pretendermos obter uma maior flexibilidade, podemos usar as variáveis que se obtêm através dos símbolos ▩ (**Wildcard variables**). Se tal acontecer, não estaremos limitados ao argumento que consta da regra definida no programa. A título de exemplo, vejamos a regra que aparece retratada na figura seguinte, que poderá ser obtida em ***Declarations/Standard Rules/Logarithms & Powers***.



A função ***Simplify*** no menu Manipulate, transformará a expressão $e^{\ln(qualquer\ var\ iável)}$ em $qualquer\ var\ iável$. Por outras palavras, a variável independente não necessita obrigatoriamente de ser ***x***. Poderá ser ***y***, ***p***, ***z*** ou mesmo uma expressão. A figura seguinte mostra isso mesmo.

$$\square\ e^{\ln(p)}$$
$$\triangle\ e^{\ln(p)} = p \quad \text{Simplify}$$
$$\square\ e^{\ln(2x+3y)}$$
$$\triangle\ e^{\ln(2x+3y)} = 2x+3y \quad \text{Simplify}$$
$$\square\ e^{\ln(\text{qualquercoisa})}$$
$$\triangle\ e^{\ln(\text{qualquercoisa})} = \text{qualquercoisa} \quad \text{Simplify}$$

De igual modo, o utilizador também poderá definir as suas regras de transformação utilizando estes caracteres especiais. Para tal, deverá escolher as funções ***Insert/Transform Rule*** no menu **Notebook**. As figuras em baixo retratam a nova regra em branco sem quaisquer valores e um exemplo de uma regra de transformação.

Upon [Transform] transform ? into ?

Upon [Transform] transform $\frac{1}{\csc(\blacksquare)}$ into $\sin(\blacksquare)$

A partir deste momento, se fizermos ***Manipulate/Transform***, depois de seleccionarmos uma expressão do tipo daquela que se acabou de definir, obtemos o pretendido, conforme se verifica na imagem seguinte.

$$\square\ \frac{1}{\csc(\theta)}$$
$$\triangle\ \frac{1}{\csc(\theta)} = \sin(\theta) \quad \text{Transform}$$

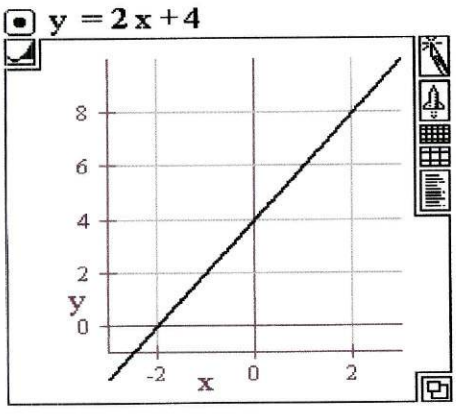
Análise gráfica

Esboço de gráficos

A representação gráfica de funções, é um dos aspectos em que o MathView apresenta melhores resultados, pela facilidade com que se obtêm os gráficos pretendidos e com que estes são manipulados. No capítulo IV deste trabalho, esse facto poderá ser constatado na prática, uma vez que contém Notebooks criados com o programa e com a possibilidade de serem manipulados. Para tal, será apenas necessária a instalação, na máquina em que se estiver a visualizar este trabalho, o respectivo *plugin*.

Mas, vejamos agora quais as instruções básicas para a introdução dos dados necessários ao esboço do gráfico de uma função e como se podem levar a cabo determinadas alterações às propriedades do mesmo.

A figura que se segue mostra o aspecto da janela que se obtém depois de se gerar o gráfico (2-D) da função $y = 2x + 4$. Depois de se introduzir a expressão analítica da função da qual se pretende o esboço gráfico, clicamos no sinal "=" para que a expressão fique seleccionada. Depois, no menu **Graph**, fazemos **y = f(x) / linear**. De referir também que o gráfico pode ser obtido de uma forma mais rápida. Basta, para isso, introduzir a expressão analítica da função, seleccioná-la (com um clique sobre o sinal "=") e depois, na palete das funções, clicar no icon  .



Relativamente aos símbolos que aparecem do lado direito do gráfico, cada um deles, depois de premido, leva a cabo uma determinada acção, a saber:



Este símbolo, que sugere uma "faca", permite fazer um ***zoom in*** em qualquer parte do gráfico. Basta clicar no símbolo e, com o cursor sobre o gráfico,

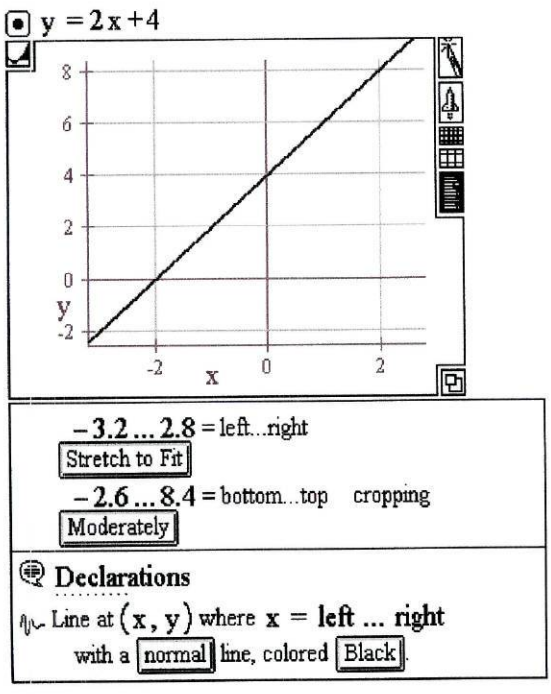
arrastá-lo ao longo da zona que queremos ver ampliada. Ao logo desta operação, o cursor toma a forma de uma "faca".

 O símbolo ao lado, que sugere um "foguetão", permite tornar o gráfico duas vezes mais reduzido (***zoom out***) por cada vez que se clica sobre ele, constituindo, por isso, a operação inversa daquela que é obtida com a "faca".

 Este rectângulo com a rede mais cerrada, aumenta o número de pontos que foram usados na elaboração do gráfico, fazendo com que este apareça esboçado de uma forma mais detalhada e perfeita.

 Por sua vez, o rectângulo ao lado com a rede menos cerrada, diminui o número de pontos que foram usados na elaboração do gráfico, fazendo com que, em determinadas situações, apareça um esboço mais imperfeito.

Por outro lado, colocando o cursor sobre o símbolo que aparece no canto inferior direito,  ,e premindo-o sem largar, poderemos aumentar e diminuir o tamanho do gráfico conforme pretendermos. Relativamente ao símbolo  , torna-se necessário quando se pretende seleccionar o próprio gráfico.



Finalmente, ao clicarmos no icon  (a que chamarei icon de detalhes), ele transforma-se em  e, automaticamente, são abertos os rectângulos que são apresentados na figura ao lado.

Ainda de referir que, quando nos aproximamos do gráfico com o ponteiro do rato, este transforma-se no símbolo e, nessa altura, podemos deslocar e alterar a janela de visualização do

gráfico, quer na vertical quer na horizontal, tão "longe" quanto nós queiramos. Basta para isso, premir sem largar o botão esquerdo do rato, com o ponteiro sobre o gráfico, deslocando-o para cima/baixo ou esquerda/direita, conforme o pretendido. Automaticamente os limites da janela de visualização (na imagem inicial – ***3,2...2,8=left...right*** e ***–2,6...8,4=bottom...top***) adaptar-se-ão à nova janela de visualização.

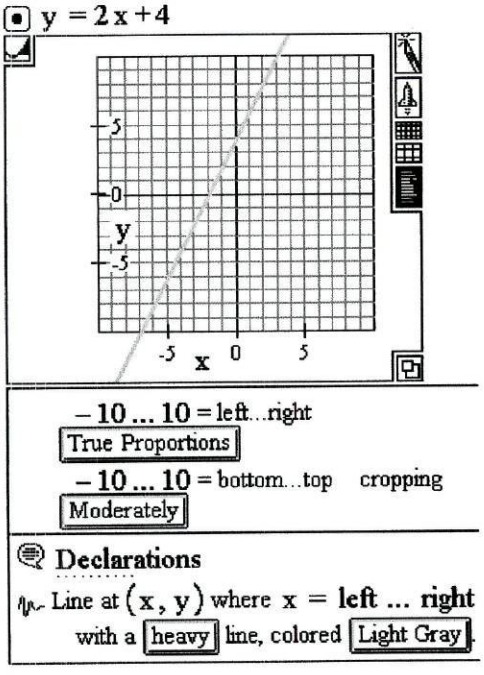
Vejamos agora alguma das alterações que se podem fazer ao gráfico anterior, depois de se clicar no botão que aparece na parte lateral direita do gráfico.

Dessas possíveis alterações, destacam-se as seguintes:

1. Os limites da janela de visualização do gráfico podem ser alterados. Para tal, deve clicar-se sobre os respectivos valores e alterá-los para os valores pretendidos. No nosso exemplo retratado na figura seguinte, tais limites foram alterados para ***–10...10.***

2. Ao clicarmos no "botão" Stretch to Fit surge a possibilidade de escolhermos "***True Proportions***", conforme se verifica no exemplo apresentado. Ou seja, há a possibilidade de colocarmos os dois eixos à mesma escala.

3. Ao clicarmos no "botão" normal, podemos escolher a espessura da linha que constitui o gráfico a esboçar. No nosso exemplo, foi escolhida a opção "***heavy***", mas há outras que estão disponíveis (***dotted*** ou ***dashed*** – dois tipos de picotado).



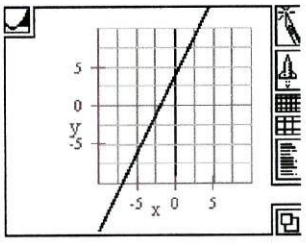
4. Por defeito, a linha que constitui o gráfico aparece representada a preto. No entanto, clicando sobre o "botão" que aparece inicialmente Black, aparecerão outras possíveis cores. No nosso exemplo, foi escolhida a "***Light Gray***".

5. Ao fazermos duplo clique sobre o balão Declarations ficamos com acesso a variadas opções. A imagem seguinte, mostra essas opções: verifica-se que é possível alterar as cores dos eixos e respectivas grelhas, podemos fazer com que as legendas dos eixos fiquem situadas this side ou other side, bem como é possível controlar a quantidade de linhas verticais e horizontais que são visíveis no gráfico. No nosso exemplo, o "0" que aparece por defeito foi substituído por "1" na parte separated by 1 colored. Para tal, seleccionou-se o "0" (com um clique sobre ele) e alterou-se para "1".

Declarations
Grid lines at (x, y) where
y = bottom ... top for each value
x = left ... right separated by 1 colored Lilac
Grid lines at (x, y) where
x = left ... right for each value
y = bottom ... top separated by 1 colored
Lilac
Axis at (left, y) where
y = bottom ... top labeled y on other side
colored Lilac
Axis at (x, bottom) where
x = left ... right labeled x on this side
colored Lilac
A Constant named left behaves as
Horizontal Minimum
A Constant named right behaves as
Horizontal Maximum
A Constant named bottom behaves as
Vertical Minimum
A Constant named top behaves as
Vertical Maximum

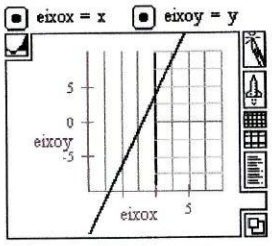
Na sequência de imagens seguinte, mostram-se alguma alterações que ainda se podem levar por diante. Destaca-se a alteração que se pode fazer à legenda dos eixos e à parte da grelha que se pode esconder ou fazer aparecer a partir de determinado limite.

Grid lines at (x, y) where y = bottom ... top for each value x = left ... right separated by 2.5 colored Black
Grid lines at (x, y) where x = left ... right for each value y = bottom ... top separated by 2.5 colored Lilac
Axis at (left, y) where y = bottom ... top labeled y on other side colored Lilac
Axis at (x, bottom) where x = left ... right labeled x on this side colored Lilac



Neste exemplo alterou-se o número de grelhas verticais a apresentar no sistema de eixos. No caso anterior, apareciam "1 em 1" unidade e agora aparecem "2.5 em 2.5" unidades.

Grid lines at (x, y) where y = bottom ... top for each value x = left ... right separated by 2.5 colored Black
Grid lines at (x, y) where x = 0 ... right for each value y = bottom ... top separated by 2.5 colored Lilac.
Axis at (left, y) where y = bottom ... top labeled eixoy on other side colored Lilac.
Axis at (x, bottom) where x = left ... right labeled eixox on this side colored Lilac.



Neste caso, teremos que declarar os novos nomes a atribuir aos eixos, como sendo os eixos tradicionais.

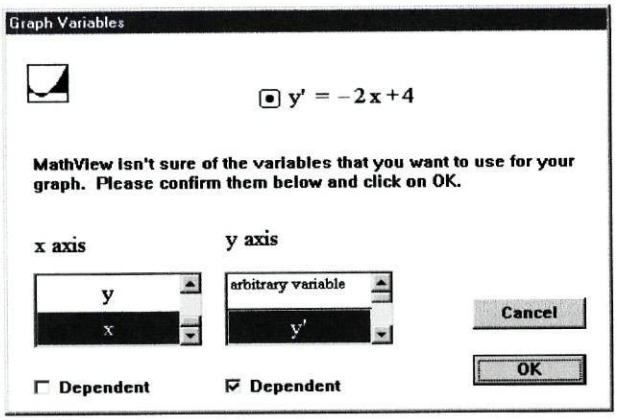
Tal como já foi referido anteriormente, com o MathView a manipulação dos dados obtidos torna-se demasiado simples de levar por diante. Suponhamos agora que queremos esboçar o gráfico da função $y = 2x + 6$. Para tal, bastará ir à expressão anterior cujo gráfico já foi esboçado, seleccionar o 4 (com um clique sobre ele) e substituir por um 6, fazendo com que o gráfico se adapte de imediato à nova função. De notar que a opção ***Always Redraw*** disponível no menu ***Graph***, terá que estar accionada, para que as alterações ao gráfico quando se altera a função sejam automáticas. Caso essa função esteja desactivada, quando alteramos a função teremos que clicar sobre o gráfico ou então escolher a função ***Redraw Now*** disponível no mesmo menu ***Graph***.

Ora, tal característica do programa, constitui uma grande vantagem, nomeadamente na abordagem de determinados conteúdos. Por exemplo, o aluno poderá chegar facilmente à conclusão de que em funções do tipo $y = mx + b$, o "*b*" (ordenada na origem) é sempre a ordenada do ponto de intersecção do gráfico da função com o eixo das ordenadas e conjecturar sobre o comportamento da inclinação do gráfico à medida que fazemos variar o valor de "*m*".

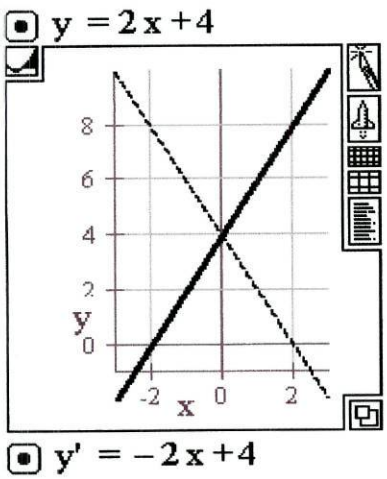
Mais do que um gráfico no mesmo sistema de eixos

Vejamos agora como é que se representam dois gráficos no mesmo sistema de eixos. Para tal, devemos introduzir a expressão analítica da primeira função e esboçar o seu gráfico, como anteriormente se explicou. Depois, premimos a tecla ***Return*** por forma a introduzirmos a expressão analítica da nova função pretendida, por exemplo $y' = -2x + 4$ (não se trata da derivada da função y mas

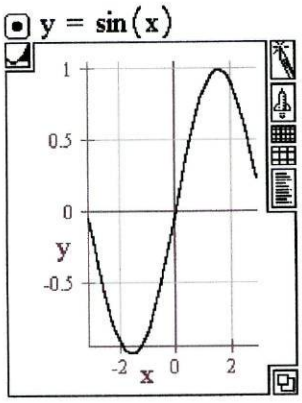
simplesmente uma outra notação – podia ser também, por exemplo, y_1). Seleccionamos esta expressão (com um clique sobre o sinal "=") e escolhemos a função ***Add Line Plot*** no menu ***Graph/Additional***, aparecendo, de imediato, uma janela como a que foi apresentada.

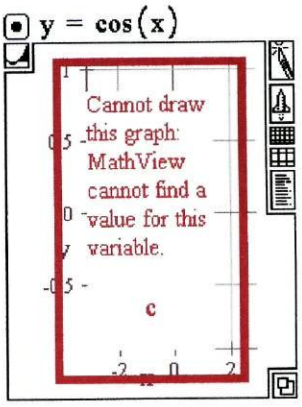


Torna-se necessário confirmar se, para o eixo dos xx's, está prevista a variável x e se para o eixo dos yy's a variável y' (com o campo referente a variável dependente devidamente accionado). Quando fazemos ***OK***, aparecem as duas funções no mesmo sistema de eixos, como mostra a figura em baixo. Por defeito, a variável independente é x e a variável dependente é y. Quando o programa tem dúvidas relativamente a essas informações (por exemplo quando se introduzem expressões com mais do que uma variável independente, como no caso $y = 2x + k$), a mesma janela que aparece retratada na figura anterior, é também aberta automaticamente no momento em que se tenta esboçar o respectivo gráfico.



De referir ainda que a linha dos dois gráficos se apresenta de formas diferentes e poderiam também ser apresentadas de cores diferentes. Tal foi possível através do que foi dito atrás, relativamente ao icon de detalhes  .

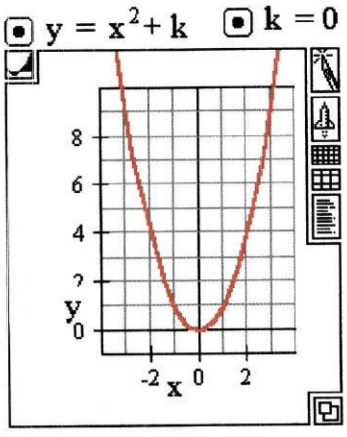




Chama-se a atenção ainda para o seguinte aspecto. Por vezes, na representação gráfica de determinadas funções, acontece o que se mostra nas figuras anteriores. Por exemplo, depois de representada a função $y = sin(x)$, se pretendermos mudar a expressão analítica para que o gráfico apareça de imediato, por exemplo para a função $y = \cos(x)$, após visualizarmos a imagem da direita, temos que ir ao menu ***Notebook*** e seleccionar a função ***Clarify***. O problema ficará, de imediato resolvido, aparecendo o gráfico da nova função pretendida.

Animação de gráficos

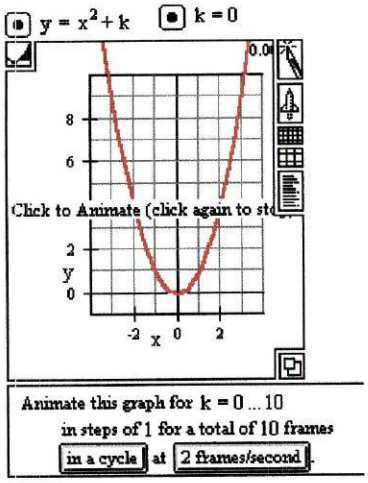
O MathView permite que se levem a cabo animações dos gráficos, o que poderá ajudar os alunos na compreensão de determinadas questões. A animação cria a ilusão de movimento através da sucessão de gráficos obtidos através da variação de um determinado parâmetro. Por exemplo, para estudarmos a deslocação do gráfico de uma parábola, na vertical, podemos animar o gráfico da função $y = x^2 + k$.



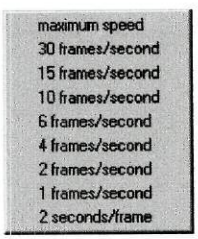
Antes de mais nada, teremos que introduzir a expressão da função da qual pretendemos obter o gráfico, neste caso $y = x^2 + k$. Depois, teremos que

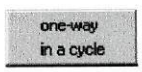
atribuir um valor a k, por exemplo, ***k=0***. De seguida, seleccionamos a nossa expressão (com um clique sobre o sinal "=") e esboçamos o gráfico conforme explicado anteriormente (escolher a função ***y=f(x)/linear*** no menu ***Graph***). A imagem anterior representa o que se obtém após se levarem a cabo tais operações.

Para se conseguir a animação, selecciona-se a expressão k=0 (com um clique sobre o sinal "=") e escolhe-se a função ***Animation/Start*** disponível no menu ***Graph***. A imagem seguinte mostra o que se obtém, após paragem da animação que se consegue com um clique sobre o gráfico.



Relativamente à parte que aparece retratada no final do gráfico, refira-se que os valores que aparecem por defeito poderão ser alterados. No nosso exemplo, fizemos variar o valor de k entre 0 e 10. Se pretendêssemos alterar estes valores para, por exemplo, ***k=0...5***, deveríamos seleccionar o 10 (com um clique sobre esse valor) e alterá-lo para 5. Podemos definir também a periodicidade com que queremos que o k varie. No nosso exemplo o valor de k aumenta uma em uma unidade (***in step of 1 for a total of 10 frames***) mas, se pretendermos que o k varie, por exemplo de duas em duas unidades, bastaria seleccionar o 1 e alterá-lo para 2. Automaticamente, o número de frames alterar-se-ia para 5.





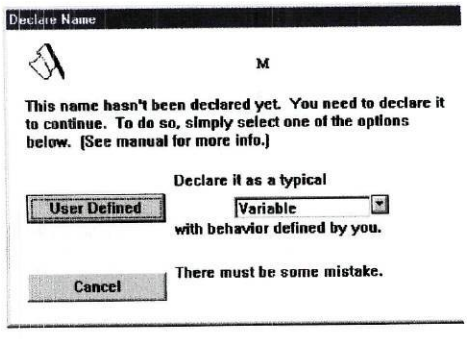
Refira-se ainda que, ao clicarmos nos "botões" [2 frames/second] e [in a cycle], surgem, respectivamente as opções que se podem visualizar nas figuras ao lado. Quanto à primeira situação, permite-nos escolher uma animação mais rápida ou mais lenta. Relativamente à segunda situação, há a referir o seguinte. As funções periódicas aparecerão representadas ***"in a cycle"***, isto é, as imagens referentes aos valores inicial e final serão coincidentes. Se a função da qual se pretende realizar uma animação do respectivo gráfico não for

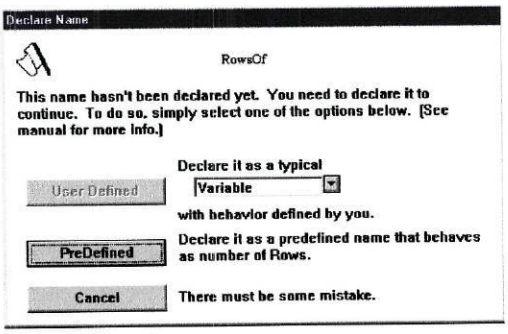
periódica, deverá usar-se a opção ***"one-way"***, em que acontecerá uma breve pausa entre a primeira e a última *frame*.

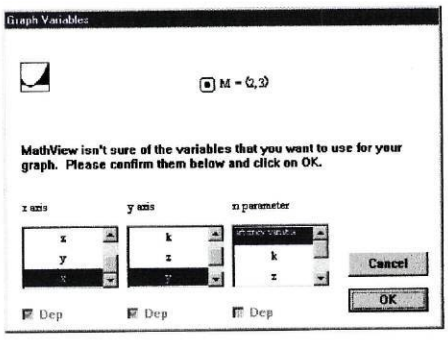
Note-se ainda que, à medida que a animação vai progredindo, o valor que k está a tomar pode ser acompanhado no canto superior direito do gráfico. Relembre-se apenas que a animação pode ser parada através de um clique sobre o gráfico e novamente activada através de um novo clique sobre o gráfico.

Marcação de pontos

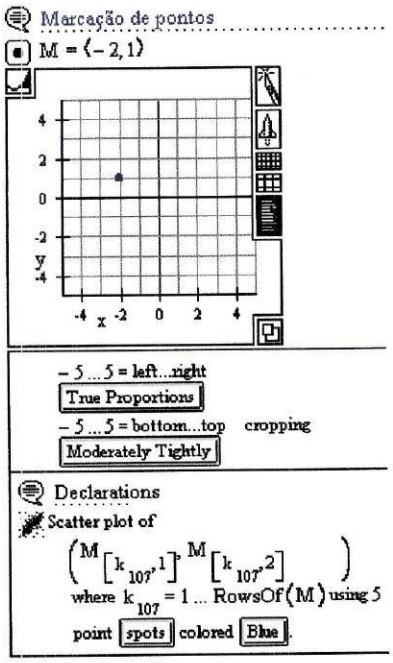
Para a marcação de pontos há que, antes de mais nada, introduzir o ponto a representar. Suponhamos, por exemplo, o ponto M(-2,1). Para a introdução de tais valores, deveremos digitar ***M = - 2 , 1*** e, automaticamente, os parêntesis serão introduzidos. De seguida, seleccionamos o ponto (com um clique no sinal "=" na expressão M=(-2,1)) e escolhemos a função ***Scatter/Linear*** no menu ***Graph***. De imediato abre-se a janela que ao lado é apresentada e, nessa altura, há que, entretanto, escolher a opção ***M-Linear Operator*** e clicar no botão **User Defined** . Aparece uma nova janela como a da direita da sequência seguinte de modo a confirmarem-se as variáveis a usar na marcação dos pontos. Por fim, na janela da esquerda da sequência seguinte, deverá clicar-se no botão ***PreDefined***, uma vez que se trata de definir uma nova variável que, entretanto, é automaticamente introduzida (***RowsOf***) e que, sobre ela, falaremos mais adiante.



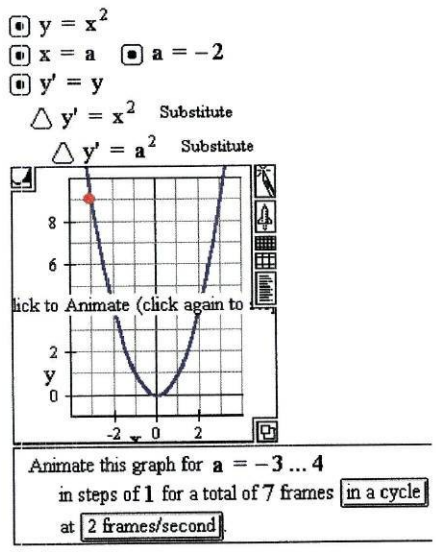




De imediato, aparecerá o nosso ponto representado num sistema de eixos que poderemos configurar ao nosso gosto, conforme referido anteriormente. A imagem do lado mostra o resultado final dessa marcação, depois de abertos os detalhes do gráfico. O "botão" spots permite escolher o formato do ponto. Para além do formato tradicional que aparece na imagem, estão disponíveis quadrados, triângulos, cruzes, entre outros. Por outro lado, é possível escolher a cor do ponto, bem como a espessura. No nosso exemplo, a espessura usada é equivalente à espessura de 5 pontos, como se pode verificar na expressão where $k_{107} = 1 \ldots \text{RowsOf}(M)$ using 5 point .



Movimento de um ponto sobre o gráfico de uma função



Depois de vista a forma de se proceder à animação de gráficos e depois de feita a abordagem da representação de pontos, seria interessante abordarmos, por exemplo, o movimento de um ponto sobre a linha que constitui um gráfico de uma dada função. A figura que se apresenta ao lado, representa essa situação, ou seja, o movimento de um ponto sobre a parábola de equação $y = x^2$.

Para conseguir tal efeito, em primeiro lugar, teremos que introduzir a expressão analítica da função e esboçar o seu gráfico (depois de introduzida a expressão, seleccioná-la com um clique sobre o sinal "=" e escolher a função ***y=f(x)/Linear*** no menu ***Graph***). De seguida, introduzimos as expressões "x = a" e "a = -2", expressões que vão definir a abcissa do ponto representado no gráfico. Depois de chegarmos à expressão $y' = a^2$, arrastando as expressões necessárias para

a expressão $y' = y$, estamos em condições de representarmos o ponto. Para tal, escolhemos a função ***Additional/With Blank Slots*** no menu **Graph**, seleccionamos de novo a função ***Additional/Add Scatter Plot*** no mesmo menu ***Graph***, aparecendo no espaço destinado aos detalhes do gráfico, uma instrução como a que de seguida se apresenta.

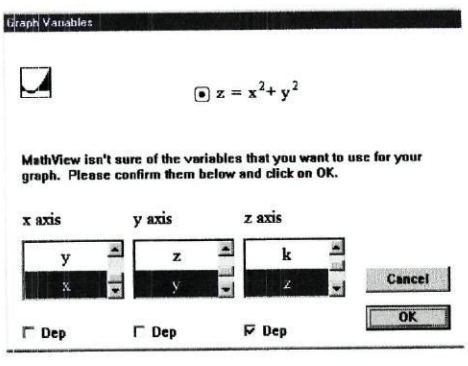
Scatter plot of (?, ?) where ? using 6 point
squares colored Blue.

Completamos os ***?*** de modo a obtermos as situações visíveis na figura ao lado, só faltando, para completar a elaboração deste exemplo, a animação do gráfico que dará origem ao movimento do ponto. Para tal, seleccionamos a expressão "a=-2" e escolhemos a função ***Animation/Start*** no menu ***Graph***. Teremos de imediato o ponto de abcissa "a" a movimentar-se sobre o gráfico da parábola representado, entre valores escolhidos pelo utilizador. Neste caso, entre –3 e 4.

Declarations
Line at (x, y) where x = −4 ... 4
with a heavy line, colored Blue.
Scatter plot of (a, y') where n = 1 ... 1
using 8 point spots colored Red.

Gráficos 3-D

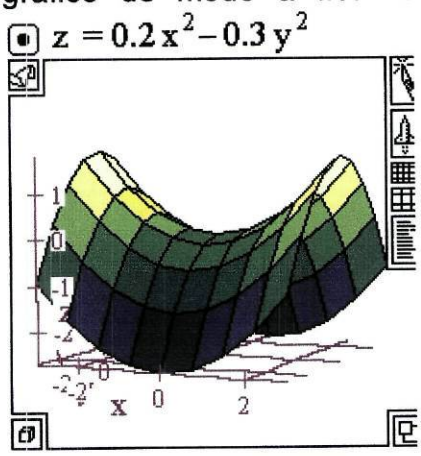
Os procedimentos a adoptar para o esboço de gráficos a três dimensões são semelhantes aos adoptados no esboço dos gráficos a duas dimensões a que já se fez referência. Antes de tudo, há que introduzir a expressão analítica da qual se pretende esboçar o respectivo gráfico. Consideremos, por exemplo, a função $z = 0.2x^2 - 0.3y^2$.



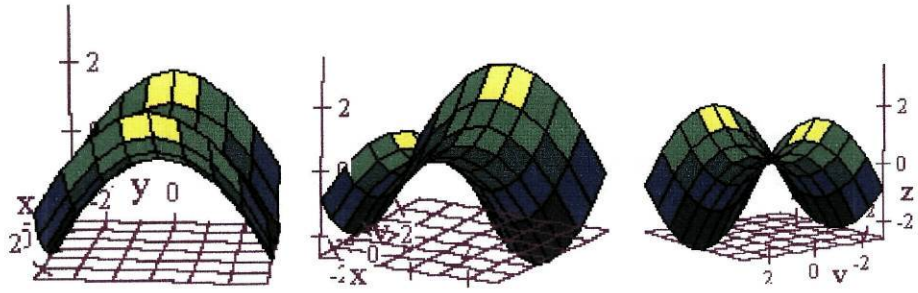
Seleccionamos a equação (com um clique sobre o sinal "=") e depois escolhemos a função ***z=f(x,y)/Color 3D*** no menu ***Graph.*** Aparece de imediato uma janela, como a do lado, a solicitar que se confirme a disposição das variáveis pelos três eixos e a pedir que

se declare a variável dependente. Na imagem seguinte aparece esboçado o gráfico da função definida anteriormente.

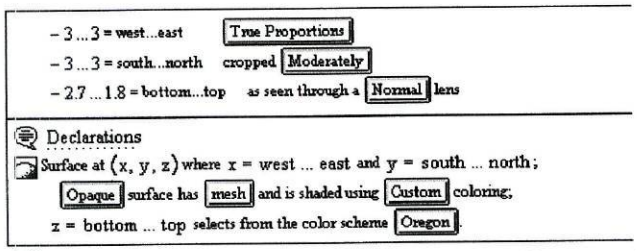
No caso dos gráficos 3-D, quando aproximamos o ponteiro do rato da zona do gráfico de modo a ficar com a forma de ☜, podemos visualizar outras perspectivas do mesmo. O símbolo ⊡ que aparece no canto inferior esquerdo do gráfico, indica que estamos a visualizar o gráfico na sua versão inicial. Caso tenhamos mudado a sua orientação, tal símbolo é substituído por ⊡ , ⊡, ⊡, ⊡ ou ⊡, dependendo da sua posição. Para voltarmos à posição inicial do gráfico, basta clicarmos sobre esse mesmo símbolo.



A sequência de imagens que de seguida se apresenta, corresponde ao gráfico da mesma função, vista em diferentes perspectivas.



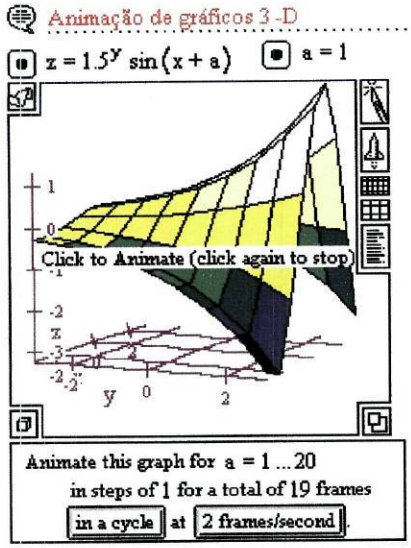
Ao clicarmos no "botão" a que dá acesso aos detalhes (⊡), aparece algo do género daquilo que aparece nos gráficos 2-D. A imagem ao lado retrata essa situação. Ao clicarmos no balão em **Declarations** poderemos, à semelhança do que fizemos para os gráficos 2-D, proceder a alterações do aspecto do gráfico.



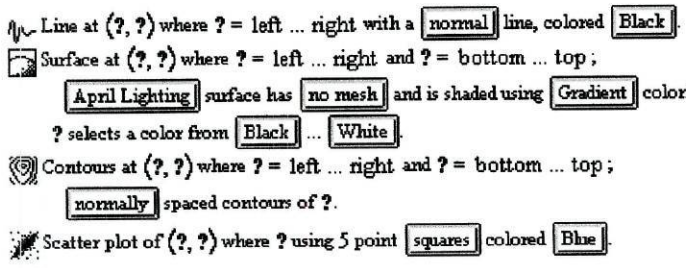
Tudo o que se disse relativamente aos gráficos 2-D no que toca à alteração da expressão analítica e da função ***Always Redraw*** continua a ser válido para estes gráficos. Também se pode, no mesmo sistema de eixos, representar

mais do que uma função, bastando para tal, depois de introduzida e seleccionada a nova função, escolher a função ***Additional/Add Surface Plot*** no menu ***Graph***.

Para a animação de gráficos 3-D, o método também é sensivelmente o mesmo daquele que foi usado para a animação de gráficos 2-D. Suponhamos que queremos, por exemplo, animar o gráfico da função $z = 1.5^{y}.sin(x+a)$, em que "a" é um parâmetro a variar. Para tal, deveremos introduzir a expressão analítica da função a representar e atribuir um valor a "a", por exemplo, a=1. De seguida, deveremos seleccionar a expressão da função a representar (com um clique sobre o sinal "=") e escolher a função ***z=f(x,y)/Color 3D*** no menu ***Graph***. Finalmente, seleccionamos a expressão "a=1" e escolhemos a função ***Animation/Start*** no mesmo menu ***Graph***. A imagem anterior representa o resultado dessas operações, mas cuja animação aparece em situação de "*pause*".



Para terminar a abordagem desta parte gráfica do programa MathView, será conveniente falarmos de uma função disponível no menu ***Graph/Additional***, que é a função ***with blank slots***, já utilizada anteriormente. Quando esta função se encontra accionada, poderemos, depois de usarmos uma qualquer função disponível no menu ***Graph***, introduzir uma outra expressão qualquer disponível na mesma função ***Additional***, começada por "***Add...***". A figura anterior representa algumas dessas opções que, obviamente, aparecem com os parâmetros representados por ***?*** e que, portanto, deverão ser preenchidos pelo utilizador. Depois de substituídos os ***?*** pelos respectivos valores, dever-se-à desactivar a função ***with blank slots***.

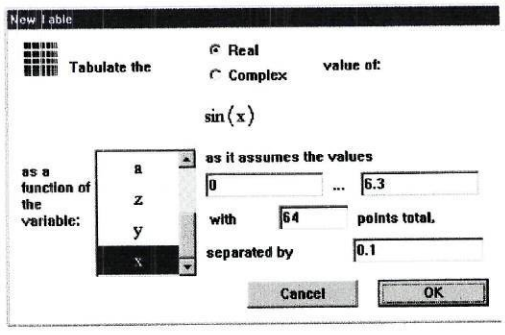


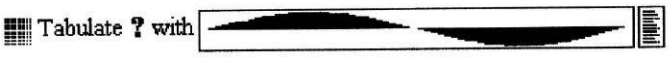
Tabelas

A utilização de tabelas com o programa MathView, poderá ser uma boa estratégia para a abordagem de temas como limites e sucessões. Podem ser directamente criadas tabelas a partir de expressões ou gráficos criados no próprio documento, ou poderão ainda ser importadas de dados externos ao próprio documento. Por outro lado, poderão também ser exportados os dados referentes a uma tabela criada no MathView, para outros programas. As tabelas podem suportar milhares de entradas, sendo apenas a memória disponível o limite do tamanho da tabela.

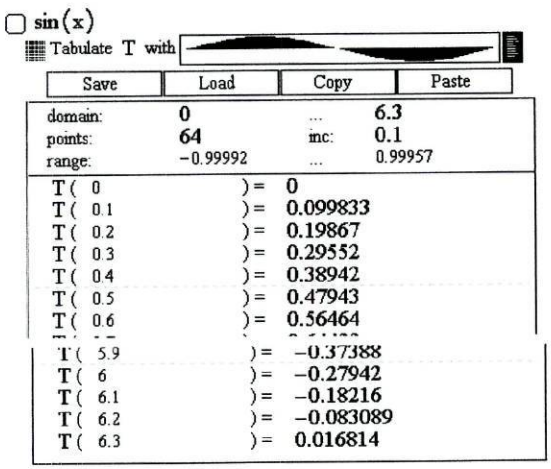
Há dois tipos de tabelas: as que dizem respeito a valores reais e as que dizem respeito a valores complexos. As primeiras contêm uma lista de valores reais para uma função e, as segundas, permitem trabalhar-se com a parte real e a parte imaginária de cada valor complexo, de uma forma individual. Ambos os tipos de tabelas contêm informações relativas ao domínio e à correspondência entre o domínio e a série de valores calculados. Vejamos um primeiro exemplo, em que está envolvida a função *sin(x)*:

Depois de introduzida a expressão ***sin(x)***, seleccionar a função ***Table/Generate*** no menu ***Manipulate***. A janela que ao lado aparece reproduzida abre-se de imediato. Nesta janela pode escolher-se o tipo de tabela (real ou complexa), o domínio (valores de x), o número de pontos com o espaçamento entre esses pontos e a variável que queremos considerar para valores do domínio. De seguida deve clicar-se em OK, aceitando-se os valores que são apresentados por defeito. A figura seguinte diz respeito à proposição referente à tabela em causa.



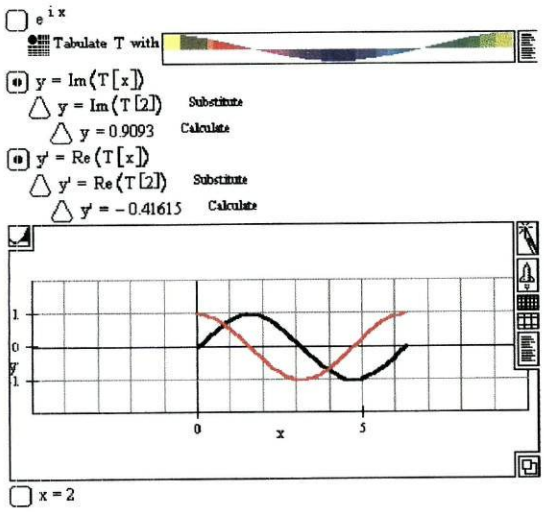


O ? que aparece na figura anterior, pode ser substituído por uma letra ou uma palavra que identificará a função que estamos a considerar. Depois de substituirmos tal símbolo por ***T***, por exemplo, deveremos escolher a função ***Clarify*** no menu ***Notebook*** e definir ***T*** como uma ***User Defined function***. Ao clicarmos no "botão" referente aos detalhes já conhecido aquando da abordagem da componente gráfica deste programa, aparecerá uma imagem como a que de seguida se apresenta.



Na realidade, a figura que aparece retratada não corresponde à totalidade da tabela que o programa criou, uma vez que essa era bastante extensa. Repare-se nos quatro "botões" situados no topo da tabela e que dizem respeito à importação e exportação de dados a que já se fez referência e que se explicará mais detalhadamente à frente.

A expressão ***domain*** define os valores a atribuir a x (ou seja, o domínio),

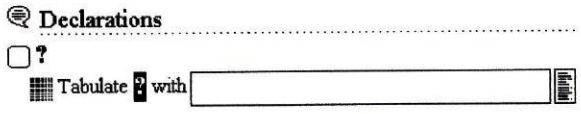


points refere-se ao número de valores do domínio usados para a criação da tabela espaçados por ***inc*** unidades e ***range*** apresenta os valores mínimo e máximo constantes da tabela. Com excepção dos valores referentes a ***range***, podemos alterar todos os outros, sendo certo que os que não foram alterados adaptar-se-ão aos novos valores. Só o número de pontos é que se mantém constante quando não é alterado directamente.

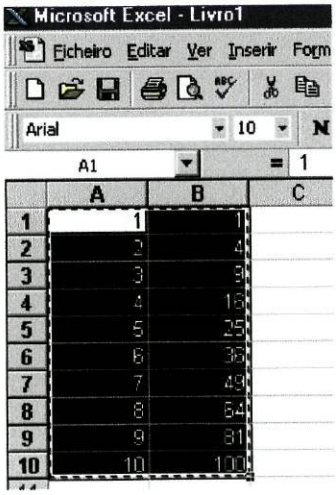
A imagem anterior, diz respeito a uma tabela complexa, referente à função $y = e^{i.x}$. Antes de mais nada, há que introduzir a expressão, seleccioná-la e gerar a tabela através da função ***Table/Generate*** disponível no menu

Manipulate. Desta vez a linha referente à tabela aparece a cores. Os valores que aparecem retratados na tabela, depois de se clicar no "botão" dos detalhes, são dois: um que diz respeito à parte real e outro que diz respeito à parte imaginária. Podemos também extrair da tabela as partes real (**Re**) e imaginária (**Im**), conforme se pode verificar na mesma figura. Compare, por exemplo, os valores obtidos através da geração da tabela com os obtidos através da definição das funções **Re** e **Im**. De referir apenas que, no momento da definição das funções, deverá garantir-se que, de facto, se tratam de funções ***Pre-defined***, aliás opção que aparece disponível por defeito.

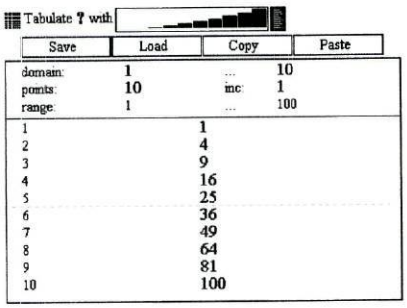
Na manipulação de tabelas, é possível a importação de dados de aplicações externas ao MathView. Podemos, por exemplo, gerar dados utilizando a folha de cálculo do MS Excel e transportar esses dados para uma tabela no MathView. Para tal, devemos fazer o seguinte: em primeiro lugar deve gerar-se uma tabela vazia, escolhendo a função ***Table/Generate*** no menu ***Manipulate***, fazendo ***OK*** na janela que surge por defeito e que diz respeito às definições da própria tabela.



De seguida vamos à folha de cálculo e seleccionamos as células referentes aos dados introduzidos, conforme se mostra na figura seguinte. De notar que a segunda coluna foi obtida a partir da primeira, calculando o quadrado dos números que constavam na coluna da esquerda. De seguida, fazemos ***copy*** no menu ***edit*** da folha de cálculo (ou clicando no *icon* respectivo). Voltamos ao programa MathView, abrimos os detalhes na nossa tabela vazia e clicamos no "botão" Paste (refira-se que não se deve escolher a função ***paste*** disponível no menu ***edit***).



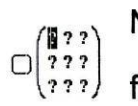
Aparecerá de imediato uma figura como a que de seguida se apresenta e que, como se vê, retrata uma função que assumiu os objectos como sendo os valores introduzidos na coluna da esquerda e as imagens os valores da coluna da direita. Uma última nota para referirmos que, caso se introduzam números com vírgulas (isto é, números com casas decimais), o programa interpreta-os como dois números. Por exemplo, se na folha de cálculo um dos números introduzidos for "38,54", o MathView interpreta-o como "38" e "54". Para evitar essa situação, deve-se configurar na folha de cálculo a vírgula como sendo "." (ponto).



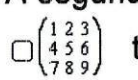
O inverso também é válido, ou seja, a exportação de dados duma tabela para ficheiros exteriores ao programa MathView também pode ser feita. Depois de se criar a tabela, e de se abrir os detalhes no botão respectivo, deve clicar-se no botão [Save]. De imediato abre-se uma caixa de diálogo na qual se deve especificar o ficheiro para onde se pretende exportar os dados. Os dados transportados serão apenas os valores referentes às imagens, ou seja, da coluna do lado direito da tabela.

Matrizes

Para a introdução de matrizes, existem dois métodos. O primeiro consiste em utilizar a palete das funções, clicando-se, sem largar, no símbolo respectivo ( ) até que se escolha a dimensão da matriz a introduzir no nosso Notebook.



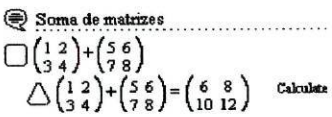
Na imagem ao lado, são visíveis os **?**, que se devem seleccionar por forma a se introduzirem os respectivos valores. A deslocação de posição em posição para a introdução dos valores, pode ser feita através da utilização do rato ou através da tecla **TAB**.

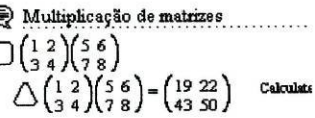


A segunda forma para a introdução dos valores, consiste em utilizar apenas o teclado. Por exemplo, para gerar a matriz que se mostra ao lado, basta introduzir os seguintes valores: 1 , 2 , 3 ; 4 , 5 , 6 ; 7 , 8 , 9 (ou seja, para se mudar para o valor da direita basta introduzir "," e para se mudar de linha deve introduzir-se ";").

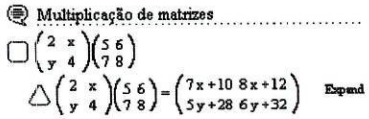
Operações com matrizes

As figuras seguintes mostram duas das operações que podem ser levadas a cabo com matrizes, bastando para tal introduzir as duas matrizes conforme se explicou anteriormente, separadas pelo sinal da respectiva operação (podia ser também efectuada a subtracção). Para efectuar a respectiva operação, deve escolher-se a função ***Calculate*** no menu ***Manipulate***, depois de se seleccionar a expressão a desenvolver.



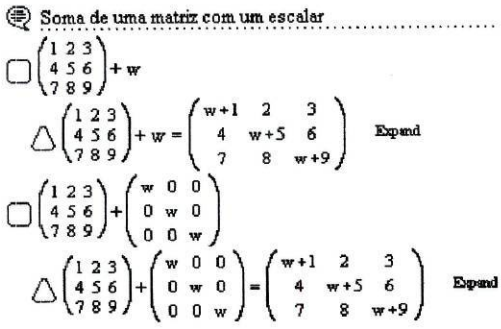


As operações adição, subtracção e multiplicação poderão ainda ser efectuadas de uma forma simbólica, ou seja, poderá visualizar-se os passos que são dados até se chegar ao resultado final. Por outro lado, a imagem ao lado mostra um exemplo em que alguns dos componentes das matrizes em causa



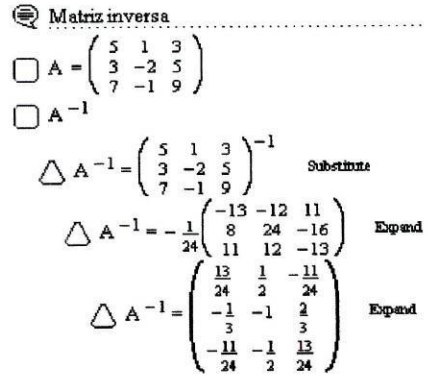
não são números mas sim parâmetros. Desta vez deve escolher-se a função ***Expand*** no menu ***Manipulate***.

O programa MathView considera um escalar **W** (definido pelo utilizador como sendo uma constante ou uma variável quando o programa pede que o utilizador o faça), como sendo uma matriz diagonal, ou seja, composta por "0" em todas as posições excepto na diagonal principal, onde aparece o escalar **W**.

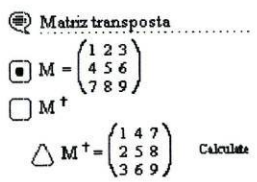


Matriz inversa e matriz transposta

O MathView permite-nos encontrar com facilidade a matriz inversa de uma matriz introduzida pelo utilizador. Podemos, como nos exemplos anteriores, operar directamente com matrizes ou podemos dar nomes às matrizes (A, B, etc). Vejamos o seguinte exemplo: antes de mais nada, há que introduzir a matriz, por exemplo **A=1,2,3;4,5,6;7,8,9**. De seguida, fazemos ***Clarify*** no menu ***Notebook*** e declaramos **A** como sendo um ***M-Linear Operator***. De seguida, numa nova proposição, introduzimos a expressão A^{-1}, referente à matriz inversa. Seleccionamos a matriz A (com um clique sobre o sinal "=") e arrastamo-la para cima da expressão A^{-1} . Depois, seleccionamos o segundo membro da expressão obtida e escolhemos a função ***Expand*** no menu ***Manipulate***. Finalmente, seleccionamos novamente o segundo membro da última expressão e fazemos, de novo, ***Manipulate/Expand***. A figura anterior retrata a situação descrita.



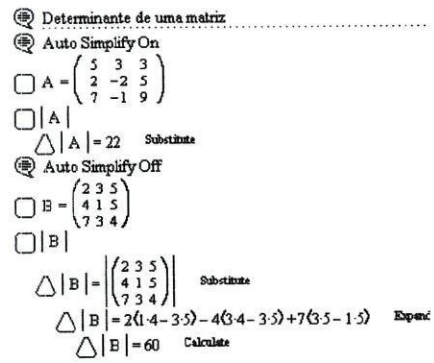
Quanto à matriz transposta de uma matriz, esta poderá ser obtida a partir de uma matriz introduzida, clicando no "botão" respectivo, disponível na palete das funções, a saber ($a^{\dagger}$).



Na figura ao lado, mostra-se como é que se pode chegar à matriz transposta da matriz M. Depois desta ter sido introduzida, numa nova proposição de introdução de dados clica-se no "botão" atrás referido e, no **?** respectivo introduz-se M. Selecciona-se essa expressão e escolhe-se a função ***Calculate*** no menu ***manipulate***.

Determinante de uma matriz

O mesmo poderá ser feito para obter o determinante de uma matriz. Depois de introduzida a matriz e, numa nova proposição, introduzida a expressão que nos dá o determinante (através do clique no botão |a|), seleccionamos a matriz (através de um clique sobre o sinal "=") e arrastamo-la para cima desta última expressão. Assim, teremos duas situações: caso esteja a função ***Manipulation Prefs/Auto Simplify*** no menu ***Manipulate*** activa, o resultado é imediato. Caso essa mesma função esteja inactiva, teremos que seleccionar o segundo membro da última expressão obtida e fazer ***Manipulate/Expand*** e, de seguida, ao segundo membro da expressão obtida ***Manipulate/Calculate***. A figura ao lado retrata essa situação. Obviamente que apenas poderemos obter determinantes em que o número de linhas é igual ao número de colunas. No caso de se tentar obter um determinante de uma matriz que não seja quadrada, o programa dá uma mensagem de erro.

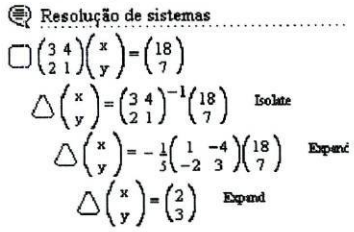


Resolução de sistemas utilizando matrizes

Para a resolução de sistemas de equações com múltiplas variáveis, para além do método de substituição que foi descrito anteriormente, podemos utilizar o cálculo matricial, embora este método possa apenas ser aplicado nos sistemas em que o número de equações é igual ao número de variáveis. Suponhamos, por exemplo, que queremos determinar a solução do sistema que de seguida se apresenta.

$$\begin{cases} 3x + 4y = 18 \\ 2x + y = 7 \end{cases}$$

Para tal, deveremos transportar o problema para a resolução de uma equação com matrizes: em primeiro lugar, deveremos introduzir as matrizes referentes aos coeficientes, às variáveis e aos valores que aparecem no segundo membro das equações, conforme se mostra na primeira linha da figura ao lado, correspondente à proposição de introdução dos dados. De seguida, seleccionamos a matriz referente às variáveis e fazemos ***Manipulate/Isolate***. Para terminar, seleccionamos o segundo membro da expressão obtida e fazemos ***Manipulate/Expand***, repetindo esta operação ainda à expressão que se irá obter. De imediato surgirá a solução pretendida. Caso pretendamos resolver outros sistemas, neste caso 2x2, basta ir às expressões inicias e alterar os coeficientes para os novos valores. De imediato a solução adaptar-se-à aos novos valores.



Quando introduzirmos coeficientes que darão origem a sistemas impossíveis ou indeterminados no corpo dos números Reais, os resultados obtidos serão os que constam das imagens seguintes. No caso dos primeiros, aparece o símbolo de "∞" em cada uma das soluções, enquanto que nos segundos, aparece um "**?**".

Sistemas impossíveis

$$\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 15 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}^{-1}\begin{pmatrix} 15 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \infty\begin{pmatrix} 1 & -2 \\ -2 & 4 \end{pmatrix}\begin{pmatrix} 15 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \infty \\ -\infty \end{pmatrix}$$

Sistemas indeterminados

$$\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 14 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}^{-1}\begin{pmatrix} 14 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \infty\begin{pmatrix} 1 & -2 \\ -2 & 4 \end{pmatrix}\begin{pmatrix} 14 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ? \\ ? \end{pmatrix}$$

Índices nas matrizes

Numa matriz, podemos fazer referência a um seu elemento qualquer, através de índices usuais. Por exemplo, dada uma matriz 4x4 M, M_3 representa a terceira linha dessa matriz. Por outro lado, $M_{(2,3)}$ representa o valor da terceira coluna que aparece na segunda posição. De referir apenas que, conforme se mostra na figura seguinte, a função que se utiliza para levar a cabo tais operações, é a função ***Calculate*** disponível no menu ***Manipulate***, depois de se introduzir e seleccionar a expressão que se pretende identificar.

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

$M_{(2,3)}$

$M_{(2,3)} = 7$ Calculate

M_3

$M_3 = (9, 10, 11, 12)$ Calculate

Produtos externo e interno

Podemos usar as potencialidades do MathView para levar a cabo operações com vectores, considerando, um vector no plano ou no espaço como uma matriz 1X2 ou 1X3, respectivamente.

A figura seguinte, mostra o resultado do produto externo de dois vectores, nas duas situações apontadas. De referir apenas que, para a introdução dos dados, podemos recorrer à palete das funções ou utilizar apenas o teclado. Na

Produto externo de dois vectores

$x = (1, 2, 4)$ $y = (1, 4, 5)$

$x \times y$

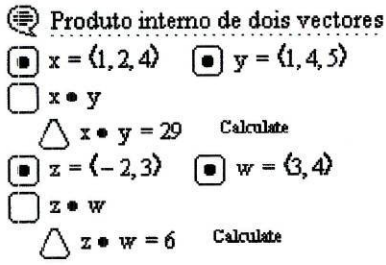
$x \times y = (-6, -1, 2)$ Calculate

$z = (-2, 3)$ $w = (3, 4)$

$z \times w$

$z \times w = -17$ Calculate

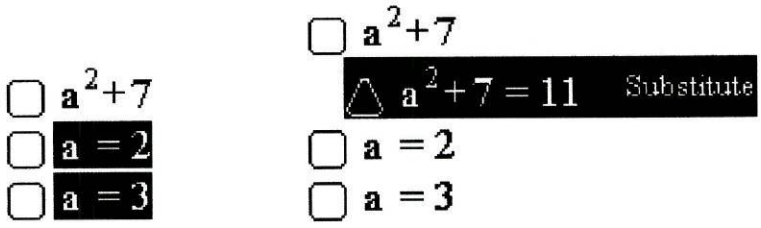
primeira situação, depois de introduzidos os dois vectores (ver introdução dos dados nas matrizes), clicamos no símbolo $\vec{a} \times \vec{b}$, numa nova proposição de introdução de dados. De seguida, basta substituir os **?** pelas letras respectivas, seleccionar a expressão e escolher a função ***Calculate*** no menu ***Manipulate***. Na segunda situação, deveremos digitar "**x * # y**", seleccionar a expressão e escolher a mesma função ***Manipulate/Calculate.***



Relativamente ao produto interno de dois vectores, poderemos também utilizar duas formas para a introdução dos dados: por um lado, à semelhança do que foi dito para o produto externo, utilizar o "botão" respectivo na palete das funções ($\vec{a} \cdot \vec{b}$) ou, por outro lado, utilizar apenas o teclado, através da sequência de teclas "**x * @ y**".

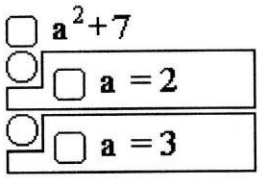
Case Theories

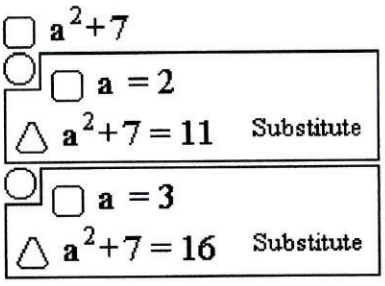
Com a utilização de um "***Case Theories***", é possível, por exemplo, trabalhar no mesmo Notebook, com áreas de trabalho diferentes. Por exemplo, no mesmo Notebook poderemos estar a considerar, em contextos diferentes, as expressões "a = 2" e "a =3". Vejamos o seguinte exemplo:



Na imagem da esquerda, estão visíveis três expressões, duas das quais estão seleccionadas (depois de se ter clicado no sinal "="). Se "empurrarmos" essas duas expressões que contêm dois valores possíveis de "a" para cima da expressão inicial, o que se obtém é o que visualiza na figura da direita, ou seja, o valor que é atribuído a "a" é o da primeira expressão seleccionada. Se

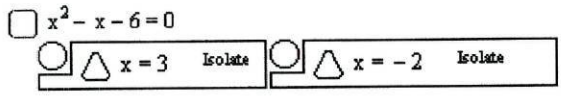
tivermos a função ***Always ReManipulate*** (no menu ***Manipulate***) activada, e se mudarmos o valor de "a" na primeira expressão de 2 para 5 (por exemplo), o resultado adapar-se-à a esse novo valor de "a". Mas, no caso de mudarmos o valor de "a" na segunda expressão de 3 para 5 (por exemplo), nada acontecerá ao resultado, já que, como se disse, o valor assumido foi apenas o da primeira expressão. Para podermos trabalhar e visualizar os resultados com ambos os valores de "a" no mesmo Notebook, deveremos criar dois ***Case Theories***, cada um deles contendo um valor para "a". Para isso, deveremos, em primeiro lugar, seleccionar a expressão "*a=2*" (com um clique sobre o sinal "=") e escolher ***Case Theory*** na função ***Insert***, no menu ***Notebook***. De seguida, repetir a mesma operação para a expressão "*a=3*", aparecendo uma figura como a que de seguida se apresenta (imagem da esquerda). Para obtermos os resultados, deveremos seleccionar os dois valores de "a" (um de cada vez) e sobrepô-los na expressão inicial.





De facto, os dois ***Case Theory*** são independentes um do outro, havendo a possibilidade de trabalhar simultaneamente com os dois valores de "a".

Ainda relativamente a esta função, há a referir que, no caso de termos activada a função ***Auto Casing*** (acessível no menu ***Manipulate/Manipulation Prefs***), cada operação que se faça que tenha mais do que uma solução, cria automaticamente um ***Case Theory***, com as duas soluções. A título de exemplo, fica a imagem seguinte, referente à resolução de uma equação do 2º grau (que desenvolveremos mais adiante).



Depois de se introduzir a expressão, deve activar-se a função ***Auto Casing*** no menu ***Manipulate/Manipulation Prefs*** e, de seguida, escolher-se a função ***Isolate*** disponível no menu ***Manipulate***, depois de se terem seleccionado os

dois "xx's". Caso a função ***Auto Casing*** não estivesse activa, apareceria apenas uma das soluções.

Formas de definir uma função

No MathView, as funções adquirem três formas: definidas explicitamente, definidas pelo utilizador e pré-definidas.

Funções definidas de uma forma explícita:

As funções definidas de uma forma explícita são obtidas através da manipulação de equações, o que, aliás, já ocorreu atrás, por exemplo, na resolução de sistemas de duas equações e duas incógnitas. Para representar graficamente a equação em baixo, deve isolar-se a variável dependente de todas as variáveis independentes. Isso consegue-se seleccionando a variável dependente "y" (com duplo clique sobre ele) e isolando-a de três formas possíveis. A primeira pode ser obtida, escolhendo no menu ***Manipulate*** a função ***Isolate***, a segunda clicando no símbolo correspondente na palete das funções e que, como já se disse anteriormente é o símbolo e, finalmente, pode isolar-se igualmente o "y", arrastando-o (com o cursor na forma de que se obtém colocando-o sobre o "y" seleccionado e premindo ao mesmo tempo a tecla **CTRL**) para cima do quadrado existente do lado esquerdo (quadrado da introdução de dados). Por um desses métodos, obtemos a seguinte função:

Exemplo de uma função definida de uma forma explícita

$-2x = x^2 - y + 12$

$y = x^2 + 2x + 12$ Isolate

Funções definidas pelo utilizador

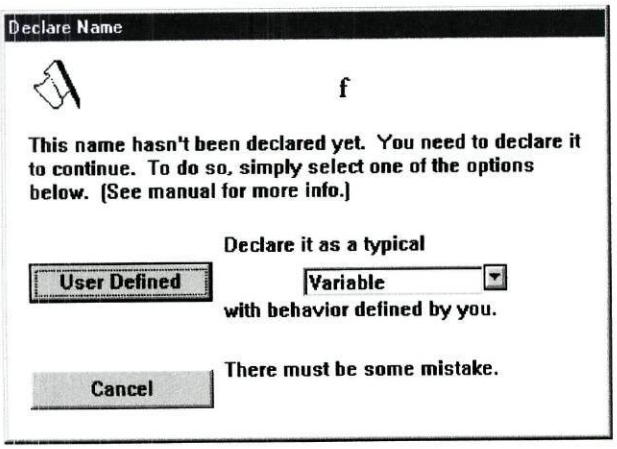
As funções definidas pelo utilizador são introduzidas com a notação mais usual nas funções. No entanto, deveremos usar uns caracteres especiais

(*Wildcards*), como aqueles que aparecem na figura em baixo, e que podem ser obtidos usando a palete das funções ou usando apenas o teclado, conforme de seguida se explica.

Exemplo de uma função definida pelo utilizador

$f(x) = x^2+3x-33$

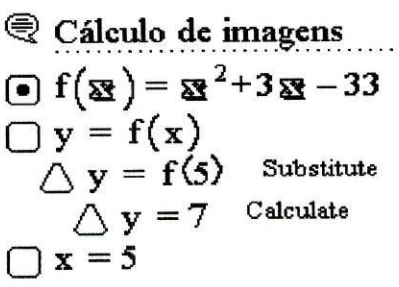
Para se obter o símbolo x basta escrever **?x** ou, na palete das variáveis, clicar sem largar no símbolo x e procurar a letra correspondente. No entanto, apesar da função se encontrar já definida, não pode ser utilizada sem antes ser declarada enquanto tal. Depois de introduzida a função como mostra a figura anterior, deverá ir-se ao menu ***Notebook*** escolher-se a função ***Clarify***, aparecendo de imediato a janela seguinte.

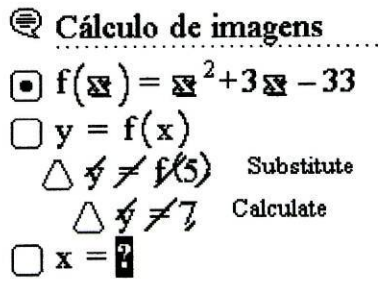


No campo que nos é apresentado na janela ao lado, escolhemos a opção ***"Function"*** nas várias opções que nos são apresentadas e clicamos no botão ***User Defined***. Fica, deste modo, a nossa função perfeitamente definida.

Para o cálculo de imagens, fazemos em primeiro lugar, y=f(x), desta vez sem os caracteres especiais que foram utilizados para definirmos a função. De seguida, teremos que introduzir o valor do qual queremos saber a imagem (por exemplo x = 5). Seleccionamos esta última expressão (com um clique sobre o sinal "=") e arrastamos para cima do "x" que aparece na expressão y=f(x) (de notar que o quadrado de introdução de dados que aparece do lado esquerdo da expressão y=f(x) não pode ficar seleccionado). Para obtermos o valor pretendido, falta apenas seleccionar o membro direito da última expressão obtida e, no menu ***Manipulate*** fazer ***Calculate***. A imagem da esquerda da

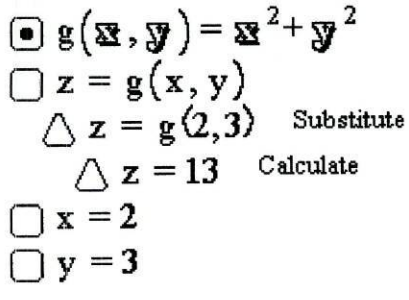
sequência seguinte, reflecte os vários passos que são obtidos através das operações atrás descritas.





A imagem da direita da sequência anterior, foi obtida quando apagamos o "5" da expressão inicial (seleccionando-o com um clique e fazendo **delete**), por forma a encontrar outras imagens de outros objectos. O rasurado que aparece na imagem, permanece até que um novo valor para o "x" seja introduzido.
À semelhança do que acontece para a substituição dos valores dos objectos, podemos também mudar a nossa função, seleccionando a anterior e alterando-a, aparecendo de imediato o valor da imagem pretendida.

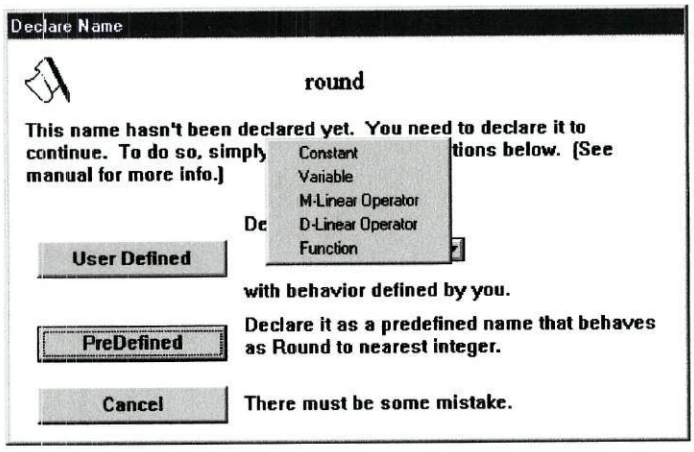
Podemos também definir funções com mais do que uma variável independente para, por exemplo, esboçarmos gráficos 3-D. Da mesma forma que se definiram as funções anteriores (com apenas uma variável independente) no que respeita aos caracteres especiais utilizados (Wildcards), definem-se estas funções, conforme se mostra na figura ao lado. Depois há que introduzir os valores de "x" e de "y" e substituir na expressão **z=g(x,y)** e escolher a função ***Calculate*** no menu ***Manipulate***, depois de seleccionado o segundo membro desta última expressão.



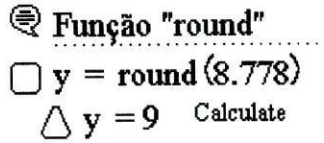
Funções Pré-definidas

O programa tem cerca de 47 funções pré-definidas, bem como várias variáveis, constantes e mesmo regras com funções conhecidas, tais como, entre outras, as trigonométricas e as logarítmicas.

Um exemplo de uma função pré-definida é a função ***round*** que consiste na obtenção do arredondamento, às unidades, do número que é dado. Depois de se introduzir a função e de se seleccionar o segundo membro da expressão, fazemos ***Manipulate/Calculate*** e obtemos a janela que ao lado se apresenta, de modo a possibilitar que se declare a função introduzida como pré-definida. Aparece por defeito a opção "***Function***" seleccionada mas, como mostra também a figura, há outras opções a considerar. No nosso caso aceitamos "***Function***" e clicamos no botão "***PreDefined***".

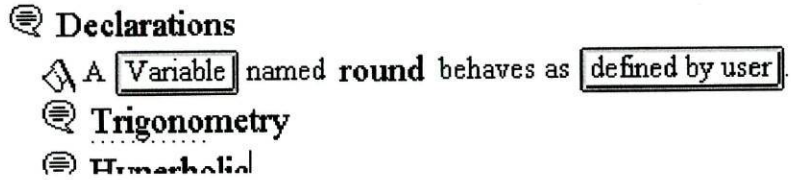


A figura seguinte retrata a operação a que anteriormente se fez referência.



Por outro lado, caso tenhamos previamente definido ***round*** como, por exemplo, uma variável definida pelo utilizador, não podemos utilizar ***round*** como uma função pré-definida para arredondar um número às unidades, conforme se descreveu anteriormente. Se o pretendermos fazer, deveremos clicar no balão Declarations de modo a obter-se algo como de seguida se apresenta. Ao clicarmos no "botão" Variable outras opções surgirão, devendo ser escolhida a

pretendida, ou seja, ***Function***. Ao clicarmos no "botão [defined by user] deveremos escolher a opção ***Numeric Funcs/Round to nearest integer***.

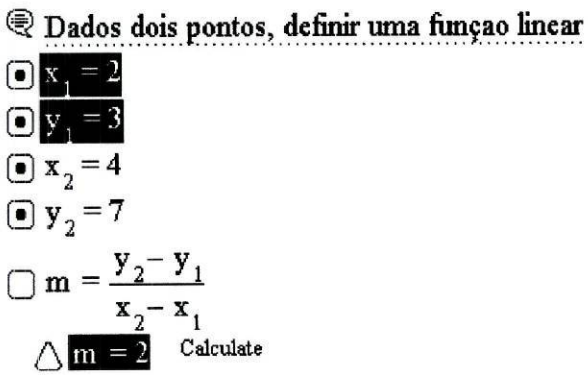


As funções no Ensino Secundário

Funções lineares

Já vimos num dos pontos anteriores a representação gráfica de funções e, inclusivamente, um dos exemplos avançados foi mesmo o de uma função do tipo $y = mx + b$. Vimos também, para esse tipo de funções, que o facto de podermos mudar, quantas vezes pretendermos, os valores de ***m*** e de ***b***, ajustando-se automaticamente o gráfico aos novos valores, permite que os alunos façam, mais facilmente, as suas conjecturas, nomeadamente no que concerne ao comportamento do declive e da ordenada da origem.

Relativamente ainda a funções lineares, vejamos agora como as definir usando apenas dois pontos, por exemplo os pontos de coordenadas (2 , 3) e (4 , 7).



Para isso, há que primeiro introduzir, separadamente, as coordenadas dos dois pontos. Apenas refiro dois pormenores que poderão ser importantes para a introdução dos dados. O primeiro tem a ver com os índices. Para os obtermos, basta escrevermos a letra correspondente (neste caso "x" ou "y"), clicar no icon a_b e introduzir o índice respectivo, fazer **ESC** para sair do índice e continuar a introdução dos dados. O segundo pormenor, relaciona-se com a introdução da fracção que aparece depois do "m=". Depois de escrevermos o numerador, conforme explicado anteriormente, seleccionámo-lo com o rato e clicamos na tecla "/" (divisão). Automaticamente, o cursor passará para o denominador.

Depois de feita a introdução desses dados, seleccionamos a fracção que aparece na última expressão com um clique sobre o sinal "=", e fazemos ***Calculate***.

De seguida, seleccionamos as três expressões como na figura anterior e, com o cursor sobre uma delas e premindo a tecla **CTRL** até aparecer o símbolo ☜, arrastamos a selecção feita para cima do quadrado da introdução de dados da primeira expressão da figura seguinte, que entretanto foi introduzida. Depois seleccionamos o "y", fazemos ***isolate*** e, finalmente, ***expand***. Fica, assim, determinada a expressão analítica de uma função linear, dados dois pontos. De referir que esta resolução consiste na mesma resolução que se efectua quando se está a utilizar papel e lápis.

$$\square\ m = \frac{y - y_1}{x - x_1}$$

$$\triangle 2 = \frac{y-3}{x-2} \quad \text{Substitute}$$

$$\triangle\ y = 2(x-2)+3 \quad \text{Isolate}$$

$$\triangle\ y = 2x - 1 \quad \text{Expand}$$

Funções quadráticas

Já vimos atrás uma forma de resolver equações do 2º grau, ou seja, de determinar as raízes de uma função quadrática, usando a função ***Manipulation Prefs/Auto Casing*** no menu ***Manipulate***. Vejamos agora outros métodos para a mesma determinação, que poderão ser utilizados conforme o que se pretender obter do aluno. Consideremos, por exemplo, a equação $2x^2 + 8x = 6$.

Método da factorização

Antes de mais nada, há que passar o 6 para o primeiro membro. Podemos fazê-lo de duas formas: seleccionar o "6" (com um clique) e com a tecla **CTRL** premida e o cursor na forma ☜, arrastá-lo para o primeiro membro (figura seguinte, à esquerda); ou então usando a função ***Apply***. Para tal, há que seleccionar a expressão (com um clique sobre o sinal "="), escolher a função ***Apply*** no menu ***Manipulate*** e acrescentar +(- 6). Depois basta fazer

Manipulate/Simplify, dando origem à sequência de expressões na figura em baixo, à direita. De referir ainda que a função ***Apply*** também pode ser obtida na palete das funções através da tecla  e a função ***Simplify*** através da tecla  .

$$\square 2x^2+8x=6$$
$$\triangle 2x^2+8x-6=0 \quad \text{Move Over}$$

$$\square 2x^2+8x=6$$
$$\triangle (2x^2+8x)-6=6-6 \quad \text{Apply}$$
$$\triangle 2x^2+8x-6=0 \quad \text{Simplify}$$

Depois de termos a equação com o segundo membro igual a "0", seleccionar todas as parcelas do primeiro membro e fazer ***Manipulate/Factor***, ou então, na palete das funções clicar na tecla respectiva,  . Depois, seleccionar um dos "x's" que aparece no primeiro membro e fazer ***Manipulate/Isolate***. De seguida, seleccionar o outro "x" e fazer também ***Manipulate/Isolate***. A imagem seguinte mostra a sequência das expressões que se obtêm.

Método da factorização

$$\square 2x^2+8x=6$$
$$\triangle 2x^2+8x-6=0 \quad \text{Move Over}$$
$$\triangle 2(x+\sqrt{7}+2)(x-\sqrt{7}+2)=0 \quad \text{Factor}$$
$$\triangle x=-\sqrt{7}-2 \quad \text{Isolate}$$
$$\triangle x=\sqrt{7}-2 \quad \text{Isolate}$$

Método "completar o quadrado"

Consideremos a mesma equação utilizada no método anterior. Depois de introduzida a expressão, fazemos ***Apply*** (depois de clicarmos no sinal "=" para que a expressão fique seleccionada). Aparecem os dois membros seleccionados e, nessa altura, dividimos ambos os membros pelo coeficiente do termo de maior grau, neste caso por 2. Depois fazemos ***Simplify*** e, de seguida, ***Expand***. A figura seguinte mostra essa sequência de passos efectuados.

Método "completar o quadrado"

$2x^2+8x=6$

$\frac{2x^2+8x}{2}=\frac{6}{2}$ Apply

$\frac{1}{2}(2x^2+8x)=3$ Simplify

$x^2+4x=3$ Expand

Seguidamente, fazemos novamente ***Apply*** e adicionamos a ambos os membros o quadrado da metade do coeficiente do termo em "x" (neste caso 2^2, escolhemos ***Simplify*** e fazemos ***Factor*** depois de seleccionarmos o primeiro membro da última expressão obtida. Finalmente, seleccionamos x e com o ***Auto Casing*** accionado, usamos a função ***Isolate***. A figura que se segue, mostra-nos estes últimos passos, que resultaram na obtenção das soluções pretendidas.

$(x^2+4x)+2^2=3+2^2$ Apply

$x^2+4x+4=7$ Simplify

$(x+2)^2=7$ Factor

$x=\sqrt{7}-2$ Isolate $\quad x=-\sqrt{7}-2$ Isolate

Método da fórmula resolvente

Como sabemos, as soluções de uma equação quadrática (do tipo $ax^2+bx+c=0$), são dadas pela expressão $x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ e, para aplicarmos este método, deveremos primeiro de tudo, introduzir os respectivos coeficientes, como na figura seguinte.

Método da fórmula resolvente

$a=2$

$b=8$

$c=-6$

Depois introduzimos a primeira expressão correspondente a uma das soluções, seleccionamos os três coeficientes simultaneamente (com um clique sobre o sinal "=" de cada uma das expressões com a tecla **SHIFT** premida) e, arrastamos com o cursor na forma de ☜ (premindo a tecla **CTRL**) os três coeficientes para cima daquela expressão. À expressão obtida fazemos ***Expand***. Finalmente, procedemos de igual modo para a expressão que nos dá a outra solução da equação inicial.

$$\square\ x = \frac{-b+\sqrt{b^2-4ac}}{2a} \qquad \square\ x = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

$\triangle\ x = \frac{1}{4}\left(\sqrt{112}-8\right)$ Substitute $\qquad \triangle\ x = \frac{1}{4}\left(-\sqrt{112}-8\right)$ Substitute

$\triangle\ x = \sqrt{7}-2$ Expand $\qquad \triangle\ x = -\sqrt{7}-2$ Expand

Método gráfico

Para podermos utilizar este método, temos, antes de mais, que manipular a equação de maneira que seja obtido algo na forma de função. Vamos, então, primeiro seleccionar o "0" e fazer ***Isolate*** para que passe para o primeiro membro. Depois, com um clique sobre ele, seleccionamos o "0" e substituímo-lo por "y", conforme mostra a figura seguinte.

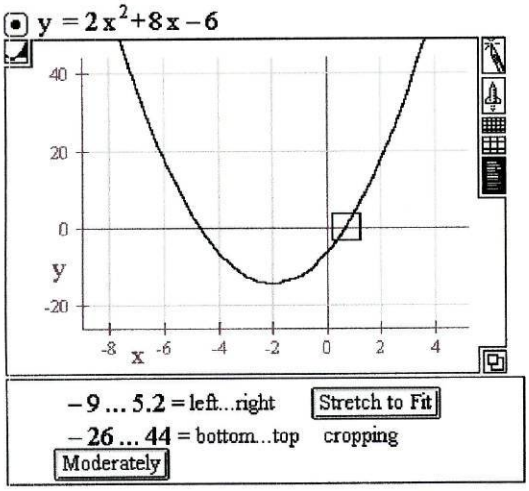
Método gráfico

$\square\ 2x^2+8x-6=0$

$\triangle\ 0 = 2x^2+8x-6$ Isolate

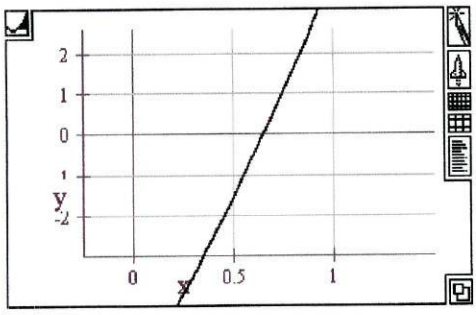
$\square\ y = 2x^2+8x-6$

De seguida, esboçamos o gráfico da função obtida e alteramos os limites de enquadramento do gráfico, de acordo, por exemplo, com aqueles valores que são mostrados na figura seguinte.



Seguidamente, deve ampliar-se a zona referente à intersecção do gráfico com o eixo dos xx's, através da utilização da "faca". Fazer isso em primeiro lugar, por exemplo, à solução positiva.

A figura da esquerda da sequência seguinte, mostra a parte do gráfico obtida depois de feita a referida ampliação.



$x = -4.6458$

$x = -\sqrt{7} - 2$ UnCalculate

$y = 0$

$x = 0.64575$

$x = \sqrt{7} - 2$ UnCalculate

$y = 0$

Para obtermos as soluções que se apresentam na figura da direita, vamos ao menu ***Manipulate*** e escolhemos a função ***Find Graph Root***. Antes de proceder a qualquer outra manipulação, escolha ***Undo*** no menu ***Edit***, aparecendo, de novo, o gráfico original antes de se ter feito a ampliação. De seguida, pelo facto da solução ser apresentada sob a forma de número decimal, seleccionamos o segundo membro da expressão $x = -4.6458$ e escolhemos, no menu ***Manipulate*** as funções ***Other/UnCalculate***. Finalmente, procedemos de igual modo para a outra solução, ficando assim encontradas as duas soluções que são apresentadas na figura da direita da sequência anterior.

De notar que, por vezes, quando fazemos a selecção com a "faca", o valor de "Y" não é exactamente zero mas sim um valor muito próximo. Nestas situações, deve-se usar a função ***UnCalculate*** para forçar o aparecimento de $y = 0$. O que o MathView faz neste caso é procurar uma solução exacta do valor que entretanto foi arredondado. De salientar que esta função, em algumas situações, chega a resultados que não correspondem à verdade.

Função exponencial e função logarítmica

Antes de usar expoentes e logaritmos, devemos familiarizarmo-nos com a introdução dos dados e perceber algumas das potencialidades do MathView, também nesta área. Há duas funções no menu ***Manipulate/Manipulation Prefs***, a função ***AutoSimplify*** e a função ***AutoCasing***, que poderão trazer grandes vantagens na manipulação de expoentes e logaritmos, nomeadamente

na apresentação das suas propriedades e que, de seguida, serão apresentadas.

Antes de mais nada, para introduzirmos um expoente no MathView, podemos fazê-lo utilizando a palete das funções, através do clique sobre o símbolo respectivo (a^c), ou através do teclado com símbolo convencional ^. Para sair do expoente deve premir-se a tecla **ESC**.

Vejamos agora algumas das capacidades das funções referidas anteriormente. Por exemplo, para se gerar os factores do monómio x^5, deve seleccionar-se a expressão, desactivar a função ***Auto Simplify***, e fazer ***Manipulate/Collect***. A imagem seguinte reflecte o resultado dessas operações.

$$\square\ x^5$$
$$\triangle\ x^5 = x\,x\,x\,x\,x \quad \text{Collect}$$

Para voltarmos à situação inicial, basta seleccionarmos o segundo membro da expressão obtida depois de se proceder à operação ***Collect***, e fazer ***Simplify***. De referir ainda que, o utilizador consegue activar ou desactivar a função ***Auto Simplify***, escolhendo no menu ***Manipulate*** a função ***Manipulation Prefs/Auto Simplify***.

A tabela seguinte resume as propriedades envolvidas com o estudo dos expoentes, através da utilização do MathView. Na primeira coluna, é apresentada a parte da expressão que é seleccionada e a função que é levada a cabo para se chegar ao resultado pretendido. As segunda e terceira colunas, indicam-nos o estado da função ***Auto Simplify*** (On ou Off) para que seja possível efectuar a manipulação pretendida. De notar que, em alguns casos, a manipulação é conseguida em ambos os casos.

Propriedade	Selecção e Manipulação	Auto Simplify ON	Auto Simplify OFF
Prop #1	$b^m b^n$ $b^m b^n = b^{m+n}$ Simplify	✓	✓
	b^{m+n} $b^{m+n} = b^m b^n$ Collect		✓
Prop #2	$\frac{b^m}{b^n}$ $\frac{b^m}{b^n} = b^{m-n}$ Simplify	✓	
	b^{m-n} $b^{m-n} = b^m b^{-n}$ Collect $b^{m-n} = b^m \frac{1}{b^n}$ Simplify	✓	✓
Prop #3	b^0 $b^0 = 1$ Simplify	✓	✓
Prop #4	$(a\,b)^m$ $(a\,b)^m = a^m b^m$ Expand	✓	✓
	$a^m b^m$ $a^m b^m = (a\,b)^m$ Factor	✓	✓
Prop #5	$(b^m)^n$ $(b^m)^n = b^{n\,m}$ Simplify	✓	✓
	$b^{m\,n}$ $b^{m\,n} = (b^m)^n$ Collect		✓

Prop #6	$\square \left(\frac{a}{b}\right)^n$ $\triangle \left(\frac{a}{b}\right)^n = \frac{a^n}{b^n}$ Expand	✓	✓
	$\square b^{-n}$ $\triangle b^{-n} = \frac{1}{b^n}$ Simplify	✓	✓
Prop #8	$\square \left(b^{\frac{1}{2}}\right)^2$ $\triangle \left(b^{\frac{1}{2}}\right)^2 = b^{\frac{1}{2}} b^{\frac{1}{2}}$ Collect $\triangle \left(b^{\frac{1}{2}}\right)^2 = b$ Simplify	✓	✓ ✓
Prop #9	$\square b^{\frac{1}{2}}$ $\triangle b^{\frac{1}{2}} = \sqrt{b}$ Simplify	✓	✓
Prop #10	$\square \sqrt{b}$ $\triangle \sqrt{b} = b^{\frac{1}{2}}$ Expand		✓

De referir ainda que, os casos notáveis da multiplicação, apresentados na figura seguinte, também podem ser conseguidos. Os dois primeiros obtêm-se fazendo ***Expand*** e a função ***Auto Simplify*** desactivada e, por sua vez, o terceiro, fazendo ***Factor*** com a função ***Auto Simplify*** activada.

Casos notáveis da multiplicação

$\square (a+b)^2$

$\triangle (a+b)^2 = a^2 + b^2 + 2ab$ Expand

$\square (a-b)^2$

$\triangle (a-b)^2 = 2(-ab) + a^2 + b^2$ Expand

$\square a^2 - b^2$

$\triangle a^2 - b^2 = (a+b)(a-b)$ Factor

Números escritos na notação científica

Vejamos primeiro a forma de introduzir os números na notação científica. Por exemplo, para introduzirmos a expressão 2.34×10^{7}, podemos fazê-lo de duas formas. A primeira, consiste na introdução habitual da expressão **2.34 * 10 ^ 7** (de notar apenas que, depois de se introduzir 2.34, a operação multiplicação poderá ser obtida através da tecla "*" ou através da tecla de espaços); a segunda consiste na introdução de **2.34 e 7**, sendo este método mais rápido.

$$\square\ 2.34\cdot10^{7}$$
$$\triangle\ 2.34\cdot10^{7} = 23400000 \quad \text{Expand}$$

A expressão obtida em ambos os casos, pode ser visualizada acima, bem como, o resultado da aplicação da função ***Expand***.

Também podemos mostrar os passos obtidos através de operações aritméticas simples com números escritos na notação científica, mas, para isso, temos que desactivar a função ***Auto Simplify***. O exemplo que está traduzido na figura seguinte, mostra a adição, multiplicação e divisão de dois números (x e y) escritos na notação científica, com uma precisão de 5 dígitos.

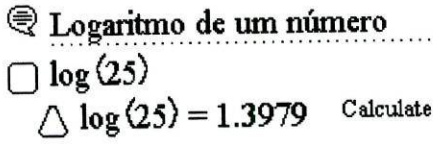
Operações com números escritos na notação científica

$$\square\ x = 1.1111\times10^{10}$$
$$\square\ y = 2.2222\times10^{10}$$
$$\square\ x + y$$
$$\triangle\ x + y = 1.1111\times10^{10} + 2.2222\times10^{10} \quad \text{Substitute}$$
$$\triangle\ x + y = 3.3333\times10^{10} \quad \text{Simplify}$$
$$\square\ x\,y$$
$$\triangle\ x\,y = 1.1111\times10^{10}\cdot 2.2222\times10^{10} \quad \text{Substitute}$$
$$\triangle\ x\,y = 2.4691\times10^{20} \quad \text{Simplify}$$
$$\square\ \frac{x}{y}$$
$$\triangle\ \frac{x}{y} = \frac{1.1111\times10^{10}}{2.2222\times10^{10}} \quad \text{Substitute}$$
$$\triangle\ \frac{x}{y} = 0.5 \quad \text{Simplify}$$

Apenas umas breves considerações sobre a sequência de operações apresentada na figura anterior. Para levar por diante a operação ***Substitute***, temos que seleccionar os dois valores introduzidos para x e para y, com um clique sobre o sinal "=" e a tecla **SHIFT** pressionada. Depois, deve pressionar-se, sem largar, a tecla **CTRL** e arrastar os dois valores para cima da expressão da qual se pretende o valor.

Logaritmos

O logaritmo comum, base 10, é uma função pré-definida neste programa. Obtém-se escrevendo ***log*** utilizando o teclado, ou premindo a tecla respectiva na palete das funções. Por exemplo, para calcularmos o valor de ***log(25)***, podemos fazê-lo das seguintes formas: em primeiro lugar, basta introduzir e seleccionar o "25", clicar em ***log(x)*** na palete das funções e, finalmente fazer ***Calculate***. A segunda forma, consiste na introdução de ***log(25***, seleccionar e fazer também ***Calculate***. A figura seguinte traduz o resultado dessas operações.



Podemos também utilizar logaritmos com outras bases, por exemplo ***b***. Para o introdução desses valores, podemos utilizar dois métodos. No primeiro, depois de escrevermos log, fazemos **SHIFT** + _ e depois introduzimos o ***b***. Premimos a tecla **ESC** para sairmos da base e, de seguida, introduzimos o número do qual queremos o logaritmo. No segundo método, escrevemos ***log***, na palete das funções clicamos na tecla correspondente aos índices a_b e premimos a tecla **ESC** para sairmos do índice. Finalmente, introduzimos o número do qual pretendemos o logaritmo.

De salientar ainda que, no caso de pretendermos usar a base Neperiana (***e***), podemos utilizar a função $\log_e(x)$ em que a base é o número de Neper **e**, ou então, podemos utilizar a função pré-definida $\ln(x)$. A imagem que se segue representa o cálculo de logaritmos com bases diferentes de 10.

Logaritmos de bases diferentes de 10

$\log_e(7)$

$\log_e(7) = 1.9459$ Calculate

$\ln(7)$

$\ln(7) = 1.9459$ Calculate

$\log_7(343)$

$\log_7(343) = 3$ Calculate

À semelhança do que se fez para as operações com expoentes, apresentam-se de seguida as propriedades dos logaritmos, que são obtidas através da operação ***Expand***, com a função ***Auto Simplify*** activa.

Propriedades dos Logaritmos

$\log_b(m\,n)$

$\log_b(m\,n) = \log_b(m) + \log_b(n)$ Expand

$\log_b\left(\frac{m}{n}\right)$

$\log_b\left(\frac{m}{n}\right) = \log_b(m) - \log_b(n)$ Expand

$\log_b(m^n)$

$\log_b(m^n) = n \log_b(m)$ Expand

$\log_b(b^x)$

$\log_b(b^x) = x$ Expand

Vejamos agora duas técnicas para a resolução de equações envolvendo expressões com expoentes e logaritmos. No primeiro exemplo que a seguir se apresenta, devemos seleccionar o "x" (com um clique sobre ele), fazer ***Isolate*** e, de seguida, ***Simplify*** (após ter-se seleccionado o segundo membro da última expressão obtida). De referir que a função ***Auto Simplify*** deve estar desactivada. No segundo exemplo, com a função ***Auto Simplify*** activada, devemos seleccionar a expressão (com um clique sobre o sinal "=") e fazer ***Apply***. De seguida, introduz-se o função ***log*** e faz-se ***Expand***. Na expressão

obtida isola-se o "x" (através da função ***Isolate***) e, finalmente, depois de se seleccionar o segundo membro da última expressão fazemos ***Calculate***.

Auto Simplify OFF

$$2^x = 16$$
$$x = \log_2(16) \quad \text{Isolate}$$
$$x = 4 \quad \text{Simplify}$$

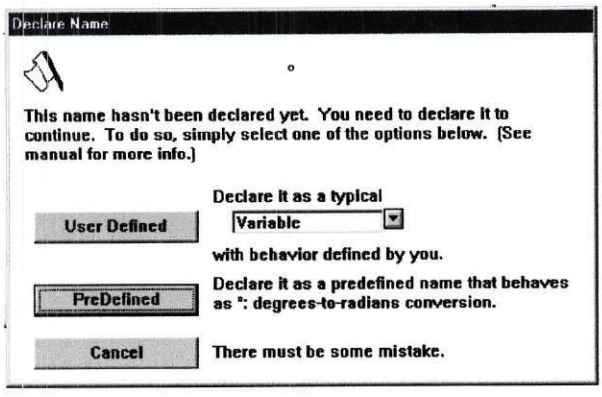
Auto Simplify ON

$$2^x = 16$$
$$\log(2^x) = \log(16) \quad \text{Apply}$$
$$\log(2)x = \log(16) \quad \text{Expand}$$
$$x = \frac{\log(16)}{\log(2)} \quad \text{Isolate}$$
$$x = 4 \quad \text{Calculate}$$

Funções trigonométricas

O MathView usa Regras de Transformação para implementar identidades trigonométricas. O programa tem 12 funções trigonométricas pré-definidas, 6 das quais tratando-se de funções inversas. Além disso, é possível acedermos a fórmulas previamente introduzidas, bem como introduzir outras que nos irão interessar para uma tarefa específica.

A medida que é usada para se fazer referência aos ângulos é dada, por defeito, em radianos. Porém, podem também usar-se os ângulos em graus, sendo necessário fazer a devida transformação. Aliás, quando se pretende, por exemplo, calcular o valor de $sin(30^o)$, após se seleccionar ***Manipulate/Calculate*** aparece a janela que ao lado se apresenta, onde teremos que declarar o símbolo de grau (°) como pré-definido. De referir ainda que, para introduzir a expressão **sin(30°)**, devemos escrever sin(30'o), em que "o" é a letra e não zero (0). A expressão *sin* também pode ser obtida, directamente, na palete das funções.



Um processo que poderia ser utilizado para se efectuar a transformação de radianos em graus e vice-versa, era através da criação de duas funções definidas pelo utilizador, de que já foi falado anteriormente. As duas figuras que se seguem mostram exactamente isso. Depois de se introduzirem as funções, deverá escolher-se a função ***Clarify*** no menu ***Notebook*** e declarar ambas as funções como ***User Defined***.

$$\text{grauspararadianos}(①) = \frac{\pi\,①}{180}$$

grauspararadianos(30)

$$\text{grauspararadianos}(30) = 0.5236$$

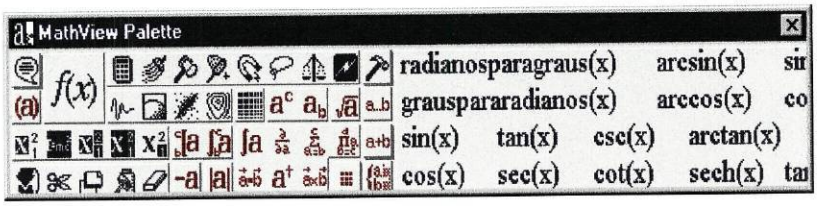
$$\text{grauspararadianos}(30) = \frac{1}{6}\pi$$

$$\text{radianosparagraus}(ℙ) = \frac{180ℙ}{\pi}$$

$$\text{radianosparagraus}\left(\frac{\pi}{6}\right)$$

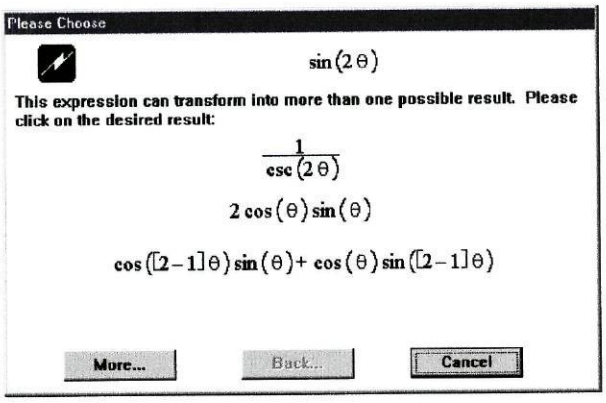
$$\text{radianosparagraus}\left(\frac{\pi}{6}\right) = 30$$

De seguida, é só introduzir os valores pretendidos, encontrando-se de imediato, o resultado procurado. Tal como se pode verificar na figura seguinte, na palete das funções, depois de terem sido declaradas as novas funções, estas aparecem visíveis.



O MathView tem as funções trigonométricas básicas definidas internamente, proporcionando ainda ao utilizador a flexibilidade suficiente para este definir mais relações trigonométricas, tais como regras de transformação.

Por exemplo, se pretendermos desenvolver a expressão $\sin(2\,\theta)$, após a sua introdução, deveremos escolher ***Transform*** no menu ***Manipulate***. De imediato, aparece a janela que de seguida se apresenta, dando ao utilizador várias opções de transformação da expressão inicial.



Funções definidas por ramos

Vejamos agora como esboçar gráficos de funções definidas por ramos. Primeiro deveremos introduzir a expressão analítica da função. Para tal, depois de escrevermos "y =", clicamos no icon da palete das funções, aparecendo de imediato a figura que de seguida se apresenta, que corresponde ao aspecto da nossa função.

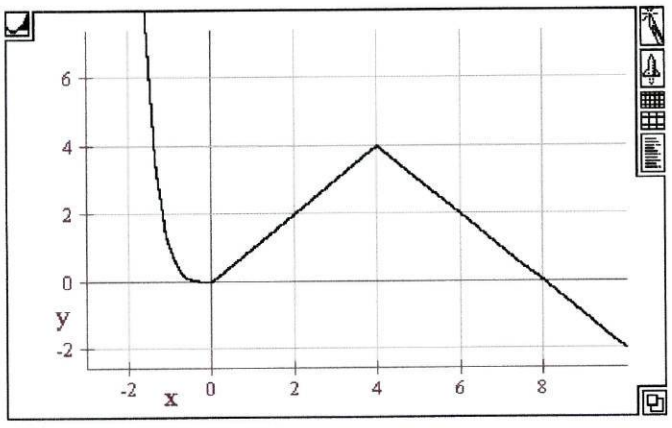
$$\square\, y = \begin{cases} ? & (? < 0) \\ ? & (? \geq 0) \end{cases}$$

A figura seguinte, representa um exemplo de uma função da qual pretendemos a representação gráfica. No entanto, antes de continuarmos, será conveniente chamar-se a atenção para alguns pormenores inerentes à sua introdução.

$$\square\, y = \begin{cases} x^4 & (x \leq 0) \\ |x| & (x \leq 4) \\ -x+8 & (x > 4) \end{cases}$$

Para escrevermos x^4, clicamos uma vez no primeiro "*?*" e introduzimos a respectiva expressão. Para o preenchimento do que aparece em (?<0), basta seleccionarmos o que aparece entre parêntesis e escrevemos a condição que queremos, por exemplo x<=0 (de notar que ao escrevermos os dois símbolos <

e = seguidos, o programa transforma-os automaticamente em $\leq$). Fazemos a mesma coisa para a segunda linha da função mas aqui há a destacar que a função $|x|$ obtém-se clicando no icon |a| da palete das funções. Finalmente, para introduzirmos uma terceira linha como no nosso exemplo, teremos que posicionar o cursor na parte final da segunda linha e introduzir ";" (ponto e vírgula). Depois é só fazer ***Graph/y=f(x)/linear*** e a função aparece representada graficamente, como de seguida se apresenta. De facto, a facilidade com que a representação gráfica de qualquer função é obtida e a facilidade com que a manipulação dos dados introduzidos é efectuada, fazem deste programa um instrumento com grandes potencialidades para o ensino da Matemática.



De salientar ainda que a expressão analítica das funções definidas por ramos não é escrita da mesma forma que estamos habituados. De facto, em condições normais, o domínio da expressão central da função em análise (|x|), seria [0,4] e não]-∞,4] como aparece escrito. Acontece que, o MathView assume que os domínios das diferentes expressões não se intersectam, ou seja, o limite esquerdo da expressão do meio nunca poderia ser inferior a "0" (limite superior da primeira expressão).

Algumas convenções de escrita

Expressão	Tecla / sequência de teclas
π	**ALT** + **CTRL** + p (PI)
45º	45 ' o (45 graus)
$25x^2+45x$	2 5 * x ^ 2 **ESC** + 4 5 * x
$x_n=2n+2$	X _ n **ESC** = 2 * n + 2
$\sqrt{45}$	Sqrt (4 5 (raíz quadrada de 45)
$\int$	? x
2x	2 * x ou 2 **SPACE** x (dois vezes x)
X x Y	X * # Y (produto externo)
X • Y	X * @ Y (produto interno)
∞	' N (infinito)

CAPÍTULO IV - Páginas interactivas com o MathView

Este capítulo será dedicado à apresentação de modelos criados com o programa MathView. Para determinados conteúdos programáticos dos Ensinos Básico e Secundário, foram criados módulos de software (Notebooks) que poderão constituir uma forma interessante de se abordar os temas em questão. Na versão em papel, estão simplesmente apresentadas as imagens que constituem esses Notebooks, não sendo possível, como é óbvio, a sua manipulação.

Preliminares

Os capítulos anteriores foram essenciais para que um professor de Matemática, independentemente do ciclo que lecciona, possa elaborar algumas tarefas a propor aos alunos. De facto, a Internet e a Matemática Interactiva podem funcionar, por um lado, como motivação e introdução de determinado conteúdo programático e, por outro, como uma ferramenta utilizável na abordagem, na íntegra, de um tema específico.

Este capítulo destina-se a mostrar alguns exemplos do que se poderá fazer com a Internet, no Ensino da Matemática. As tarefas que aqui são apresentadas destinam-se, na sua grande maioria, a alunos do Ensino Básico e Ensino Secundário, e para a sua realização, foram utilizados apenas os comandos da linguagem HTML, de forma directa ou indirecta, e algumas rotinas em Javascript de modo a que se controlassem algumas questões de aspecto. Foi usado o programa MathView na elaboração da generalidade dos exemplos, nomeadamente ao nível da representação gráfica de funções e manipulação algébrica de expressões matemáticas.

Ao longo deste capítulo, à medida que vão sendo necessárias manipulações específicas de algum comando, serão apresentadas as devidas instruções. De referir ainda que, conforme se disse anteriormente, para que o utilizador possa

manipular este tipo de cenários criados com o programa Mathview, não necessita de adquirir o próprio programa. Bastará instalar o *plugin* no seu computador, para conseguir visualizar as figuras que aqui são apresentadas, figuras essas que foram criadas com o referido programa. Para tal, poderá aceder ao site www.livemath.com, onde poderá, gratuitamente fazer o seu download.

Uma das vantagens deste programa reside precisamente no facto de se poder, a qualquer momento, alterar os valores que surgem na imagem quando a página que a contém é carregada, obtendo-se de imediato a actualização dos cálculos ou do gráfico correspondente. Chama-se a atenção de que, em certos casos, poderão surgir alguns problemas de visualização dos resultados depois de alterados os valores iniciais. Já se fez referência a este facto no Capítulo III.

Falou-se também no Capítulo III que, muitas das operações poderiam ser obtidas através das paletes de variáveis e das funções. Ora, quando se utiliza o *plugin* para a simples visualização de ficheiros criados com o programa, como está ocorrer no momento em que se visualiza este texto, obviamente que essas paletes não estão disponíveis. Assim, as alterações aos cenários apresentados só são possíveis através da utilização do teclado e/ou do rato.

Quanto aos exemplos que aqui são apresentados, preocupamo-nos em reunir neste capítulo não só os mais elementares, que se resumem a simples cálculos numéricos, mas também passando pela representação gráfica de funções, resolução de equações até ao terceiro grau, entre outros. Tratam-se apenas de alguns exemplos daquilo que a Internet e a WWW, neste caso com o auxílio do *plugin* do MathView, nos permite fazer, a nós, professores de Matemática.

De salientar que, a generalidade dos exemplos aqui apresentados são acompanhados de um pequeno texto cuja finalidade será enquadrá-los. Sempre que se abre determinada página que contém imagens criadas com o MathView a que correspondem ficheiros "*.thp", este é carregado da forma inicial como foi criado. Se fizermos quaisquer alterações, depois de fecharmos a respectiva página, ela volta à fase inicial. Ora, se porventura tiver problemas

na alteração de determinados valores, se se enganou e estragou o ficheiro inicial não sendo mais possível qualquer alteração, pode voltar a uma outra página e, de seguida, regressar à mesma.

Informações importantes

À medida que serão apresentados os exemplos, são expostas algumas das instruções básicas necessárias à alteração dos dados que são fornecidos. No entanto, neste espaço, antes de avançarmos para qualquer caso concreto, chamar-se-à a atenção do utilizador para alguns aspectos específicos e necessários à edição dos ficheiros do MathView que são carregados por defeito. Em primeiro lugar, uma referência aos símbolos que se seguem.

	Este balão acompanha um comentário que surge com o ficheiro.
	Este quadrado de cantos arredondados indica uma proposição na qual foi introduzida determinada expressão.
	Este triângulo de cantos arredondados indica uma conclusão. Aparece a acompanhar uma expressão que surge após se ter realizado determinada operação.

Selecção de objectos

Para se proceder a qualquer alteração de um ficheiro originário do MathView, de modo a que os cálculos se adaptem ao novo cenário, podemos fazer de duas formas. Por exemplo, na expressão

$$\square 2x+3+4=5,$$

se pretendermos seleccionar o algarismo "3", basta fazermos um clique sobre ele. Para introduzirmos outro valor, ou apagamos com a tecla **BACKSPACE** ou introduzimos por cima o novo valor. Se pretendermos seleccionar, por exemplo, a expressão "3+4", fazemos um clique sobre o "3" e, sem largar o botão esquerdo do rato, deslocamos até ao "4". Se pretendermos seleccionar a expressão na sua totalidade, basta fazermos um clique sobre o sinal "=".

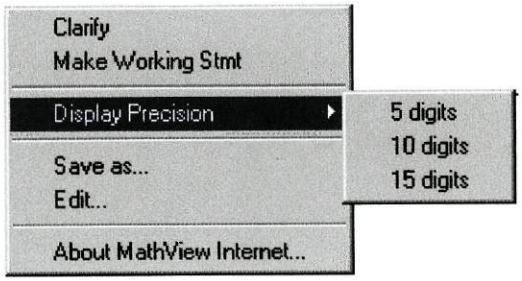
Introdução de expoentes

Para a introdução de expressões que envolvam expoentes, por exemplo 3^2+4.x, a sequência de teclas a utilizar é a seguinte: 3 ^ 2 ESC + 4 * x. De ressaltar o facto de, em certos casos, a introdução de expoentes não ser possível ou ser susceptível de provocar erros. Isto ocorre quando, por exemplo, a expressão inicial a ser alterada, não tiver qualquer expoente.

Botão direito do rato

A janela que de seguida se apresenta surge quando, sobre uma imagem que diz respeito a um ficheiro criado com o programa MathView (de extensão "*.thp"), se pressiona o botão do lado direito do rato. Das operações a que o utilizador poderá ter acesso, destacam-se:

- *Clarify* que permite, em certos casos, actualizar a informação do gráfico que não é conseguida de uma forma automática;
- *Display Precision* que permite escolher o número de dígitos com que os resultados são apresentados;
- *Save as...* que permite guardar o ficheiro com o nome que pretendermos;
- *Edit* que permite alterar o ficheiro inicial, caso o utilizador tenha instalado na sua máquina o programa MathView e que, portanto, não seja detentor apenas do *plugin* que permite visualizar este tipo de ficheiros.



Esboço de gráficos

Com a utilização do programa MathView, o esboço de gráficos é bastante bem conseguido. A manipulação é de extrema facilidade e as conjecturas a tirar pelos alunos poderão ser obtidas de uma forma mais rápida, eficaz e motivadora.

Do lado direito de cada gráfico, aparecem os botões que de seguida se reproduzem e que permitem levar a cabo determinadas operações no próprio gráfico que é apresentado, nomeadamente:

	Esta faca permite fazer um zoom (zoom in) em qualquer parte do gráfico. Depois de se clicar neste símbolo, deve colocar-se o cursor sobre o gráfico, pressionar sem largar o botão esquerdo do rato e arrastá-lo ao longo da zona do gráfico que se pretende ampliar.
	Este foguetão permite reduzir o gráfico (zoom out), ou seja, faz o inverso daquilo que é conseguido com a faca. A diferença é que, para accionar tal operação basta clicar sobre esse símbolo.
	Este rectângulo com a rede mais cerrada, aumenta o número de pontos que foram usados na elaboração do gráfico, aumentando a sua resolução.
	Este rectângulo com a rede menos cerrada, diminui o número de pontos que foram usados na elaboração do gráfico.
	Colocando, sem largar, o cursor sobre este símbolo, podemos ajustar o tamanho do gráfico, conforme o nosso gosto e/ou necessidade.

 Na versão integral do programa, para além dos símbolos anteriores, existe o símbolo mostrado ao lado e que permite manipular algumas propriedades do gráfico, como, por exemplo, a escala de cada um dos eixos, a cor e espessura da linha que constitui o gráfico.

Cálculos algébricos

Com o MathView, os cálculos tornam-se bastante mais compreensíveis do que se fossem efectuados com uma calculadora (simples ou científica). De facto, a actualização desde os resultados de cálculos intermédios até aos resultados finais, sempre que se alteram os valores iniciais, é automática o que, poderá constituir um grande auxílio na aprendizagem de certos conceitos.

Operações com fracções

Neste primeiro exemplo, pretende-se que o aluno efectue operações (adição e subtracção) em que as parcelas são fracções. Para tal, bastará ao aluno alterar os valores do numerador e/ou do denominador, de uma ou das duas parcelas que são apresentadas no Notebook, obtendo de imediato o resultado final.

Para alterar os valores que são apresentados neste exemplo, deverá seleccionar o valor que pretende alterar e substituí-lo pelo valor pretendido. Repare que o resultado adapta-se de imediato aos novos valores. Além disso, o resultado poderá ainda ser obtido da forma que mais nos convém, isto é, da forma de número decimal ou fraccionário.

Um processo simples para levar a cabo subtracções, consiste em seleccionar a fracção respectiva (através de um clique sobre o sinal "+") e digitar o sinal "-".

Experimente mudar também o número de dígitos com que é apresentado o resultado, escolhendo a opção *Display Precision* depois de pressionar o botão direito do rato em cima do espaço da tela ocupado pelo Notebook. Terá então a oportunidade de escolher uma das três opções disponíveis (5, 10 ou 15 dígitos), apesar de, o programa permitir, na altura da concepção dos Notebooks iniciais, a utilização de mais opções.

Resultado obtido sob a forma de número decimal

$\frac{1}{5}+\frac{3}{7}$

$\frac{1}{5}+\frac{3}{7}=0.62857$

Resultado obtido sob a forma de fracção

$\frac{1}{5}+\frac{3}{7}$

$\frac{1}{5}+\frac{3}{7}=\frac{22}{35}$

Operações com polinómios

Com este Notebook, o aluno poderá trabalhar os casos notáveis da multiplicação de polinómios, conteúdo programático do 3º ciclo do Ensino Básico. O aluno poderá modificar quaisquer valores que são apresentados, em qualquer dos casos notáveis. Poderia optar-se por, caso seja oportuno, substituir os dois primeiros casos notáveis apenas por um, uma vez que a única diferença entre os dois é o sinal que separa as duas parcelas. Para tal, experimente substituir no segundo caso notável a segunda parcela, seleccionando "-4" e apagando-o com a tecla **DELETE** e, de seguida, introduza o novo valor, desta vez com sinal positivo.

Casos notáveis da multiplicação

$(2x+3)^2$ Quadrado da soma

$(2x+3)^2=(2x+3)(2x+3)$

$(2x+3)^2=4x^2+12x+9$

$(3x-4)^2$ Quadrado da diferença

$(3x-4)^2=(3x-4)(3x-4)$

$(3x-4)^2=9x^2-24x+16$

x^2-9 Diferença de quadrados

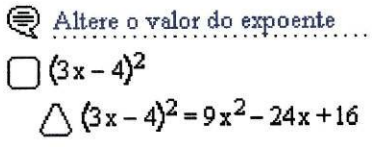
$x^2-9=(x+3)(x-3)$

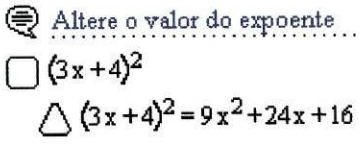
Experimente também substituir a primeira e segunda parcelas por *a* e *b* respectivamente, obtendo-se, desta forma, o caso geral do desenvolvimento dos casos notáveis da multiplicação.

De referir ainda que, no 3º caso notável, se se substituir a segunda parcela por um valor positivo, irão aparecer no desenvolvimento do caso notável valores complexos. O aluno deverá, portanto, estar prevenido para esse facto.

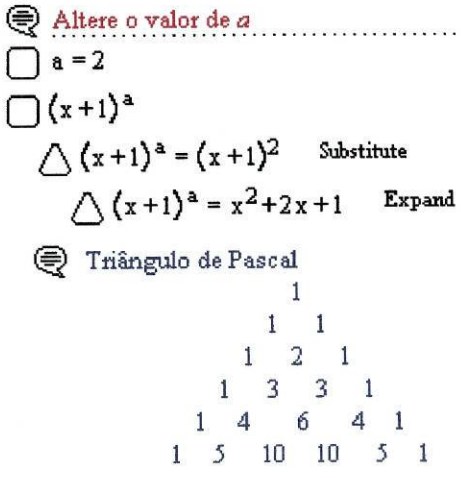
Poderá ainda experimentar introduzir, em cada um dos casos notáveis, valores fraccionários. De notar que, para tal, deverá seleccionar o número (introduzido por defeito), pressionar a tecla "/" e depois introduzir o denominador.

Nos Notebooks seguintes, são apresentados os desenvolvimentos do quadrado de uma diferença e do quadrado de uma soma. Poderá ainda, por exemplo, alterar o valor do expoente por outro número inteiro e ver o desenvolvimento de tal polinómio. Sugere-se que, para os expoentes, por questões de memória do próprio programa se utilizem números inteiros menores que 19.





De seguida, apresenta-se o desenvolvimento da expressão $(x+1)^a$, em que *a* é um número inteiro (para não exceder os limites de memória do programa, sugere-se que não atribua a *a* valores superiores a 18). Ora , os coeficientes do polinómio obtido do desenvolvimento da expressão $(x+1)^a$, correspondem aos algarismos que aparecem na (a+1)-ésima linha do Triângulo de Pascal. No Notebook que de seguida se apresenta, o aluno poderia obter o polinómio referente à expressão referida, alterando, ele próprio, o valor dos expoentes. É apresentado na parte inferior do Notebook, o Triângulo de Pascal com 19 linhas.



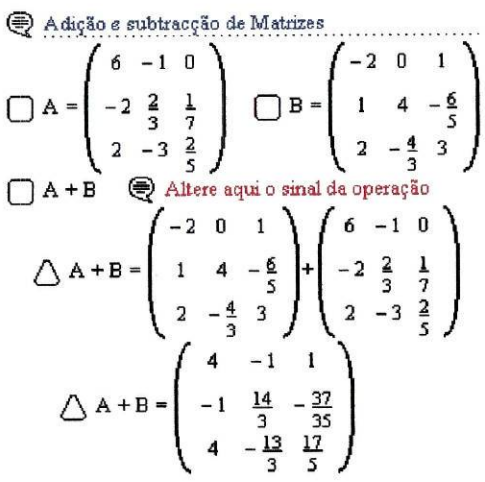
Cálculo Matricial

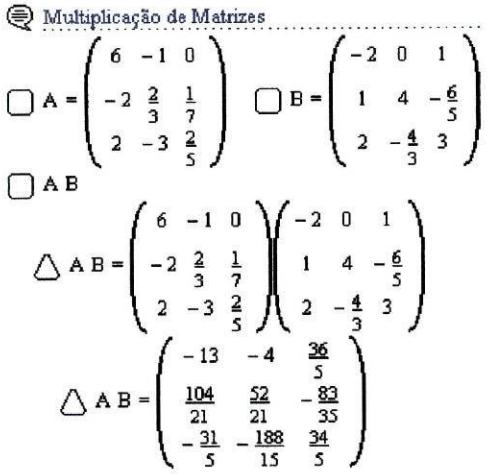
Nesta secção fica claro que cálculos que geralmente são aborrecidos e fastidiosos de efectuar, como são o caso de cálculos em que estão envolvidas matrizes, tornam-se bastante mais simples com o uso do MathView. De

seguida são apresentados alguns exemplos de operações em que estão envolvidas matrizes *3 X 3*.

Soma e produto de matrizes

No primeiro exemplo, são dadas duas matrizes e o resultado da sua soma. O aluno poderá, não só, mudar os valores que constituem qualquer das matrizes, mas também mudar a operação de adição para subtracção. Para levarmos a cabo multiplicação de matrizes, devemos ter em atenção que o número de colunas da primeira matriz seja igual ao número de linhas da segunda matriz. A matriz resultante desta multiplicação terá o número de colunas da primeira matriz e o número de linhas da segunda. Poderá ver também um exemplo de multiplicação de matrizes *3 X 3*.

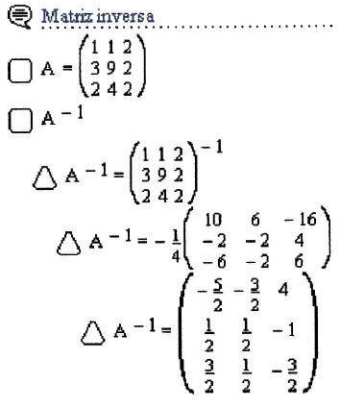




Inversa de uma matriz

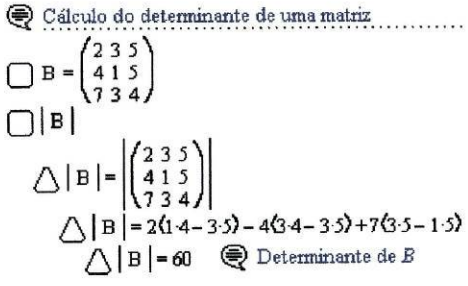
No exemplo seguinte, dada uma matriz obtém-se a respectiva matriz inversa. Altere qualquer um dos valores que constituem a matriz *A* e, de imediato, a matriz inversa adapta-se a esses novos valores. De notar que a matriz inversa de uma dada matriz *A*, é uma matriz designada por A^{-1}, tal que $A.A^{-1}$ é igual à matriz identidade (todos os valores são 0, excepto na diagonal principal que são 1). Ainda de referir que para uma matriz possuir matriz inversa, deverá ser

uma matriz quadrada, isto é, uma matriz cujo número de linhas é igual ao número de colunas.



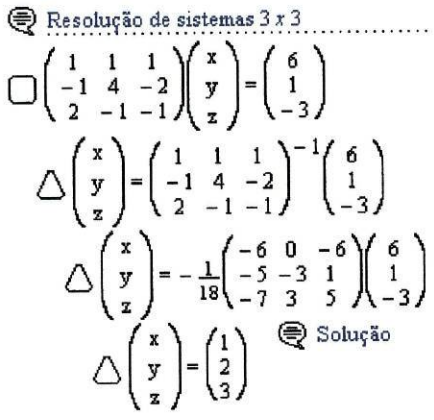
Determinante de uma matriz

No exemplo seguinte, calcula-se o determinante de uma matriz. Experimente mudar os valores que são apresentados por defeito.



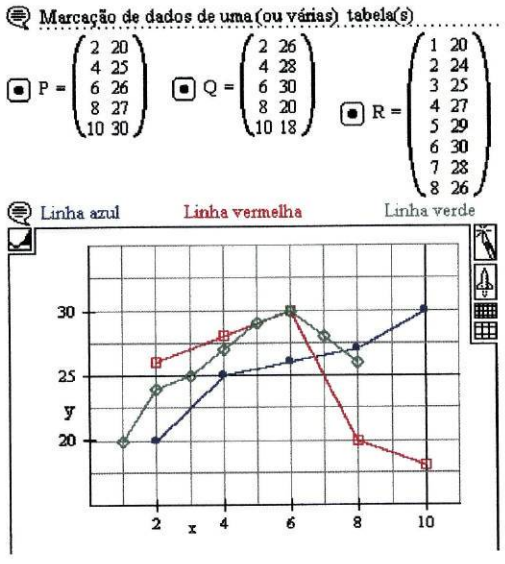
Resolução de sistemas

A resolução de sistemas escritos na forma de matrizes, é o exemplo que de seguida se mostra. Experimente alterar os valores incluídos na primeira matriz e que dizem respeito aos coeficientes envolvidos no respectivo sistema. Experimente alterar esses coeficientes e veja que a solução se adapta de imediato aos novos valores.



Representação gráfica de matrizes de duas colunas

Relativamente à representação gráfica de dados de matrizes, no Notebook seguinte apresenta-se um exemplo de três matrizes, cujos valores aparecem representados num mesmo sistema de eixos. Além disso, os diferentes pontos referentes à mesma matriz são unidos por uma linha. Experimente alterar alguns valores que são fornecidos por defeito e visualize as alterações produzidas.



Resolução de equações

Tal como noutros conteúdos programáticos, na resolução de equações podemos optar por mostrar ao aluno maior ou menor número de passos necessários à sua resolução. Isto para equações de todo o tipo: 1º, 2º, 3º graus ou superior, equações trigonométricas, exponenciais e logarítmicas.

Equações do 1º grau

Seleccione um qualquer dos valores que são apresentados e substitua-o por um valor à sua escolha. Evidentemente que, em termos de poupança de tempo na resolução deste tipo de equações, não se obtêm grandes vantagens com esta abordagem. No entanto, quando o aluno vai alterando os valores que aparecem na equação, poderá, na fase de aprendizagem, permitir que ele verifique as suas próprias conjecturas.

Por outro lado, pode ainda seleccionar toda a expressão (no primeiro caso), e introduzir outra do mesmo tipo.

Altere os valores na equação inicial

$\square\ 2x-5=3$

$\triangle\ 2x=8$

$\triangle\ x=4$

Altere os valores de a, b e c

$\square\ ax+b=c$ $\square\ a=2$ $\square\ b=-5$ $\square\ c=3$

$\triangle\ 2x-5=3$

$\triangle\ 2x=8$

$\triangle\ x=4$

Equações do 2º grau

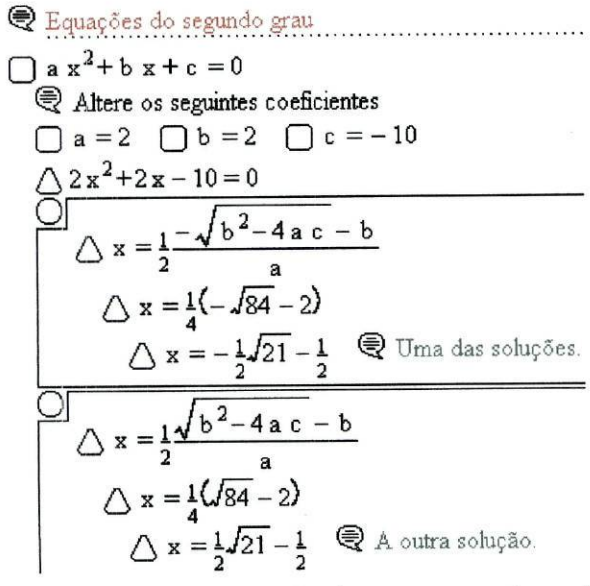
As equações do 2º grau podem ser resolvidas de variadíssimas formas, conforme se referiu no Capítulo III. Definidos os objectivos que se pretendem alcançar com os alunos, a forma de abordagem deste (e doutros) conteúdo programático é decisiva. Podemos simplesmente pretender as soluções de uma equação do segundo grau, bastando para isso introduzir os coeficientes,

ou podemos explicar como surge a fórmula resolvente e, consequentemente, como se obtêm as soluções. Veja então as diferentes formas de abordagem.

Este primeiro exemplo, pretende apenas dar a conhecer as soluções de uma equação do segundo grau. Claro que, para alunos do Ensino Secundário, não será, na generalidade das vezes, necessário proceder à dedução da fórmula resolvente, tanto mais que qualquer calculadora gráfica, cuja utilização na sala de aula e em exames de índole nacional é já permitida, já fornece as soluções após a introdução dos respectivos coeficientes.

Contudo, o estudo da função quadrática torna-se bastante enriquecedor de modo a que o aluno perceba muito do que acontece ao nível gráfico das funções em geral, nomeadamente, das funções polinomiais.

No exemplo seguinte, as soluções são obtidas de uma forma directa, bastando para isso, introduzirem-se os três coeficientes. Assim, seleccione o(s) coeficiente(s) que pretende modificar e introduza o(s) novo(s) valor(es). Repare que as soluções aparecerão de acordo com os novos coeficientes e, consequentemente, de acordo com a nova equação.

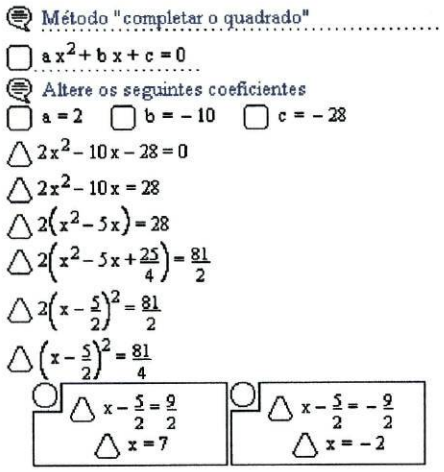


De notar que, para alguns casos, as soluções que aparecem são valores complexos. Ou seja, a respectiva equação é impossível no conjunto dos números reais. Trata-se de uma das particularidades do programa: o utilizador não pode restringir-se apenas ao corpo dos números reais. Poderá ser uma primeira oportunidade de se abordar com os alunos o facto de haver equações impossíveis no corpo dos números reais.

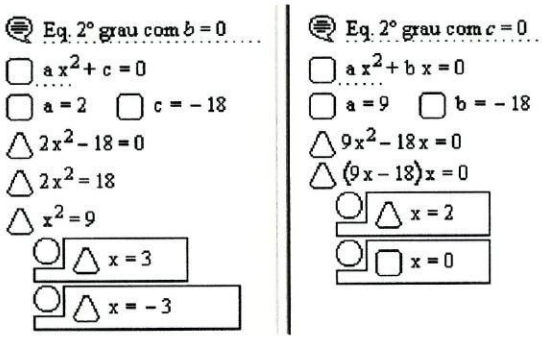
De salientar que, em algumas equações a expressão que dá cada uma das soluções aparece repetida. Isso acontece pelo facto do programa não

necessitar, para esses casos, de fazer simplificações ao resultado e, como tal repete a solução (experimente, por exemplo, atribuir os valores de a=1, b=1 e c=-6).

O segundo exemplo, poderá constituir uma forma mais didáctica de abordar este tema. O aluno, através do método que consiste em completar o quadrado que aparece a dada altura na resolução da equação do segundo grau, visualiza os passos necessários à determinação das soluções. De notar que, o caso genérico foi escondido com um duplo clique sobre a primeira expressão, que aparece seguida de reticências.



Nos Notebooks que se seguem, são retratados os casos de equações do 2º grau em que *b = 0* ou *c = 0*.



Equações do 3º grau

Foi criado um Notebook para a resolução de equações do terceiro grau. Apesar do seu desempenho não se mostrar de todo eficaz, é possível determinar as soluções exactas de determinadas equações do terceiro grau, nomeadamente

aquelas em que as soluções são inteiras, apesar de uma aproximação decimal das raízes ser sempre possível.

A leitura da respectiva página torna-se, de certa forma, difícil, na medida em que não estão escondidos alguns passos intermédios para que o utilizador possa visualizar o desempenho do programa na resolução da equação. Experimente alterar os coeficientes que são dados por defeito e veja os resultados obtidos. É apresentada apenas parte do Notebook, por não ser possível, devido ao seu tamanho, apresentar a totalidade da figura.

Equações do terceiro grau

◉ a = 1 ◉ b = −5 ☐ c = 7 ☐ d = −3

△ $x^3-5x^2+7x-3=0$

☐ $a\,x^3+b\,x^2+c\,x+d=0$

△ $\frac{a\,x^3+b\,x^2+c\,x+d}{a}=\frac{0}{a}$

△ $\frac{1}{a}\,d+\frac{c\,x}{a}+\frac{b\,x^2}{a}+x^3=0$

☐ $x=y-\frac{\frac{b}{a}}{3}$

△ $\frac{1}{a}\,d+\left(y-\frac{1}{3}\frac{b}{a}\right)^3+\frac{c\left(y-\frac{1}{3}\frac{b}{a}\right)}{a}+\frac{b\left(y-\frac{1}{3}\frac{b}{a}\right)^2}{a}=0$

△ $\frac{1}{a}\,d+\left(\frac{1}{3}\frac{b^2\,y}{a^2}-\frac{b\,y^2}{a}+y^3-\frac{1}{27}\frac{b^3}{a^3}\right)+\frac{c\left(y-\frac{1}{3}\frac{b}{a}\right)}{a}+\frac{b\left(y-\frac{1}{3}\frac{b}{a}\right)^2}{a}=0$

△ $\frac{1}{a}\,d+\left(\frac{1}{3}\frac{b^2\,y}{a^2}-\frac{b\,y^2}{a}+y^3-\frac{1}{27}\frac{b^3}{a^3}\right)+\frac{c\left(y-\frac{1}{3}\frac{b}{a}\right)}{a}+\frac{b\left(-\frac{2}{3}\frac{b\,y}{a}+y^2+\frac{1}{9}\frac{b^2}{a^2}\right)}{a}=0$

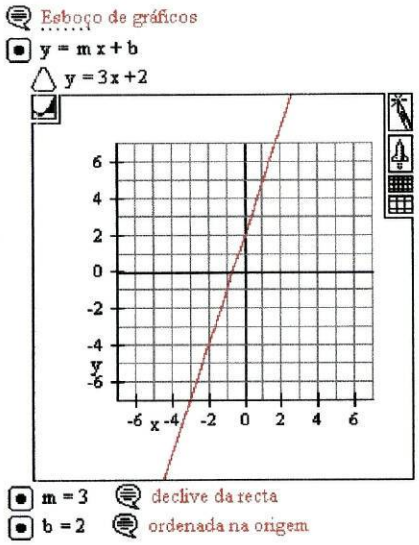
Estudo das funções

A função afim e as rectas não verticais

Para o estudo da função afim (função do tipo $y=m.x+b$), o uso de determinados Notebooks poderá ser uma boa solução. Os que de seguida se apresentam, são apenas algumas sugestões daquilo que se poderá fazer, de modo a ajudar o aluno, quer do 3º ciclo do EB quando se aborda pela primeira vez esta questão, quer mesmo os do ES onde a abordagem é feita de uma forma mais profunda, a compreender algumas das propriedades tanto do declive, como da ordenada na origem.

Esboço gráfico

Este primeiro exemplo explora a vertente gráfica do programa. Pretende-se com ele, que o aluno, ao mudar os valores dos parâmetros *m* e *b*, verifique o que acontece ao gráfico (recta) da função obtida com esses valores. Por este meio, o aluno poderá chegar à relação que existe entre *m* e a inclinação da recta, bem como a relação que existe entre o valor de *b* e o ponto de intersecção da recta com o eixo dos yy's.



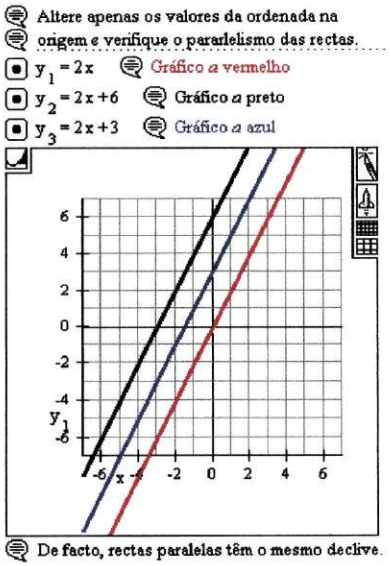
Altere os valores de m e de b que lhe são apresentados e tente estabelecer relações entre esses valores e a inclinação da recta e sua intersecção com o eixo dos yy's.

Rectas paralelas

No segundo exemplo, tenta-se levar o aluno a concluir que duas rectas paralelas têm o mesmo declive. De facto, ao mostrarmos no mesmo sistema de

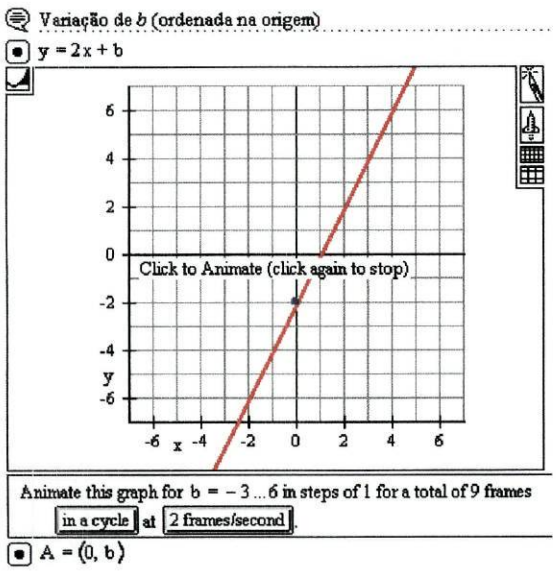
eixos o gráfico de, por exemplo, três (ou mais) funções afins todas com o mesmo declive, e pedir ao aluno que mude apenas a ordenada na origem, poderá ajudar o aluno a conjecturar sobre estas questões.

Alterando o valor de, por exemplo, um declive de uma das rectas em baixo, facilmente se visualiza que a recta respectiva deixa de ser paralela às outras duas.



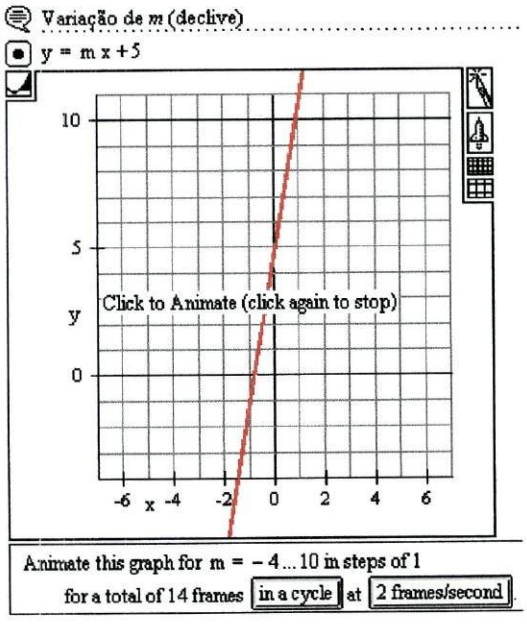
A ordenada na origem

No exemplo seguinte, conseguimos fazer variar a ordenada na origem *b* de uma forma automática, isto é, com um simples clique sobre a imagem, o valor de *b* vai crescendo da forma que nós pretendermos (no nosso exemplo, *b* varia entre -3 e 6). Os valores que vão aparecendo no canto superior direito da imagem mostram a evolução do valor de *b*. É possível escolher os valores inicial e final para *b*, bem como o valor de crescimento de *b*, ajustando-se automaticamente o número de frames necessárias à representação de todas as hipóteses para *b*. Pressionado o botão esquerdo



do rato em cima de 2 frames/second poderá escolher o número de frames por segundo.

O declive



Desta vez fazemos uma animação da variação de *m*, o declive da recta. Visualiza-se perfeitamente o que vai acontecendo à recta, nomeadamente, quando o declive muda de sinal. Visualiza-se também o facto da recta se tornar horizontal quando m = 0.

De notar que ainda pode clicar em in a cycle e escolher alterar essa opção para "one-way", podendo visualizar, continuamente ou apenas uma vez, a animação em curso.

Rectas perpendiculares

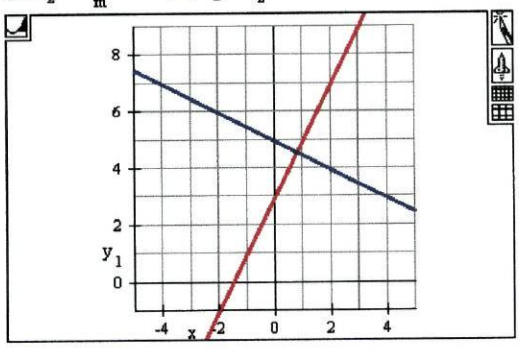
O Notebook seguinte poderá ser usado para que os alunos cheguem às relações existentes entre duas rectas perpendiculares. Deverá alterar o valor do declive da recta dada (valor de *m*) e verificar que as duas rectas se mantêm perpendiculares. Experimente alterar também os valor das ordenadas na origem, uma vez que obtém, como já foi visto anteriormente, rectas paralelas às rectas dadas.

Rectas perpendiculares

$m = 2$ Valor do declive

$y_1 = mx + 3$ $\triangle\ y_1 = 2x + 3$ Gráfico *a* azul

$y_2 = -\frac{1}{m}x + 5$ $\triangle\ y_2 = -\frac{1}{2}x + 5$ Gráfico *a* vermelho



Dois pontos definem uma recta

No exemplo que de seguida se mostra, o aluno visualiza como se chega à equação reduzida de uma recta, dados dois pontos. Quando o aluno modifica as coordenadas dos pontos que são dados no exemplo, o resultado, bem como todos os passos intermédios, adaptar-se-ão a esses novos valores. Por outro lado, pode já não ser necessário para o aluno, por já ter essas matérias compreendidas, a visualização de todos os passos. Pode haver simplesmente o interesse em utilizar um Notebook para facilitar os cálculos. De seguida apresentam-se as duas situações.

Equação de uma recta recta que passa por dois pontos

☐ $A(x_1, y_1)$

Altere as coordenadas do ponto *A*

☐ $x_1 = 2$ ☐ $y_1 = 3$

$\triangle\ A(x_1, y_1) = A(2,3)$

☐ $B(x_2, y_2)$

Altere as coordenadas do ponto *B*

☐ $x_2 = 5$ ☐ $y_2 = 7$

$\triangle\ B(x_2, y_2) = B(5,7)$

☐ $m = \frac{y_2 - y_1}{x_2 - x_1}$

$\triangle\ m = \frac{4}{3}$ Declive da recta

☐ $y = m(x - x_1) + y_1$

$\triangle\ y = \frac{4}{3}(x - 2) + 3$

$\triangle\ y = \frac{4}{3}x + \frac{1}{3}$ Equação reduzida da recta que passa por *A* e por *B*

Equação de uma recta recta que passa por dois pontos

☐ $A(x_1, y_1)$

Altere as coordenadas do ponto *A*

☐ $x_1 = 2$ ☐ $y_1 = 3$

$\triangle\ A(x_1, y_1) = A(2,3)$

☐ $B(x_2, y_2)$

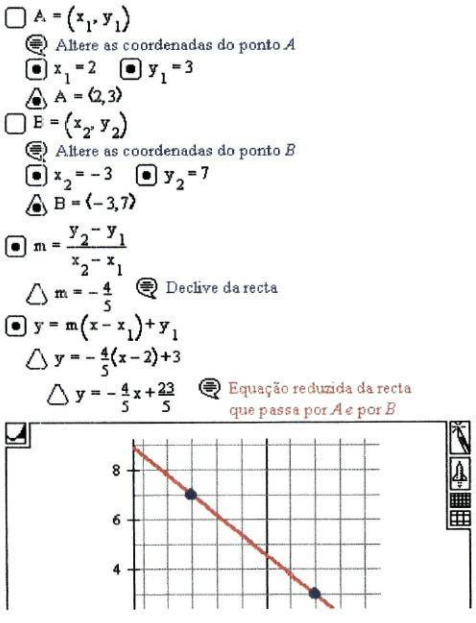
Altere as coordenadas do ponto *B*

☐ $x_2 = 4$ ☐ $y_2 = 7$

$\triangle\ B(x_2, y_2) = B(4,7)$

☐ $y = \frac{y_2 - y_1}{x_2 - x_1}x + \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$

$\triangle\ y = 2x - 1$ Equação reduzida da recta que passa pelos pontos *A* e *B*.



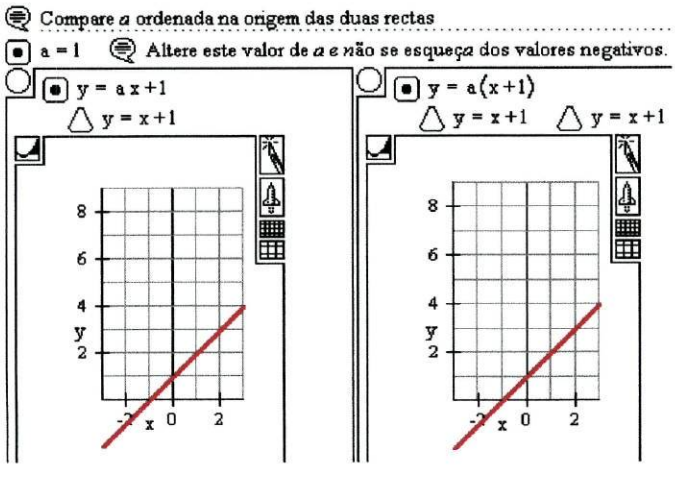
Poderá ainda visualizar um Notebook em que, para além da vertente algébrica da determinação da equação reduzida de uma recta que passa por dois pontos dados, poderá ainda acompanhar o raciocínio sob o ponto de vista gráfico.

No Notebook ao lado, o aluno poderá chegar a conclusões relativamente à determinação de uma recta (na equação reduzida) que passa por dois pontos dados. Ao mesmo tempo, o aluno vai acompanhando a representação gráfica dessa mesma recta, o que poderá contribuir para que conjecture sobre o declive e a ordenada na origem.

Experimente alterar os valores das coordenadas dos pontos A (x_1,y_1) e B(x_2,y_2) e visualize os resultados produzidos, quer ao nível algébrico, quer mesmo ao nível gráfico.

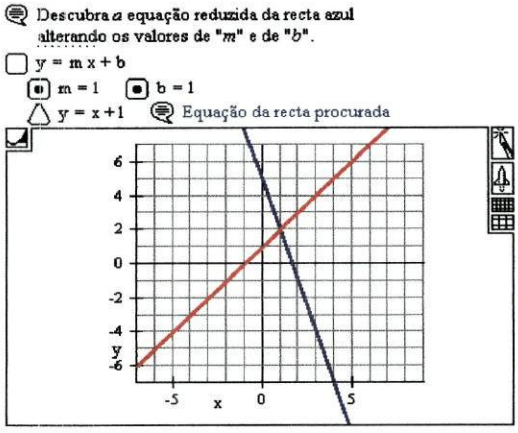
A recta *y = a.(x+1)*

Este Notebook, relaciona-se com o erro que, por vezes os alunos cometem, por não reconhecerem diferenças entre *y=a.x+1* e *y = a.(x+1)*. Ora, depois do estudo analítico das mesmas, a abordagem gráfica poderá ser também uma alternativa.

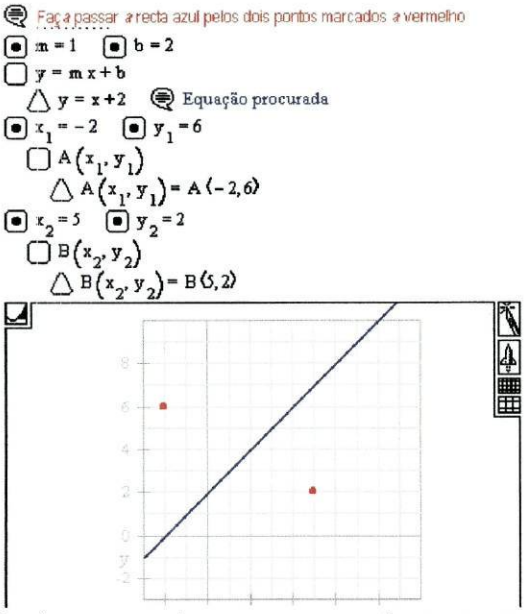


Descobrir a equação de uma recta

Nos Notebooks seguintes, o aluno tentará encontrar a equação reduzida (da forma $y = m.x + b$) de uma recta, devendo, para tal, atribuir valores a m e a b. Poderão constituir alguns exercícios práticos para a abordagem deste tema das funções afins.



No primeiro exemplo, deverão ser atribuídos valores a m e a b de modo a que a recta a vermelho se sobreponha à recta a azul. A solução é encontrada quando no ecrã for visível apenas a recta a vermelho. Caso seja visível apenas a recta a azul, é sinal que a recta a vermelho está numa parte do ecrã não visível devido à escala considerada (por exemplo para $m = 2$ e $b = 70$).

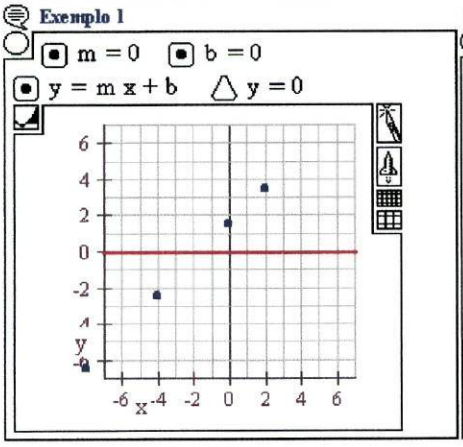


Com este segundo Notebook, pretende-se que o aluno descubra a equação da recta que passa por dois pontos dados. No exemplo inicial, os dois pontos são A(- 2 , 6) e B(5 , 2). O aluno deverá alterar os valores de m e b de modo a que a recta esboçada a azul passe por esses dois pontos. Depois de encontrada a recta, o aluno poderá ainda mudar os pontos A e B (alterando os valores de x_1, y_1, x_2 e y_2) e voltar a tentar encontrar a nova recta que passa por esses dois novos pontos.

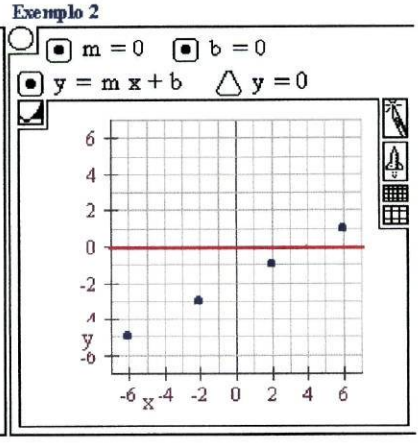
Finalmente, para cada um dos casos seguintes, tente fazer passar a recta pelos pontos que aparecem representados, alterando os valores de m e de b.

Tente encontrar os valores de "*m*" e o "*b*" capazes de fazer passar *a* recta *a* vermelho pelos pontos que aparecem marcados

Exemplo 1



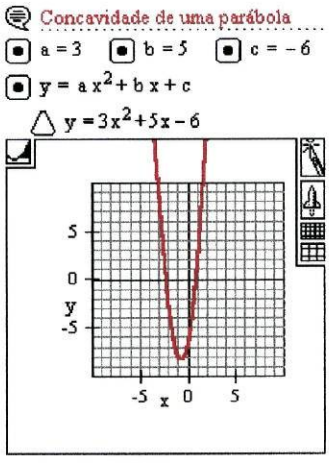
Exemplo 2



Função Quadrática

Um dos primeiros tópicos no estudo da função quadrática é a sua representação gráfica, nomeadamente a percepção do tipo de concavidade da parábola que lhe está associada.

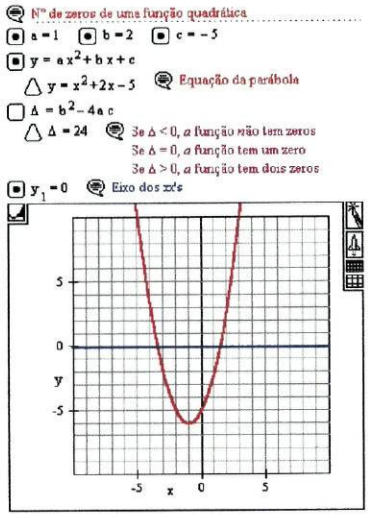
No primeiro Notebook que se apresenta, sugere-se que o aluno altere os valores de *a*, *b* e *c* e vá registando a concavidade (para baixo ou para cima) da função respectiva. Ao fim de uns tantos valores atribuídos, o aluno deverá ser capaz de concluir que a concavidade de uma parábola depende do valor de *a*: Se $a > 0$, a concavidade é voltada para cima e, por outro lado, se $a < 0$, a concavidade é voltada para baixo. De notar que o caso de $a = 0$ não se coloca uma vez que a função obtida ($y = b \cdot x + c$) não se trata de uma função polinomial do 2º grau.



Determinação do número de raízes

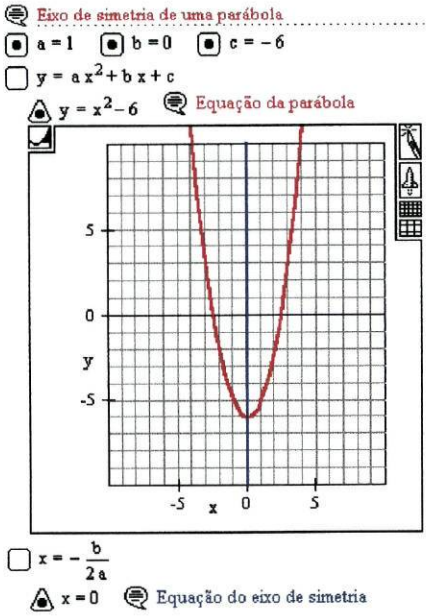
A determinação do número de zeros de uma parábola, antes de se avançar com o seu estudo propriamente dito, torna-se importante para que os alunos conjecturem sobre o comportamento do gráfico. Com este intuito, criou-se o Notebook seguinte no qual é apresentada a determinação do binómio discriminante, ao mesmo tempo que se visualiza o gráfico da respectiva função, onde são visíveis as raízes da função.

Por outro lado, a determinação de tais "zeros" de uma parábola, está relacionada com a resolução de uma equação do 2º grau, conteúdo que está retratado na parte de resolução de equações.



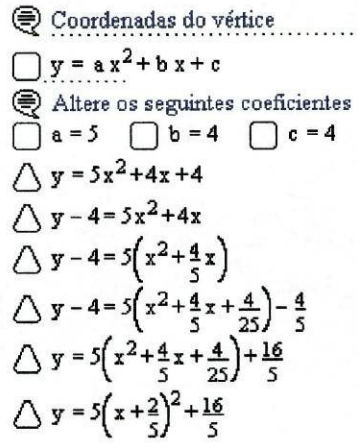
Eixo de simetria

O Notebook seguinte, pretende mostrar ao aluno a determinação do eixo de simetria de uma parábola, quer sob o ponto de vista geométrico, quer sob o ponto de vista analítico. Altere os valores dos coeficientes da função quadrática e visualize o respectivo eixo de simetria obtido, quer ao nível da sua equação, quer ao nível da sua representação gráfica.



Vértice da parábola

Para o estudo da localização do vértice, será importante o aluno concluir que qualquer expressão do tipo $y=ax^2+bx+c$ pode ser escrita da forma $\mathbf{y = a\,(x - k)^2 + h}$. O Notebook seguinte conduz o aluno a esta transformação algébrica.

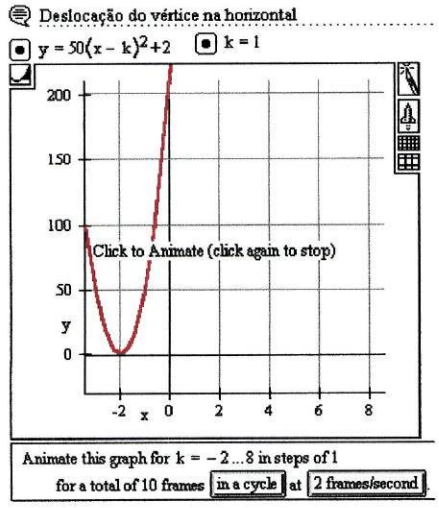


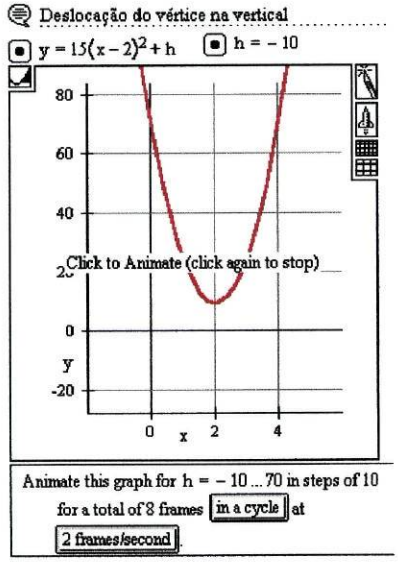
A representação gráfica de funções quadráticas é de extrema importância no Ensino Secundário, pois podem interiorizar-se determinadas noções que se mostrarão decisivas na representação gráfica de outro tipo de funções. A localização do vértice, por exemplo, torna-se bastante simples de compreender com Notebooks animados como os que de seguida se apresentam.

A influência dos parâmetros no esboço gráfico

De facto, depois de concluir que qualquer expressão do tipo $y=ax^2+bx+c$ pode ser escrita da forma $\mathbf{y = a\,(x-k)^2 + h}$, para levar o aluno a concluir que as coordenadas do vértice da parábola são precisamente (k,h), poderemos fazer variar automaticamente o valores de k e h e ver o gráfico da função deslocar-se, respectivamente, na horizontal e na vertical.

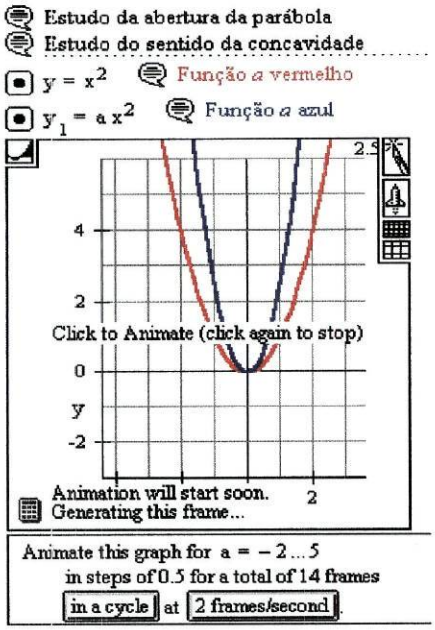
Em ambos os casos, verifique no canto superior direito os valores de k e h a aumentarem, à medida que o gráfico se desloca, respectivamente, na horizontal e na vertical. Conforme referido anteriormente, pode também aumentar e diminuir a velocidade de processamento da nova posição do gráfico. As imagens seguintes dizem respeito a essas duas situações.





Na figura seguinte, tenta mostrar-se duas coisas. Em primeiro lugar, mantendo fixa a função a vermelho e animando a função a azul fazendo crescer o valor de *a*, mostra-se que à medida que *a* aumenta, a parábola vai ficando cada vez mais estreita.

Em segundo lugar, mostra-se também que, quando *a* é negativo a parábola tem a concavidade virada para baixo, enquanto que quando *a* é positivo, a parábola tem concavidade voltada para cima.

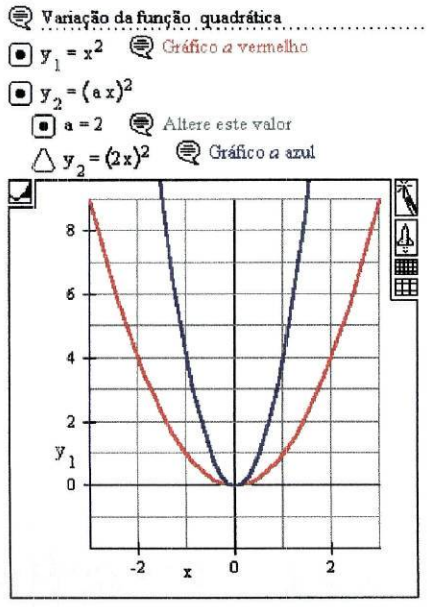


Há ainda a destacar os casos de $a = 0$ e $a = 1$. No primeiro caso, é fácil ver que estamos perante uma função ($y = 0$) cujo gráfico é uma recta horizontal e, no segundo caso, facilmente se conclui que as duas funções são coincidentes.

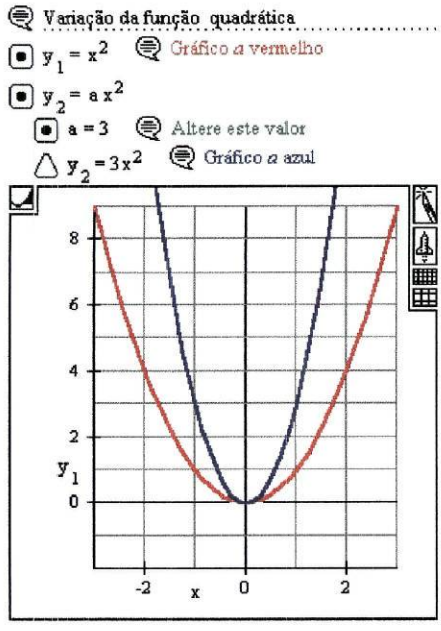
Podemos acompanhar a evolução do valor de *a*, depois de se iniciar a animação do gráfico, no canto superior direito do mesmo. Neste caso, o valor de *a* varia entre - 2 e 5, e cresce na razão de 0.5.

Variação de $y = x^2$

Relativamente à função quadrática, vejamos agora algumas variações da função $y = x^2$. Cada um dos Notebooks seguintes, tentará mostrar ao aluno as diferenças existentes, sob o ponto de vista analítico e geométrico, entre as funções referenciadas.



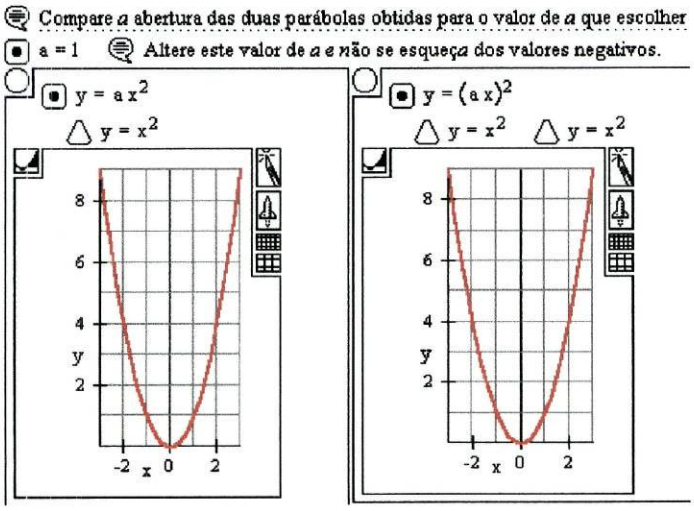
Neste primeiro Notebook, pode verificar-se que ao alterar o valor de *a*, a concavidade da parábola também sofre alterações. De facto, a concavidade da parábola (gráfico da função $y = (ax)^2$) fica mais estreita ou mais larga, conforme $|a| > 1$ ou $|a| < 1$. Experimente modificar o valor de *a*, atribuindo-lhe valores quer positivos quer negativos. De referir que a concavidade da parábola obtida será sempre voltada para cima, uma vez que o valor de *a* aparece também levantado ao quadrado.

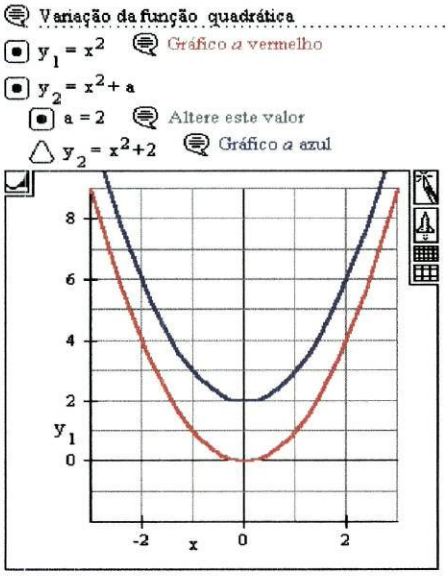


Enquanto que no Notebook anterior o valor de *a* aparecia também elevado ao quadrado, neste exemplo isso não acontece, o que provoca, por exemplo, que para valores de *a* negativos, a concavidade da parábola obtida seja voltada para baixo.

Para uma melhor visualização da diferença entre as funções $y = ax^2$ e $y = (ax)^2$, foi criado o Notebook que se segue.

Com este Notebook, poderá mostrar-se ao aluno a diferença entre as duas funções referenciadas, uma vez que são frequentes as confusões entre estas duas funções, por haver "esquecimento" dos parêntesis. Altere o valor que está atribuído a *a* e, de imediato, as expressões analíticas das duas funções adaptar-se-ão ao novo valor, assim como o respectivo gráfico. Obviamente que não devem ser esquecidos os valores negativos de *a*, altura em que a primeira parábola mudará de concavidade.

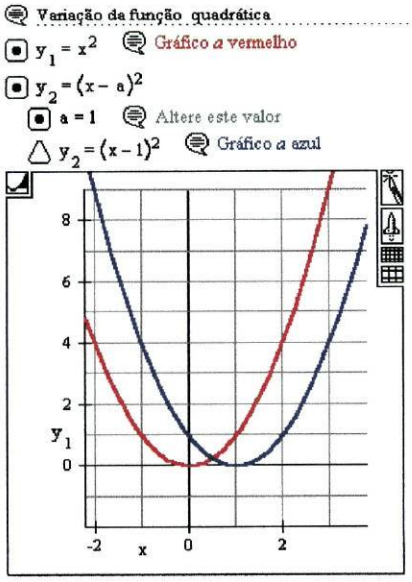




Neste Notebook, visualiza-se a deslocação, na vertical, do vértice da parábola. Quando, na função $y = x^2 + a$, o valor de *a* é positivo a deslocação é "para cima" (ou seja, no sentido da semi-recta [OA], em que A=(0,1)) e, no caso de *a* ser negativo a deslocação é "para baixo" (ou seja, no sentido da semi-recta [OB], em que B=(0,-1)).

Neste último caso, visualiza-se a deslocação, na horizontal, do vértice da parábola. Quando, na função $y = (x - a)^2$, o valor de *a* é positivo a deslocação é "para a direita" (ou seja, no sentido da semi-recta [OA], em que A=(1,0) e, no

caso de *a* ser negativo a deslocação é "para a esquerda" (ou seja, no sentido da semi-recta [OB], em que B=(-1,0)).

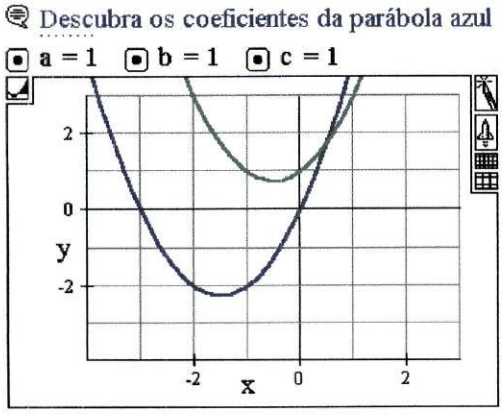


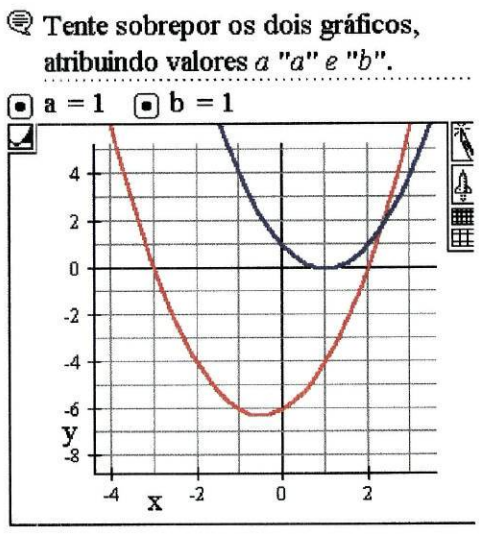
Alguns exercícios práticos

De seguida apresentam-se alguns exercícios práticos de aplicação de funções quadráticas. Para os resolver, o aluno, através da alteração dos valores de *a*, *b* e *c*, tentará chegar à expressão analítica de uma determinada função quadrática.

No primeiro exemplo, o aluno deverá mudar os coeficientes *a, b* e *c*, de modo a descobrir a parábola (escrita na forma $y = a.x^2+b.x+c$) cujo gráfico aparece esboçado a azul.

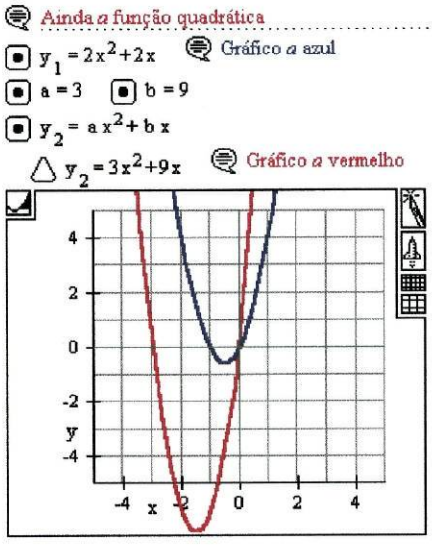
De salientar que quando são escolhidos novos coeficientes, a parábola respectiva aparece esboçada a verde e, como tal, o aluno descobrirá a solução quando no ecrã apenas é visível esse gráfico a verde (se for visível apenas o gráfico a azul, é sinal que o gráfico a verde está numa zona do ecrã não visível, não estando, portanto, neste caso, o problema resolvido).





Neste Notebook, a função cujo gráfico é a parábola esboçada a vermelho, da qual se pretende a respectiva equação, está escrita na forma $y = (x - a).(x - b)$. O aluno deverá alterar os valores de *a* e de *b* de modo a que o gráfico a azul fique sobre o gráfico a vermelho (isso acontece apenas quando é visível no ecrã o gráfico a azul; se for visível apenas o gráfico a vermelho, é sinal que o gráfico a azul está numa zona do ecrã não visível).

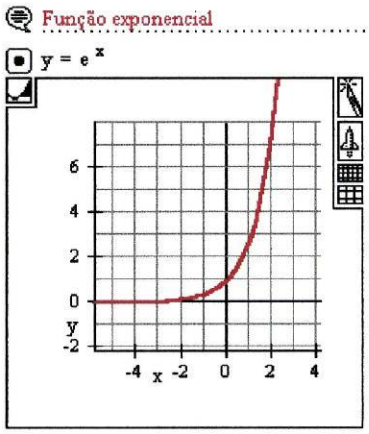
Neste último exemplo, sugere-se que o aluno se debruce sobre a família de funções reais de variável real do tipo $y = a.x^2+b.x$, no que diz respeito a um dos zeros. Para tal, deverá alterar os valores de *a* e de *b* que são apresentados, e verificar o que acontece sempre a uma das raízes da função cujo gráfico aparece esboçado a vermelho.



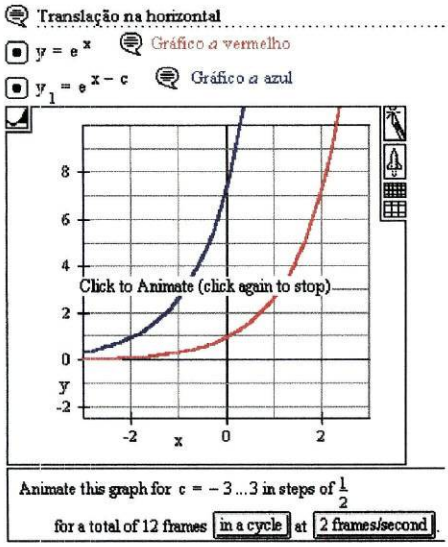
Função Exponencial

No programa do 12º ano de escolaridade, prevê-se o estudo das propriedades analíticas e gráficas da família de funções definidas por $f(x)=a^x$, com $a > 1$.

No Notebook que se segue, faz-se a apresentação da função $y=e^x$, em que a base é o número de Neper (aproximadamente 2.718). Experimente alterar o valor da base, seleccionando-o e introduzindo um outro valor, e veja as alterações produzidas.



Nos exemplos que se seguirão, tenta-se ajudar o aluno a tirar algumas conclusões acerca da função exponencial, comparando o gráfico da função $y=e^x$ (gráfico anterior), com os gráficos de funções que contemplam determinados parâmetros.

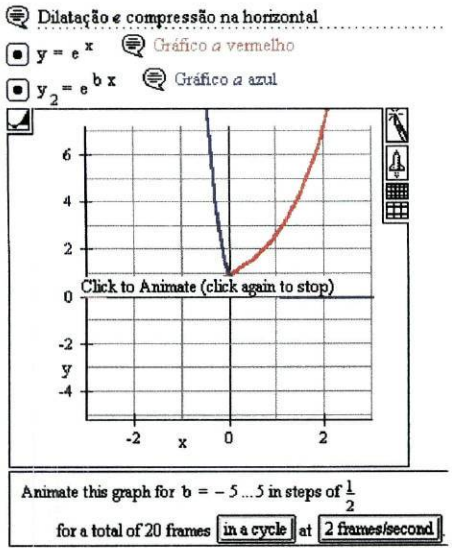


Este primeiro caso, que compara as duas funções que aparecem definidas na imagem, provoca as alterações que o parâmetro *c* numa função **f(x-c)** provoca numa função qualquer **f(x)**.

De facto, quando subtraímos x por um número positivo *c*, a função sofre uma translação horizontal, para a direita, *c* unidades. Quando *c* é negativo, a translação, desta vez, é para a

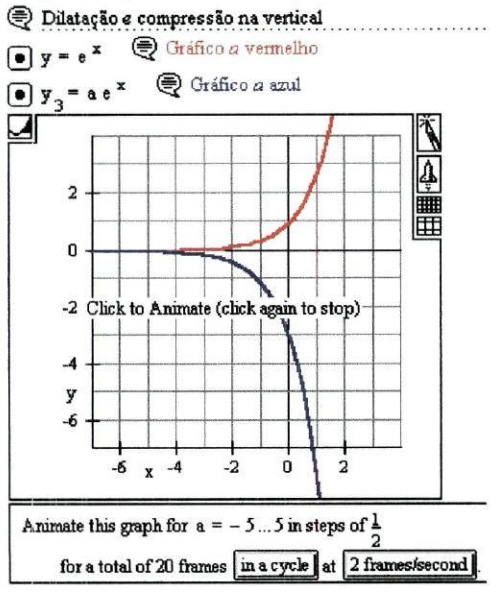
esquerda, |c| unidades.

Neste segundo exemplo, apresenta-se um Notebook para que o aluno deduza as transformações geométricas que o gráfico sofre quando se multiplica a função exponencial por um número real não nulo, transformações essas que ocorrem com qualquer outra função.



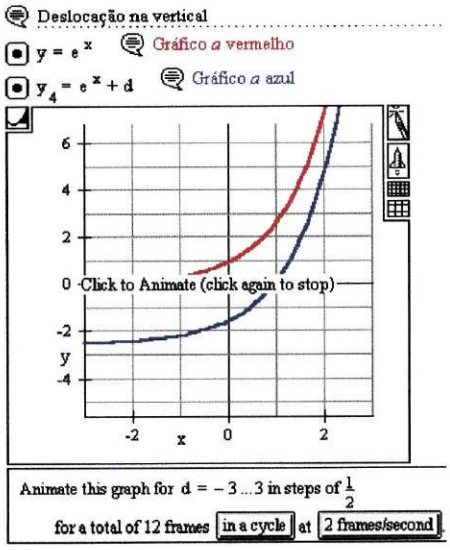
Conforme se pode verificar, quando multiplicamos x na função $y=e^x$ por um número positivo b, teremos duas situações: se $0 < b < 1$, teremos uma contracção na horizontal da função e, se $b > 1$, teremos uma dilatação na horizontal da função inicial. Por sua vez, quando $b < 0$, a contracção ou dilatação na horizontal, serão seguidas de uma reflexão sobre o eixo dos yy's.

Neste terceiro caso, à semelhança do que aconteceria com qualquer outra função, quando multiplicamos a função $y=e^x$ por um número positivo a, teremos duas situações: se $0 < a < 1$, teremos uma contracção na vertical da função e se $a > 1$ teremos uma dilação na vertical da função inicial.



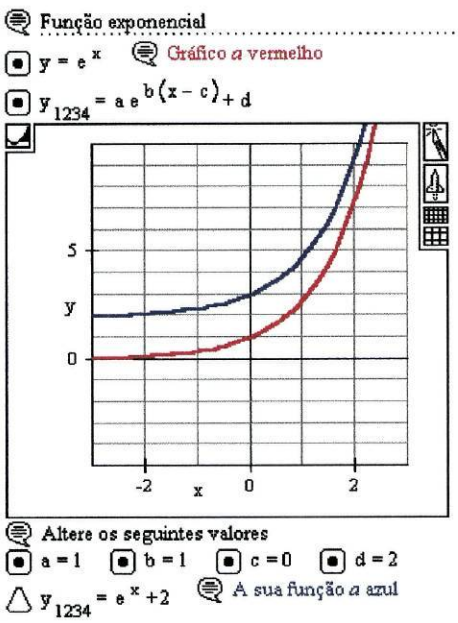
Por sua vez, quando $a < 0$, a contracção ou dilatação na vertical, serão seguidas de uma reflexão sobre o eixo dos xx's. De salientar que isto acontece para qualquer função e não apenas para a função exponencial.

Finalmente, um último caso que seria comum a qualquer outra função. Quando somamos à função inicial um número positivo *d*, a função sofre uma translação vertical, "para cima", *d* unidades. Quando *d* é negativo, a translação, desta vez, é "para baixo", |*d*| unidades. O Notebook seguinte caracteriza essa situação.



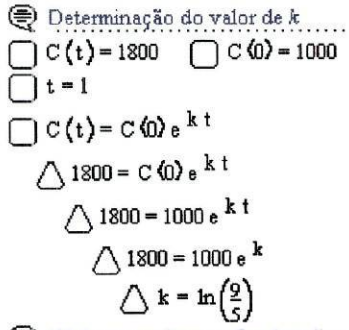
Noutras secções deste trabalho, na versão on-line, foram criados outros Notebooks que retratam as mesmas transformações que se acabou de fazer referência, para outras funções, designadamente função logarítmica e funções trigonométricas. Neste versão em papel, não serão mostradas as imagens referentes a cada uma das transformações pois estaríamos, perante a mesma situação. Será mostrada apenas um "Notebook resumo", ao mesmo tempo que se fará referência ao que se acabou de dizer para a função exponencial.

O Notebook que de seguida se apresenta, contempla num só caso todas as situações que foram apresentadas anteriormente em Notebooks diferentes. O aluno poderá manipular qualquer dos parâmetros à sua escolha e ver, imediatamente, a alteração produzida na função inicial.



Para finalizar, de seguida apresenta-se um Notebook que permite a resolução de um problema, em que está envolvida a resolução de uma equação em que aparece uma função exponencial.

Suponha que o crescimento da população de uma colónia de mosquitos obedece à seguinte lei: $C(t)=C(0).e^{k.t}$ (k constante), em que t representa o número de mosquitos ao fim de t dias e N(0) a população inicial da colónia. Determine o número de mosquitos da colónia ao fim de três anos, sabendo que inicialmente a colónia tem 1000 mosquitos e que há 1800 mosquitos ao fim de um dia.



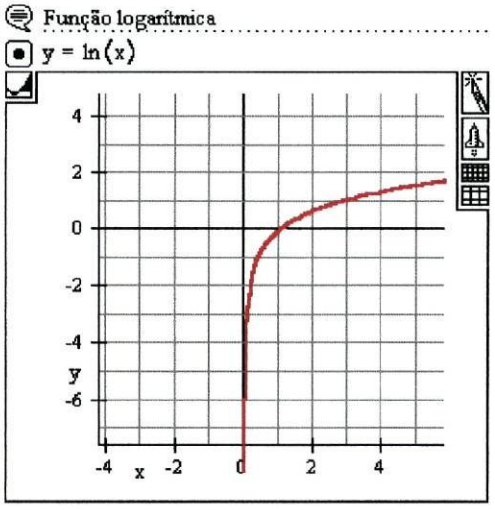


Experimente mudar as hipóteses iniciais, ou seja, o valor de C(0) ou o valor de C(t) ao fim de um dia. Verifique a actualização imediata quer do valor de k (calculado na primeira parte do Notebook) quer do valor final de mosquitos ao fim de t dias, que também poderá ser alterado.

Função Logarítmica

No programa do 12º ano de escolaridade, está também previsto o estudo das propriedades analíticas e gráficas da família de funções definidas por $f(x)=\log_a x$, com $a > 1$.

No primeiro Notebook que se apresenta, mostra-se a função $y=ln(x)$ (logaritmo de base e - Número de Neper).



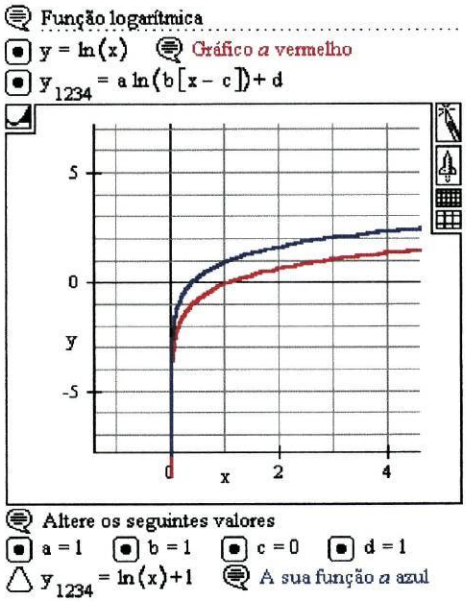
De seguida, tenta-se ajudar o aluno a tirar algumas conclusões acerca da função logaritmo. A ideia será comparar o gráfico da função $y=ln(x)$ (gráfico anterior), com os gráficos das seguintes funções:

- $y_1=ln(x-c)$
- $y_2=ln(b.x)$
- $y_3=a.ln(x)$
- $y_4=ln(x)+d$

Não se apresentarão aqui os Notebooks que constam da versão on-line deste trabalho, uma vez que bastaria utilizar aqueles que foram criados para a função exponencial, alterando a respectiva função. As propriedades mantinham-se exactamente as mesmas.

Contudo, apresenta-se um único Notebook que resume as quatro situações das transformações ocorridas na função inicial, através da intervenção de um

parâmetro. O aluno poderá manipular qualquer dos parâmetros à sua escolha e ver, imediatamente, a alteração produzida na função inicial.



Funções trigonométricas

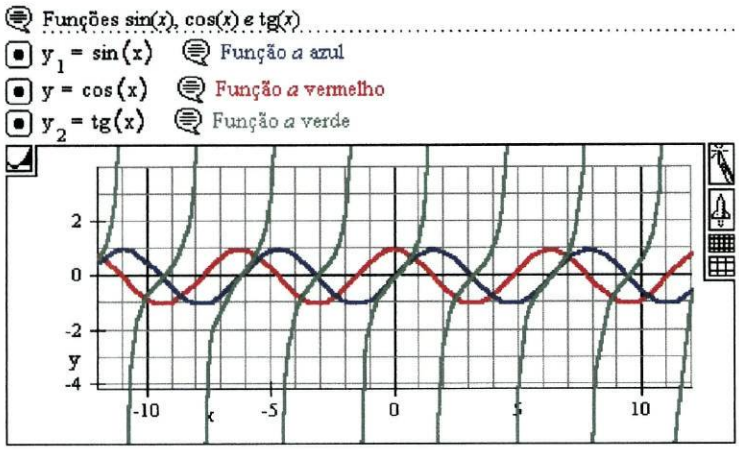
Este espaço destina-se a exemplificar o uso do MathView no estudo das funções trigonométricas *sin(x)*, *cos(x)* e *tg(x)*. Nos Notebooks construídos, faz-se também uma comparação entre as funções *sin(x)* e *cos(x)*, por se tratarem de funções cujos gráficos se comportam de forma semelhante, fazendo com que, frequentemente, os alunos apresentem dificuldades na sua identificação. Com os Notebooks que de seguida se apresentam, os alunos poderão retirar as suas conjecturas, através da manipulação dos valores que são apresentados inicialmente.

Na figura seguinte faz-se uma primeira apresentação dessas três funções, cujo gráfico é esboçado no mesmo sistema de eixos. Sugere-se neste primeiro Notebook que, por exemplo, o aluno posicione o ponteiro do rato (sob a forma de uma "mão") sobre a figura, pressione o botão esquerdo do rato e, sem largar, arraste os gráficos para a esquerda e para a direita, para cima e para baixo. Daqui poderá levar-se o aluno a chegar a conclusões referentes às principais características destas três funções.

Outra actividade que se poderia propor aos alunos (por exemplo, apenas a um deles se a aula estiver a decorrer apenas com uma imagem projectada através de um *data show*) seria o professor esconder os gráficos das funções *sin(x)* e *cos(x)* (arrastando-as pelo mesmo processo atrás descrito) numa parte do sistema de eixos bem longe da origem do referencial cartesiano. O aluno teria, portanto, que encontrar as funções, levando-o a ter a noção de domínio e contradomínio.

Apenas relembrar mais um pormenor que se prende com as funções *sin(x)* e *tg(x)*. No programa MathView a notação utilizada para a primeira função é *sin(x)* e para a segunda *tan(x)*. Como os programas portugueses adoptam as notações *sen(x)* e *tg(x)* haveria a possibilidade de se modificar no próprio programa essas notações. Aliás, foi o que se fez com a função *tg(x)*. Já no caso da função *sen(x)*, optou-se por não se fazer qualquer alteração por ser pouco susceptível a confusões por parte dos alunos.

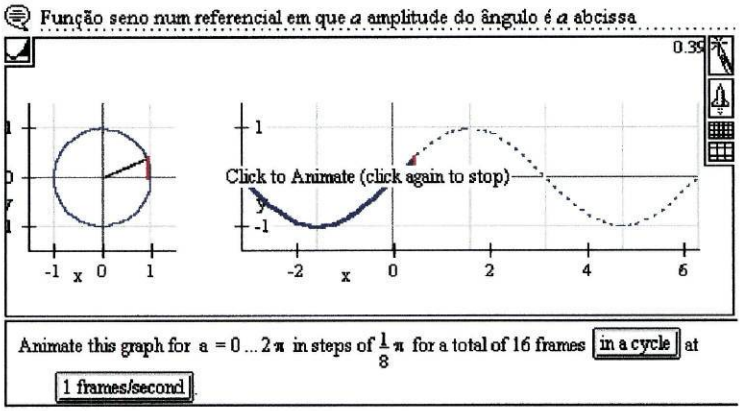
Vejamos então as três funções no mesmo sistema de eixos.



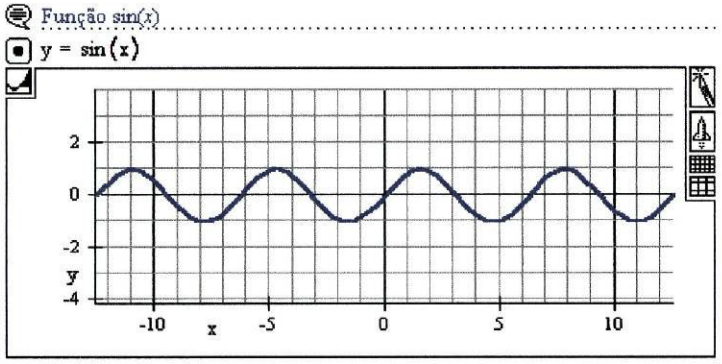
Na caixa que se segue, poderão levantar-se algumas questões que ajudem os alunos a distinguir as funções *sin(x)* e *cos(x)*. Para que isso seja possível, sugere-se que se leve o aluno a visualizar os dois gráficos esboçados em sistemas de eixos distintos de modo a que possa tirar as suas conclusões e, por fim, que se mostre ao aluno as duas funções esboçadas no mesmo sistema de eixos. Visualize a função *sin(x)* e, de seguida, visualize a função *cos(x)*. Depois de retirar as suas conclusões, visualize as duas funções representadas no mesmo sistema de eixos. Nesta versão em papel, é mostrado apenas o Notebook que contém as duas funções representadas simultaneamente, não sendo, portanto, perceptível a interactividade desta situação.

A função *sin(x)*

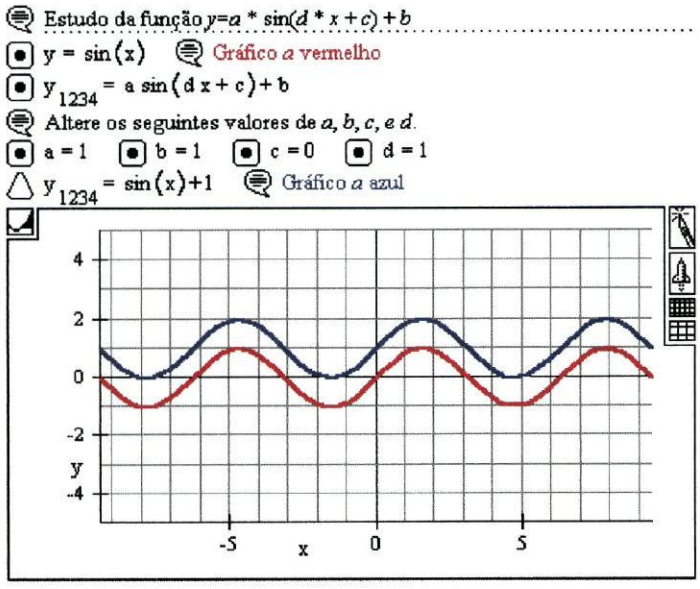
Será de toda a conveniência mostrar aos alunos, através de um mecanismo dinâmico, a proveniência desta função. O exemplo que de seguida é apresentado, pretende ilustrar essa situação. Na parte esquerda da figura é visível um ponto a rodar sobre o círculo trigonométrico e na parte direita da mesma figura, ao mesmo tempo, verifica-se que o ponto vai percorrendo o gráfico da função *y=sin(x)*. Para animar o Notebook seguinte, bastaria, na versão on-line, clicar sobre ele.



No Notebook seguinte mostra-se o gráfico da função *y=sin(x)*, esboçado no intervalo [-4.π , 4.π]. Poderá ser utilizado, por exemplo, para se chegar a conclusões relativamente ao contradomínio, à paridade, zeros, entre outros aspectos.



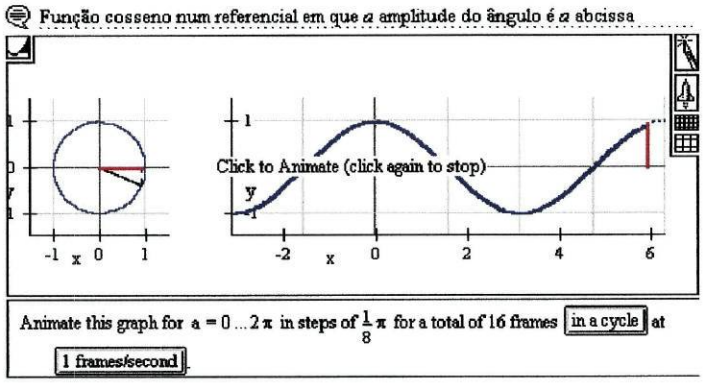
Nos Notebooks que de seguida se apresentam, tentar-se-à ajudar o aluno a constatar as alterações à função *y=sin(x)*, através da sua comparação com as funções $y_1=a.sin(x)$, $y_2=b+sin(x)$, $y_3=sin(x+c)$ e $y_4=sin(d.x)$. Estes quatro casos que provocam "alterações" distintas à função inicial, alterações essas que ocorrem independentemente da função que se está a considerar. Aliás, o mesmo já tinha sido feito para a função exponencial e, como tal, não vamos apresentar os Notebooks que retratam essa situação, para esta função. Simplesmente se apresentará o Notebook em que o aluno poderá proceder às alterações que falámos anteriormente, todas ao mesmo tempo. Desse modo, poderá visualizar de imediato, a alteração produzida à função inicial.



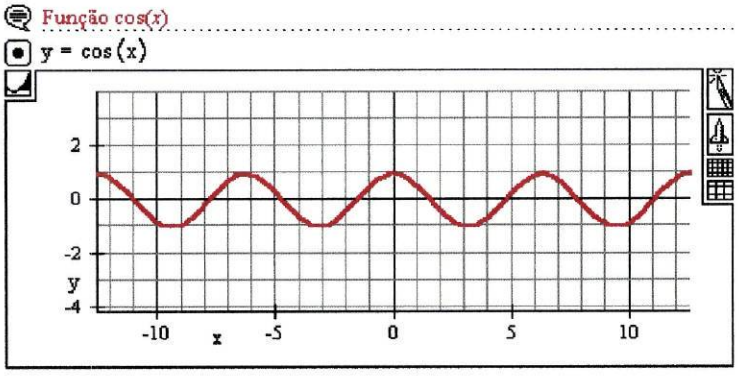
A função *cos(x)*

Neste espaço, as actividades propostas são exactamente as mesmas que as que foram propostas para a função *sin(x)*. Aliás, em certos casos, remete-se mesmo o aluno para os mesmos Notebooks que foram apresentados aquando da abordagem da função *sin(x)*, sugerindo-se que apenas se substitua a expressão "sin" pela expressão "cos".

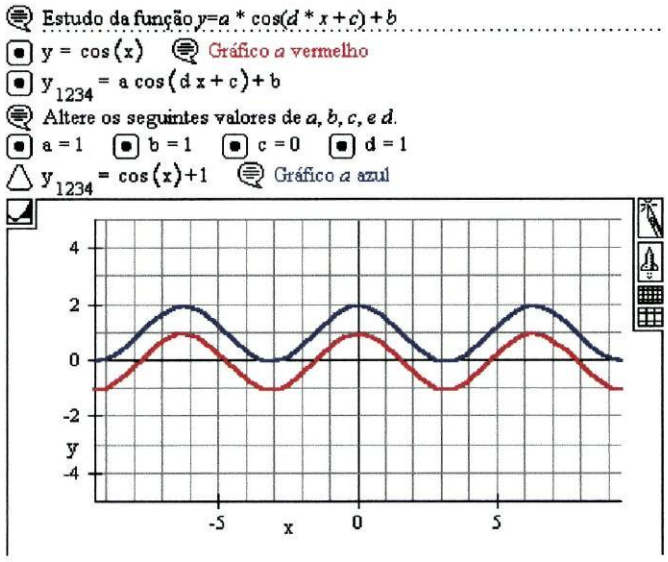
À semelhança do que se fez com a função *sin(x)*, também seria conveniente mostrar aos alunos, através de um mecanismo dinâmico, a proveniência desta função. O exemplo que de seguida é apresentado, pretende ilustrar essa situação. Na parte esquerda da figura é visível um ponto a rodar sobre o círculo trigonométrico e na parte direita da figura, ao mesmo tempo, verifica-se que o ponto vai percorrendo o gráfico da função *y=cos(x)*.



No Notebook seguinte mostra-se o gráfico da função *y=cos(x)*, esboçado no intervalo [-4.π , 4.π]. Poderá ser utilizado, por exemplo, para se chegar a conclusões relativamente ao contradomínio, à paridade, zeros, entre outros aspectos.

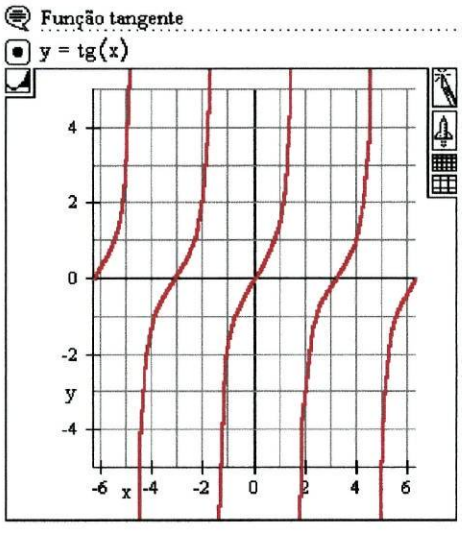


De seguida, tentar-se-à ajudar o aluno, a constatar as alterações à função *y=cos(x)*, através da sua comparação com as funções $y_1=a.cos(x)$, $y_2=b+cos(x)$, $y_3=cos(x+c)$ e $y_4=cos(d.x)$. Estes quatro casos poderiam ter sido criados através do aproveitamento dos que foram realizados para o mesmo estudo que foi feito para a função *y=sin(x)*. Também não serão apresentados esses Notebooks, mas antes um resumo dessas quatro situações.

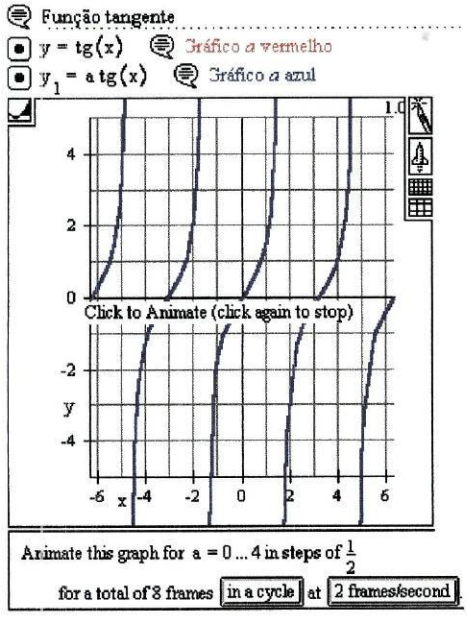


A função *y=tg(x)*

Neste primeiro Notebook, apresenta-se a função *tg(x)*, de onde já se poderão tirar algumas conclusões, nomeadamente, no que diz respeito ao domínio, contradomínio, zeros, período, entre outras.

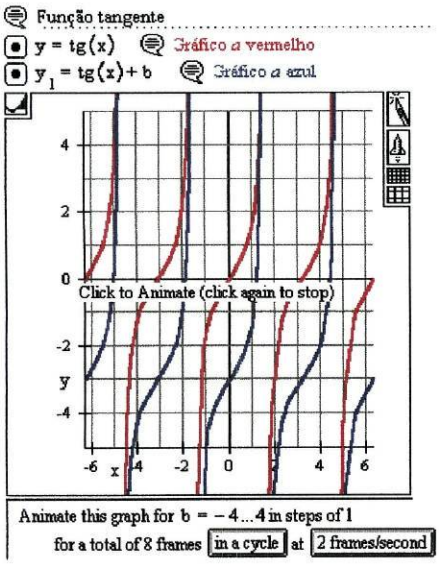


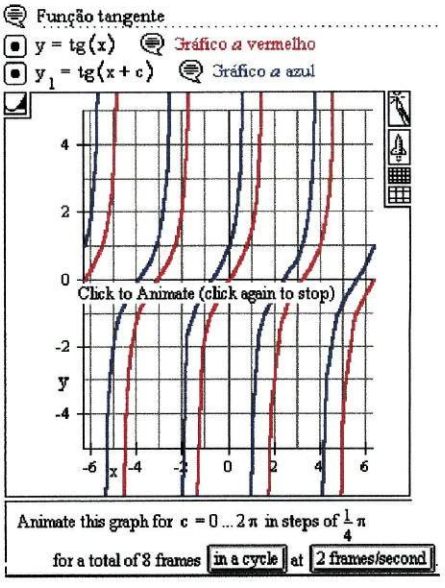
De seguida, tentar-se-à ajudar o aluno a constatar algumas das propriedades da função *y=tg(x)*, através da comparação desta função com as funções y_1=*a.tg(x)*, y_2=*b+tg(x)*, y_3=*tg(x+c)* e y_4=*tg(d.x)*. Optou-se desta vez pela apresentação de Notebooks animados, ao contrário do que se tinha feito para as função *sin(x)* e *cos(x)* e, como tal, são apresentadas as imagens referentes a essas situações.



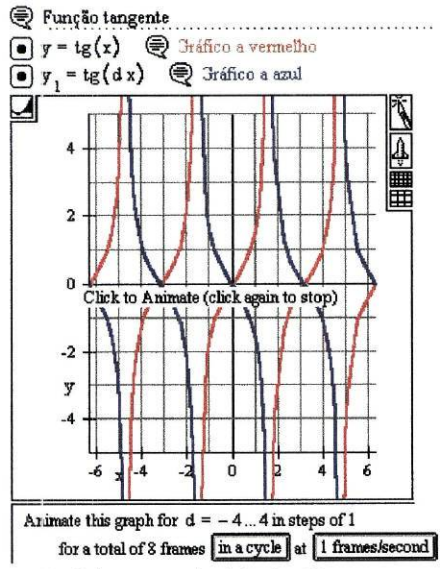
Neste primeiro Notebook, compara-se o gráfico da função $y=tg(x)$ com o gráfico da função $y_1=a.tg(x)$. À medida que o valor de a vai aumentando, o gráfico da função inicial vai esticando, como se estivesse a aproximar-se de uma recta vertical. É óbvio que quando $a = 0$, estamos perante a função $y = 0$ (cujo gráfico é uma recta horizontal). De referir ainda que é possível modificar os valores a atribuir a a, bastando para isso seleccionar na parte ***"Animate this graph..."*** os valores de a (0...4) e substituí-los pelos valores pretendidos. Se pretendermos também que a vá aumentando de uma outra forma que não ½, basta seleccionar esse valor e introduzir o pretendido.

No segundo Notebook (imagem em baixo, à esquerda), comparam-se as funções $y=tg(x)$ e $y_1=tg(x)+b$. Constata-se que a função inicial, à semelhança com o que acontece com a generalidade das funções, sofre uma translação na vertical | b | unidades. Essa translação é para cima ou para baixo, conforme se $b > 0$ ou $b < 0$, respectivamente.

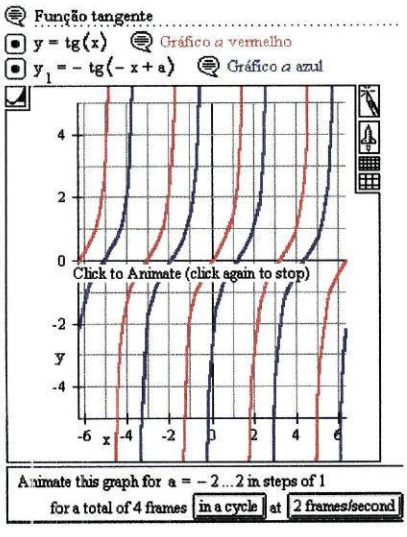




No terceiro exemplo (imagem em cima, à direita) leva-se o aluno a concluir sobre o período da função *y=tg(x)*. Através deste Notebook animado, constata-se que os dois gráficos se sobrepõem apenas em dois momentos: quando *c = 0* e *c = 3,142* (valor aproximado de π). Isto quer dizer que quando *a* toma esses dois valores, os gráficos das funções *y=tg(x)* e y_1*=tg(x+c)* são iguais. Para *c=0* é óbvio que assim seria uma vez que *f(x)=f(x+0)*. Já para o segundo valor de *c*, fica claro tratar-se do período da função *tg(x)*. Deverá ainda mostrar-se ao aluno que o mesmo acontece noutras partes do gráfico que não as visíveis no ecrã. Para tal, deverá clicar-se sobre a figura e, sem largar o rato, deverá arrastar-se o gráfico quer para a esquerda/direita quer para cima/baixo. De notar que o valor de *c* pode ser acompanhado no canto superior direito da imagem.

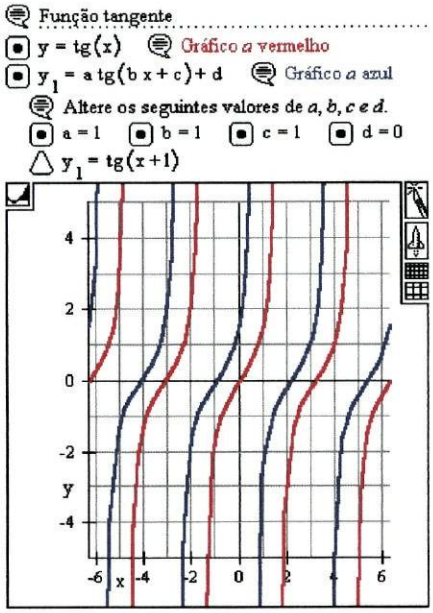


Finalmente, o quarto exemplo (ao lado) constitui uma comparação entre as funções *y=tg(x)* e y_1*=tg(d.x)*. Como se pode verificar, para o caso de *d=0*, teremos que y_1*=tg(0.x)*, ou seja, y_1*=0* a que corresponde uma recta horizontal. Para os outros casos, verifica-se que à medida que *d* vai crescendo, a curvatura do gráfico da função correspondente é menos acentuada.



Este Notebook destina-se a verificar a paridade da função *y=tg(x)*, através da comparação do seu gráfico com o gráfico da função y_1*= - tg(-x+a)*. Constata-se que apenas quando *a = 0* é que os dois gráficos se sobrepõem. Isso quer dizer que quando *a = 0* as duas funções são iguais, ou seja, *tg(x)=-tg(-x)*, o que é o mesmo que dizer que *tg(-x) = - tg(x)*, isto é, *y=tg(x)* é ímpar.

No Notebook que de seguida se apresenta, o aluno poderá proceder às alterações que falámos anteriormente, todas ao mesmo tempo. Desse modo, poderá visualizar de imediato, a alteração produzida à função inicial.



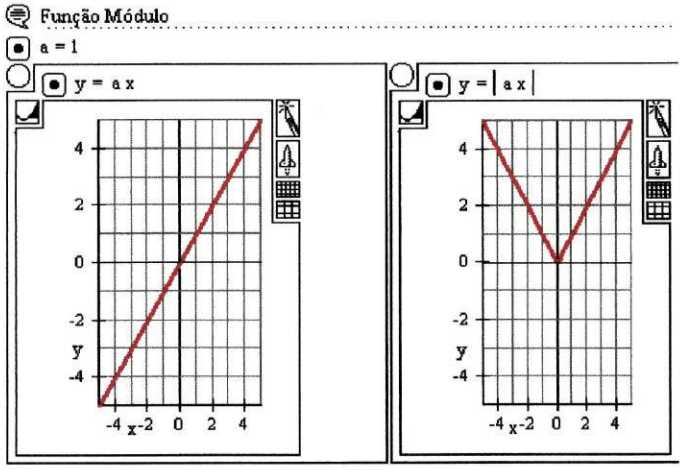
Função Módulo

$$|x| = \begin{cases} x & (x \geq 0) \\ -x & (x < 0) \end{cases}$$

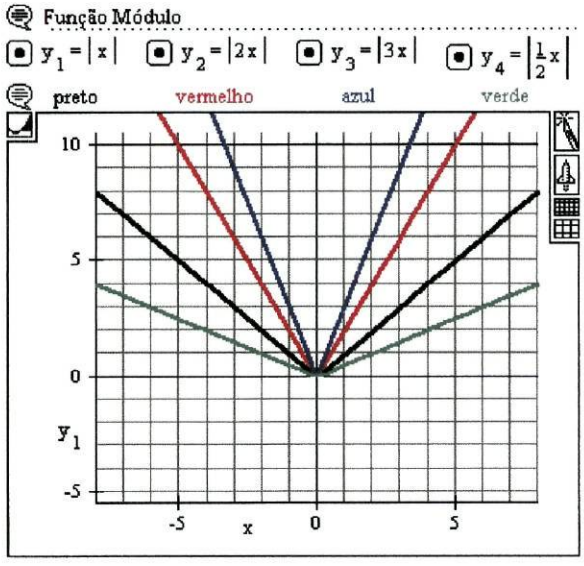
Antes de mais nada, relembremos a noção de *módulo* ou *valor absoluto* de um número. Sugerem-se dois métodos: um deles consiste numa tabela que, através de um *script* criado em Javascript, nos dá o resultado e, o outro, consiste na apresentação de um Notebook, criado com o MathView, onde o aluno apenas tem que substituir o valor que é apresentado por defeito pelo valor do qual pretende calcular o seu módulo. Vejamos então tais métodos:

x: |x| calcular apagar	Função Módulo $f(x) = \|x\|$ Altere o valor seguinte f(2) $f(2) = 2$
Introduza o número do qual pretende saber o valor absoluto na caixa respectiva e pressione o botão "calcular". Ao premir o botão "apagar", os dois campos voltam a ficar vazios.	No Notebook em cima, o aluno deverá alterar o valor do objecto do qual se pretende calcular o valor absoluto. Para tal, deverá seleccionar o "2" (com um duplo clique sobre ele) e introduzir o novo valor.

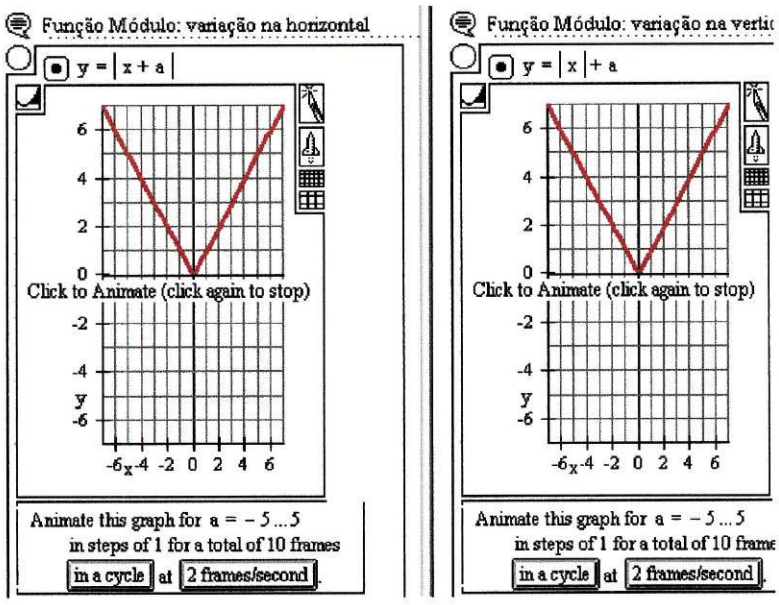
Depois de relembrada a noção de módulo de um número, comecemos por estudar a representação gráfica da função módulo. O Notebook que se segue, tenta fazer uma comparação entre as funções $y=ax$ e $y=|ax|$. Experimente mudar o valor de a que é apresentado no exemplo seguinte e que tenta mostrar isso mesmo.



O exemplo seguinte poderá ser usado para mostrar ao aluno a relação existente entre *a* na função *y*=|*ax*| e a "abertura" do gráfico dessa função. Facilmente se conclui que quanto maior for o valor de |*a*|, menor será a abertura do gráfico.

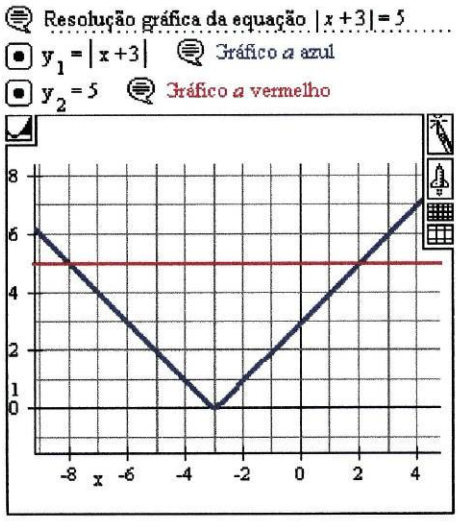


Nos Notebooks seguintes, tenta-se mostrar a diferença entre as duas funções *y*=|*x*+*a*| e *y*=|*x*|+*a*. Para activar alguma das animações, basta clicar sobre o gráfico respectivo. De referir ainda que, no canto superior direito de cada uma das animações, depois de activadas, aparece o valor actual que *a* está a tomar.



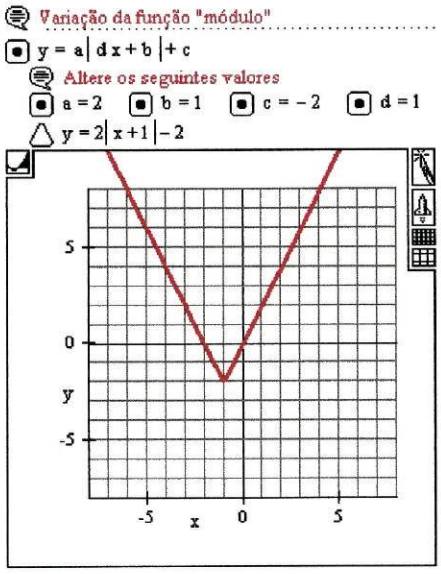
Equações com a função Módulo

De seguida apresenta-se uma resolução gráfica de equações em que está envolvida a função módulo. O exemplo que é apresentado (e que pode, obviamente, ser alterado pelo utilizador), resolve graficamente a equação $|x+3|=5$. Para tal, representaram-se graficamente as duas funções referentes aos dois membros da equação ($y_1=|x+3|$ e $y_2=5$). As soluções situam-se precisamente nos pontos de intersecção dos dois gráficos. Altere os valores envolvidos em cada uma das funções e veja, de imediato, os gráficos obtidos. Repare-se que, por um lado, quando os dois gráficos não se intersectam a equação é impossível e, por outro, quando temos $y_2=0$, a equação apenas tem uma solução. Experimente todas essas situações.



A função $y = a.| d.x + b | + c$

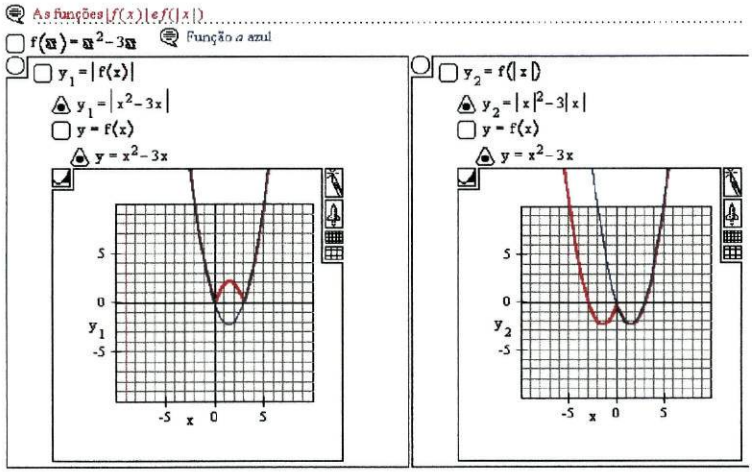
A influência destes parâmetros no comportamento gráfico da função módulo, poderá ser abordada duma forma interessante, através da utilização do Notebook seguinte. De facto, neste Notebook, tenta-se que o aluno conjecture sobre o efeito da variação dos parâmetros *a*, *b*, *c* e *d* na função $y = a.| d.x + b | + c$. Para isso, o aluno deverá alterar os valores de *a*, *b* *c* e *d*, que são inicialmente apresentados e, verificar no gráfico respectivo, as alterações produzidas à função inicial.



As funções $y = | f (x) |$ e $y = f (| x |)$

A composição de funções, em que uma delas é a função módulo, poderá ser abordada com um Notebook como aquele que aqui é apresentado. Este Notebook, constitui um exemplo para tentar mostrar que estamos perante funções diferentes, quando procedemos à composição de funções, em que uma delas é a função módulo. Seleccione, por exemplo, o "3" na função inicial e substitua-o por outro valor à sua escolha, verificando as alterações produzidas. Como se pode verificar, o gráfico da função $y_1 = | f (x) |$ obtém-se do gráfico $y = f (x)$ mantendo os pontos de ordenada positiva ou nula e transformando os pontos de ordenada negativa por uma simetria em relação ao eixo dos xx's. Por seu turno, o gráfico da função $y_2 = f (| x |)$ obtém-se do

gráfico de $y = f(x)$ mantendo os pontos de abcissa positiva e transformando os pontos de abcissa negativa de modo a que pontos de abcissas simétricas sejam simétricos em relação ao eixo dos yy's.



Funções definidas por ramos

As funções definidas por ramos também podem, de certa forma, ser exploradas utilizando o MathView. Dizemos de certa forma, na medida em que, neste tipo de gráficos, o programa apresenta certas limitações. Por exemplo, no primeiro caso que aqui é apresentado, entre um ramo da função e outro ramo (no valor $x = a$) aparece traçada, entre o final da parábola e o início da recta, uma linha, como que de uma parte do gráfico se tratasse. Isto acontece sempre que a função definida por dois (ou mais) ramos, apresenta uma descontinuidade num determinado ponto *a*. Neste caso, o computador traça sempre um segmento de recta a unir os dois pontos $(a,f(a^+))$ e $(a,f(a^-))$, em que $f(a^+)$ é o limite à direita do ponto *a* e $f(a^-)$ é o limite à esquerda do ponto *a*. Isto não acontece quando $f(a^+)$ ou $f(a^-)$ é infinito.

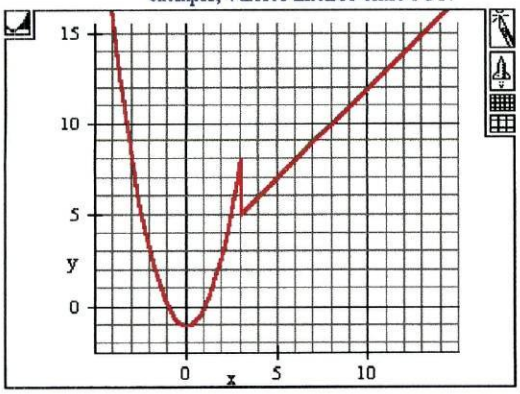
Experimente substituir o valor de *a* por números inteiros entre 1 e 5. Ora, para o último valor pedido, 5, a parte do gráfico referente ao ponto em que há "um salto" da função, não é visível. Antes de a tornarmos visível, poderíamos pedir aos alunos que tentassem prever o que é que se passará nessa parte do gráfico da função estudada. Para deslocar a função para baixo, basta aproximar o ponteiro do rato do gráfico de modo a que se transforme numa mão, premir o botão esquerdo do rato e deslocar o gráfico para a posição pretendida. Para além de alterar o valor de *a*, pode também alterar a expressão analítica de cada um dos ramos, obtendo, dessa forma, uma nova função. De seguida apresenta-se o Notebook respectivo, bem como uma animação desta situação.

Funções definidas por ramos

$y = \begin{cases} x^2 - 1 & (x < a) \\ x + 2 & (x \geq a) \end{cases}$

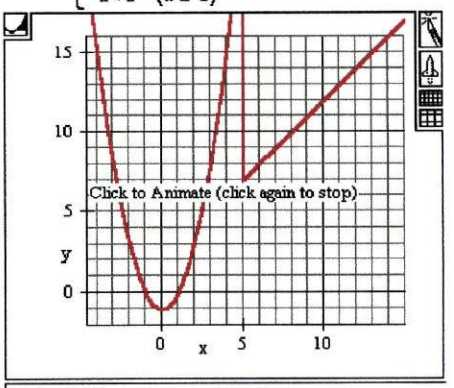
$a = 3$

Altere o valor de "a" atribuindo-lhe, por exemplo, valores inteiros entre 1 e 5.



Funções definidas por ramos

$y = \begin{cases} x^2 - 1 & (x < a) \\ x + 2 & (x \geq a) \end{cases}$ $a = 1$

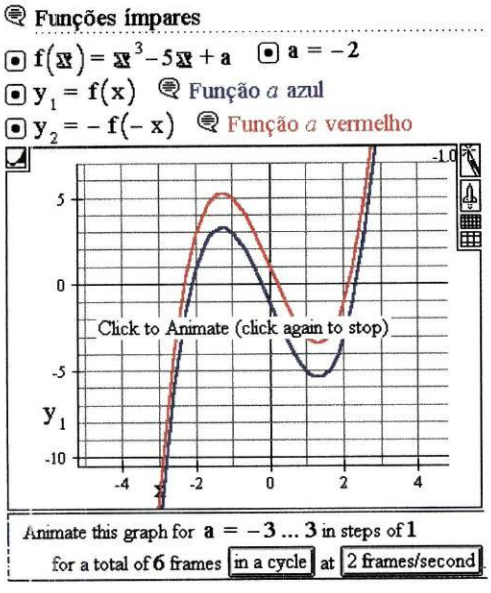


Animate this graph for a = 0 ... 6 in steps of 1
for a total of 6 frames in a cycle at 1 frames/second

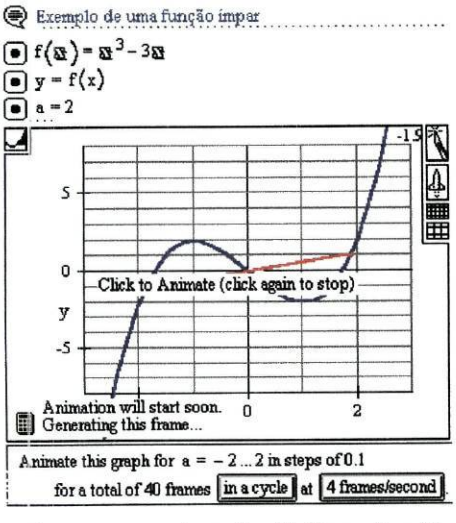
Paridade de funções

Neste espaço, aborda-se a paridade de funções. Pode alterar as expressões analíticas das funções que são apresentadas por defeito, tendo acesso, dessa forma, ao estudo da paridade dessas novas funções.

Funções ímpares



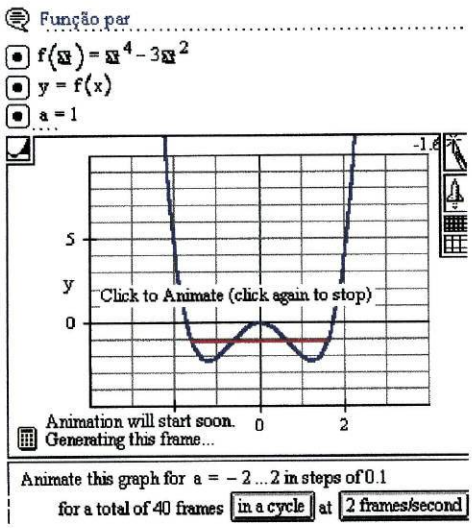
No primeiro exemplo (ao lado), podem ver-se os gráficos de duas funções: o da função *y = f (x)* (traçado a azul) e o da função *y = - f (- x)* (traçado a vermelho). A animação que se mostra, corresponde às várias funções que se obtêm à medida que se vai alterando o valor de "a". Quando os dois gráficos forem coincidentes (isso acontece quando *a = 0*), obtemos, de facto, uma função ímpar.



No exemplo ao lado, pode ver-se o gráfico da função *y = f (x)* (traçado a azul) e um segmento de recta (traçado a vermelho) compreendido entre os pontos (a,f(a)) e (-a,f(-a)) pertencentes ao gráfico da função inicial. De notar que tal segmento contém a origem do referencial (o ponto (0,0)) que corresponde exactamente, para cada valor de "a", ao ponto médio do segmento de recta, ou seja, os pontos (a,f(a)) e (-a,f(-a)) são simétricos em relação à origem, isto é, estamos perante uma função ímpar.

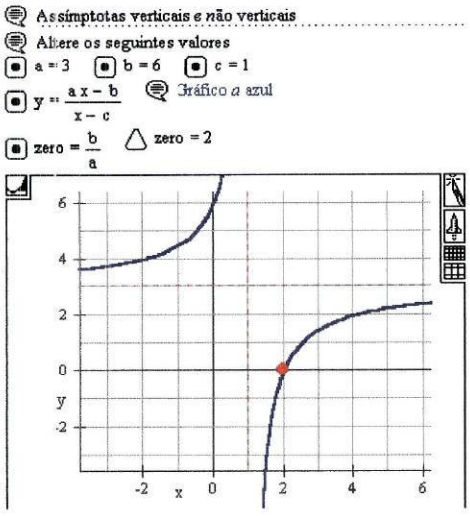
Funções pares

No exemplo seguinte, pode ver-se o gráfico da função $y = f(x)$ (traçado a azul) e um segmento de recta (traçado a vermelho) compreendido entre os pontos (a,f(a)) e (-a,f(-a)) pertencentes ao gráfico da função inicial. De notar que tal segmento é perpendicular ao eixo dos *yy's*, ao mesmo tempo que é bissectado pelo mesmo eixo. Na realidade, os pontos (a,f(a)) e (-a,f(-a)) são simétricos relativamente ao eixo dos *yy's*, isto é, estamos perante uma função par.



Assímptotas de funções racionais

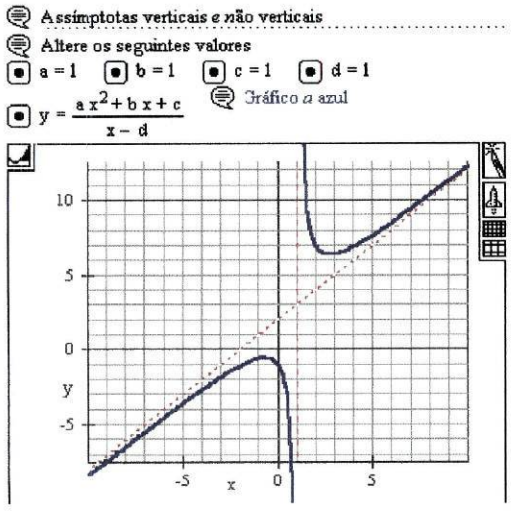
Neste Notebook apresentam-se as assímptotas de uma função racional, em que o numerador e o denominador são funções lineares. Como se pode observar, neste primeiro caso estamos perante uma assímptota vertical e uma assímptota horizontal.



O valor de *a* determina a equação da assímptota horizontal (*y=a*) e o valor de *c* determina a assímptota vertical (*x=c*).

O valor de *b/a* determina o zero da função e, como é óbvio, sempre que *b/a=c*, estamos perante a recta horizontal *y=a*, que não tem assímptotas. Experimente alterar os diferentes valores que são apresentados por defeito e visualize os efeitos produzidos.

No Notebook seguinte, apresentam-se as assímptotas de uma função racional, em que o numerador é uma função quadrática e o denominador uma função linear. Ora, neste caso, estamos perante, como se pode observar, de uma assímptota vertical e de uma assímptota oblíqua.



O valor de *d* determina a equação da assímptota vertical (*x=d*) e o valor de *a* determina o declive da assímptota oblíqua (é óbvio que se *a=0* a assímptota será horizontal).

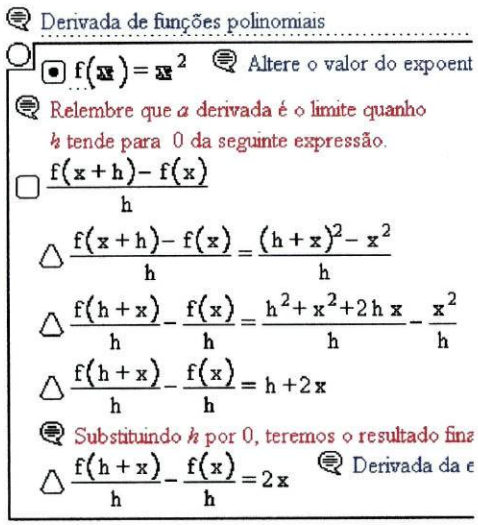
Experimente alterar os diferentes valores que são apresentados por defeito e visualize os efeitos produzidos.

Estudo de derivadas

Neste espaço serão desenvolvidas questões relacionadas com o cálculo algébrico de derivadas, bem como a sua componente geométrica.

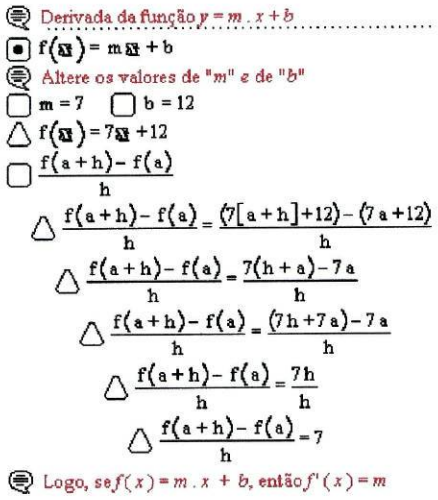
A derivada de x^a

Neste primeiro exemplo, o aluno poderá calcular derivadas de variadíssimas expressões do tipo x^a, com a um número inteiro. Sugere-se que se oriente o aluno a substituir a por números inteiros menores que 19, para que não aconteçam alguns problemas com os cálculos devido à sua complexidade. Depois do aluno substituir a por alguns valores, poderá facilmente chegar à conclusão que a derivada de x^a é $a.x^{a-1}$.



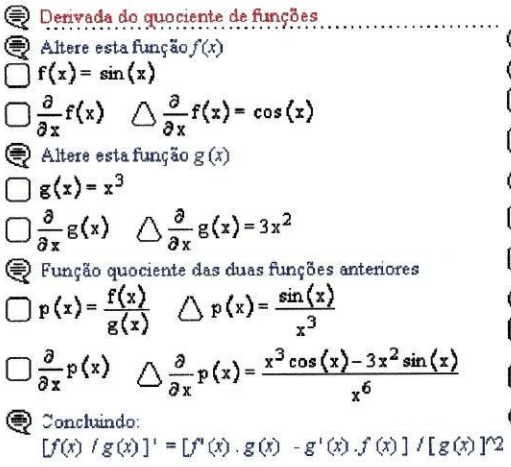
A derivada da função afim

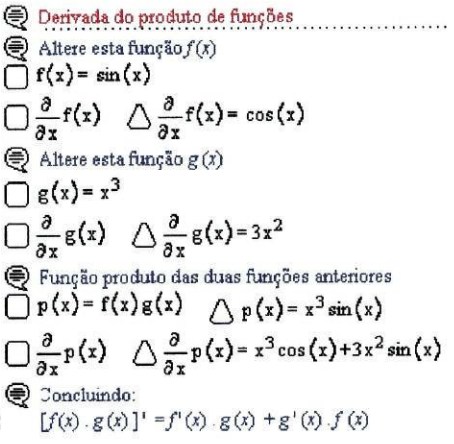
No Notebook seguinte, trata-se a derivada da função afim. O aluno deverá alterar, à sua vontade, os valores de m e de b na expressão $y=m.x+b$, de modo a poder concluir que $y'=m$.

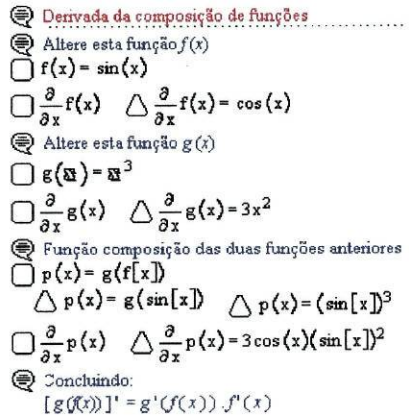


Regras de derivação

Relativamente às regras de derivação, de seguida apresentam-se alguns Notebooks que ajudarão o aluno a deduzir algumas delas, nomeadamente as que dizem respeito ao produto, ao quociente e à composição de funções, conforme imagens em baixo. Experimente alterar a expressão analítica das funções que são apresentadas e verifique os resultados obtidos. Relativamente a estas situações, há que ter em atenção a notação para designar a derivada de uma dada função, utilizada pelo programa e, consequentemente, nestes Notebooks, por não ser conhecida pelos alunos do Ensino Secundário.

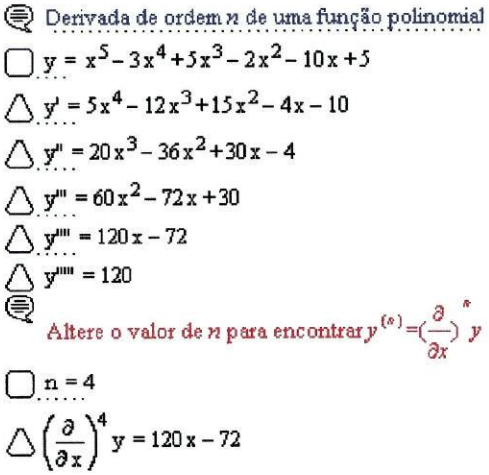






Derivadas de ordem *n*

O Notebook seguinte, poderá constituir uma forma interessante para o cálculo de derivadas de ordem *n*, de funções polinomiais. No exemplo que é apresentado, a função que aparece retratada é um polinómio do 5º grau, em que são visíveis as expressões referentes às cinco primeiras derivadas. Em alternativa, poderemos no mesmo Notebook calcular a derivada de ordem *n*, alterando o valor de *n* no respectivo espaço. Experimente também alterar a função polinomial que é dada, introduzindo por exemplo funções polinomiais de outros graus.



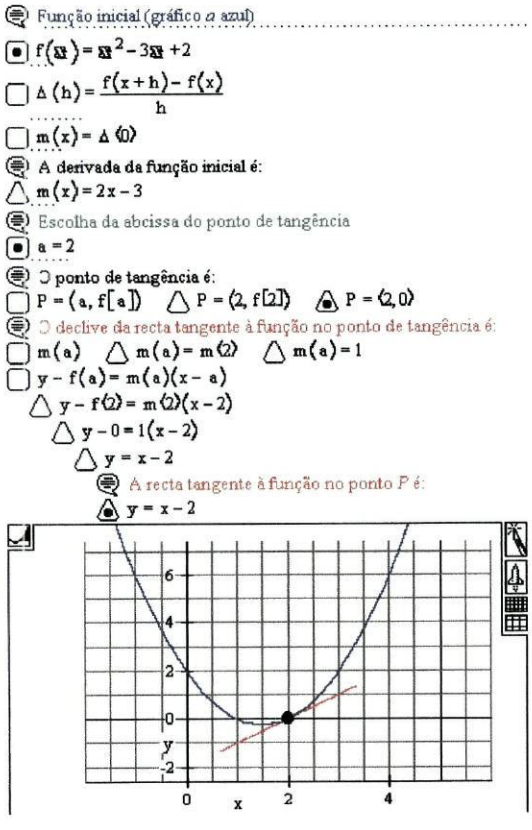
De seguida, sugere-se a análise de alguns Notebooks para um estudo mais aprofundado das derivadas sob o ponto de vista geométrico.

O significado geométrico de derivada

A derivada de uma função num ponto é o declive da recta tangente à função nesse mesmo ponto. Isto para funções regulares, ou seja, que são deriváveis numa vizinhança do ponto considerado. Esta interpretação geométrica da noção de derivada é importantíssima para se perceberem determinados conceitos que envolvem derivadas. No Notebook seguinte, a partir de uma função polinomial (do segundo grau), tenta-se que o aluno visualize esse aspecto geométrico, depois de ter aprendido a calcular derivadas sob o ponto de vista algébrico.

Relativamente a esse Notebook, há a referir o seguinte:

- A primeira expressão diz respeito à função inicial, cujo gráfico aparece representado a azul. Poderá alterar os coeficientes que são apresentados por defeito e, com esses novos coeficientes, obterá uma nova função e, consequentemente, uma nova função derivada.



- No passo seguinte, inicia-se o cálculo da derivada da função, através da definição. Estão escondidos os cálculos intermédios mas, como é óbvio, como estamos a calcular o limite quando *h* tende para zero, determinando $\Delta(0)$, fica calculada a derivada da função inicial (neste caso, estamos perante uma função afim, que designamos por *m(x)*).

- Seguidamente, pede-se ao aluno que introduza a abcissa do ponto que se pretende que seja o ponto de tangência.

Depois de calculada a imagem desse ponto, designaremos por P esse ponto de tangência. Experimente alterar o valor da abcissa e visualize as alterações produzidas.

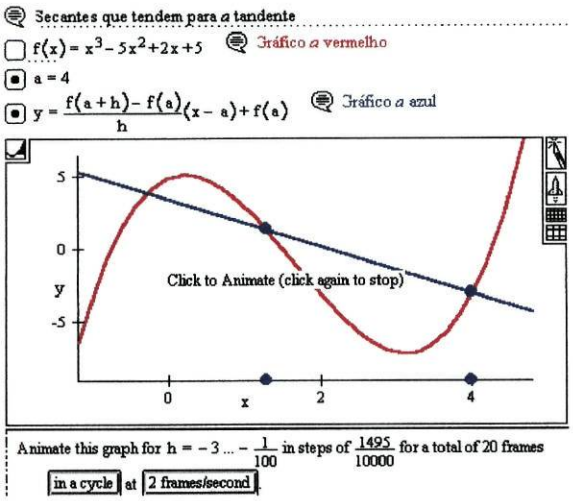
- De seguida, através do cálculo de m(a), obtém-se o declive da recta tangente à função no ponto de tangência considerado, o que permite determinar analiticamente, a equação reduzida dessa mesma recta.

- Finalmente, apresenta-se no mesmo referencial cartesiano, o gráfico da função inicial, o ponto de tangência e a recta tangente à função nesse mesmo ponto (cujo declive constitui a derivada da função no ponto de tangência).

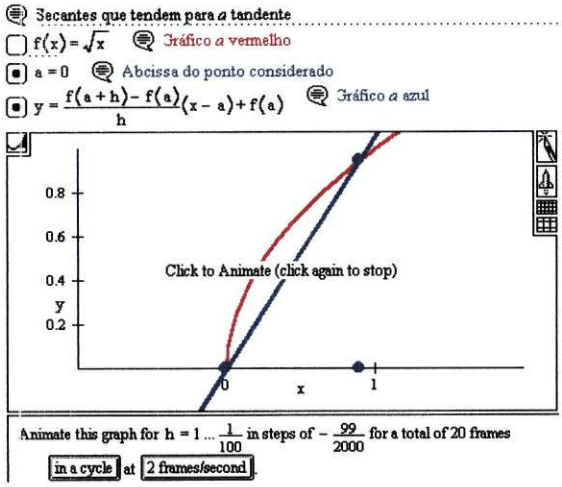
- Uma chamada de atenção para referir que as escalas consideradas nos dois eixos cartesianos não são iguais.

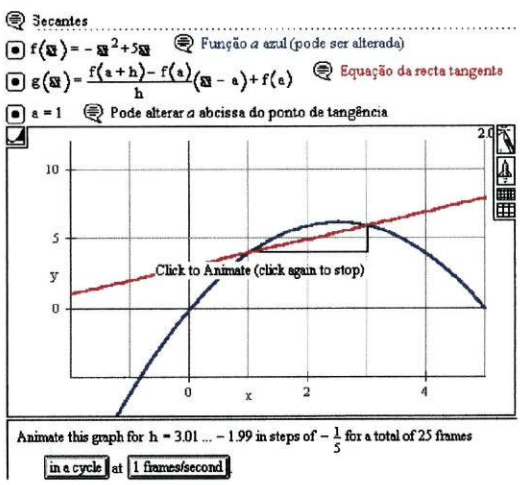
Secantes e tangentes

De seguida, apresenta-se um Notebook cuja finalidade é sugerir ao aluno o facto das rectas secantes a uma função em que um dos pontos é fixo, tendem a aproximar-se da recta tangente por esse ponto, quando fazemos diminuir a distância entre os pontos (*h*). De facto, o declive da recta que está a ser considerada, quando *h* se aproxima cada vez mais de zero, será, por definição, a derivada da função no ponto de tangência. Experimente alterar o valor de "a" (abcissa do ponto de tangência) e verifique que os resultados se mantêm.



De seguida apresentam-se mais dois exemplos desta situação, em que são consideradas outras duas funções, uma das quais do 2º grau.

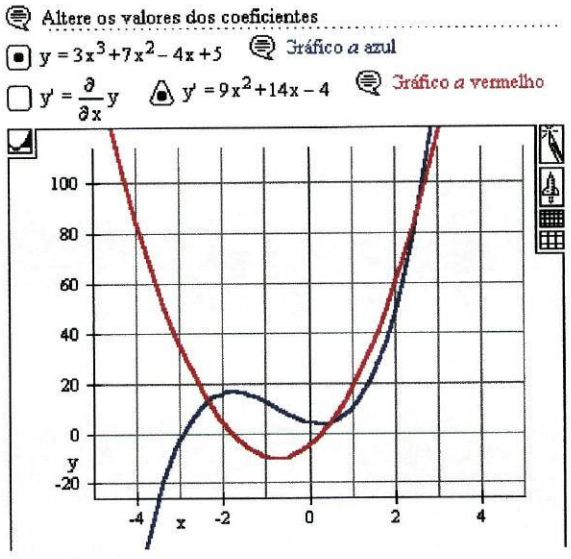




O sinal da derivada e a monotonia da função

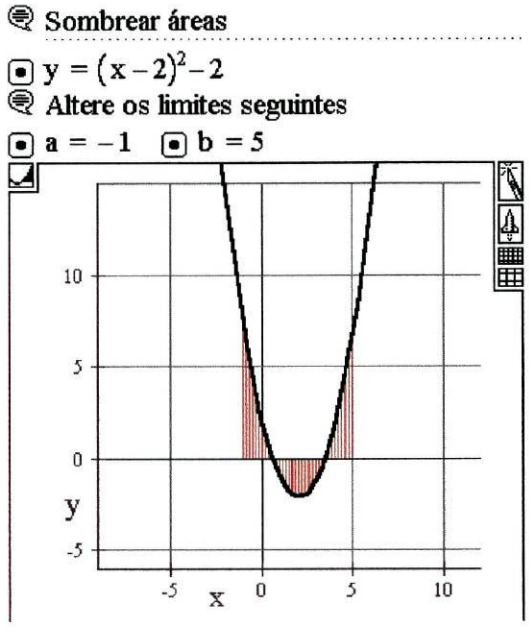
No Notebook seguinte, tentar-se-à que o aluno verifique, através de um exemplo de uma função polinomial do 3º grau bem escolhida, que uma função é crescente apenas quando a sua derivada for positiva. A visualização dos dois gráficos (função e respectiva derivada) no mesmo sistema de eixos, poderá ser uma forma para motivar a demonstração desse importante resultado. Refira-se ainda que o aluno poderá alterar os coeficientes que inicialmente foram dados e visualizar o mesmo resultado para as novas funções obtidas. Além disso,

poderá relembrar-se ao aluno que a derivada de uma função polinomial do 3º grau é uma função polinomial do 2º grau.

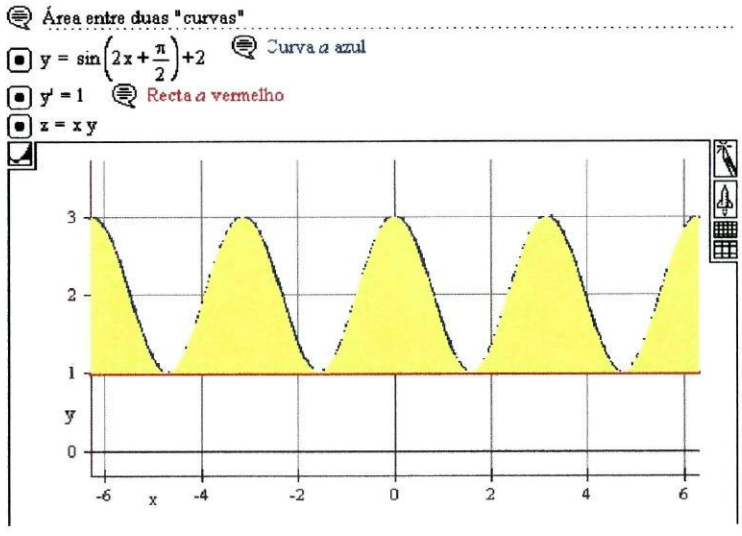


Sombrear áreas

No Notebook seguinte, define-se uma área entre o gráfico referente a uma parábola e uma recta (eixo dos xx's). Experimente alterar quer a parábola quer os limites entre os quais se pretende definir a área e veja as alterações produzidas.



No Notebook seguinte também se define a área entre o gráfico de uma função trigonométrica e uma recta, definida pelo utilizador. Experimente alterar as duas expressões e visualize as alterações produzidas.



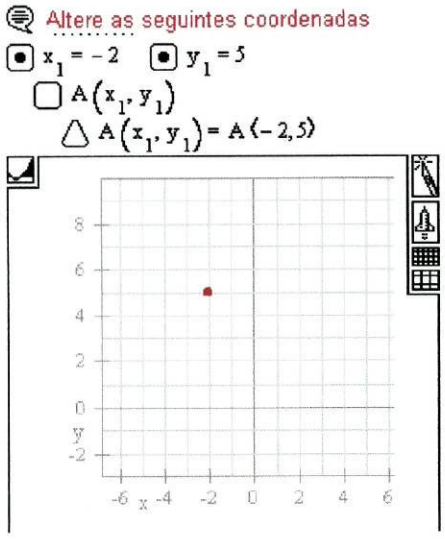
Geometria no plano

É referido nos programas do Ensino Secundária que o "Ensino da Geometria reveste-se da maior importância devendo desenvolver no aluno uma intuição geométrica e um raciocínio espacial assim como capacidades para explorar, conjecturar, raciocinar logicamente, usar e aplicar a Matemática, formular e resolver problemas abstractos ou numa perspectiva de modelação matemática". Neste espaço, abordam-se algumas questões relacionadas com este tema, mais propriamente com Geometria no Plano.

Apresentam-se igualmente algumas questões que, não fazendo parte dos programas do ensino secundário, mostravam também interesse em serem abordadas, com vista a que se desse conta do desempenho do MathView.

Coordenadas no plano

Relembremos a marcação de pontos num referencial cartesiano. No Notebook seguinte, o aluno pode mudar as coordenadas x_1 e y_1 e, de imediato, o ponto marcado no referencial apresentado muda de posição, de acordo com as novas coordenadas.



Distância entre dois pontos

Tal como em vários conteúdos programáticos, a abordagem desta questão pode ser feita de uma forma mais directa (para aqueles alunos que já dominam o tema, funcionando para eles o Notebook como algo que servirá para evitar os cálculos manuais), ou de uma forma indirecta (para aqueles alunos que iniciam o tema e que, portanto, deverão ter acesso a todos os passos). O Notebook seguinte mostra a expressão analítica que nos dá, directamente, a distância entre dois pontos.



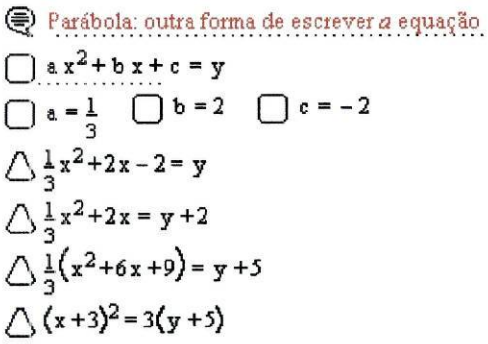
Cónicas

O estudo das cónicas também poderá ser feito utilizando este programa. A representação gráfica das cónicas, costuma ser de alguma complexidade para os alunos, designadamente os do 10º ano, que iniciam o seu estudo. Vejamos os Notebooks que se seguem e que ajudarão os alunos a visualizar melhor o comportamento das cónicas, à medida que vão variando os parâmetros envolvidos.

Parábola

Já vimos aquando do estudo da função quadrática, algumas das propriedades das parábolas. Vejamos agora o caso de outras parábolas.

É importante referir que qualquer expressão do tipo $y=a.x^2+b.x+c$ pode ser escrita da forma $(x+m)^2=n.(y+p)$ e, por seu turno, as expressões do tipo $x=a.y^2+b.y+c$, que também dão origem a parábolas, podem ser escritas da forma $(y+m)^2=n.(x+p)$.

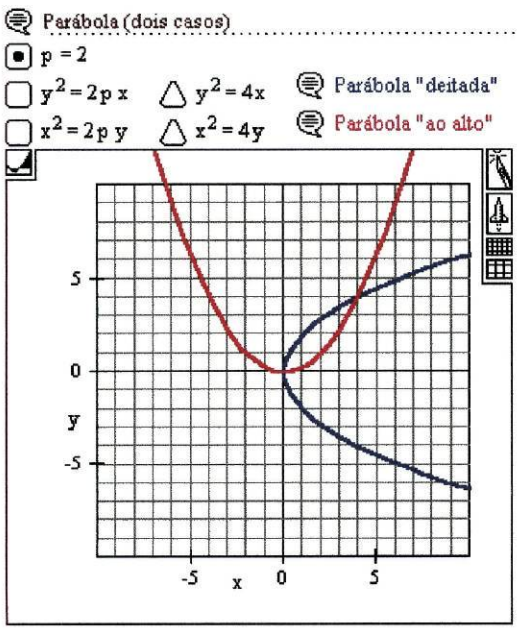


No Notebook ao lado, tenta mostrar-se isso mesmo. Experimente alterar os valores de *a*, *b* e *c* no caso que é apresentado, para expressões do tipo $y=a.x^2+b.x+c$. O outro caso (referente a expressões do tipo $x=a.y^2+b.y+c$), depois de visto este, torna-se idêntico.

De uma forma geral, podemos então apresentar a seguinte definição:

Uma parábola é um conjunto de pontos do plano equidistantes de um ponto fixo (foco) e de uma recta (directriz) que não o contém (os elementos da parábola são: o foco, o vértice e a directriz).

O Notebook que de seguida se apresenta, mostra dois casos possíveis de parábolas, cujas directrizes são rectas verticais ou horizontais: o primeiro caso, em que a directriz é uma recta horizontal (parábola "ao alto"), cujo gráfico está esboçado a vermelho e, o segundo caso, em que a directriz é uma recta vertical (parábola "deitada"), cujo gráfico está esboçado a azul. De referir que estamos a considerar ainda apenas as parábolas cujo vértice é a origem do referencial: O(0,0).



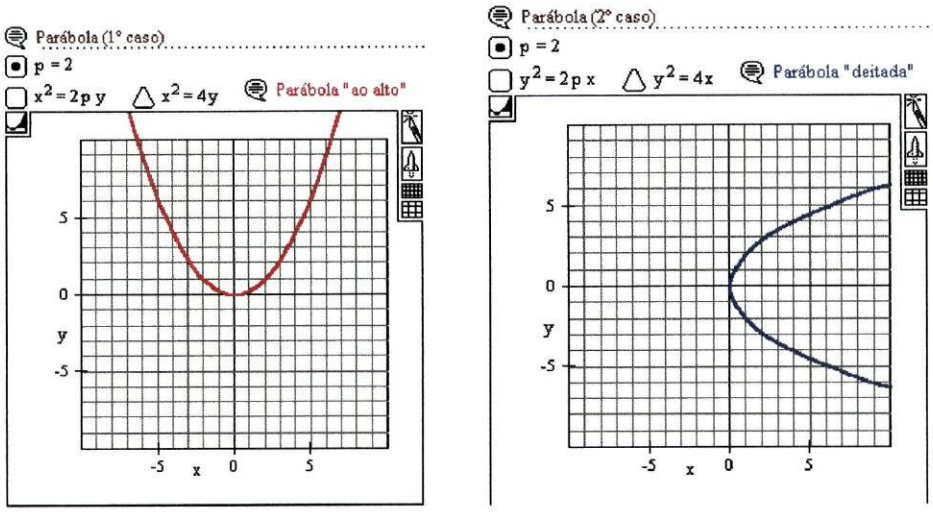
As parábolas do tipo $x^2 = 2.p.y$ e $y^2 = 2.p.x$

Vejamos agora os dois casos em separado, para que possam ser visualizadas com maior pormenor as suas propriedades.

No Notebook seguinte (da esquerda), mostra-se uma parábola com vértice na origem, ou seja, em O(0,0). Trata-se de uma parábola do tipo $x^2 = 2.p.y$, em que p, também chamado parâmetro da parábola, representa o módulo da distância do foco à directriz.

Experimente agora alterar os valores de p e visualize as alterações produzidas. Como se verifica, ao alterarmos o valor de p, a parábola fica mais larga ou mais estreita.

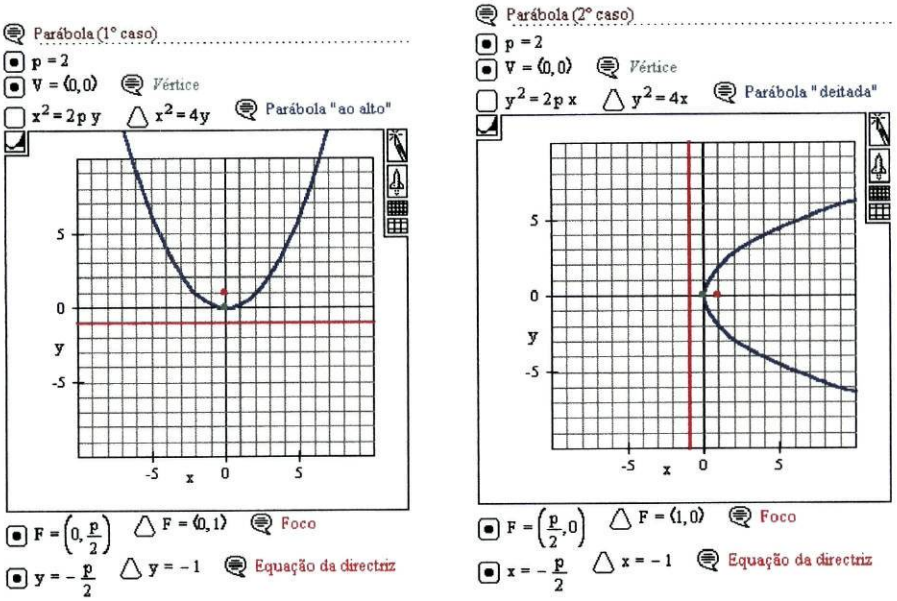
Refira-se ainda que para valores positivos de p teremos parábolas com a concavidade voltada para cima e para valores negativos de p teremos parábolas com a concavidade voltada para baixo.



O caso das parábolas do tipo $y^2 = 2.p.x$ aparece retratado na imagem em cima (à direita). Experimente também alterar os valores de p. Como se pode verificar, para valores positivos de p teremos parábolas com a concavidade voltada para a direita e para valores negativos de p teremos parábolas com a concavidade virada à esquerda.

A localização do foco, do vértice e da directriz

No Notebook seguinte, mostram-se os elementos da parábola, analítica e geometricamente: a localização do foco e respectivas coordenadas, a localização do vértice e respectivas coordenadas, bem como a representação da directriz da parábola e respectiva equação. Experimente mudar o valor de *p* no Notebook e visualize as alterações produzidas. São apresentados os casos das parábolas do tipo $x^2 = 2.p.y$, bem como das parábolas do tipo $y^2 = 2.p.x$.

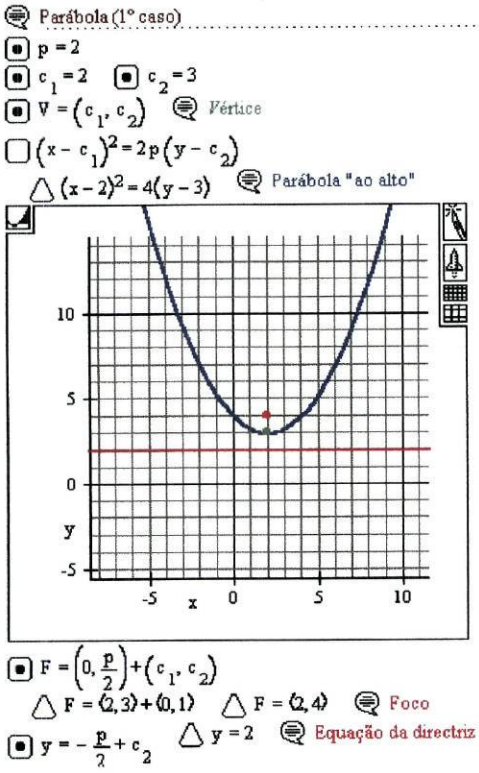


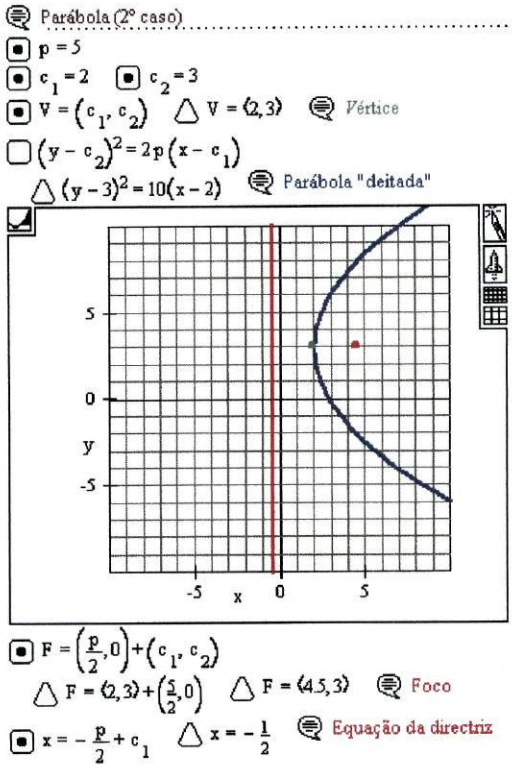
Parábolas não centradas

Vejamos agora o caso das parábolas que não estão centradas, isto é, que não têm o vértice no ponto V(0,0). Consideremos então o vértice da parábola como sendo o ponto de coordenadas $V(c_1,c_2)$.

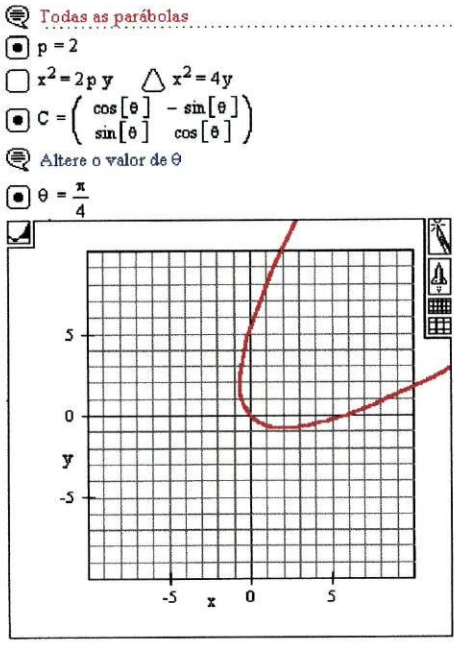
Nos Notebooks seguintes, são apresentados todos os casos de parábolas "ao alto" (respectivamente "deitadas"), ou seja, com a concavidade voltada para cima ou para baixo (respectivamente, para a direita ou para a esquerda) , dependendo do sinal de *p* (claro que o caso anterior se resume a atribuir os valores de $c_1 = 0$ e $c_2 = 0$). A imagem da esquerda, traduz as parábolas do tipo $(x-c_1)^2=2.p.(y-c_2)$, cujo vértice é o ponto $V(c_1,c_2)$ e a da direita as parábolas do tipo $(y-c_2)^2=2.p.(x-c_1)$.

Experimente alterar os valores de *p* e/ou as coordenadas do vértice (c_1 e c_2) e veja as alterações produzidas.





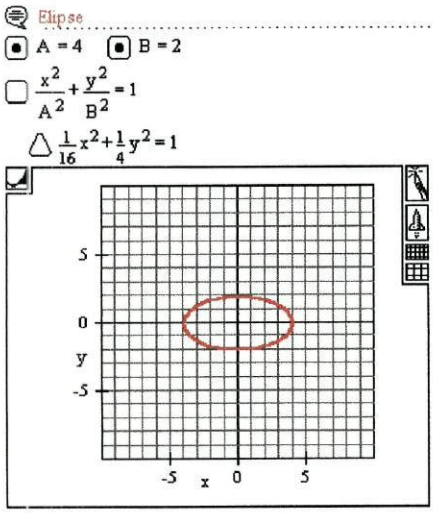
Vejamos agora outras parábolas, para além daquelas que foram atrás representadas. O Notebook seguinte, contempla todos os casos de parábolas, incluindo os anteriores, que são obtidos por uma rotação (no sentido positivo) de ângulo θ.



Elipse

Uma elipse é um conjunto de pontos do plano cuja soma das distâncias a dois pontos fixos (a que chamamos focos) é constante e maior do que a distância entre eles (os elementos da elipse são: o centro, o eixo maior, o eixo menor, os focos e os vértices).

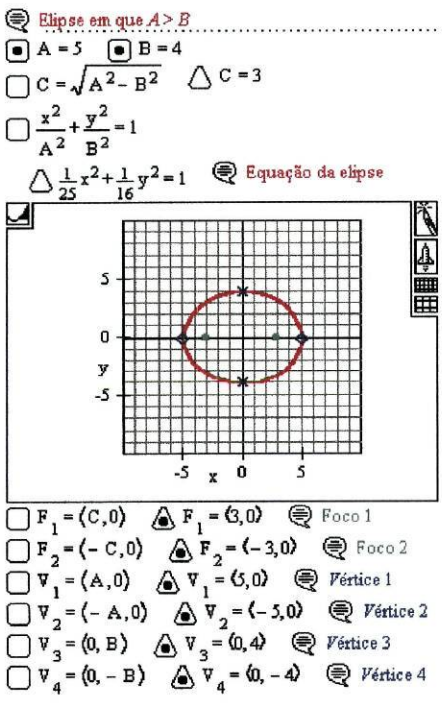
No Notebook seguinte, mostra-se uma elipse com centro na origem, ou seja, em C(0,0), em que A e B são os comprimentos dos semi-eixos, horizontal e vertical, respectivamente.

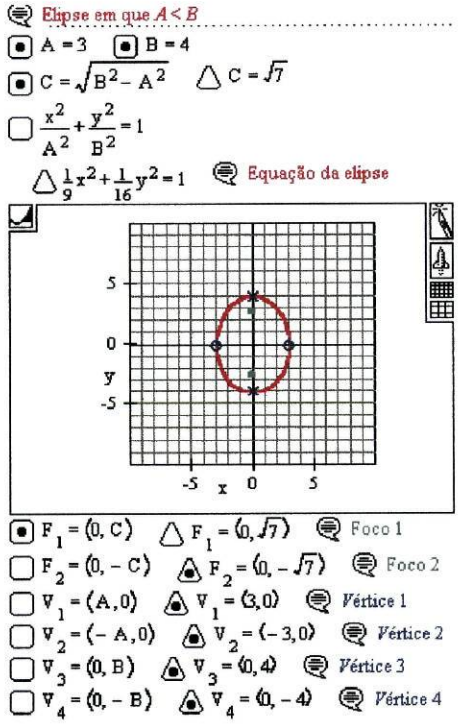


Experimente atribuir outros valores a A e a B e visualize as alterações produzidas em termos de posição da elipse obtida.
Convém ainda referir que apenas poderão atribuir-se valores positivos a A e a B, uma vez que se tratam de comprimentos.

A localização dos focos e dos vértices

Nos Notebooks seguintes, mostram-se os elementos da elipse, analítica e geometricamente: a localização dos focos e respectivas coordenadas e a localização dos vértices bem como as respectivas coordenadas. Experimente mudar os valores de A e de B no Notebook e visualize as alterações produzidas. Note-se que foram separados os casos $A > B$ e $A < B$ para que não surgissem problemas no cálculo do valor de C.



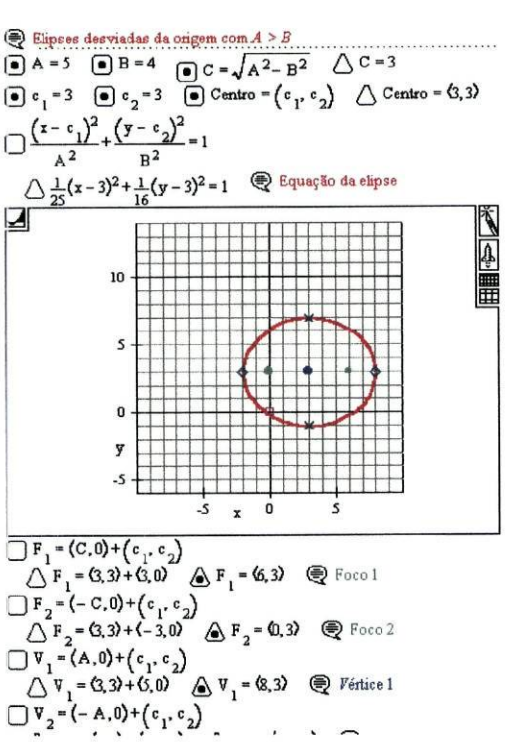


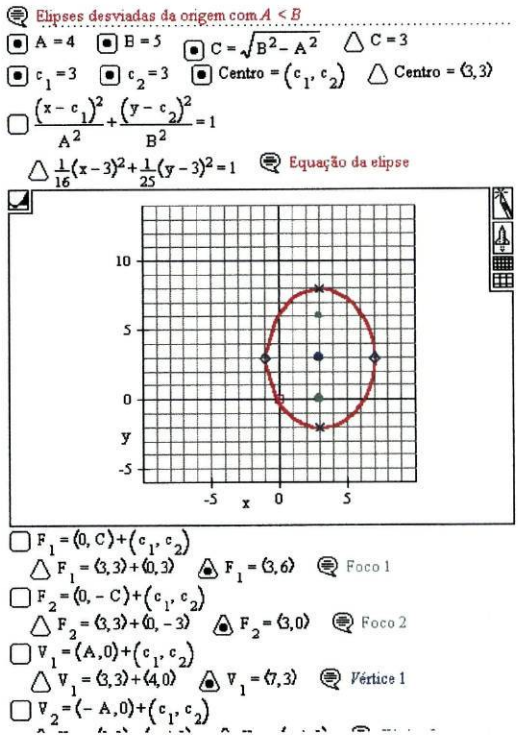
Elipses não centradas

Vejamos agora o caso das elipses que não estão centradas, isto é, que não têm centro no ponto C(0,0). Consideremos o centro da elipse como sendo o ponto de coordenadas $C(c_1,c_2)$.

Nos Notebooks seguintes, são apresentados todos os casos de elipses cujos eixos maior e menor são paralelos aos eixos cartesianos (claro que o caso anterior se resume a atribuir os valores de $c_1 = 0$ e $c_2 = 0$). De notar ainda que, para não existirem problemas no cálculo de C, optou-se novamente por se separar os casos de $A > B$ e $A < B$.

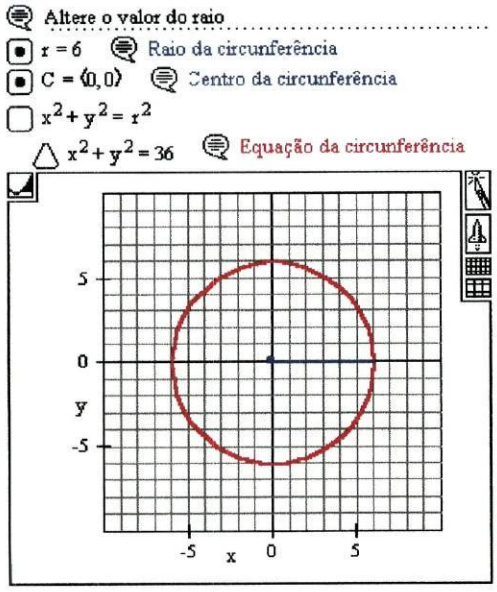
Experimente alterar os valores de A e de B, bem como os valores das coordenadas do centro da elipse (c_1 e c_2).

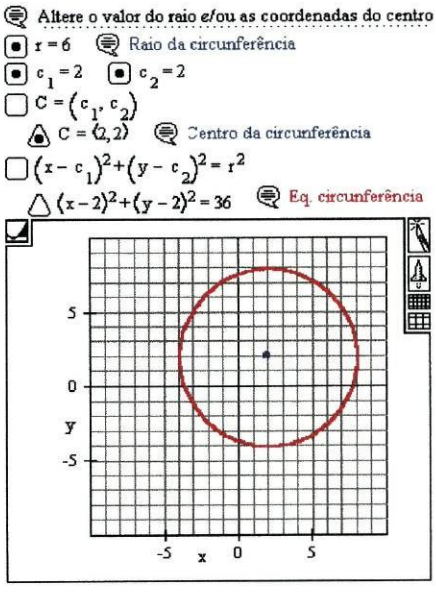




Um caso particular: a circunferência

Antes de terminarmos a abordagem das elipses, seria conveniente uma referência ao caso em que $A = B$, caso esse em que a elipse é uma circunferência. Os Notebooks que de seguida se apresentam retratam as circunferências de centro C(0,0) e raio r e as circunferências não centradas na origem, isto é, de centro C(c_1 ,c_2) e raio r.





Vejamos ainda outras elipses, para além daquelas que foram atrás apresentadas. O Notebook seguinte, contempla os restantes casos de elipses, que são obtidas das anteriores por uma rotação (no sentido positivo) de ângulo θ.

Outras elipses

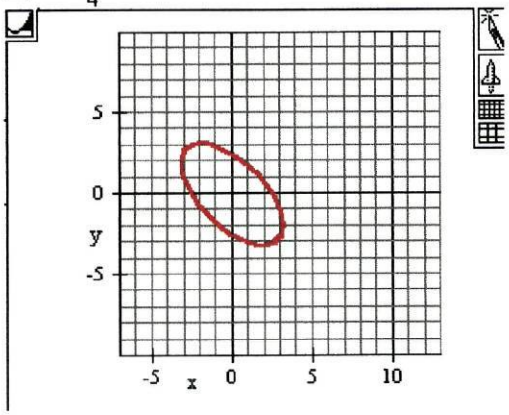
$A = 4$ $B = 2$

$\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1$ $\frac{1}{16}x^2 + \frac{1}{4}y^2 = 1$

$C = \begin{pmatrix} \cos[\theta] & -\sin[\theta] \\ \sin[\theta] & \cos[\theta] \end{pmatrix}$

Altere o valor de θ

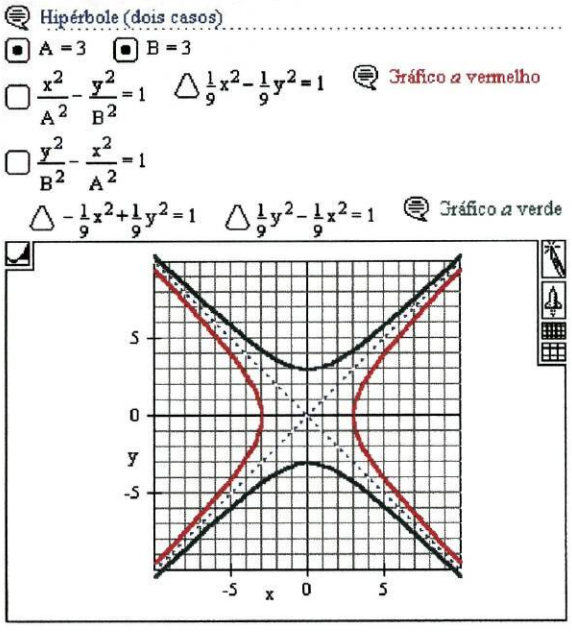
$\theta = \frac{\pi}{4}$



Hipérbole

Uma hipérbole é um conjunto de pontos do plano tais que o módulo da diferença das distâncias a dois pontos fixos (a que chamamos focos) é constante e menor que a distância entre eles (os elementos da hipérbole são: os focos, os vértices e as assímptotas).

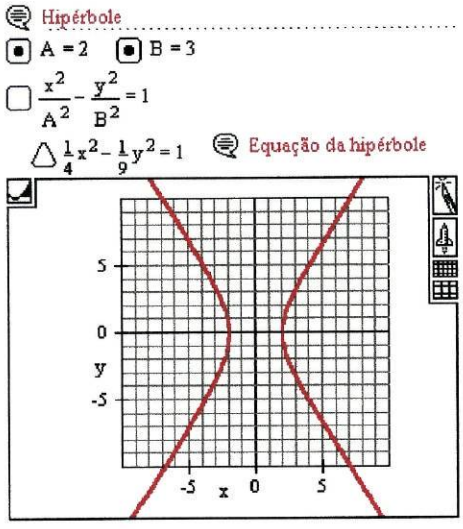
O Notebook que de seguida se apresenta, mostra dois casos possíveis de hipérboles: o primeiro caso, em que os vértices das duas parábolas que a constituem se situam sobre o eixo dos *xx's* (gráfico a vermelho) e, o segundo caso, em que os vértices das duas parábolas que constituem a hipérbole se situam sobre o eixo dos *yy's* (gráfico a verde). De referir que estamos a considerar ainda apenas as hipérboles cujo ponto médio entre os focos é a origem do referencial: O(0,0).

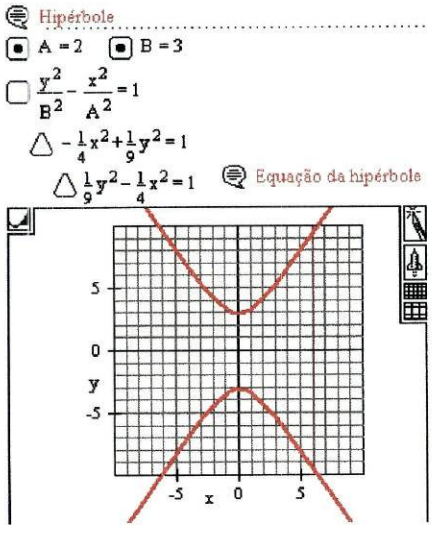


Vejamos agora os dois casos em separado, no sentido de se visualizarem com mais pormenor as suas propriedades.

No Notebook seguinte, mostra-se uma hipérbole com centro na origem, ou seja, em C(0,0), em que *2*A* é a constante a que se refere a definição.

Como facilmente se verifica, é irrelevante se $A > B$ ou se $A < B$, uma vez que a hipérbole aparece exactamente na mesma posição. Experimente visualizar esse facto atribuindo valores diferentes a A e a B. Como se verifica, ao alterarmos o valor de A o efeito provocado é um afastamento ou aproximação dos dois vértices, enquanto que a alteração ao valor de B faz alargar ou apertar a concavidade das duas "parábolas".





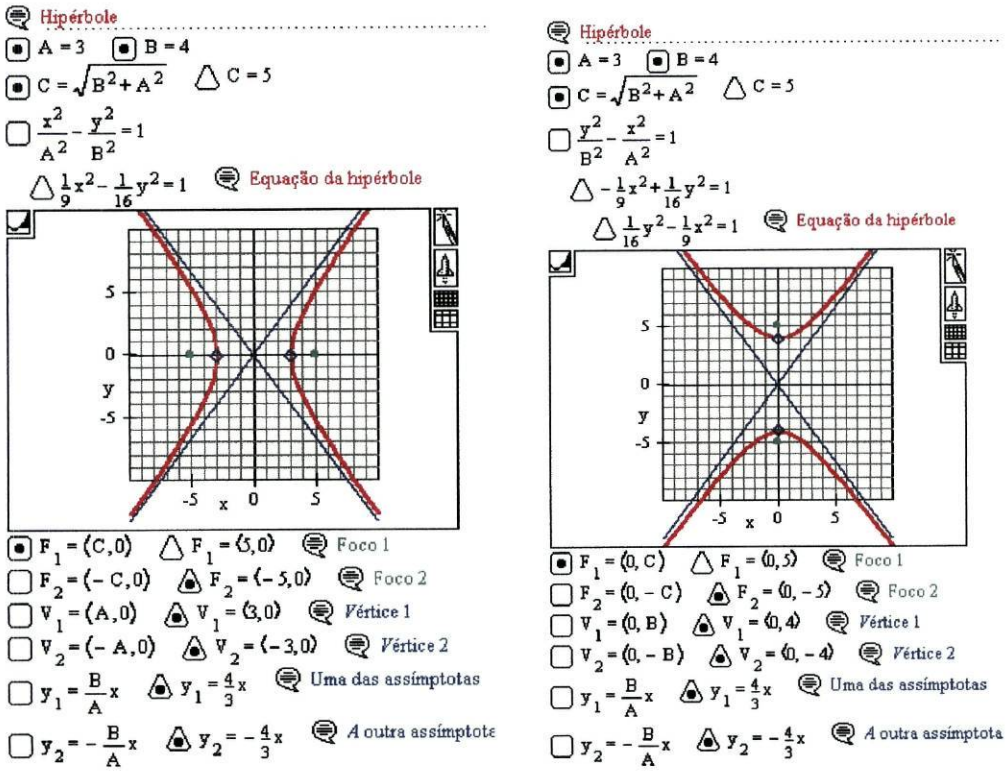
Há ainda outro caso a considerar. Trata-se da hipérbole cujos focos estão situados sobre o eixo dos *yy's*, ao contrário do caso anterior em que os focos se situavam sobre o eixos dos *xx's*. A imagem ao lado retrata essa situação.

Convém ainda referir que apenas poderão atribuir-se valores positivos a A e a B, uma vez que se trata de comprimentos.

A localização dos focos, dos vértices e das assímptotas

Nos Notebooks seguintes, mostram-se os elementos da hipérbole, analítica e geometricamente: a localização dos focos e respectivas coordenadas, a localização dos vértices das duas parábolas que constituem a hipérbole e respectivas coordenadas, bem como a representação das assímptotas da hipérbole e respectivas equações. Experimente mudar os valores de A e de B

nos Notebooks e visualize as alterações produzidas. Note-se que são apresentados o caso em que os focos estão sobre o eixo dos *xx's* e o caso em que os focos se encontram sobre o eixo dos *yy's*.



As hipérboles não centradas

Vejamos agora o caso das hipérboles não centradas, ou seja, que não têm o ponto médio dos focos na origem do referencial. Consideremos então o ponto médio dos focos como sendo o ponto $C(c_1,c_2)$.

Nos Notebooks seguintes, são apresentados os casos de hipérboles em que os segmentos de recta que unem os focos são paralelos aos eixos coordenados (claro que os casos anteriores se resumem a atribuir os valores de $c_1 = 0$ e $c_2 = 0$) e que se encontram alinhados sobre uma recta vertical (imagem à esquerda) ou horizontal (imagem à direita). De notar ainda que, em ambos os casos as assímptotas das hipérboles intersectam-se no ponto médio do segmento de recta cujos extremos são os focos da hipérbole.

Experimente alterar os valores de *A* e de *B*, bem como os valores das coordenadas do centro da hipérbole (c_1 e c_2).

Hipérbole

A = 3 B = 4 $C = \sqrt{B^2 + A^2}$ C = 5

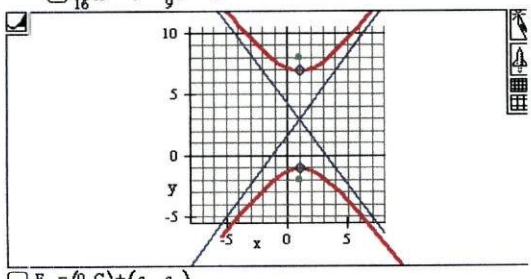
$c_1 = 1$ $c_2 = 3$

Centro = (c_1, c_2) Centro = (1, 3)

$\frac{(y - c_2)^2}{B^2} - \frac{(x - c_1)^2}{A^2} = 1$

$-\frac{1}{9}(x-1)^2 + \frac{1}{16}(y-3)^2 = 1$

$\frac{1}{16}(y-3)^2 = \frac{1}{9}(x-1)^2 + 1$ Equação da hipérbole



$F_1 = (0, C) + (c_1, c_2)$

$F_1 = (0, 5) + (1, 3)$ $F_1 = (1, 8)$ Foco 1

$F_2 = (0, -C) + (c_1, c_2)$

$F_2 = (1, 3) + (0, -5)$ $F_2 = (1, -2)$ Foco 2

$V_1 = (0, B) + (c_1, c_2)$

$V_1 = (0, 4) + (1, 3)$ $V_1 = (1, 7)$ Vértice 1

$V_2 = (0, -B) + (c_1, c_2)$

$V_2 = (1, 3) + (0, -4)$ $V_2 = (1, -1)$ Vértice 2

$y_1 = \frac{B}{A}(x - c_1) + c_2$

$y_1 = \frac{4}{3}(x-1) + 3$ $y_1 = \frac{4}{3}x + \frac{5}{3}$ Uma das assímptotas

$y_2 = -\frac{B}{A}(x - c_1) + c_2$

$y_2 = -\frac{4}{3}(x-1) + 3$ $y_2 = -\frac{4}{3}x + \frac{13}{3}$ A outra assímptota

Hipérbole

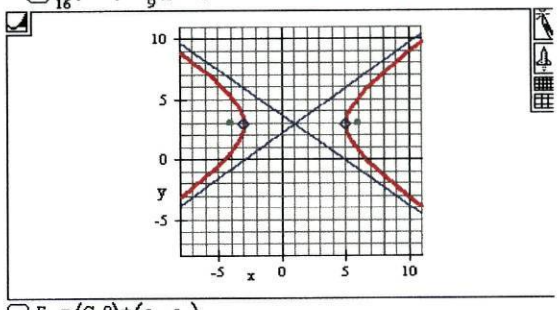
A = 4 B = 3 $C = \sqrt{B^2 + A^2}$ C = 5

$c_1 = 1$ $c_2 = 3$

Centro = (c_1, c_2) Centro = (1, 3)

$\frac{(x - c_1)^2}{A^2} - \frac{(y - c_2)^2}{B^2} = 1$

$\frac{1}{16}(x-1)^2 - \frac{1}{9}(y-3)^2 = 1$ Equação da hipérbole



$F_1 = (C, 0) + (c_1, c_2)$

$F_1 = (1, 3) + (5, 0)$ $F_1 = (6, 3)$ Foco 1

$F_2 = (-C, 0) + (c_1, c_2)$

$F_2 = (1, 3) + (-5, 0)$ $F_2 = (-4, 3)$ Foco 2

$V_1 = (A, 0) + (c_1, c_2)$

$V_1 = (1, 3) + (4, 0)$ $V_1 = (5, 3)$ Vértice 1

$V_2 = (-A, 0) + (c_1, c_2)$

$V_2 = (1, 3) + (-4, 0)$ $V_2 = (-3, 3)$ Vértice 2

$y_1 = \frac{B}{A}(x - c_1) + c_2$

$y_1 = \frac{3}{4}(x-1) + 3$ $y_1 = \frac{3}{4}x + \frac{9}{4}$ Uma das assímptotas

$y_2 = -\frac{B}{A}(x - c_1) + c_2$

$y_2 = -\frac{3}{4}(x-1) + 3$ $y_2 = -\frac{3}{4}x + \frac{15}{4}$ A outra assímptota

Vejamos ainda outras hipérboles, para além daquelas que foram atrás apresentadas. O Notebook seguinte, contempla os restantes casos de hipérboles, que são obtidas das anteriores por uma rotação (no sentido positivo) de ângulo θ.

Outras hipérbole

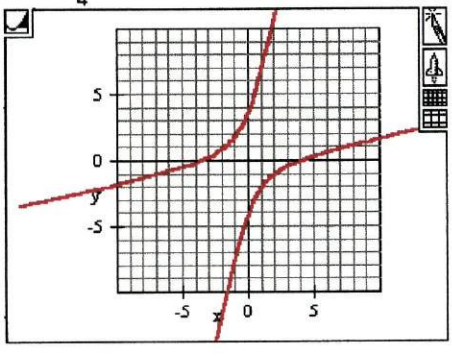
A = 2 B = 3

$\frac{x^2}{A^2} - \frac{y^2}{B^2} = 1$ $\frac{1}{4}x^2 - \frac{1}{9}y^2 = 1$

$C = \begin{pmatrix} \cos[\theta] & -\sin[\theta] \\ \sin[\theta] & \cos[\theta] \end{pmatrix}$

Altere o valor de θ

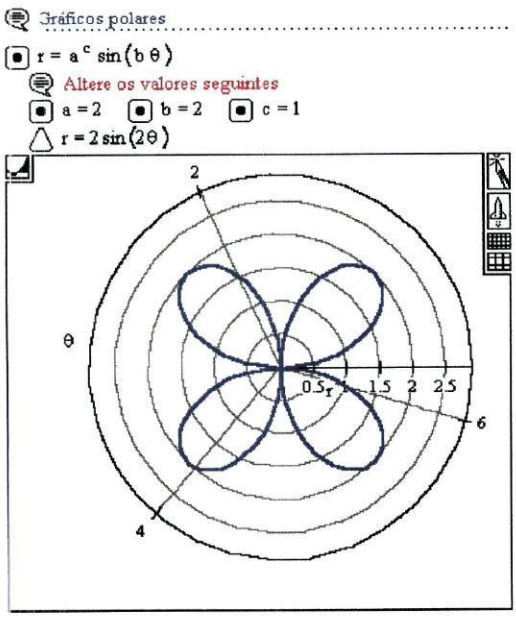
$\theta = \frac{\pi}{4}$

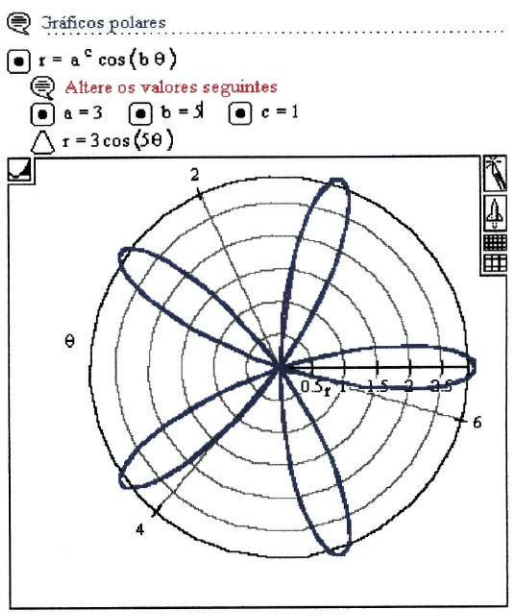


Gráficos polares

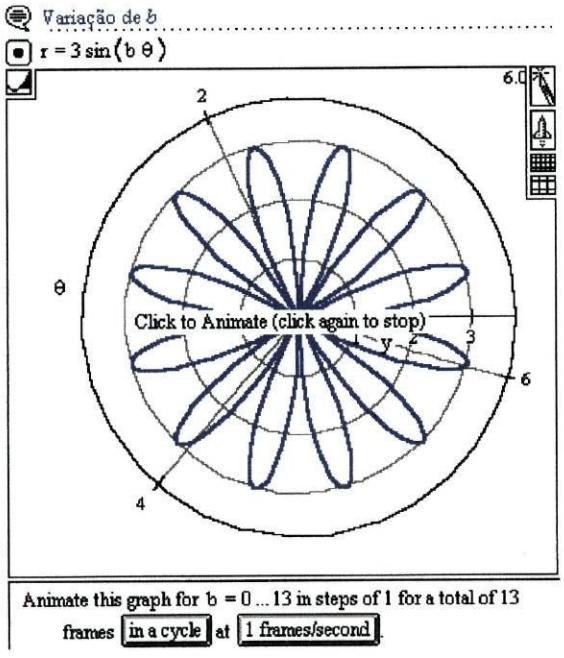
Neste espaço serão apresentadas algumas representações gráficas de funções escritas utilizando coordenadas polares, funções essas que designaremos por funções polares. No Notebook da esquerda são referenciadas as funções polares da forma $r=a^{c}.sin(\ b.\theta\ )$, sendo que as funções do tipo $r=a^{c}.cos(\ b.\theta\ )$ (imagem em baixo, à direita) se apresentam muito semelhantes. Para confirmar tal facto, poderá substituir a expressão "sin" por "cos". Caso não seja visível o gráfico respectivo, deve-se clicar em cima do gráfico com o botão do lado direito do rato e escolher a função ***Clarify*** que aparece disponível.

Experimente alterar os valores de *a*, *b* e *c* e veja as alterações produzidas ao respectivo gráfico.





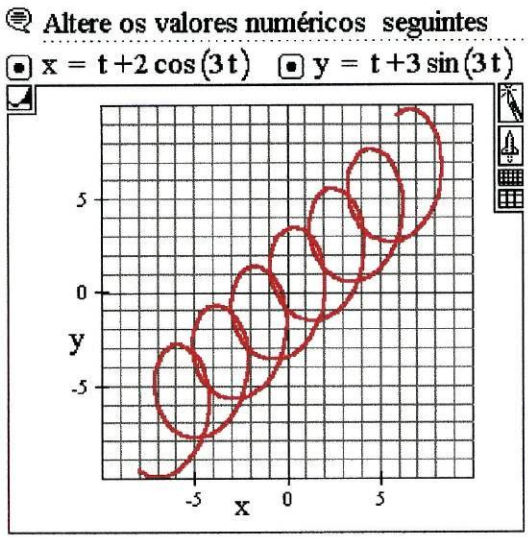
Poderá ainda visualizar uma animação desta situação como a que é apresentada de seguida, onde se faz variar o parâmetro *b*.


Variação de b
r = 3 sin(b θ)
6.0
2
θ
Click to Animate (click again to stop)
4
6
Animate this graph for b = 0 ... 13 in steps of 1 for a total of 13
frames in a cycle at 1 frames/second

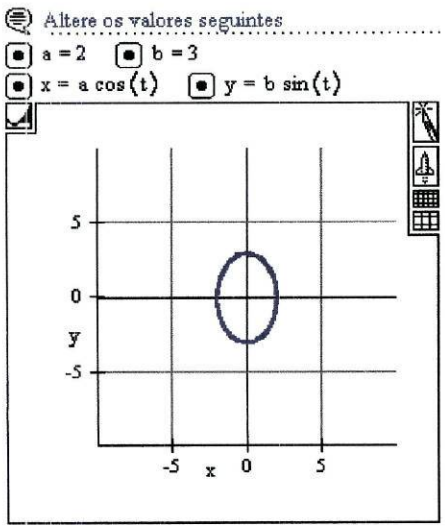
Equações paramétricas no plano

Neste espaço serão visualizadas algumas representações gráficas de funções escritas através da utilização de equações paramétricas. Em cada um dos exemplos, altere as respectivas equações e visualize as respectivas alterações.

O primeiro Notebook diz respeito a uma curva simples cuja expressão analítica aparece escrita em equações paramétricas. Sugere-se que altere as expressões que a constituem para serem visualizados os efeitos provocados.



De seguida apresenta-se uma elipse. O aluno deverá alterar os valores de *a* e de *b* que são apresentados por defeito e, através da visualização imediata dos resultados provocados por essa alteração, deverá conjecturar sobre a influência dos valores em causa no respectivo gráfico.

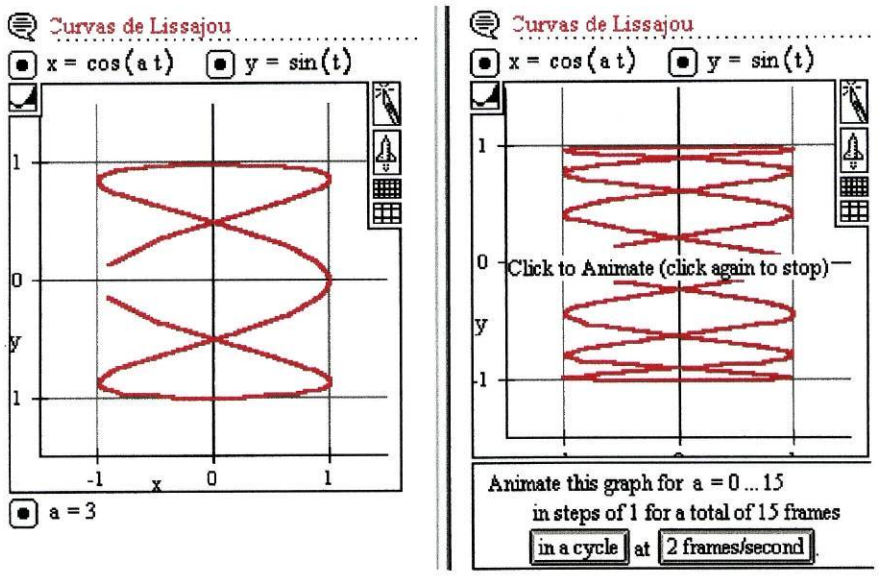


Experimente fazer as seguintes substituições:

- x=2.cos(t) e y=2.sin(t)
- x=5.cos(t) e y=3.sin(t)
- x=2.sec(t) e y=3.tan(t)

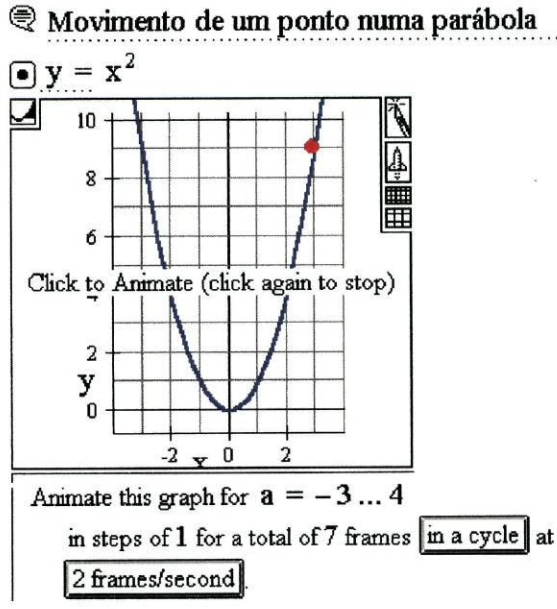
Neste espaço, são apresentadas as Curvas de Lissajou. Na figura da esquerda, poderá alterar o valor de *a* que é apresentado e verificar, no respectivo gráfico, as alterações produzidas.

Por sua vez, no gráfico da direita, poderá provocar uma animação, clicando sobre uma zona qualquer do gráfico. O valor de *a* é alterado automaticamente, podendo ser acompanhada a sua evolução no canto superior direito da respectiva imagem.

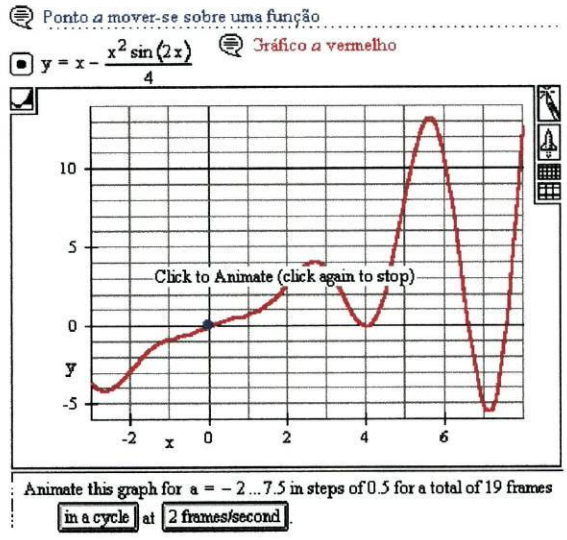


Movimento de um ponto ao longo de um gráfico

Neste primeiro Notebook é visível um ponto percorrer o gráfico de uma função quadrática. Poderá alterar a expressão analítica da função e visualizar a mesma situação.



No Notebook seguinte, a função sobre a qual o ponto "circula" é bem mais complexa que no caso anterior. Poderá também alterar a expressão analítica da expressão apresentada.

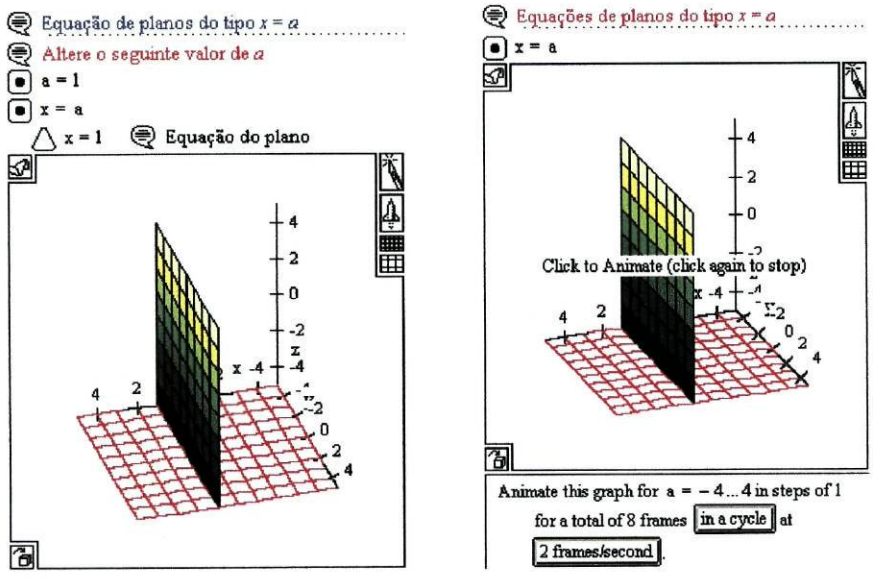


Geometria no espaço

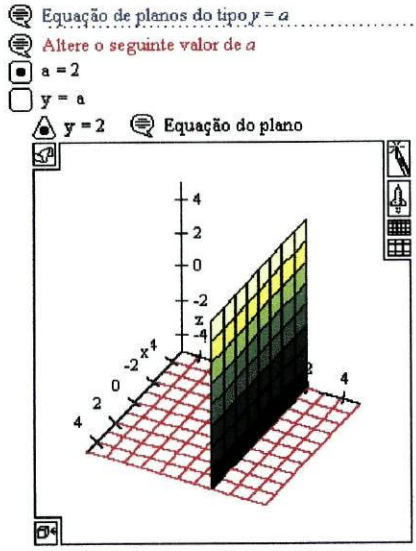
Nos programas do Ensino Secundário, já desde o 10º ano de escolaridade, o estudo da Geometria no Plano e no Espaço assume uma grande importância. No entanto, os alunos demonstram, geralmente, algumas dificuldades em conseguirem visualizar, no espaço, algumas propriedades geométricas. Os Notebooks seguintes, poderão servir como uma forma de abordar, no espaço, os planos perpendiculares aos eixos coordenados.

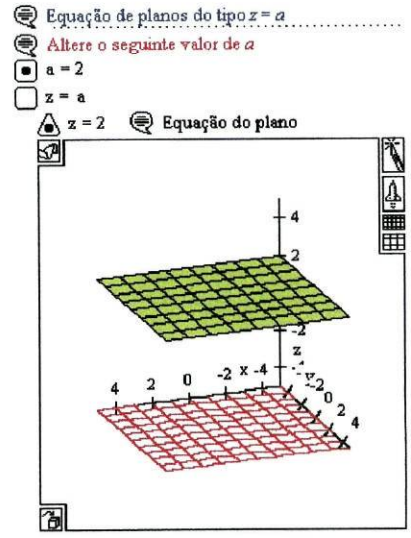
Planos do tipo *x=a*, *y=a* e *z=a*

No Notebook seguinte, pretende-se que o aluno estude o que acontece com os planos do tipo $x = a$. Para tal, deverá alterar o valor de a e ver, no gráfico, as alterações produzidas. Como se pode verificar, os planos do tipo $x = a$ são paralelos ao plano zOy. Poderá ainda visualizar uma animação desse Notebook (representada lateralmente), que, automaticamente, mostra a mesma situação.



Pode ainda ver o que acontece com os planos do tipo $y = a$. Conforme se pode verificar, os planos deste tipo são paralelos ao plano xOz. Analogamente ao que acontece com os outros dois casos, os planos do tipo $z = a$ são paralelos ao plano xOy. Os Notebooks seguintes retratam essas duas situações.


Equação de planos do tipo y = a
Altere o seguinte valor de a
a = 2
y = a
y = 2
Equação do plano
4
2
0
-2
-4
z
x
-2
0
2
4


Equação de planos do tipo z = a
Altere o seguinte valor de a
a = 2
z = a
z = 2
Equação do plano
4
2
-2
z
4
2
0
-2
x
-4
-2
0
2
4

Sólidos de revolução

Neste espaço serão apresentados alguns sólidos de revolução, bem como, em alguns dos exemplos, a forma como são obtidos.

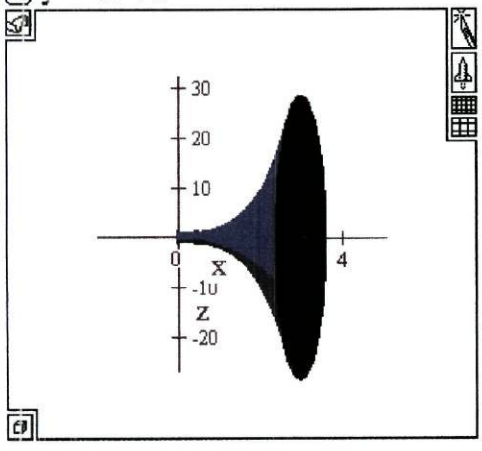
Neste exemplo, apresenta-se um cone (sólido de revolução) em que poderão ser alterados os valores (do eixo dos xx's) entre os quais está representado. Experimente alterar os valores de *a* e de *b* e visualize as alterações produzidas no sólido.

Experimente também alterar o valor da expressão que aparece no Notebook e visualize também as respectivas alterações no gráfico.

Altere os valores de "*a*" e de "*b*"

a = 0 b = 3

$y = x^3+1$



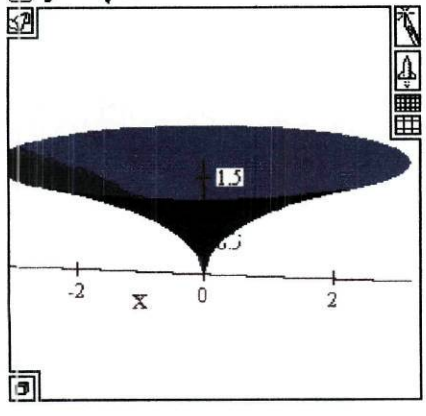
Experimente substituir a expressão que é apresentada pelas seguintes expressões:

- $y=e^x$
- $y=x^3-4.x$
- $y=x^2$

Neste exemplo, apresenta-se um sólido de revolução sobre o eixo dos yy's.

Sólido de revolução

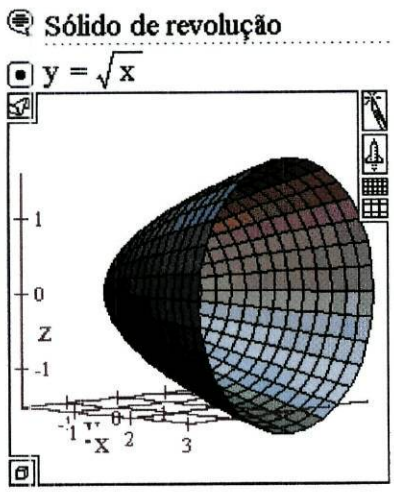
$y = \sqrt{x}$



Experimente substituir a função que é apresentada pelas seguintes funções:

- *y=sin(x)*
- *y=cos(x)*
- *y=ln(x)*
- *y=1/x*

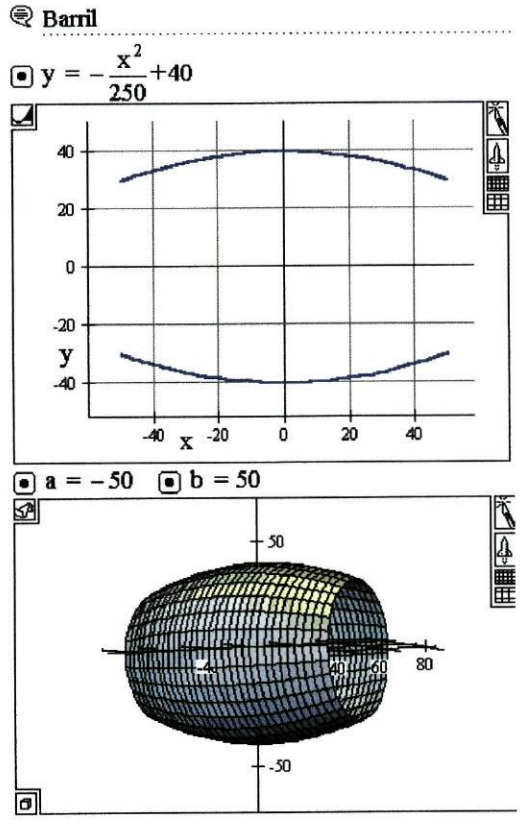
Neste exemplo, apresenta-se um outro sólido de revolução sobre o eixo dos xx's.



Experimente substituir a expressão que é apresentada pelas seguintes expressões:

- *y=sin(x)*
- *y=cos(x)*
- *y=ln(x)*
- *y=1/x*
- *y=1*

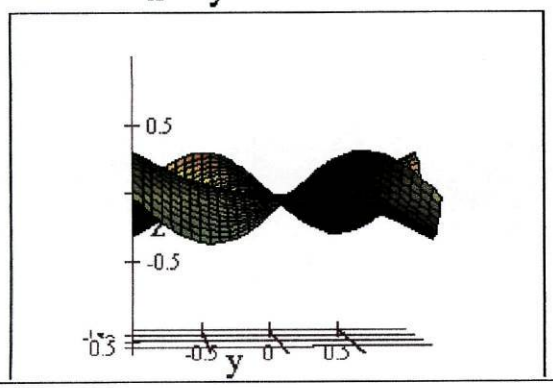
No exemplo seguinte, apresenta-se um sólido de revolução que representa um barril. A primeira imagem diz respeito à função que, ao rodar sobre uma recta horizontal, dá origem ao respectivo barril. Experimente alterar o valor dos limites *a* e *b* entre os quais o barril é representado.



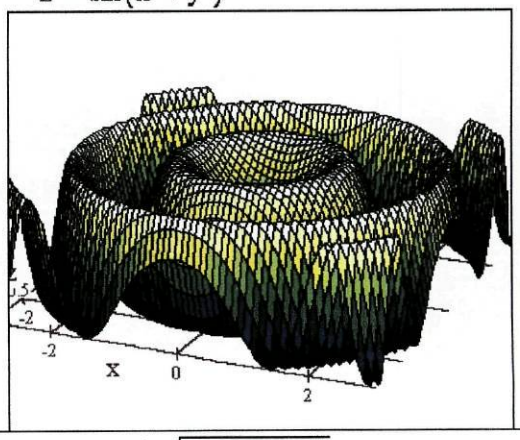
Superfícies

Este espaço é dedicado à apresentação de alguns exemplos de superfícies criadas a partir de Notebooks do MathView. Para cada um dos exemplos, altere a expressão que está envolvida e visualize as alterações provocadas.

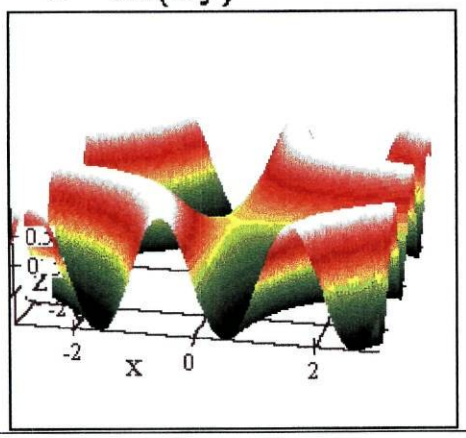
$$z = \frac{x\,y\left(x^2 - y^2\right)}{x^2 + y^2}$$



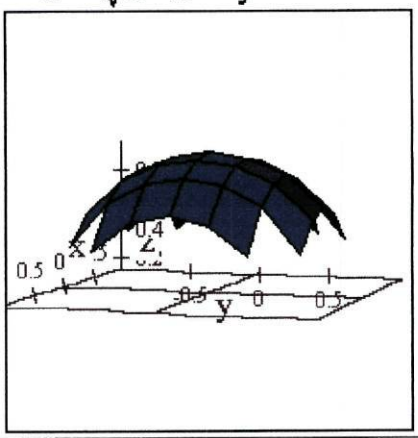
$$z = \sin\left(x^2 + y^2\right)$$



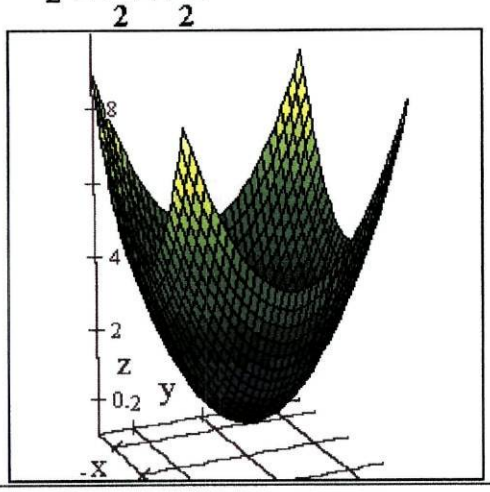
$$z = \sin(x\,y)$$



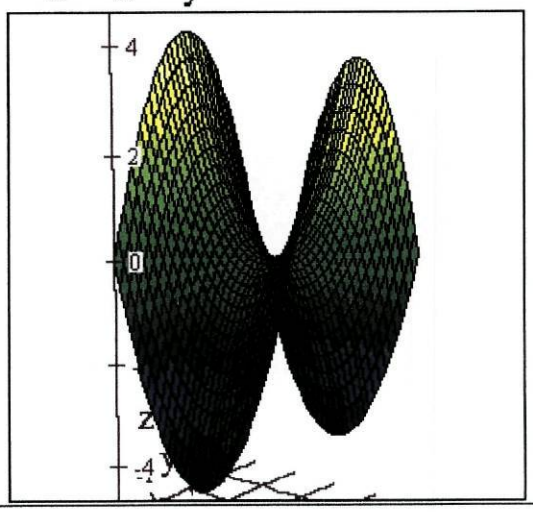
$$z = \sqrt{1 - x^2 - y^2}$$



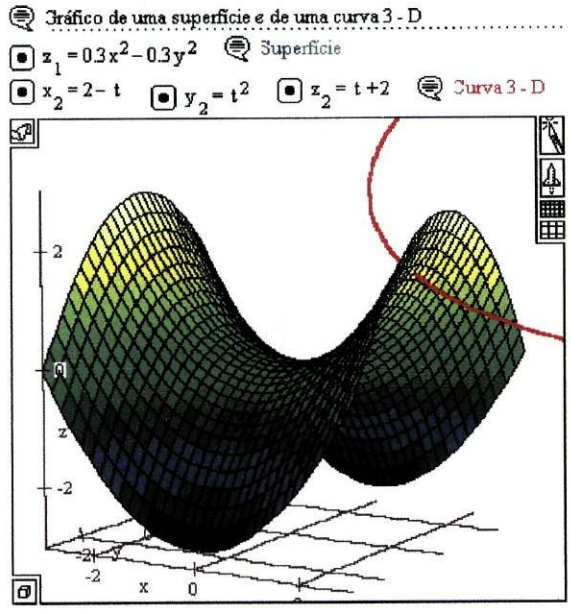
$$z = \frac{x^2}{2} + \frac{y^2}{2}$$



$$z = x^2 - y^2$$



Um último exemplo para dar conta da possibilidade de esboçar, no mesmo sistema de eixos em que foi esboçada a superfície, um gráfico 3-D.

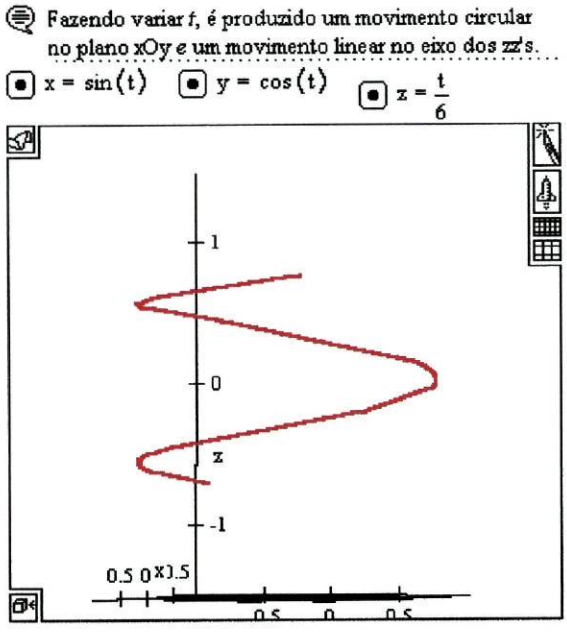


Equações paramétricas no espaço

Neste espaço serão visualizadas algumas representações gráficas de expressões escritas através da utilização de equações paramétricas. Em cada um dos exemplos, altere as respectivas expressões e visualize as respectivas alterações.

O primeiro exemplo diz respeito a uma espiral. Posicione o cursor imediatamente antes de *t* na expressão referente a *z* e digite um número à sua escolha. Veja as alterações produzidas. Experimente também alterar, na mesma expressão, o valor do denominador. Seleccione o valor que aparece por defeito ("6") com um clique sobre ele e introduza outro valor. Faça o mesmo para as outras expressões, ou seja, altere-as.

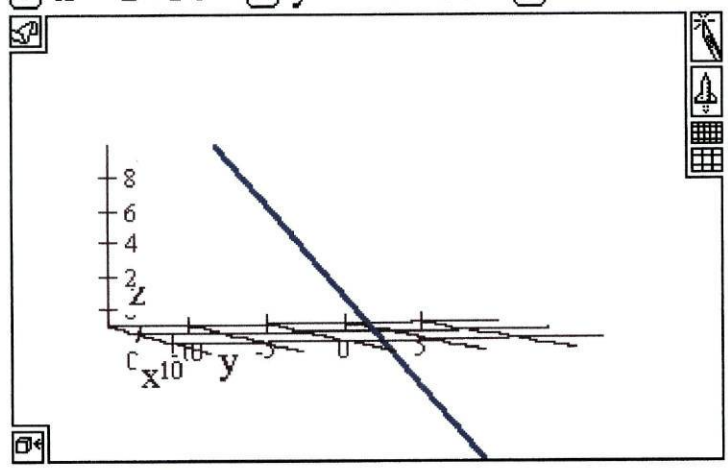
Para uma melhor percepção das alterações produzidas, posicione o cursor do rato sobre o gráfico, clique no botão esquerdo do rato e, sem largar, mude-o de posição.



Nos exemplos seguintes, é apresentado o esboço de uma recta, no espaço, cuja expressão está escrita através de equações paramétricas. Altere os valores em qualquer uma das expressões e visualize as alterações produzidas.

Equações paramétricas (gráfico 3 - D)

$x = 2+3t$ $y = -3+2t$ $z = 1-3t$

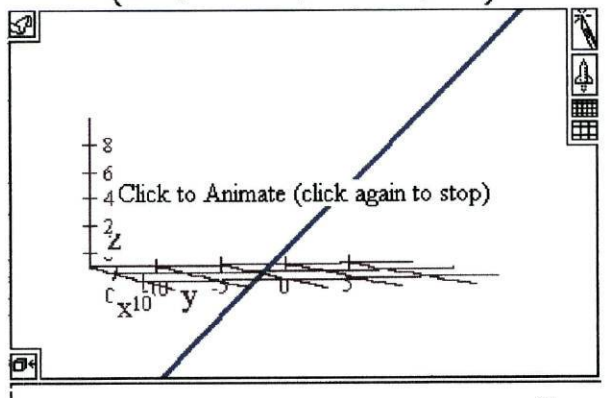


Seguidamente apresenta-se uma animação desta situação que resulta da variação do valor do ângulo θ. Experimente também alterar os valores que constam em cada uma das expressões, bem como da matriz multiplicativa e visualize as alterações produzidas.

Equações paramétricas (gráfico 3 - D)

$x = 2+3t$ $y = -3+2t$ $z = 1-3t$

$$A = \begin{pmatrix} \cos[\theta] & \sin[\theta] & 0 \\ \sin[\theta] & -\cos[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Animate this graph for $\theta = 0 \ldots 2\pi$ in steps of $\frac{\pi}{5}$

for a total of 10 frames in a cycle at 2 frames/second

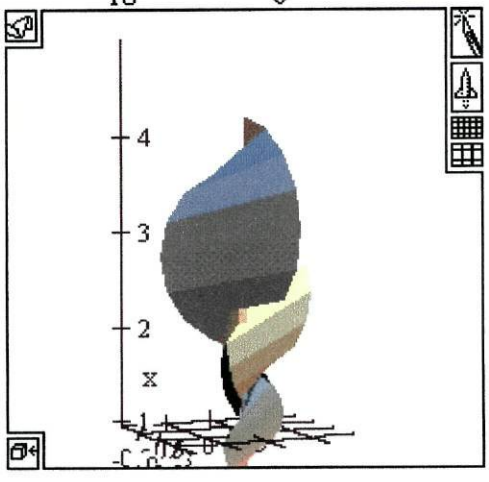
No Notebook seguinte, a expressão do gráfico representado contém dois parâmetros. Altere as expressões que a constituem e veja os resultados produzidos.

Gráficos 3 - D com dois parâmetros

$$x = \frac{t}{18}\sin(s)\cos(t)$$

$$y = \frac{t}{18}\sin(s)\sin(t)$$

$$z = \frac{t}{18}\cos(s)+\frac{t}{6}$$



Capítulo V – Alternativas ao MathView

Neste capítulo estarão alojados vários outros recursos à disposição de quem utiliza a Internet, para ensinar ou aprender Matemática. Aqui, tentou-se prescindir da utilização do *plugin* do MathView, ou seja, apenas usar código HTML, linguagem Java e JavaScript.

Preliminares

Para cada um dos exemplos criados, o utilizador poderá, através do documento que contém determinada *applet*, ter acesso ao código que permite apresentar os respectivos resultados. Para tal, poderá pressionar no botão direito do rato e escolher a função "Ver Html". Ao escolher tal função, o utilizador tem então a disponibilidade de ver o código respectivo e, se pretender, utilizá-lo para a concepção das suas próprias páginas.

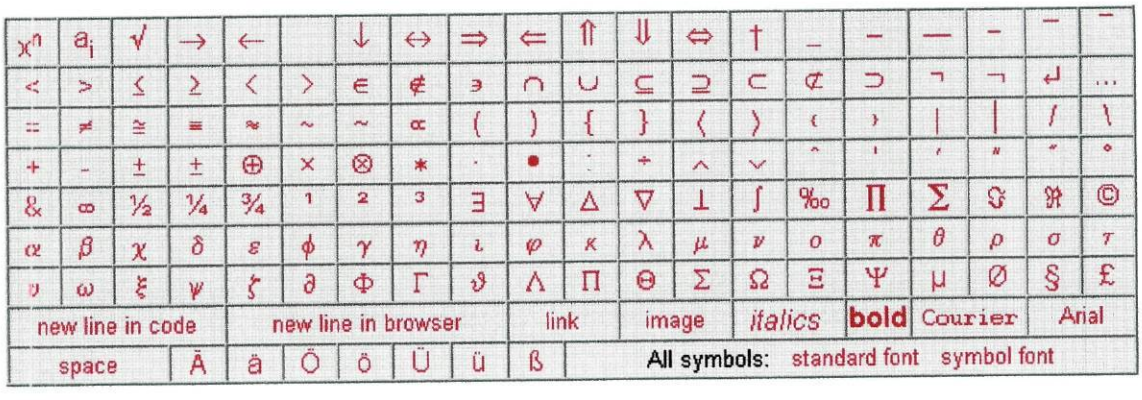
De salientar ainda que cada *applet* tem a sua sintaxe própria, relativamente à especificação de determinadas características. Para além das características comuns a todas as *applets*, como seja a forma de as invocar num documento, existem determinadas instruções específicas em cada uma delas, sem o conhecimento das quais se torna bastante difícil a obtenção de bons resultados interactivos. Para cada uma delas, optou-se por uma explicação das instruções mais básicas e, como tal, se pretender obter mais informações, poderá recorrer ao site disponível na Internet para cada uma delas.

Simbologia Matemática

Neste espaço, apresentam-se algumas formas de se conseguirem as instruções necessárias à obtenção de alguns símbolos tipicamente matemáticos e expressões escritas em linguagem matemática, na elaboração de páginas WWW.

Caracteres especiais

O primeiro exemplo, diz respeito a simbologia matemática. Obtém-se determinado símbolo clicando-se sobre ele, ficando o seu código disponível na caixa superior esquerda.



Visualizador de LaTex

A HotEqn é uma *applet* em linguagem Java, cuja utilização por parte de quem trabalha com Matemática, poderá trazer importantes soluções em termos de visualização de expressões que envolvem simbologia tipicamente matemática.

Esta *applet* usa a notação LaTex na codificação das expressões. Quando uma página, em que está envolvida esta *applet*, é carregada, a *string* introduzida na área de texto é interpretada e as fontes envolvidas são carregadas do servidor onde a *applet* se encontra armazenada. Claro que poderemos fazer o download desta *applet*, a título gratuito, no site http://www.esr.ruhr-uni-bochum.de/VCLab/software/HotEqn/HotEqnDevelopersEdge.html, por forma a

elaborarmos os nossos próprios documentos offline. Depois de feito isso, com a *applet* alojada na máquina do utilizador, este poderá criar os seus próprios exemplos. E foi isso que também fizemos neste trabalho. Utilizámos esta *applet* para levar a cabo determinadas alterações de aspecto ao texto que queríamos introduzir nas nossas páginas.

A partir da versão 2.01 desta *applet*, surgiram duas subclasses (as *applets* dHotEqn e mHotEqn) que trouxeram melhores desempenhos e outras potencialidades à *applet* inicial, nomeadamente no que concerne a animações. De facto, a *applet* sem estas duas extensões, não incluía métodos de edição de expressões, substituição de variáveis, nem outros elementos mais sofisticados sob o ponto de vista interactivo. Quanto à dHotEqn, apesar de ter as mesmas funcionalidades do que a *applet* base, acrescenta aspectos que dizem respeito a manipulação de equações, substituição dinâmica de variáveis em expressões, inclusivamente oriundas de partes da *applet* presentes na mesma página, através de cliques no rato. Relativamente à mHotEqn, trata-se de uma extensão da anterior que acrescenta o facto de se poder interagir com o MatLab.

De seguida, serão apresentados alguns exemplos da utilização desta *applet*, nos primeiros dos quais se apresentará a sintaxe necessária à obtenção dos resultados que são visualizados no browser.

Suponhamos, por exemplo, que queremos introduzir no nosso documento a "raíz quadrada de 125". Recorrendo simplesmente a código HTML, o melhor que poderíamos obter era a expressão √(125), o que, manifestamente, não satisfaz as necessidades de rigor inerentes à linguagem matemática. Ora, o recurso à *applet* que estamos a considerar poderá ser uma boa alternativa. O resultado obtido será $\sqrt{125}$, o que constitui, de facto, uma boa solução.

Relativamente ao fundo amarelo que aparece a envolver a nossa expressão, poderá ser alterada tal cor. Evidentemente que, na generalidade dos casos, deverá ter interesse que a cor do fundo da expressão seja da mesma cor do documento que está a ser lido, de modo a que não sejam perceptíveis

quaisquer diferenças de aspecto. Contudo, para que se perceba onde estão os elementos, neste presente documento, que foram introduzidos através da utilização da *applet* agora em análise, optou-se pelo uso de outra cor de fundo para a *string*, diferente daquela que está a ser usada como fundo do documento.

No que concerne à sintaxe necessária à obtenção deste aspecto para o exemplo que estamos a considerar, antes de mais nada refira-se que estamos perante uma *applet*, ou seja, o código será limitado pelas *tags* <*applet*> e </*applet*>. O código que foi necessário apresenta-se de seguida:

```
<APPLET CODE="HotEqn.class" WIDTH="50" HEIGHT="22" ARCHIVE="HotEqn.zip"
NAME="eqn1"
        ALIGN="top">
  <PARAM NAME="equation" VALUE=" \sqrt{125}"
  <PARAM NAME="Fontsize1" VALUE="18">
  <PARAM NAME="Fontsize2" VALUE="14">
  <PARAM NAME="Fontsize3" VALUE="12">
  <PARAM NAME="Fontsize4" VALUE="8">
  <PARAM NAME="bgcolor" VALUE="ffff00">
  <PARAM NAME="fgcolor" VALUE="000000">
  <PARAM NAME="halign" VALUE="center">
  <PARAM NAME="valign" VALUE="top">
</APPLET>
```

Da análise do código anterior, destacamos o facto do parâmetro CODE assumir o valor de "HotEqn.class" e o ARCHIVE o valor "HotEqn.zip". De salientar ainda que a parte que aparece a vermelho ("\sqrt{125}") diz respeito à sintaxe da expressão matemática que pretendemos introduzir na página HTML, neste caso, "raíz quadrada de 125".

De seguida apresentam-se mais três exemplos de expressões que poderão ser introduzidas em documentos HTML através da utilização desta *applet*.

Exemplo 1:

$$\sqrt{12} = \sqrt{4 \times 3} = \sqrt{2^2 \times 3} = 2\sqrt{3}$$

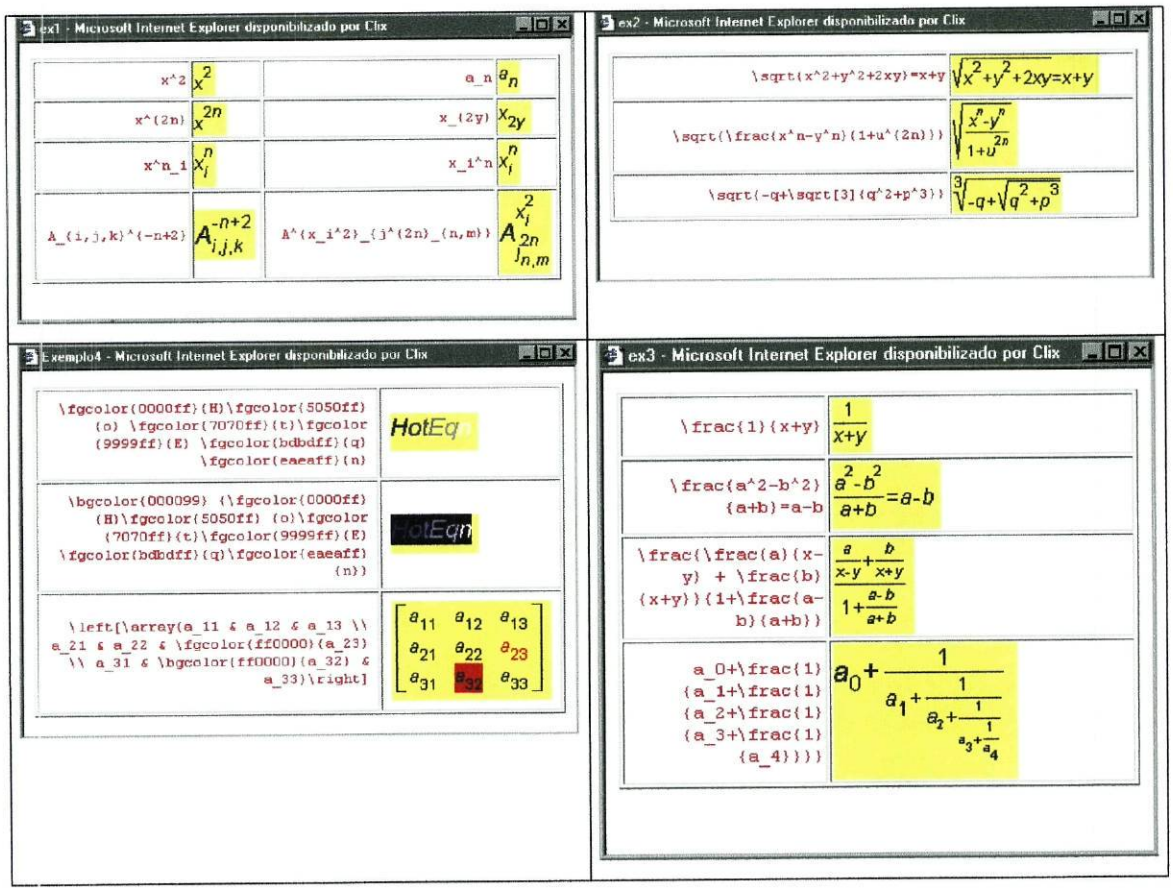
Exemplo 2:

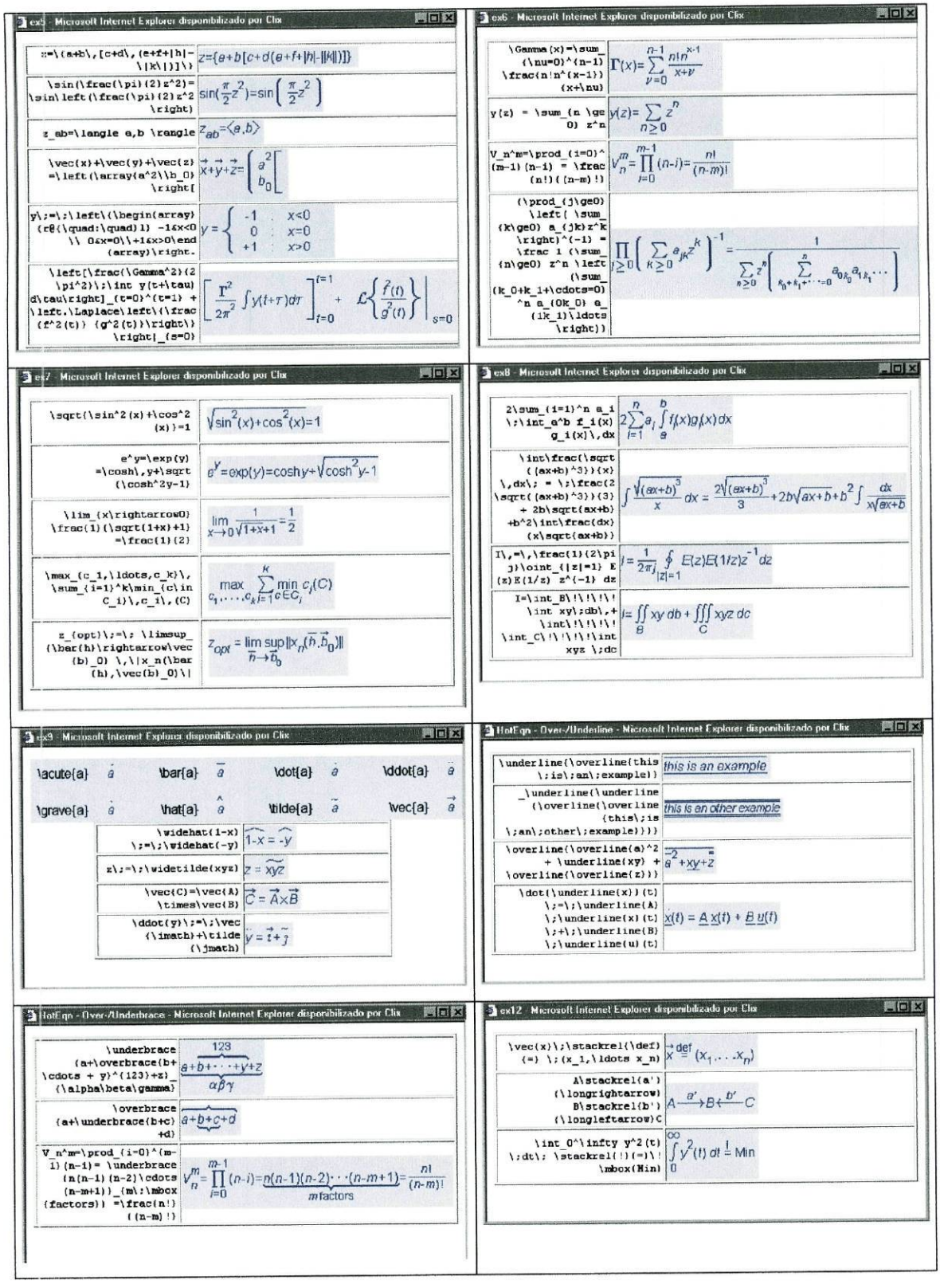
$$\sqrt{50} = \sqrt{25 \times 2} = \sqrt{5^2 \times 2} = 5\sqrt{2}$$

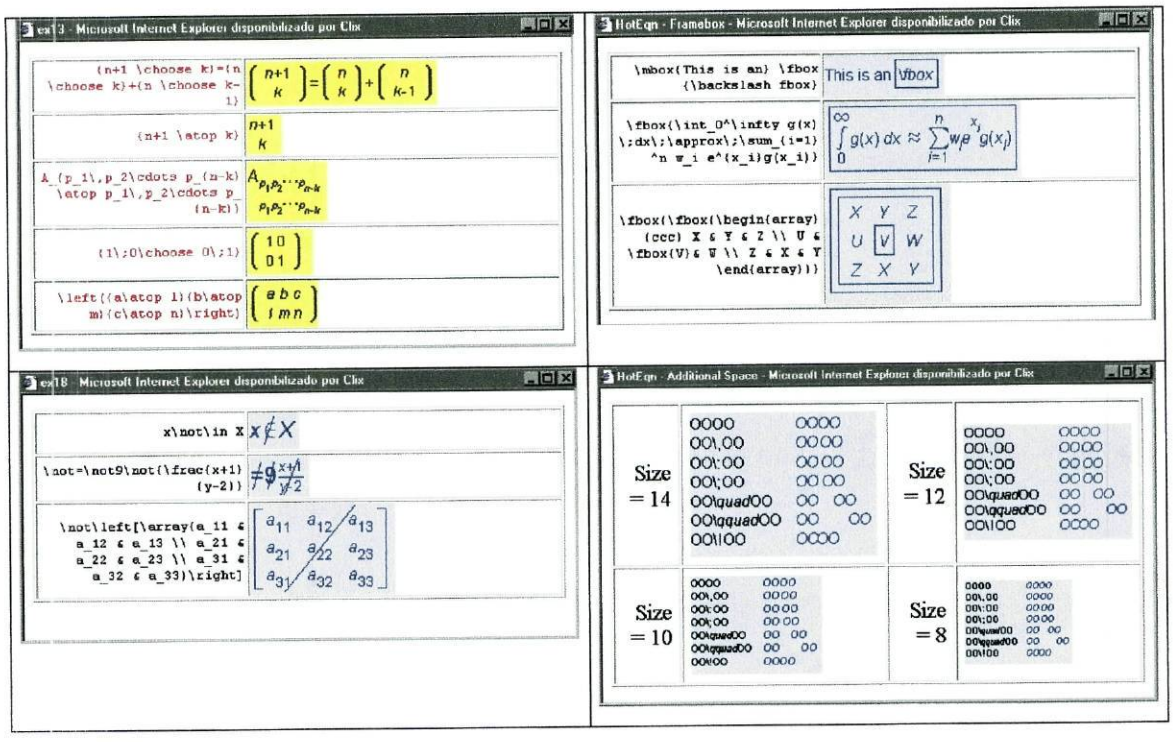
Exemplo 3:

$$3.5*x+\frac{5}{x+y}+4$$

De seguida, apresentam-se alguns exemplos do que poderá ser feito em termos de escrita em linguagem matemática, através da utilização desta *applet*. Na coluna da esquerda de cada um dos exemplos, é visível a instrução específica para obtermos o resultado visível na coluna da direita de cada um dos exemplos.




ex5 - Microsoft Internet Explorer disponibilizado por Clix
z=\{a+b\,[c+d\,(e+f+|h|-\|k\|)]\}
z={a+b[c+d(e+f+|h|-||k||)]}
\sin(\frac(\pi)(2)z^2)=\sin\left(\frac(\pi)(2)z^2\right)
z_ab=\langle a,b \rangle
z_ab=⟨a,b⟩
\vec(x)+\vec(y)+\vec(z)=\left(\array(a^2\\b_0)\right[
x+y+z=
y\;=\;\left\(\begin(array)(r@{\quad:\quad}l) -1&x<0\\ 0&x=0\\+1&x>0\end(array)\right.
y = -1 : x<0; 0 : x=0; +1 : x>0
\left[\frac(\Gamma^2)(2\pi^2)\;\int y(t+\tau)d\tau\right]_(t=0)^(t=1) + \left.\Laplace\left\(\frac(f^2(t)) (g^2(t))\right\)\right|_(s=0)
ex6 - Microsoft Internet Explorer disponibilizado por Clix
\Gamma(x)=\sum_(\nu=0)^(n-1)\frac(n!n^(x-1))(x+\nu)
Γ(x)=Σ_{ν=0}^{n-1} n!n^{x-1}/(x+ν)
y(z) = \sum_(n \ge 0) z^n
y(z)=Σ_{n≥0} z^n
V_n^m=\prod_(i=0)^(m-1)(n-i) = \frac(n!)((n-m)!)
V_n^m=Π_{i=0}^{m-1}(n-i)=n!/(n-m)!
(\prod_(j\ge0)\left(\sum_(k\ge0) a_(jk)z^k\right)^(-1) = \frac 1 (\sum_(n\ge0) z^n \left(\sum_(k_0+k_1+\cdots=0)^n a_(0k_0) a_(1k_1)\ldots\right))
ex7 - Microsoft Internet Explorer disponibilizado por Clix
\sqrt(\sin^2(x)+\cos^2(x))=1
√(sin²(x)+cos²(x))=1
e^y=\exp(y)=\cosh\,y+\sqrt(\cosh^2y-1)
e^y=exp(y)=coshy+√(cosh²y-1)
\lim_(x\rightarrow0)\frac(1)(\sqrt(1+x)+1)=\frac(1)(2)
lim_{x→0} 1/(√(1+x)+1)=1/2
\max_(c_1,\ldots,c_k)\,\sum_(i=1)^k\min_(c\in C_i)\,c_i\,(C)
max_{c_1,...,c_k} Σ_{i=1}^{k} min_{c∈C_i} c_i(C)
z_(opt)\;=\; \limsup_(\bar(h)\rightarrow\vec(b)_0) \,\|x_n(\bar(h),\vec(b)_0)\|
z_opt = lim sup_{h→b_0} ||x_n(h,b_0)||
ex8 - Microsoft Internet Explorer disponibilizado por Clix
2\sum_(i=1)^n a_i\;\int_a^b f_i(x)g_i(x)\,dx
2Σ_{i=1}^{n} a_i ∫_a^b f_i(x)g_i(x)dx
\int\frac(\sqrt((ax+b)^3))(x)\,dx\; = \;\frac(2\sqrt((ax+b)^3))(3) + 2b\sqrt(ax+b) +b^2\int\frac(dx)(x\sqrt(ax+b))
∫√((ax+b)³)/x dx = 2√((ax+b)³)/3 + 2b√(ax+b) + b² ∫ dx/(x√(ax+b))
I\,=\,\frac(1)(2\pi j)\oint_(|z|=1) E(z)E(1/z) z^(-1) dz
I = 1/(2πj) ∮_{|z|=1} E(z)E(1/z)z^{-1} dz
I=\int_B\!\!\!\!\int xy\;db\,+\int\!\!\!\!\int_C\!\!\!\!\int xyz \;dc
I= ∬_B xy db + ∭_C xyz dc
ex9 - Microsoft Internet Explorer disponibilizado por Clix
\acute{a} \bar{a} \dot{a} \ddot{a}
\grave{a} \hat{a} \tilde{a} \vec{a}
\widehat(1-x)\;=\;\widehat(-y)
1-x = -y
z\;=\;\widetilde(xyz)
z = xyz
\vec(C)=\vec(A)\times\vec(B)
C = A×B
\ddot(y)\;=\;\vec(\imath)+\tilde(\jmath)
y = i + j
HotEqn - Over-/Underline - Microsoft Internet Explorer disponibilizado por Clix
\underline(\overline(this\;is\;an\;example))
this is an example
_\underline(\underline(\overline(\overline(this\;is\;an\;other\;example)))
this is an other example
\overline(\overline(a)^2 + \underline(xy) + \overline(\overline(z)))
a²+xy+z
\dot(\underline(x))(t)\;=\;\underline(A)\;\underline(x)(t)\;+\;\underline(B)\;\underline(u)(t)
x(t) = A x(t) + B u(t)
HotEqn - Over-/Underbrace - Microsoft Internet Explorer disponibilizado por Clix
\underbrace(a+\overbrace(b+\cdots + y)^(123)+z)_(\alpha\beta\gamma)
a+b+···+y+z
123
αβγ
\overbrace(a+\underbrace(b+c)+d)
a+b+c+d
V_n^m=\prod_(i=0)^(m-1)(n-i)= \underbrace(n(n-1)(n-2)\cdots(n-m+1))_(m\;\mbox(factors)) =\frac(n!)((n-m)!)
V_n^m=Π_{i=0}^{m-1}(n-i)=n(n-1)(n-2)···(n-m+1)=n!/(n-m)!
m factors
ex12 - Microsoft Internet Explorer disponibilizado por Clix
\vec(x)\;\stackrel(\def)(=) \;(x_1,\ldots x_n)
x ≝ (x_1,...x_n)
A\stackrel(a')(\longrightarrow) B\stackrel(b')(\longleftarrow)C
A —a'→ B ←b'— C
\int_0^\infty y^2(t)\;dt\; \stackrel(!)(=)\!\mbox(Min)
∫_0^∞ y²(t) dt ≝ Min


ex13 - Microsoft Internet Explorer disponibilizado por Clix
{n+1 \choose k}={n \choose k}+{n \choose k-1}
{n+1 \atop k}
A_{p_1\,,p_2\cdots p_{n-k} \atop p_1\,,p_2\cdots p_{n-k}}
{1\;0\choose 0\;1}
\left({a\atop l}{b\atop m}{c\atop n}\right)
HotEqn - Framebox - Microsoft Internet Explorer disponibilizado por Clix
\mbox{This is an} \fbox{\backslash fbox}
This is an \fbox
\fbox{\int_0^\infty g(x) \;dx\;\approx\;\sum_{i=1}^n w_i e^{x_i}g(x_i)}
\fbox{\fbox{\begin{array}{ccc} X & Y & Z \\ U & V & W \\ Z & X & Y \end{array}}}
ex18 - Microsoft Internet Explorer disponibilizado por Clix
x\not\in X
\not=\not9\not{\frac{x+1}{y-2}}
\not\left[\array{a_11 & a_12 & a_13 \\ a_21 & a_22 & a_23 \\ a_31 & a_32 & a_33}\right]
HotEqn - Additional Space - Microsoft Internet Explorer disponibilizado por Clix
Size = 14
Size = 12
Size = 10
Size = 8

Cálculos elementares

Nesta secção vamos utilizar somente código HTML e JavaScript para a criação de exemplos que poderão substituir o MathView nos cálculos elementares e noutras tarefas básicas.

Calculadora simples

O primeiro exemplo que se apresenta, é o de uma calculadora simples, em que os valores a introduzir poderão ser obtidos através de botões, criados em HTML. Refira-se que para a criação de uma máquina de calcular deste tipo, poderemos também optar pela elaboração de um "image map" (tratados no Capítulo II), em que, em vez de botões, os números e operações seriam obtidos pelo pressionar de uma zona da tela que seria um link para determinada operação. Veja, por exemplo as calculadoras que são apresentadas no endereço http://www.calculator.com.

A calculadora que aqui se apresenta, foi criada usando apenas código HTML e JavaScript. Trata-se de uma calculadora que apenas contém as operações mais básicas, mas outras operações poderiam ser introduzidas, de modo a que, por exemplo, pudéssemos criar uma calculadora científica. Ora, a criação da nossa própria calculadora que responda às nossas reais necessidades operativas, está ligada, de certa forma, ao ano lectivo que estamos a leccionar. Desde os alunos mais novos até aos mais velhos, o uso de calculadoras é aconselhado pelos programas oficiais e, a sua construção, poderá ser uma boa forma de se perceber o seu funcionamento em termos operativos.

Uma última nota para referir que, no nosso exemplo, os valores deverão ser introduzidos através da utilização dos botões que são apresentados, pressionando com o ponteiro do rato sobre eles e, por outro lado, que a função sqrt ("square root") representa a raíz quadrada, apesar do "input" não fazer aparecer o símbolo respectivo.

Ao pressionar sobre a imagem seguinte, terá acesso à introdução de dados na calculadora que é apresentada.



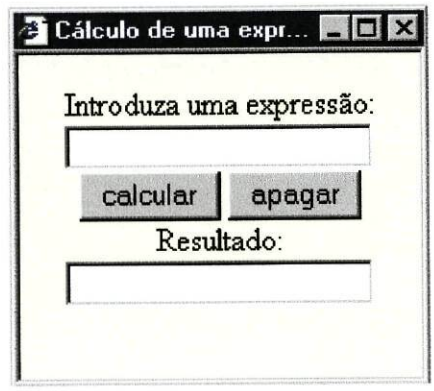
Cálculo de expressões

Neste exemplo, o aluno introduz o valor da expressão que pretende calcular, de uma forma diferente daquela que é utilizada numa calculadora, como a do exemplo anterior. Enquanto que no exemplo anterior o aluno tinha que pressionar sobre os botões respectivos para proceder à introdução dos dados, neste exemplo, o aluno terá que posicionar o cursor na caixa referente à introdução de dados e, usando o teclado, deverá introduzir a expressão a calcular.

De salientar que, neste caso, não se torna necessário definir, à partida, todas as funções associadas às operações aritméticas. Foi definida, em JavaScript, uma função (calcula()) que, através da utilização da função **eval** devolve, na caixa inferior, o resultado da expressão introduzida na caixa superior. A sintaxe para a definição de tal função é a seguinte:

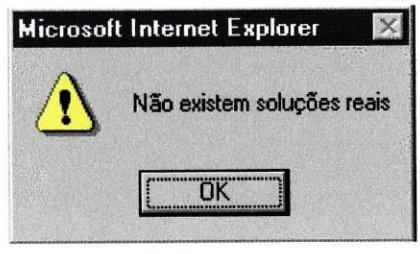
```
function calcula(valor)
{
valor.resultado.value=eval(valor.expressao.value)
}
```

Ao pressionar sobre a imagem seguinte, terá acesso à introdução de dados nas caixas disponibilizadas na figura que é apresentada.



Raízes de equações do 2º grau

Este exemplo, constitui uma forma de obter as raízes de equações do 2º grau, com a precisão que a linguagem JavaScript permite (10 casas decimais). Ao contrário do que acontecia quando se utilizou um Notebook do MathView para resolver este tipo de equações, em que não se podia evitar que aparecessem as soluções quando estas eram complexas, desta vez, temos uma forma de, quando não existirem soluções reais, fazer aparecer uma janela (através de um alert) alertando para isso mesmo, conforme mostra a imagem seguinte.



A parte da função que foi definida para extrair as soluções de uma equação do segundo grau e que permite o aparecimento de tal janela de alert é a seguinte:

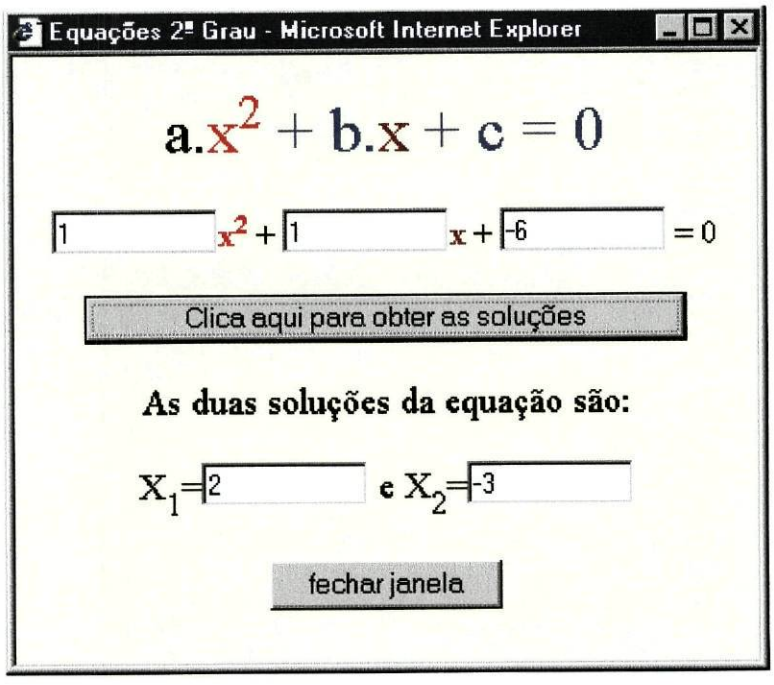
```
function calculate (form){
a=eval(form.a.value);
b=eval(form.b.value);
c=eval(form.c.value);
```

```
if ((Math.pow(b, 2)-4*a*c) <0){
alert ("Não existem soluções reais");
```

onde form.a.value (respectivamente form.b.value e form.c.value) é o coeficiente *a* (respectivamente *b* e *c*) da equação $a.x^2 + b.x + c = 0$

De salientar ainda que, ao contrário do que acontecia no caso anterior em que era introduzida a expressão completa da equação a resolver, desta vez, apenas são introduzidos os coeficientes.

Pressione sobre a imagem seguinte e resolva as suas equações do 2º grau, introduzindo os respectivos coeficientes.



Aritmética de inteiros

Depois de introduzido um número pelo utilizador, este poderá escolher a operação que pretende efectuar, bastando para tal, pressionar sobre o respectivo botão. Pode testar se o número é ou não primo, pode determinar os divisores do número introduzido, ou pode ainda calcular o factorial desse número. De referir apenas que, neste último caso, deverá ter-se em atenção à

precisão utilizada. De facto, os resultados são obtidos correctamente até ao 21!, inclusivé.

A saída dos dados é feita através de uma caixa de "alert", depois de se clicar no botão respectivo.

Clique sobre a imagem seguinte para poder introduzir o valor que pretende saber se é ou não primo, conhecer os seus divisores, ou calcular factorial desse número. Depois de aberta tal janela, poderá visualizar o código que foi utilizado para a obtenção destes resultados, através do já conhecido método do botão direito do rato para accionar a função "Ver HTML".

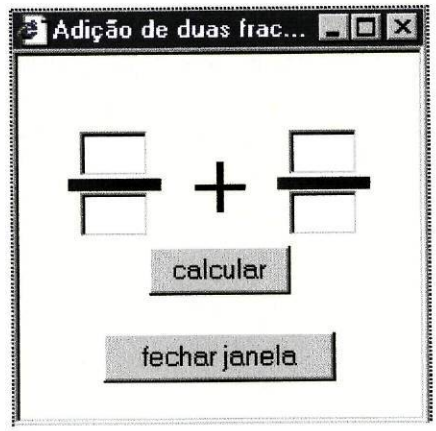


Cálculos com fracções

Com o *script* seguinte, poderemos efectuar adições e subtracções envolvendo números fraccionários. Após a introdução dos numeradores e denominadores, pressiona-se sobre o botão "calcular", para que a respectiva operação seja efectuada, seguida das respectivas simplificações.

Clique sobre a imagem seguinte para poder introduzir as fracções que pretende adicionar. De salientar que a introdução de fracções negativas (e com isso a obtenção de subtracções), pode ser feita introduzindo o respectivo sinal negativo no numerador ou denominador da fracção respectiva.

À semelhança do que foi feito para a Aritmética de inteiros, também aqui os resultados são obtidos numa janela de "alert".



Álgebra simbólica

Nesta secção apresentam-se algumas *applets* capazes de desenvolverem determinadas operações algébricas. Para além de ter sido usada a linguagem HTML e algumas instruções em JavaScript, desta vez, as *applets* usadas foram criadas com a linguagem JAVA. Neste espaço, aproveitam-se essas *applets* disponíveis na Web para a obtenção de determinados resultados.

Hartmath

Trata-se de mais uma *applet* com um bom desempenho em termos de resultados. No entanto, em termos de *output*, ou seja, da forma com que os resultados são apresentados, fica um pouco aquém das expectativas. Por outro lado, a sintaxe necessária à introdução dos dados é um pouco específica, o que, faz com que se deva compreender a *applet* com alguma profundidade.

Esta *applet* também poderá ser obtida gratuitamente (desde que não seja para fins comerciais), no endereço http://www.hartmath.com. O código que de seguida se apresenta, é o código necessário à introdução dessa *applet* numa página Web. Foi usada a versão referente aos mini-computadores PSION, apenas por questões de grafismo.

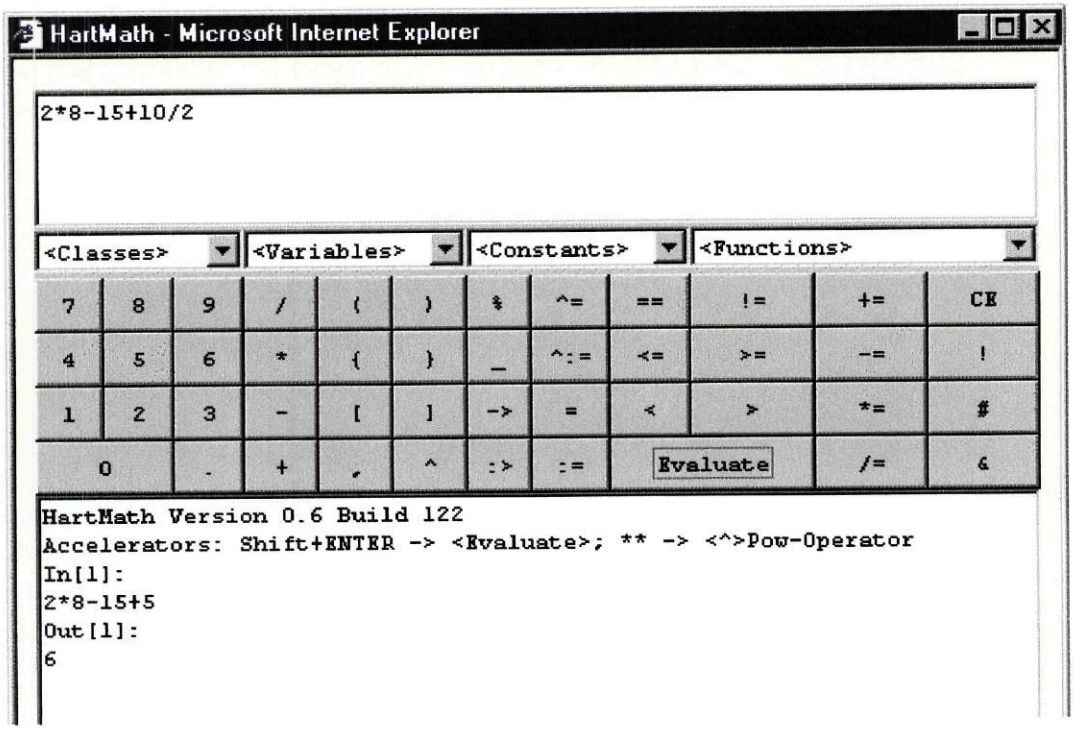
```
<HTML>
<HEAD>
<TITLE>HartMath</TITLE>
</HEAD>
<BODY bgcolor=cornsilk>
<APPLET CODE="HMPsionApplet.class" archive="hartmath.jar" width="500" height="480" >
<PARAM NAME="Width" VALUE="500">
<PARAM NAME="Height" VALUE="480">
</APPLET>
</BODY>
</HTML>
```

Como se pode verificar através da análise do código anterior, o parâmetro CODE invoca o ficheiro "HMPsion*Applet*.class", situado no ARCHIVE "hartmath.jar". Será importante salientar que aquele ficheiro "*.class" refere-se ao sistema PSION mas as potencialidades são as mesmas que a versão

tradicional (para tal deveríamos invocar o ficheiro "HMWeb*Applet*.class" disponível no mesmo ficheiro "hartmath.jar").

A imagem seguinte, mostra o aspecto da *applet*. Na zona branca situada na parte superior dos botões, o utilizador introduz a expressão que deseja simplificar e os resultados aparecerão na parte branca situada na zona inferior aos botões.

Depois de introduzida a expressão, para se determinar o resultado pretendido, deverá pressionar-se na tecla **RETURN** (ENTER da parte não numérica do teclado), tendo a tecla SHIFT pressionada, ou ainda no botão "EVALUATE".



Vejamos agora alguns aspectos referentes à sintaxe própria desta *applet*.

Comentários

Os caracteres // iniciam um comentário. Para finalizar o comentário, utilizam-se os caracteres \n.

Marcas Identificativas

Um identificador é uma sequência de caracteres, em que o primeiro deverá ser uma letra ou, em especial, o carácter **$**. Todos os outros caracteres na sequência podem ser letras ou algarismos.

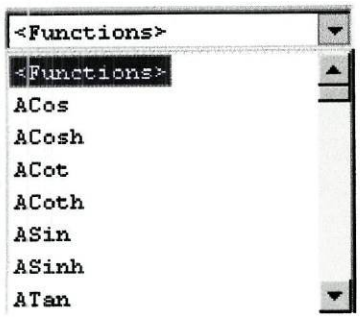
Constantes

As constantes podem ser numéricas (de base dez, dois ou hexadecimal - nos dois últimos casos, a constante deverá ser precedida de, respectivamente, Ob e Ox); constituídas por partes inteiras, decimais ou fraccionárias, carácter "e" ou "E" e expoentes inteiros; ou podem ainda ser expressões iniciadas e finalizadas por aspas duplas.

Símbolos

Existem determinados símbolos que representam constantes (p.e. E, p, True, False), variáveis (por exemplo x, y, z) ou funções (por exemplo f, Sin, Cos, Multitply). Por outro lado, estes símbolos poderão conter determinados atributos (dos quais falaremos mais adiante), como por exemplo Listable, Flat, Orderless, OneIdentity, HoldFirst, HoldRest, HoldAll).
Os símbolos seguidos de "_" (por exemplo x_, y_ e z_), notações já utilizadas no programa Mathematica, têm a mesma finalidade que as *Wildcards* têm no MathView, nomeadamente no que toca à definição de funções.

Funções



A forma de definir funções é a tradicional, por exemplo Sin(0), f(x,y) ou List(). No entanto, existem determinadas funções (que poderão ser acedidas no menu existente na *applet* para o efeito retratado pela imagem ao lado) que poderão igualmente ser invocadas a partir de determinados operadores.

Vejamos o seguinte exemplo. A função AddTo adiciona a um determinado valor um outro valor. Para levar a cabo tal operação, podemos fazê-lo de três seguintes formas, em que é apresentada a sintaxe necessária à obtenção do respectivo resultado:

- a=2;a+=5
- a=2;AddTo(a,5)
- 2+5

Em todos os casos o resultado será 7, uma vez que a 2 estamos a adicionar 5.

De seguida, apresenta-se uma lista de outras funções que poderão ser invocadas a partir de um operador:

Operador	Função
{ }	List (com as chavetas, poderão ser definidos vectores ou matrizes)
[ ]	Part
#	Slot
.	Dot
^ ou **	Pow
/	Divide
*	Multiply
-	Subtract
+	Add
==	Equal
!=	Unequal
<	Less
<=	LessEqual
>	Greater
>=	GreaterEqual
!expressão	Not
&&	And
||	Or
->	Rule
:>	RuleDelayed
/.	ReplaceAll
+=	AddTo

-=	SubtractFrom
*=	TimesBy
/=	DivideBy
&	Lambda
=	Set
:=	SetDelayed
^=	UpSet
^:=	UpSetDelayed
;	Statement

Através do recurso ao operador **%**, podemos obter o resultado do desenvolvimento anterior. Usando n vezes este operador, podemos obter o resultado obtido n-passos antes do actual.

De seguida apresenta-se uma lista de algumas das funções que podem ser acedidas na caixa de selecção cuja imagem aparece anteriormente ou através da digitação da respectiva expressão, no espaço destinado à introdução de dados. Na lista seguinte, o símbolo → separa o input do respectivo output, ou seja, separa a expressão que é preciso introduzir (representada a azul) e o respectivo resultado (representado a vermelho).

ACos(<arg1>) - Arcosseno

ACos(-1) → Pi

Add(<arg1>,...,<argN>) - Adiciona

Add(3,4,5) → 12

Check(<condition>,<expr>) - Verifica condições

f(x_):=Check(x>0,x+1) → Null

f(y_):=Check(y<0,y*3) → Null

f(10) → 11

f(-1) → -3

f(0) → f(0)

D(<expr>,...,<symbol>) - Deriva

D(x^7+4*x,x) → 4+7*x^6

Definition(<symbol>) - retorna as definições associadas

Definition(Sin) → definições associadas à função Sin, nomeadamente:

Definition(Sin)

// *** System Attributes:

Attributes(Listable) // *** System Rules:

Sin(I)=I*Sinh(1)

Sin(Pi)=0

Sin(1/8*Pi)=1/2*Sqrt(2-Sqrt(2))

Sin(3/8*Pi)=1/2*Sqrt(2+Sqrt(2))

Sin(3/10*Pi)=1/4+1/4*Sqrt(5)

Sin(2/5*Pi)=1/4*Sqrt(2)*Sqrt(5+Sqrt(5))

Sin(1/10*Pi)=-1/4+1/4*Sqrt(5)

Sin(1/6*Pi)=1/2

Sin(1/4*Pi)=1/2*Sqrt(2)

Sin(1/3*Pi)=1/2*Sqrt(3)

Sin(1/2*Pi)=1

Sin(5/12*Pi)=1/4*(1+1/3*Sqrt(3))*Sqrt(6)

Sin(1/5*Pi)=1/4*Sqrt(2)*Sqrt(5-Sqrt(5))

Sin(1/12*Pi)=1/4*(1-1/3*Sqrt(3))*Sqrt(6)

Sin(0)=0

Sin(x_IsNumber*y_):=Check(SignumCmp(x)<0,-Sin(-x*y))

Sin(Pi*x_IsNumber):=Check(x>=1/2,If(x<1,Sin((1-x)*Pi),If(x<2,-Sin((2-x)*Pi),Sin((x-2*Div(Trunc(x),2))*Pi))))

Sin(ACos(x_)):=Sqrt(1-x^2)

Sin(x_IsNumber):=Check(SignumCmp(x)<0,-Sin(-x))

Sin(ASin(x_)):=x

Sin(ATan(x_)):=x*1/Sqrt(1+x^2)

Div(<arg1>,...,<argN>) - Divide

Div(23,7) → 3

DivP(<polynomial1>,<polynomial2>,<variable>) - Divide polinómios

DivP(x^2+x-12,x-3,x) → 4+x

Exp(<arg1>) - exponencial

Exp(0) → 1

Exp(4) → E^4

Expand(<arg1>) - expande expressão

Expand((a+b)*(c+d)) → a*c+a*d+b*c+b*d

Factorial(<arg1>) - devolve o factorial de um número

Factorial(5) → 120

GCD(<arg1>,...,<argN>) - máximo divisor comum

GCD(15,27) → 3

GCDP(<polynomial1>,<polynomial2>,<variable>) - máximo polinómio comum

GCDP(x^3+3*x^2-10*x-24,x^2+4*x-21,x) → -3+x

Greater(<arg1>,arg2>) - compara números

Greater(7,7) (7 > 7) → False

GreaterEqual(<arg1>,arg2>) - compara números

GreaterEqual(7,7) (7 ≥ 7) → True

IdentityMatrix(<length>) - devolve a matriz identidade

IdentityMatrix(2) → {{1,0},{0,1}}

IsRational(<arg>) - teste de racionais

IsRational(I) → False

LengthStr(<str>) - mede comprimento de uma *string*

LengthStr(miguel) → 6

Less(<arg1>,<arg2>) - compara números

Less(7,9) (7 < 9) → True

LessEqual(<arg1>,<arg2>) - compara números

LessEqual(7,7) (7≤ 7) → True

N(<expr>) - devolve o valor numérico

N(Sin(1/2)) → 0.47942553604203

Pow(<arg1>,<arg2>) - potência

Pow(3,2) (3^2) → 9

ReplaceStr(<string>,<char1>,<char2>) - substitui parte de uma *string*

ReplaceStr(miguel,i,o) → moguel

SolveP(<polynomial>,<variable>) - resolve equações

SolveP(x^2+x-12,x) → {-1/2+Sqrt(49/4),-1/2-Sqrt(49/4)}*

SolveP(x^2-4*x+2,x) → {2+Sqrt(2),2-Sqrt(2)}

ToLowerCaseStr(<string>) - transforma em minúsculas

ToLowerCaseStr(MiGuel) → miguel

ToUpperCaseStr(<string>) - transforma em maiúsculas

ToUpperCaseStr(MiGuel) → MIGUEL

Existem ainda, como não poderia deixar de ser, as funções e operadores "tradicionais", cuja sintaxe para a sua invocação é a seguinte: Função(<arg>). De salientar que a "Função" deverá ser iniciada por maiúsculas e <arg> deverá ser o valor do qual se pretende calcular a respectiva imagem. Alguns exemplos dessas funções:

Sin(<arg>) - função seno;

ASin(<arg>) - função arco seno;

Sinh(<arg>) - função seno hiperbólico;

ASinh(<arg>) - função arco seno hiperbólico;

Cos(<arg>) - função cosseno;

ACos(<arg>) - função arco cosseno

Cosh(<arg>) - função cosseno hiperbólico;

ACosh(<arg>) - função arco cosseno hiperbólico

Tan(<arg>) - função tangente;

ATann(<arg>) - função arco tangente;

Tanh(<arg>) - função tangente hiperbólica;

ATanh(<arg>) - função arco tangente hiperbólica;

Cot(<arg>) - função cotangente;

ACot(<arg>) - função arco cotangente;

Coth(<arg>) - função cotangente hiperbólica;

ACoth(<arg>) - função arco cotangente hiperbólica;

Sqrt(<arg>) - Função raíz quadrada;

Abs(<arg>) - função módulo;

A exploração da vertente gráfica desta *applet* será feita na sub-secção PlotPanel do Hartmath, disponível no espaço dedicado às Representações Gráficas.

Formel Manipulator

Esta *applet* Java, criada por Jens-Uwe Dolinsky em 17/08/2000, permite que o utilizador introduza uma determinada expressão e que, através do clique num botão, execute determinadas operações matemáticas, designadamente:

- operações simples;
- funções básicas;
- substituições;
- resolução de equações;
- derivação;
- integração;
- aproximações de Taylor;

- Séries de Fourrier;
- aproximações.

	Sintaxe	descrição
operadores	+,-,*,/,^	operações básicas
constantes	PI	p
funções básicas	SQRT(x)	raíz quadrada
	EXP(x)	função exponencial-base e
	LN(x)	função logaritmo-base e
	SIN(x),COS(x), TAN(x),COT(x)	funções trigonométricas
	ASIN(x),ACOS (x),ATAN(x)	funções trigonométricas inversas
	SINH(x),COSH (x)	funções hiperbólicas
	ABS(x)	valor absoluto de x
	FAK(x)	factorial de x
substituições	SUBST(expr,x, a)	substitui na expressão expr, a variável x pelo valor a
Resolução de equações	SOLVE(expr1= expr2,x)	Resolve equações lineares (ou quadráticas) expr1-expr2=0 em ordem a x.
derivação	DIFF(f,x)	deriva f em ordem a x
	DIFF(f,x,n)	calcula a n-ésima derivada em ordem a x
integração	INT(f,x)	integral indefinido de f em ordem a x
	INT(f,x,i1,i2)	integral definido de f em ordem a x no intervalo [i1 , i2]
Aproximações de Taylor	TAYLOR(f,x,x0 ,n)	Aproximação de Taylor de f em ordem a x em x=x0 na n-ésima ordem
Séries de Fourier	FOURIER(f,x,t 1,t2,n)	Série de Fourier de f em ordem a x no intervalo [t1,t2] na n-ésima ordem
Aproximações	APPROX(x)	determina o valor de x

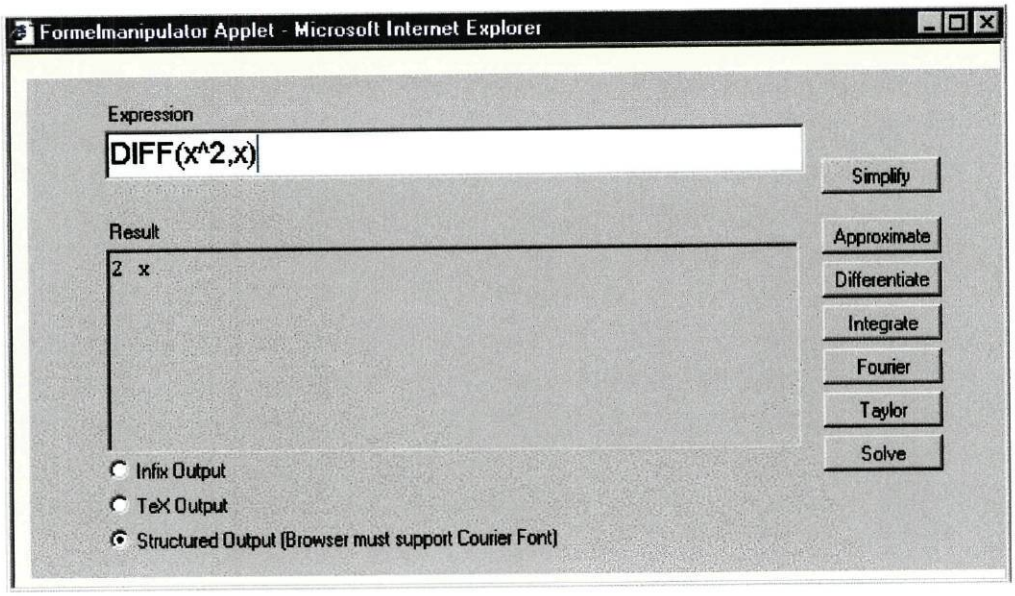
Experimente introduzir na caixa destinada à introdução de dados as seguintes expressões e simplifique-as:

Derivação	DIFF(-1/5*x^2*TAN(LN(x)*/2),x) DIFF(SIN(t^x),t)
Derivação parcial	DIFF(DIFF(1/(x^2+y^2)x),y)
Integração	INT(x^4+SIN(PI*x),x) INT(3*x^4+SIN(x),x,0,4)
Cálculo numérico	(8+564)*98-6 APPROX(SIN(23)*COS(3*0.7))
Substituições	SUBST(2*x^2+x-3,x,4) valor do polinómio 2*x^2+x-3 no ponto x=4
Aproximações de Taylor	TAYLOR(SINH(x^2),x,0,5)
Séries de Fourier	Fourier(x,x,0,1,6)
Resolução de equações	SOLVE(3*x^2-8=x-9+x^2,x)

De salientar ainda que esta *applet* foi utilizada para conseguir o output de determinadas situações neste capítulo, designadamente na secção referente à derivação de expressões (em que foi usada a função "DIFF") e na secção que diz respeito à resolução de equações (em que foi usada a função "SOLVE").

Poderá fazer o download gratuito desta *applet* no endereço http://www.mb.hs-wismar.de/mitarbeiter/pawletta/00Uwe/formel.html.

Clique sobre a imagem seguinte para ter a possibilidade de introduzir as expressões que pretende desenvolver.



Derivação de expressões

Este exemplo diz respeito a uma aplicação criada em Java que permite calcular a derivada de uma função, introduzida algebricamente pelo utilizador. Quando se pressiona na imagem que é apresentada de seguida, é aberta uma janela com os seguintes elementos: uma caixa de texto onde o utilizador poderá introduzir a expressão da qual pretende calcular a derivada e uma zona branca na parte inferior da tela onde aparecerá representada a expressão analítica da derivada da função introduzida. Para se chegar ao resultado deverá pressionar-se o respectivo botão, com a inscrição "derivar". Poderão introduzir-se novas funções, depois de se apagarem os valores que são visíveis nos dois campos.

Relativamente a este exemplo há alguns aspectos que importa ressaltar. Em primeiro lugar, dizer que a apresentação do output é feita através da utilização da *applet* Visualizador de LaTex.

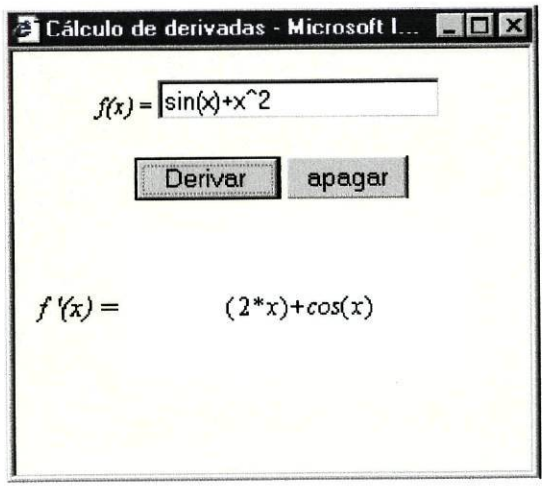
Por outro lado, a janela que se abrirá é constituída por duas partes: a superior que contém uma caixa de texto onde o utilizador faz o input da expressão da qual pretende calcular a derivada, e a inferior onde aparecerá o resultado, no espaço da tela a branco. É claro que para expressões maiores esse espaço não será suficiente para tornar visível a totalidade da expressão mas, como já se referiu para outras situações, o controlo dessa zona também é possível.

Foi usada uma expressão cuja sintaxe é do tipo **expressao=expressao.replace("sen","sin")** (contendo "expressao" uma cadeia de caracteres), para que se conseguisse substituir as ocorrências de "sen" por "sin", com o intuito de se "aportuguesarem" algumas expressões. Isto quer dizer que, com esta transformação, o aluno poderá fazer referência à função seno da forma que está habituado, sem ter que usar as notações usuais, em Inglês, pré-definidas. Esta é uma técnica que valerá a pena ser usada, tanto mais que não são necessárias mais do que uma "meia dúzia" de linhas de código.

Para que se compreenda melhor esta situação, experimente também introduzir, por exemplo, a função *tg(x)*. O resultado da sua derivada que é apresentado é *diff(tg(x),x)*, uma vez que não foi feita a transformação através da expressão *replace*. Para calcularmos a derivada de expressões que envolvam a função tangente, teremos, portanto, que usar uma expressão que a *applet* reconheça: neste caso *tan(x)*.

De salientar ainda que neste exemplo foi utilizada a *applet* Formel Manipulator referida nesta mesma secção, em particular através da função DIFF.

A imagem seguinte diz respeito ao aspecto do documento que foi criado para levar a cabo o que se disse anteriormente. Pressione sobre ela, experimente introduzir algumas expressões analíticas de funções e calcule as suas derivadas.

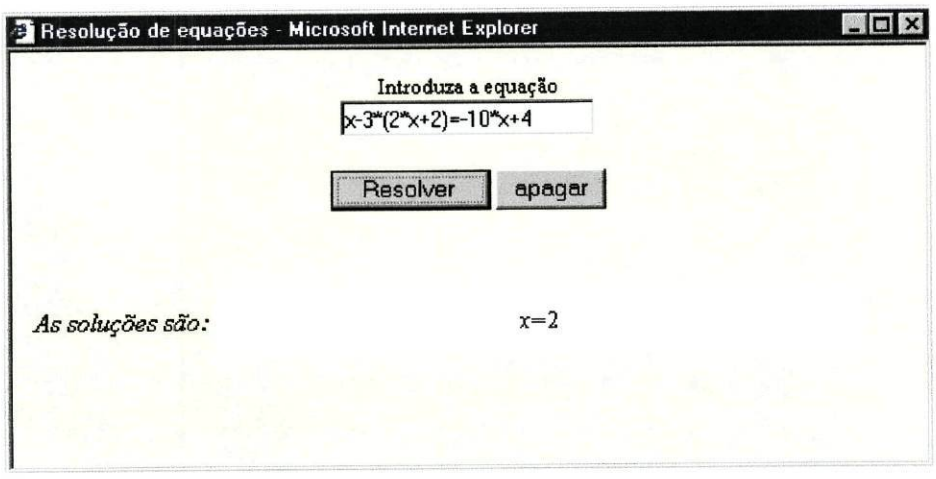


Resolução de equações dos 1º e 2º graus

Este é um exemplo que foi criado através da utilização de duas *applets* de que já se fez referência. Uma delas é a *applet* Formel Manipulator referida nesta mesma secção, que permite efectuar determinadas manipulações matemáticas, em particular com a função particular SOLVE. Por outro lado, para a apresentação dos resultados, foi utilizada a *applet*, descrita no espaço "Visualizador de LaTex". Trata-se, de facto, de um exemplo em que foram usadas duas *applets* completamente distintas e independentes uma da outra mas que, para a execução de determinada tarefa, podem ser complementadas e usadas simultaneamente no mesmo documento.

De salientar ainda o facto de, com o auxílio da linguagem JavaScript, foi usado novamente um método para modificar determinadas expressões. A sintaxe **expressao=expressao.replace("cao","gato")** (contendo "expressao" uma cadeia de caracteres), substitui as ocorrências de "cao" por "gato". Por exemplo, se a expressao="o cao vai a correr", o resultado será "o gato vai a correr".

Pressione com o ponteiro do rato sobre a imagem seguinte e determine as soluções de equações do 1º e 2º graus introduzidas por si.



Representações gráficas

Neste espaço apresentam-se algumas aplicações Java que permitem a análise gráfica. À semelhança do que foi feito na secção anterior, simplesmente se utilizarão *applet*s em linguagem Java já existentes na Web, se bem que com algumas adaptações.

XFunctions

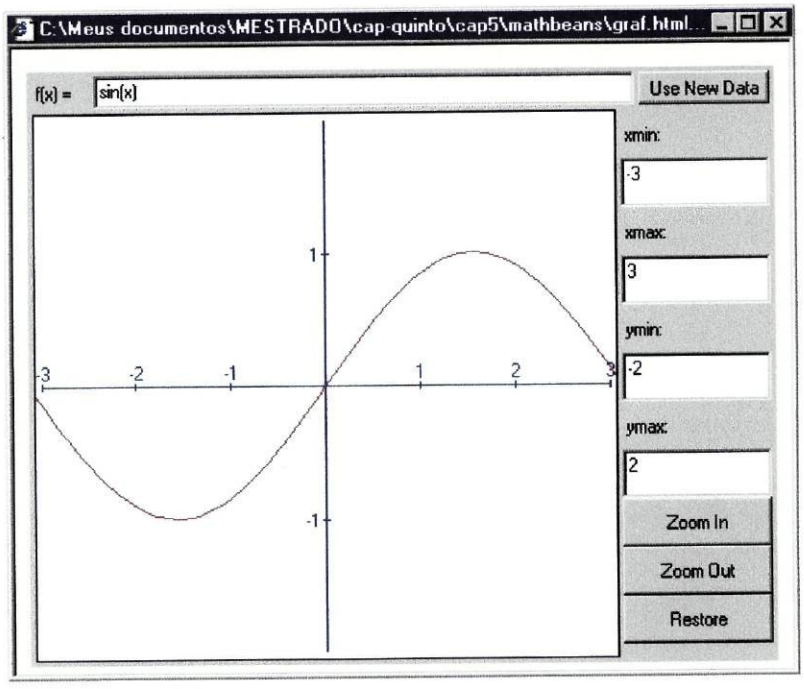
Com esta *applet*, criada por David Eck, o utilizador poderá esboçar gráficos de funções, das quais introduz apenas a expressão analítica. Tem a vantagem de não necessitar de qualquer software especial mas, em termos de output, ou seja, de resultado final, não são conseguidos os mesmos resultados como aqueles que se obtêm através da utilização do MathView, nomeadamente em termos de manipulação da escala. Este aspecto mostra-se essencial na visualização do comportamento de determinadas funções. Por outro lado, é possível controlar os limites de variação das abcissas e das ordenadas, em termos de campo de visualização. É possível também utilizar as funções zoom in e zoom out.

A *applet* foi invocada no documento referente à janela retratada em baixo da seguinte forma:

```
<applet code="mathbeans.GraphApplet.class"
archive="mathbeans.jar" width=480 height=380>
<param name="function" value="sin(x)">
<param name="limits" value="-3 3 -2 2">
<param name="userInput" value="yes">
<param name="userLimits" value="yes">
<param name="userZoom" value="yes">
</appet>
```

De salientar que esta *applet* poderá ser obtida gratuitamente, no endereço: http://math.hws.edu/xFunction, onde também poderão ser obtidas mais informações para a sua manipulação, numa versão mais recente daquela que aqui é apresentada.

Pressione sobre a imagem seguinte para poder introduzir as suas próprias funções, das quais pretende esboçar o respectivo gráfico.



Java Graph Package

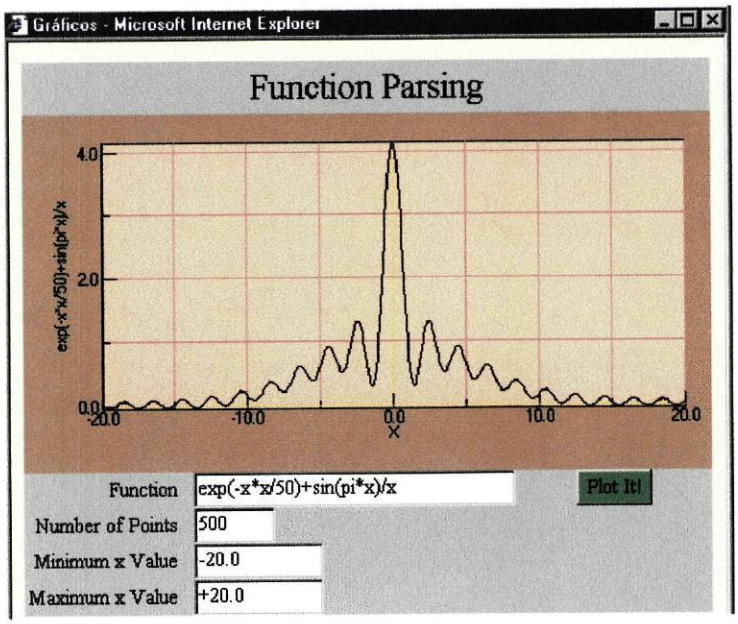
Criada por Leigh Brookshaw, esta *applet* apresenta boas capacidades em termos de representação gráfica de funções. Para além de se poder mudar a expressão analítica da função da qual se pretende esboçar o gráfico, é possível escolher a variação das abcissas, bem como o número de pontos com que se pretende o gráfico. Pode ainda pressionar o botão esquerdo do rato, com o ponteiro sobre uma zona qualquer do gráfico e, sem largar, formar uma caixa na zona que pretende ampliar. Para obter mais informações sobre o seu funcionamento ou se pretender o seu download, visite o site http://www.sci.usq.edu.au/staff/leighb/graph/top.html.

A *applet* foi invocada através das instruções que aparecem de seguida, onde são visíveis, por exemplo, as dimensões do espaço da tela ocupado pela *applet*, o número de pontos que são marcados para a representação gráfica da função, o domínio da função, a janela de visualização e a expressão que

aparece representada por defeito. O valor do parâmetro CODE é o que aparece referenciado a vermelho.

```
<APPLET code=parse1d.class codeBase=classes height=400 width=500>
<PARAM NAME="function" VALUE="exp(-x*x/50)+sin(pi*x)/x">
<PARAM NAME="points" VALUE="500">
<PARAM NAME="minimum" VALUE="-20.0">
<PARAM NAME="maximum" VALUE="+20.0">
<PARAM NAME="code" VALUE="parse1d.class">
<PARAM NAME="height" VALUE="500">
<PARAM NAME="width" VALUE="600">
</APPLET>
```

Pressione sobre a janela e introduza as expressão analíticas das funções das quais pretende esboçar o gráfico.



PlotPanel do Hartmath

Tivemos oportunidade de nos referirmos às instruções básicas para a manipulação desta *applet* no espaço existente na Álgebra Simbólica, na secção Hartmath. Nesta secção, iremos apenas abordar a componente gráfica desta mesma aplicação Java.

Existem, então, outras funções para além daquelas que foram descritas na secção anterior, que permitem explorar a componente gráfica desta *applet*. Ao invocarmos tais funções, abre-se uma janela que contém o gráfico da função

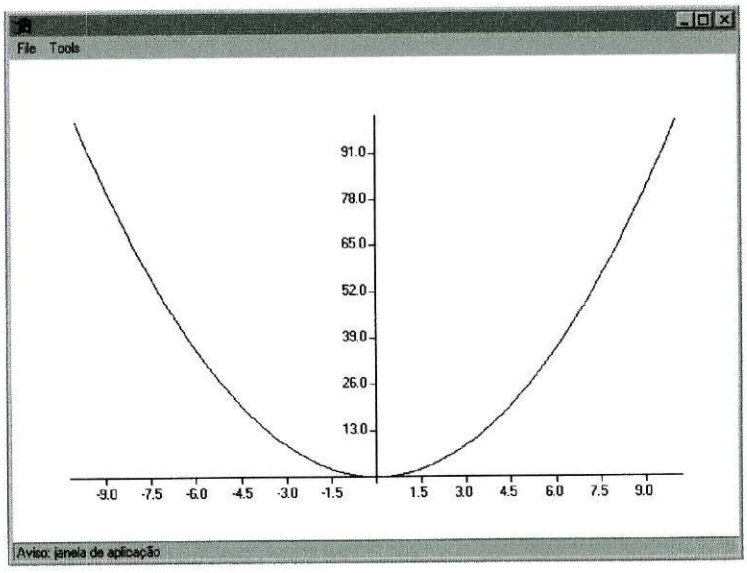
que se introduziu inicialmente, a variar entre os valores também fixados à priori. As expressões seguintes constituem a sintaxe necessária ao esboço de gráficos, inclusivamente gráficos tri-dimensionais e referentes a expressões paramétricas.

Plot(x^2,{x,-10,10})

Plot3D(Sin(x^2*y^2),{x,-5,5},{y,-2,2})

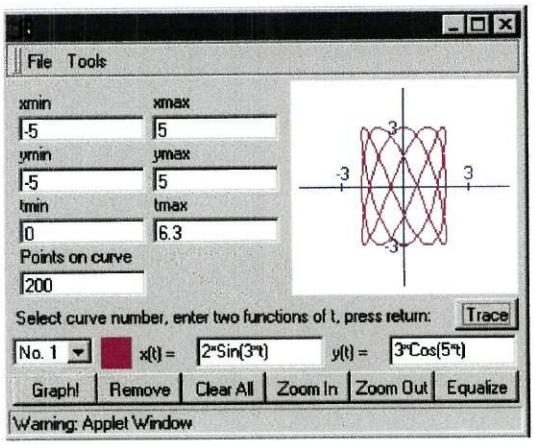
ParametricPlot({2*Sin(x),3*Cos(x)},{x, -Pi, Pi})

A imagem seguinte, representa a janela que é aberta quando se esboça um gráfico de uma dada função, por exemplo, a primeira que aparece na lista anterior.

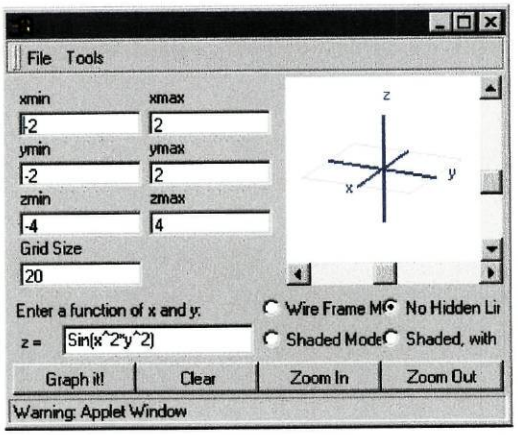


Por outro lado, é possível abrir uma janela destinada a um estudo mais aprofundado de funções (designadamente as segunda e terceira da lista anterior) em que se podem controlar os limites de variação da mesma, número de pontos com que é traçada a linha, zoom in e zoom out. A sintaxe necessária à abertura de tais janelas é a que aparece de seguida a azul, onde, por baixo de cada uma dessas instruções aparece a imagem da respectiva janela que é aberta. Relembre-se que, depois de introduzida a expressão na caixa disponível na *applet* para esse efeito, terá que se pressionar a tecla **RETURN** (ENTER da parte não numérica do teclado), com a tecla SHIFT pressionada.

ParametricPlotPanel()



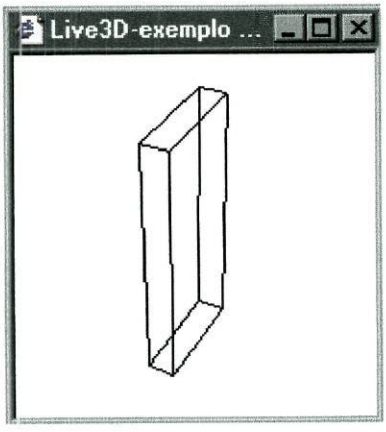
Plot3DPanel()



LiveGraphics3D

A LiveGraphics3D é uma *applet* Java criada sem fins comerciais, cuja utilização pode trazer bons resultados na manipulação de gráficos tri-dimencionais, gráficos esses produzidos através do programa *Mathematica*. Para trabalhar com ela, o utilizador, à semelhança do que acontece com outras *applets* apresentadas ao longo deste Capítulo V, não necessita de conhecer linguagem de programação Java.

Vejamos um pequeno exemplo: o código que de seguida se apresenta, é o código necessário à introdução da imagem que aparece em baixo.




```
<html><head>
<title>
Live3D-exemplo
</title>
</head>
<BODY bgcolor=cornsilk>
<APPLET code="Live.class" archive="Live.jar"  height=150
width=150>
<PARAM name=BGCOLOR value=#ffffff>
<PARAM name=INPUT value="Graphics3D[{},
BoxRatios->{1,4,9} ]">
</APPLET>
</body></html>
```

O exemplo diz respeito a uma caixa que, geralmente, aparece a circunscrever o gráfico esboçado. De salientar que a tag *<APPLET>* contém determinados parâmetros que convertem o código apresentado numa determinada imagem gráfica. Isto quer dizer que a *applet* em causa pode ser controlada por parâmetros, alguns dos quais se revelam, como na generalidade das *applets*, opcionais. Em termos de obrigatoriedade para o funcionamento desta *applet* temos ARCHIVE ("Live.jar"), CODE ("Live.class"), WIDTH e HEIGHT (largura e altura) e o parâmetro INPUT_FILE (que pode ser substituído pelo parâmetro INPUT).

Para conseguirmos efectuar construções com esta *applet*, teremos que introduzir determinadas instruções no parâmetro INPUT, que constitui uma *string* que, por sua vez, contém instruções do tipo InputForm, referentes a objectos do tipo Grafics3D ou ShowAnimation já conhecidos do programa *Mathematica*. No nosso exemplo, essas instruções específicas foram as seguintes:

```
<PARAM name=INPUT value="Graphics3D[{},
BoxRatios->{1,4,9} ]">
```

Para além dos parâmetros de Java, existem ainda outros parâmetros capazes de controlar determinados aspectos das construções obtidas que, para uma abordagem mais completa, sugere-se uma visita ao site respectivo (http://wwwvis.informatik.uni-stuttgart.de/~kraus/livegraphics3D/index.html).
Desses parâmetros, destacam-se, por exemplo:

parâmetro	descrição
AUDIO_ENTER	nome de ficheiro audio (de extensão ".au"), que é accionado quando o ponteiro do rato se situa na zona da *applet*
AUDIO_LOOP_ENTER	o mesmo que o AUDIO_ENTER mas executado em loop
AUDIO_FIRST_FRAME	o mesmo que o AUDIO_ENTER mas executado enquanto a primeira frame de uma animação é visualizada
BACKGROUND	nome do ficheiro de imagem com extensão "*.JPEG" ou "*.GIF" para ser visualizado enquanto fundo fixo
BGCOLOR	cor do fundo na forma hexadecimal
CYLINDRICAL_BACKGROUND	nome do ficheiro de imagem com extensão "*.JPEG" ou "*.GIF" para ser visualizado enquanto fundo cilíndrico
INPUT_ARCHIVE	nome de um ficheiro zip que contém o INPUT_FILE.
INPUT_FILE	nome de um ficheiro que contém instruções do tipo InputForm referentes a objectos do tipo Graphics3D ou ShowAnimation. Este ficheiro deverá estar no ficheiro zip (veja INPUT_ARCHIVE).

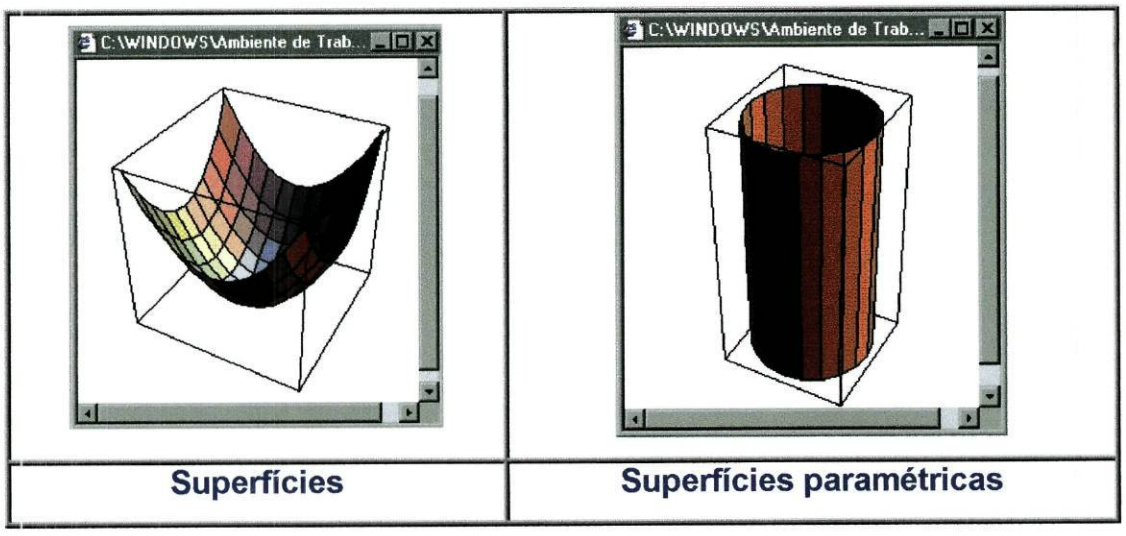
MAGNIFICATION	factor referente ao zoom (que pode ser modificado interactivamente premindo-se na tecla SHIFT e arrastando o ponteiro verticalmente)
POINT_EDGE_COLOR	cor das linhas de todos os pontos na forma hexadecimal
SPHERICAL_BACKGROUND	nome do ficheiro de imagem JPEG ou GIF que será apresentado como fundo esférico
SPIN_X	velocidade do movimento giratório em torno de um eixo vertical, especificado por correspondentes movimentos do rato, em pixels, por frame.
SPIN_Y	o mesmo que o SPIN_X para uma velocidade de movimento giratório em torno de um eixo horizontal
VISIBLE_FACES	determina as faces a desenhar, do polígono. (Por defeito, são representadas ambas as faces; o valor FRONT representará apenas as faces frontais e o valor BACK apenas as faces de trás.)

Relativamente aos comandos capazes de desenvolverem algum tipo de interactividade no gráfico apresentado, há a referir os seguintes aspectos:

acção do utilizador	reacção da *applet*
1. arrastar o rato com o botão esquerdo pressionado	rotação da imagem sobre um eixo
2. largar o botão esquerdo do rato enquanto este é arrastado	faz girar a imagem sobre um eixo
3. premir tecla SHIFT enquanto se executa 1. na vertical	amplia ou reduz a imagem (zoom)
4. premir tecla SHIFT enquanto se executa 1. na horizontal	rotação da imagem sobre um eixo perpendicular aquele que, ficticiamente, está a ser percorrido
5. premir a tecla ALT enquanto se executa 1. na vertival	Retira partes do gráfico

Nas duas secções seguintes, serão apresentados dois exemplos de aplicação da LiveGraphics3D.

A primeira refere-se à representação gráfica de superfícies e a segunda à representação de superfícies em coordenadas paramétricas. Em ambos os casos, ao pressionar na imagem respectiva, abre-se uma janela que permite, entre outras coisas, a introdução de expressões analíticas das quais se pretende visualizar o gráfico. Para cada função representada graficamente, pode pressionar o botão direito do rato com o ponteiro situado sobre a janela referente ao gráfico e ver o código, através da escolha da função "Ver Html" e daí, verificar como se efectuam, em termos de sintaxe, as respectivas construções com o auxílio desta *applet*.



JavaSketchpad

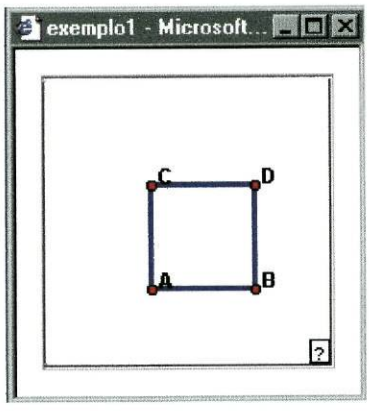
Esta *applet*, que também se encontra disponível gratuitamente (desde que não seja com intuitos comerciais), no endereço http://www.keypress.com/sketchpad/java_gsp/index.html, poderá ser usada para a concepção e apresentação on-line de construções geométricas, em que o utilizador poderá manipular os objectos representados, de uma forma interactiva. À semelhança do que acontece com a generalidade das *applets*, esta também é mergulhada nas páginas HTML, bastando invocá-la para a utilizarmos.

Vejamos um pequeno exemplo: o código que de seguida se apresenta, é o código necessário à introdução de uma construção através da utilização desta *applet*.

```
<HTML>
<HEAD>
<TITLE>exemplo1</TITLE>
</HEAD>
<BODY bgColor=cornsilk>
<center>
<APPLET align=center code=GSP.class height=170 width=170 ARCHIVE="JSPDR3.JAR">
<PARAM NAME="offscreen" VALUE="0">
<PARAM NAME="DirectedAngles" VALUE="0">
<PARAM NAME="Backgreen" VALUE="255">
<PARAM NAME="Backred" VALUE="255">
<PARAM NAME="Backblue" VALUE="255">
<PARAM NAME="Construction" VALUE="
```

```
{1} Point(120,120)[label('B')];
{2} Point(60,120)[label('A')];
{3} Point(60,60)[label('C')];
{4} Point(120,60)[label('D')];
{5} Segment (3,4) [thick];
{6} Segment (4,1) [thick];
{7} Segment (1,2) [thick];
{8} Segment (2,3) [thick];
">
<PARAM NAME="align" VALUE="center">
<PARAM NAME="height" VALUE="250">
<PARAM NAME="width" VALUE="250">
<PARAM NAME="ARCHIVE" VALUE="JSPDR3.JAR">
</APPLET>
</center>
</BODY>
</HTML>
```

A construção obtida será, então a seguinte:



Pela análise do código anterior, em relação aos parâmetros necessários à sua visualização, podemos referir o seguinte:

- O parâmetro CODE deverá ser "GSP.class";

- O parâmetro ARCHIVE "JSPDR3.JAR");

- Entre as *tags* **<APPLET>** e **</APPLET>**, estão definidos os parâmetros específicos de cada *string*, ou seja, em cada uma das marcas do tipo <PARAM NAME="nomedoparametro" VALUE="valor"> são definidas as características específicas da *applet* que está a ser conseguida. De seguida, abordar-se-ão os parâmetros essenciais à construção do exemplo agora em análise, bem como outros parâmetros necessários à obtenção de determinados resultados. De salientar ainda que o

parâmetro "Construction" é o que contém as instruções específicas de cada construção.

Aparecem ainda outros parâmetros que fazem com que o aspecto tome esta ou aquela característica. Por exemplo, se nada se disser em contrário, a tela que diz respeito à expressão conseguida através da *applet*, aparece a cinzento. No entanto, à semelhança do que acontece com a tradicional linguagem HTML, poderemos controlar a intensidade de verde, vermelho e azul, dentro de uma escala que vai de 0 a 255. No nosso exemplo, atribuímos o valor aos três tons de 255, ou seja, o fundo da tela aparecerá a branco. Os parâmetros seguintes e os respectivos valores, dizem respeito a essa situação.

```
<PARAM NAME="Backgreen" VALUE="255">
<PARAM NAME="Backred" VALUE="255">
<PARAM NAME="Backblue" VALUE="255">
```

O Parâmetro **CONSTRUCTION**

Este parâmetro é o mais importante do JavaSketchpad, uma vez que é através dele que se conseguem os efeitos e as construções pretendidas. O código que de seguida se apresenta, diz respeito ao exemplo que estamos a considerar, ou seja, um quadrado representado na tela, constituído por quatro pontos (os vértices) e quatro segmentos de recta (os lados). Uma particularidade deste tipo de construções é que os pontos podem ser movidos pelo utilizador (desde que se diga o contrário através de uma instrução específica), criando uma interactividade interessante para quem visualiza este tipo de construções.

```
<PARAM NAME="Construction" VALUE="
{1} Point(120,120)[label('B')];
{2} Point(60,120)[label('A')];
{3} Point(60,60)[label('C')];
{4} Point(120,60)[label('D')];
{5} Segment (3,4) [thick];
{6} Segment (4,1) [thick];
{7} Segment (1,2) [thick];
{8} Segment (2,3) [thick];
">
```

Relativamente ao código anterior, destacamos os seguintes aspectos:

- Qualquer texto que for escrito entre chavetas ({}) é ignorado pelo JavaSketchpad. No nosso exemplo, optou-se por levar a cabo uma numeração para que fosse mais fácil a identificação de cada uma das linhas de código.

- A sintaxe referente à definição da construção é **{nome_do_objecto} {argumento_do_objecto} {formatos_opcionais}**. Por exemplo, na linha de construção {2} Point(60,120)[label('A')];, **{nome_do_objecto}** corresponde a **Point**, **{argumento_do_objecto}** corresponde a **(60,120)** e **{formatos_opcionais}** corresponde a **[label('A')]**.

De uma forma geral, {nome_do_objecto} poderá representar pontos ("point"), rectas ("line"), segmentos de recta ("segment") ou mesmo determinadas funções sobre objectos (por exemplo, a que diz respeito ao ponto médio- "midpoint").

Por sua vez, {argumento_do_objecto} depende do anterior, é uma lista de inteiros entre parêntesis e separados por vírgulas a que correspondem a definição de objectos de construção.

Vejamos de seguida as formas de definir algumas diferentes construções. Para obter mais informações acerca das construções obtidas a partir do JavaSketchpad, poderá visitar o site referido no início desta página ou o trabalho realizado por José Manuel dos Santos dos Santos.

Pontos

Point(num1,num2) - Através desta expressão é definido um ponto, em que (num1,num2) é um par ordenado que define a posição do ponto na tela. A coordenada num1 representa a distância na horizontal, em pixels, do ponto à margem esquerda da tela e num2 representa a distância na vertical, em pixels,

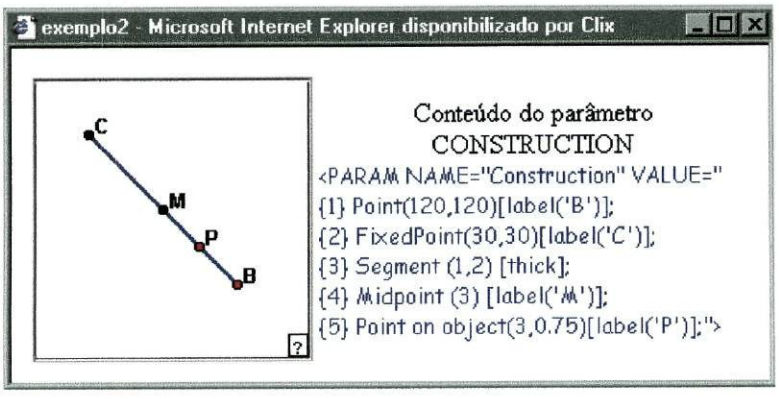
do ponto ao topo da tela. De referir que o ponto (0,0) representa o canto superior esquerdo da tela e (120,100) o canto inferior direito da tela.

FixedPoint(num1,num2) - ponto fixo

MidPoint(segm) - ponto médio do segmento de recta "segm"

Point on object(obj,num) - ponto num lugar geométrico (recta ou círculo)

A imagem seguinte constitui um exemplo.



Objectos rectilíneos

Segment(Point1,Point2) - Segmento de recta por Point1 e Point2

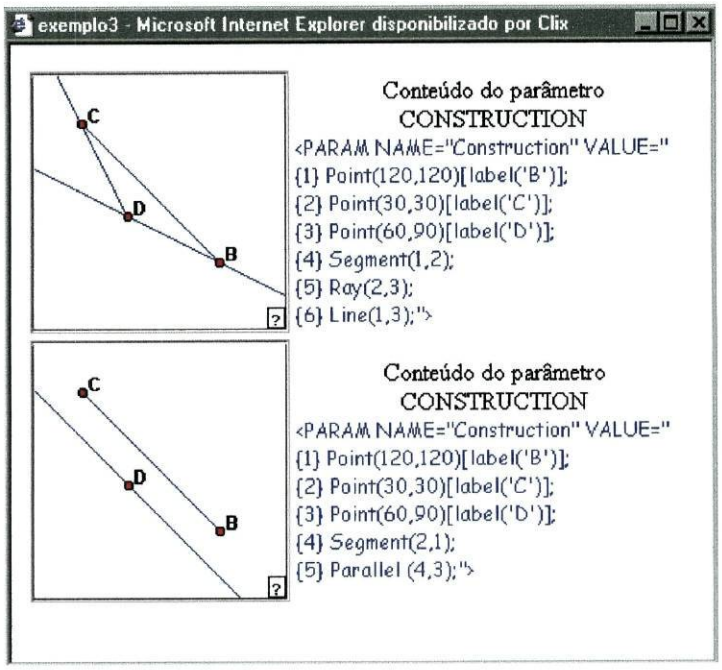
Ray(Point1,Point2) - Semi-recta com origem em Point2

Line(Point1,Point2) - Recta por Point1 e Point2

Perpendicular(Point1,Point2) - Recta perpendicular ao segmento por Point1 e Point2

Parallel(object,Point2) - Recta paralela ao object (segmento, recta ou semi-recta) que passa por Point1

A imagem seguinte constitui um exemplo.



Construções geométricas

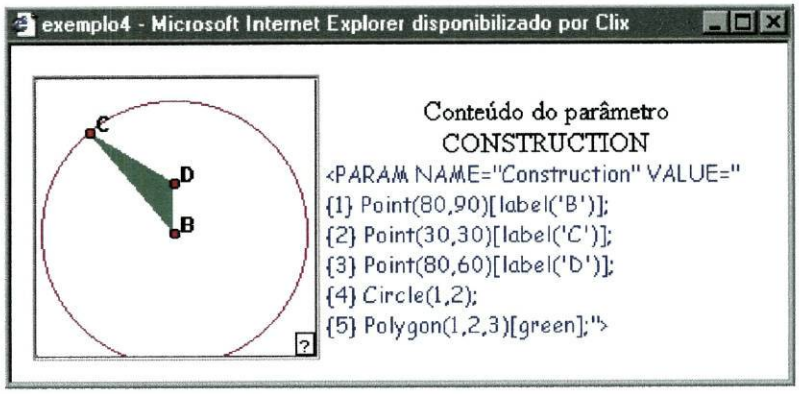
Circle(Point1,Point2) - Circunferência de centro Point1 e que contém Point2

Circle by radius(Point1,Segm) - Circunferência de centro Point1 e raio igual ao comprimento de "segm"

Polygon(Point1,Point2,...,PointN) - Interior do polígono de vértice Point1, Point2,...,PointN

Circle interior(circ) - Círculo definido pela parte interior da circunferência "circ"

A imagem seguinte constitui um exemplo.



Construções de medidas

Length(Segm,posx,posy,string) - Medida do segmento de recta Segm, em que a legenda string aparece posicionada na tela com início no ponto de coordenadas (posx,posy);

Angle(Point1,Point2,Point3,posx,posy,string) - Medida do ângulo definidos pelos três pontos, em que a legenda aparece situada como na Length;

Perimeter(obj1,posx,posy,string) - Perímetro do polígono obj1, em que a legenda aparece situada como na Length;

Circunference(obj1,posx,posy,string) - Perímetro da circunferência obj1, em que a legenda aparece situada como na Length;

Radius(circ,posx,posy,string) - Medida do raio da circunferência ou círculo circ, em que a legenda aparece situada como na Length;

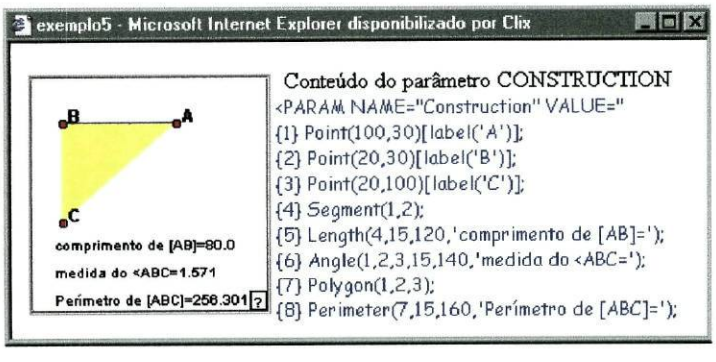
Area(obj1,posx,posy,string) - Área do objecto obj1, em que a legenda aparece situada como na Length;

Slope(obj1,posx,posy,string) - Declive, em relação à horizontal, do objecto obj1 (pode ser recta, semi-recta ou segmento de recta), em que a legenda aparece situada como na Length;

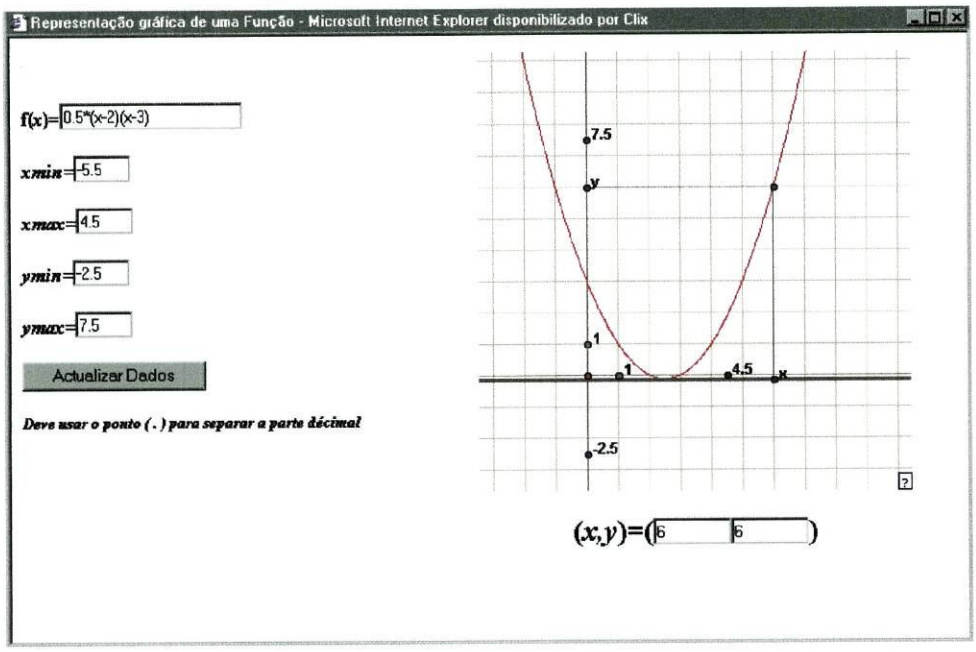
Coordinates(Point,obj,posx,posy,string) - Determina o par ordenado que representa as coordenadas do ponto Point relativamente ao referencial designado por obj, em que a legenda aparece situada como na Length;

Calculate(posx,posy,prefixstring,exprestring)(obj1,...,objN) - Determina o valor de uma fórmula contida em exprestring, em que obj1,...,objN são as referências dos objectos cujas propriedades intervêm no cálculo da expressão. A legenda aparece situada como na Length;

A imagem seguinte constitui um exemplo.



De seguida apresenta-se um exemplo em que aparecem retratados alguns dos aspectos interessantes que se podem fazer com esta *applet*, designadamente, no que diz respeito à leitura de partes da tela, neste caso de determinados pontos após a representação gráfica de uma função introduzida pelo utilizador. Este exemplo, foi retirado do trabalho realizado por José Manuel dos Santos dos Santos.



Bibliografia:

Associação de Professores de Matemática (1997). *Histórias da aula de Matemática*. Lisboa: Autor.

Associação de Professores de Matemática (1998). *Matemática 2001. Diagnóstico e Recomendações para o Ensino e Aprendizagem da Matemática*. Lisboa: Autor.

Benavente, A. (1996). Os projectos "Inovar Educando/Educar Inovando". In B. Campos (Ed.), *Investigação e inovação para a qualidade das escolas* (pp. 47-51). Lisboa: Instituto de Inovação Educacional.

Bentes Paulo, J. M. (1999). *Web & Internet e o Ensino da Matemática*. Artigo apresentado na I Conferência Internacional Challanges'99.

Brophy, T. (1997). *Mathematics & Science with MathView*. Londres: John F. Marshall.

Callixto, J. C. (1998).*Como utilizar o Windows 98*. Amadora: Edições Gustave Eiffel.

Carriço, J. A: (1998). *A Linguagem HTML*. Lisboa: CTI – Centro de Tecnologias de Informação, Lda.

Coelho, P. (1998). *Criação de Páginas na World Wide Web com HTML 4 & Java*. Lisboa: FCA – Editora de Informática.

Departamento de Avaliação, Prospectiva e Planeamento - Ministério da Educação (1999). *Programa Nónio – Século XXI. Alguns números sobre as tecnologias de informação e comunicação na educação*. Lisboa: Autor.

Exame Informática (1999). *Como Construir Páginas Web com o Microsoft Office 97*. Linda-a-Velha: Abril/Controljornal.

Feather, S. (1997). *Javascript em Exemplos* (Tradução, A. Rodrigues Antunes). São Paulo: Makron Books. (Original publicado em 1996).

Freitas, C. (1997). A integração das NTI no processo de ensino – aprendizagem. In C. Freitas et al. (Eds), *Tecnologias de Informação e Comunicação na Aprendizagem* (pp.11-20). Lisboa: Instituto de Inovação Educacional.

Freitas, J. (1999). De onde vimos e para onde vamos: o futuro da Internet na escola. In J. A. Alves, P. Campos e P. Brito (Eds), *O Futuro da Internet. Estado da arte e tendências de evolução* (pp.183-196). Lisboa: Centro Atlântico.

Hoffner, N. S. (1997). *MathView User's guide*. Ontario: Waterloo Maple Inc.

Kahn, B. (1991). *Os Computadores no Ensino da Matemática. Lisboa: Publicações Dom Quixote.*

Lemos, M. (1998). *Estar na Internet. Tudo o que precisa de saber sobre a Internet.* Lisboa: McGraw-Hill.

Milner, A., Burrows, T. (Eds.). (1997). *Internet* (Tradução, A. Silva e P. Taipas). Londres: DK Direct Limited. (Original publicado em 1996).

Ministério da Educação (1997). *Matemática: Programas 10º, 11º e 12º Anos.* Lisboa: Autor.

Nelson, S.L. (1998). *Microsoft Frontpage 98 em Imagens* (Tradução, HCR – estudos e Consultadoria Informática). Alfragide: McGraw-Hill. (Original publicado em 1997).

Novais, M. (1997). Aprender (com) os media para com os media. In C. Freitas et al. (Eds), *Tecnologias de Informação e Comunicação na Aprendizagem* (pp.21-29). Lisboa: Instituto de Inovação Educacional.

Pais, F. (1999). *Multimédia e Ensino. Um novo paradigma*. Lisboa: Instituto de Inovação Educacional.

Papert, S. (1996). *A Família em Rede* (Tradução, F. Nunes e F. Melo). Lisboa: Relógio D'Água.

Ponte, J. P. (1994). O Professor de Matemática: um balanço de dez anos de investigação. *Quadrante. Revista teórica de Investigação, vol.3, nº2*, pp.79-115.

Ponte, J. P. (1997). *As Novas Tecnologias e a Educação*. Lisboa: Texto Editora.

Ponte, J. P. e Canavarro, A. P. (1997). *Matemática e Novas Tecnologias*. Lisboa: Universidade Aberta.

Proteste (2000). *Manual de Sobrevivência na Internet*. Lisboa: Autor.

Ramos, J.L. (1997). Da História dos Descobrimentos ao Descobrimento da História: a utilização de micromundos de aprendizagem baseados em computador nos processos de ensino – aprendizagem da História. In C. Freitas et al. (Eds), *Tecnologias de Informação e Comunicação na Aprendizagem* (pp.45-62). Lisboa: Instituto de Inovação Educacional.

Santos, J. M. (1999). *Construção de Páginas Interactivas com o JavaSketchpad*. Dissertação de mestrado não publicada, Faculdade de Ciências, Universidade do Porto, Portugal.

Serge, P., Riché-Magnier, M. (1998). *A Escola na Era da Internet* (Tradução P. Vidalinc). Paris: Éditions Nathan.

Endereços electrónicos:

http://www.livemath.com

http://www.webprimitives.com

http://www.edwardspenney.com

http://cms.dt.uh.edu/Faculty/Becerral/mvtitlepage.htm

http://www.mtsac.edu/math/mathview/plugin.htm

http://www.calculus.net

http://archives.math.utk.edu

http://www.anglia.co.uk/education/mathsnet/mathview/index.html

http://www.distancemath.com

http://otterbein.edu/home/fac/DVDLDVR/mathview/index.htm

http://math.ggu.edu/mathview.html

http://alvin.cc.tx.us/gc4m/mathview.htm

http://www.keypress.com/sketchpad/java_gsp/index.html

http://www.calculator.com

http://www.hartmath.com

http://www.univie.ac.at/future.media/moe

http://www.esr.ruhr-uni-bochum.de/VcLab/software/hoteqn/hoteqn.html

http://www.mathworks.com

http://www.mb.hs-wismar.de/mitarbeiter/pawletta/00Uwe/formel.html

http://math.hws.edu/XFunctions

http://www.sci.usq.edu.au/staff/leighb/graph

http://wwwvis.informatik.uni-stuttgart.de/~kraus/livegraphics3D/index.html

http://www.jsworld.com

http://java.sun.com

http://www.w3.org

http://htmlprimer.com

http://www.pageresources.com

http://www.softlord.com/comp

http://www.apm.pt

http://www.mct.pt

http://www.dapp.min-edu.pt/nonio/nonio.htm