Faculdade de Engenharia e Faculdade de Ciências da
Universidade do Porto

# RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES ALGÉBRICAS, EM MECÂNICA DOS FLUIDOS COMPUTACIONAL, POR MÉTODOS ITERATIVOS NÃO-ESTACIONÁRIOS

Carlos Jorge Rocha Balsa
(Licenciado em Engenharia de Minas)

Dissertação submetida para a obtenção do grau de
Mestre em Métodos Computacionais em Ciência e Engenharia
pela Universidade do Porto

Dezembro, 2000

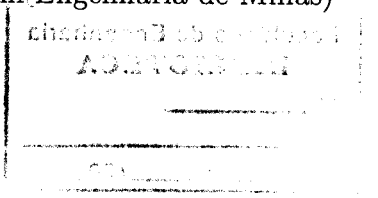
Faculdade de Engenharia e Faculdade de Ciências da

Universidade do Porto

# RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES ALGÉBRICAS, EM MECÂNICA DOS FLUIDOS COMPUTACIONAL, POR MÉTODOS ITERATIVOS NÃO-ESTACIONÁRIOS

Carlos Jorge Rocha Balsa

(Licenciado em Engenharia de Minas)



Dissertação submetida para a obtenção do grau de
Mestre em Métodos Computacionais em Ciências e Engenharia
pela Universidade do Porto

Dezembro, 2000

# Resumo

Devido ao elevado tempo de computação que normalmente exige, a resolução de sistemas de equações algébricas lineares tem uma importância vital em muitos problemas de computação científica. A escolha adequada do método de resolução pode traduzir-se numa grande melhoria do desempenho global de programas de cálculo.

Neste trabalho testa-se o desempenho computacional de diversos métodos iterativos para resolução de sistemas de equações provenientes de um problema de CFD (*Computational Fluid Dynamics*). Para esse efeito, utilizaram-se as matrizes produzidas por um algoritmo de resolução das equações do escoamento (viz SIMPLE), na simulação do escoamento de um fluido, em regime laminar, no interior de uma caixa com tampa deslizante. Estas matrizes têm dimensões de acordo com o número de pontos utilizados para discretizar o domínio ($40 \times 40$, $80 \times 80$, $120 \times 80$ e $120 \times 120$) e diziam respeito à resolução do campo de pressão e velocidade. Os sistemas derivados do cálculo das pressões tem características diferentes dos sistemas derivados da velocidade que tornam a sua resolução mais difícil.

O estudo abrangeu 10 métodos iterativos diferentes, dos quais 1 é um método estacionário (TDMA) e os restantes 9 são métodos não-estacionários, 3 dos quais de projecção oblíqua (BICG, BICGSTAB e TFQMR) e 6 de projecção ortogonal (CG, CGNR, FOM, GMRES, PGMRES e DQGMRES). Os pré-condicionadores utilizados para os sistemas das pressões foram o ILU($\ell$) e o ILUDP(`tol`,$\alpha$).

Na escolha do melhor método convém distinguir entre métodos com e sem pré-condicionamento. Os nossos resultados mostram que, no caso de não se utilizar pré-condicionamento, o método mais adequado é DQGMRES(2). Este método foi o único que conseguiu resolver todos os sistemas de teste sem pré-condicionamento, no limite máximo de iterações estabelecido. Caso se opte pela utilização de um pré-condicionador, a nossa escolha recai sobre o TFQMR associado ao pré-condicionador ILU(6) ou, em alternativa, sobre o BICGSTAB associado ao pré-condicionador ILUDP(0.01,0.75). Estes dois métodos com os respectivos pré-condicionadores convergiram em todas as situações analisadas, em tempos próximos ou iguais aos melhores.

# Abstract

The solution of systems of linear equations is a time consuming task of vital importance in scientific computing. The right choice of solver may lead to drastic reductions of computing time.

In this thesis the computational performance of a variety of iterative solvers were tested for solving the system of linear algebraic equations originated by a computer code for Computational Fluid Dynamics. Matrices from an algorithm for solving the fluid flow equations governing the laminar flow inside a cavity with a sliding lid, provided the framework for a series of tests. These matrices for solution of the pressure and velocity field varied in dimension between $40 \times 40$, $80 \times 80$, $120 \times 80$ and $120 \times 120$. The matrices for solving the pressure field required longer computing times compared with the matrices for solving the velocity field.

The study covered 10 different iterative methods, among which 1 was a stationary method (TDMA) and the remaining 9 were non-stationary, 3 of oblique projection (BICG, BICGSTAB e TFQMR) and 6 of orthogonal projection (CG, CGNR, FOM, GMRES, PGMRES e DQGMRES). ILU($\ell$) and ILUDP(`tol`,$\alpha$) were the preconditioners used.

When choosing the most appropriate method one must distinguish betheen methods with and without pré-condicioner. Ours results showed that in case of methods without preconditioner, DQGMRES(2) appeared as the most suitable because it was the only one that converged in all the cases being tested. In case of methods with preconditioner, TFQMR with ILU(6) preconditioner or BICGSTAB with ILUDP(0.01,0.75) preconditioner appeared to be superior. These two methods always reached convergence and with computing times close or identical to the best in each of the test cases.

# Résumé

La solution des systèmes d'équations lineaires dans le calcule scientifique est très importante à cause du temps élevé nécessairement utilisé. Un choix correct de la méthode à utiliser peut se traduire par une meilleur vitesse d'exécution de l'algorithme géneral.

Les testes éffectués dans ce travail ont pour objectif de comparer les résultats obtenus par différentes méthodes itératives, dans la résolution de système d'équations provenant d'un problème de CFD (*Computational Fluid Dynamics*). Les matrices utilisées sont générées par um algorithme qui résout les équations de l'écoulement (viz SIMPLE), dans la simulation d'un fluide, en régime laminaire, à l'intérieur d'une caisse avec un couvercle glissant. Ces matrices ont une dimension d'accord avec la quantité de points utilisés pour diviser le domaine en volumes finis ($40 \times 40$, $80 \times 80$, $120 \times 80$ et $120 \times 120$ éléments) et sont en relation avec le champs de vitesse et de pression. Les systèmes provenant du calcul de champs de pression ont des propriétées, opposées à des systèmes provenant du calcul de champs de vitesse, qui sont responsables des grandes difficultées de résolution.

L'étude a abordé 10 méthodes itératives différentes. L'une d'entre est stationnaire et les neufs autres sont non-stationnaires. Ces dernières se divisent en deux groupes: les méthodes de projection oblique (BICG, BICGSTAB e TFQMR) et les méthodes de projection orthogonale (CG, CGNR, FOM, GMRES, PGMRES e DQGMRES). Les pré-conditionneurs utilisés pour les systèmes en pression sont: ILU($\ell$) et ILUDP(`tol`,$\alpha$).

Le choix de la méthode à utiliser dépend de l'utilisation ou pas d'un pré-conditcionneur. Dans le cas où le pré-condicioneur n'est pas utilisé, notre choix va être le DQGMRES(2). Cette méthode est la seule qui resout tous les systèmes sans pré-conditionneur, dans la limite maximum d'itérations permit. Dans le cas ou le pré-condicioneur est utilisé notre choix est le TFQMR avec le pré-condicioneur ILU(6) ou alors le BICGSTAB avec le pré-condicioneur ILUDP(0.01,0.75). Ces deux méthodes avec les respectifs pré-condicioneurs ont résolus tout les systèmes analysés dans des temps très près ou egaux au meilleurs méthodes de résolutions.

# Agradecimentos

Estou muito grato ao Professor José Laginha Palma pela sua orientação, trabalho de revisão desta tese assim como por todos os conhecimentos transmitidos.

Agradeço ao Doutor Fernando Aristides Castro pelo seu apoio na utilização do sistema operativo LINUX, indispensável para a execução desta tese.

Agradeço ao Professor João Macedo pela sua disponibilidade e interesse no trabalho realizado nesta tese.

Este trabalho foi efectuado no âmbito do projecto PRAXIS XXI 3/3.1/CEG/2511/95, intitulado "*Algoritmos Paralelos para Mecânica dos Fluidos*".

Agradeço ao departamento de matemática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança por me ter dispensado do serviço docente ao longo do segundo semestre do ano lectivo 1999/2000.

Uma palavra de agradecimento ao meu colega Albano Alves pelo seu voluntarismo em proporcionar as melhores condições de utilização dos meios informáticos da E.S.T.G. de Bragança.

Finalmente, o meu profundo agradecimento à minha mulher e ao meu filho por terem sido tão pacientes comigo ao longo do intenso ano decorrido.

# Conteúdo

# Lista de Tabelas

# Lista de Figuras

# Nomenclatura

## Símbolos Matemáticos

| | |
|---|---|
| $\alpha, \beta, \ldots, \omega$ | escalares |
| $\delta_{ij}$ | símbolo de Kronecker |
| $\kappa_2(A)$ | número de condição espectral de $A$ |
| $\kappa_\infty(A)$ | número de condição infinito de $A$ |
| $\lambda_{\max}(A)$ | maior dos valores próprios de $A$ |
| $\lambda_{\min}(A)$ | menor dos valores próprios de $A$ |
| $\psi_i, \phi_i, \pi_i$ | polinómios de grau $i$ |
| $\rho(A)$ | raio espectral de $A$ |
| $\varrho$ | massa específica |
| $\sigma(A)$ | conjunto dos valores próprios (espectro) de $A$ |
| | |
| $A, \ldots, Z$ | matrizes |
| $A_{i,j}$ | bloco de uma matriz |
| $A^T$ | matriz transposta |
| $A^{-1}$ | matriz inversa |
| $D$ | matriz diagonal |
| $I$ | matriz identidade |
| $\mathcal{K}_m$ | sub-espaço de procura de dimensão $m$ |
| $L$ | matriz triangular inferior |
| $\mathcal{L}_m$ | sub-espaço das restrições de dimensão $m$ |
| $M$ | pré-condicionador |
| $\mathcal{R}^n$ | conjunto dos números reais |
| $Re$ | número de Reynolds do domínio |
| $Re_m$ | número de Reynolds da malha |
| $T$ | matriz triangular |
| $U$ | matriz triangular superior |
| $V_m, W_m$ | matrizes de dimensões $n \times m$ |
| $a, \ldots, z$ | vectores |
| $a_{i,j}$ | elemento de uma matriz |
| $e_1, \ldots, e_m$ | base canónica |
| $h$ | afastamento dos nós da malha |
| $p$ | pressão |
| $r^{(i)}$ | vector resíduo à i-ésima iteração |
| $u$ | velocidade horizontal |
| $u_m$ | velocidade horizontal na malha |
| $u_{tampa}$ | velocidade horizontal na tampa |
| $v$ | velocidade vertical |
| $v_m$ | vector coluna $m$ de $V_m$ |
| $x$ | solução exacta do sistema $Ax = b$ |
| $x_*$ | solução exacta do sistema dual $A^T x_* = b_*$ |
| $x^{(i)}$ | vector solução à i-ésima iteração |

| | |
|---|---|
| $\|A\|_1$ | norma-1 da matriz $A$ |
| $\|A\|_2$ | norma-2 da matriz $A$ |
| $\|A\|_\infty$ | norma infinita da matriz $A$ |
| $\|A\|_F$ | norma Frobenius da matriz $A$ |
| $\|b\|_2$ | norma euclidiana do vector $b$ |
| $\det(A)$ | determinante de $A$ |
| $\partial/\partial$ | derivada parcial |
| $(x, y), x^T y$ | produto interno de dois vectores |

**Programação**

| | |
|---|---|
| `LCD` | número de colunas diagonalmente dominantes |
| `LRD` | número de linhas diagonalmente dominantes |
| `N` | número de pontos interiores ao domínio |
| `NNZ` | número de elementos nulos |
| `atol` | tolerância absoluta |
| `lfill` | nível de enchimento |
| `nbloc` | número limite de colunas a permutar |
| `ni`×`nj` | dimensões da malha discretizante |
| `ni`$p$`nj` | sistema da pressão num domínio com `ni`×`nj` nós |
| `ni`$u$`nj` | sistema da velocidade num domínio com `ni`×`nj` nós |
| `permtol` | tolerância de permutação de colunas |
| `rtol` | tolerância relativa |
| `tol` | tolerância |

**Estruturas de Dados**

| | |
|---|---|
| BND | *linpack Banded format* |
| CSC | *Compressed Sparse Column format* |
| CSR | *Compressed Sparse Row format* |
| DNS | *Dense format* |
| RGS | *Regular Grid Stencil storage format* |

**Bibliotecas**

| | |
|---|---|
| LAPACK | *Linear Algebra PacKage* |
| PSPARSLIB | *portable Library of Parallel Sparse iterative solvers* |
| SPARSKIT | *basic tool Kit for Sparse matrix computation* |

**Métodos Iterativos**

| | |
|---|---|
| BICG | *BiConjugate Gradient* |
| BICGSTAB | *BiConjugate Gradient Stabilized* |
| CG | *Conjugate Gradient* |
| CGNR | *Conjugate Gradient Normal Residual* |
| CGNE | *Conjugate Gradient Normal Error* |

| | |
|---|---|
| CGS | *Conjugate Gradient-Squared* |
| CR | *Conjugate Residuals* |
| DQGMRES | *Direct Quasi-GMRES* |
| FOM | *Full Ortogonalization Method* |
| GMRES | *Generalized Minimal Residual* |
| PGMRES | *Precondicioned Generalized Minimal Residual* |
| QMR | *Quasi-Minimal Residual* |
| SOR | *Successive Over-Relaxation* |
| SSOR | *Symmetric Successive Over-Relaxation* |
| TDMA | *Tri-Diagonal Matrix Algorithm* |
| TFQMR | *Tanspose-Free Quasi-Minimal Residual* |

**Pré-Condicionadores**

| | |
|---|---|
| ILU | factorização Incompleta LU |
| ILUD | ILU com compensação Diagonal |
| ILUDP | ILUD com Pivotagem das colunas |
| ILUT | ILU por Truncatura |
| ILUTP | ILUT com Pivotagem das colunas |
| MILU | ILU Modificada |

**Outras Abreviaturas**

| | |
|---|---|
| CFD | *Computational Fluid Dynamics* |
| PDEs | *Partial Differential Equations* |
| SIMPLE | *Semi Implicit Pressure Linked Equations* |

# Capítulo 1

# Introdução

Grande parte dos sistemas de equações algébricas, resolvidos em computação científica, decorrem da resolução numérica de equações diferenciais às diferenças parciais (PDEs), (Dongarra and Eijkhout, 1999). As técnicas utilizadas, que consistem em discretizar o domínio através do método das diferenças finitas, elementos finitos, etc, originam um conjunto de equações que relacionam os pontos pertencentes ao domínio. No caso de um domínio bidimensional, a resolução da equação diferencial que descreve o comportamento de uma variável física, através do método das diferenças finitas centrais, conduz a uma equação para cada ponto, interior ao domínio, que relaciona este com os quatro pontos vizinhos (figura 1.1).

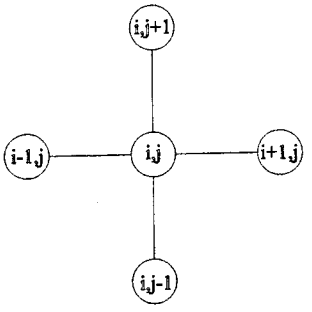



Figura 1.1: Pontos vizinhos de um domínio plano

A equação para o ponto genérico de coordenadas $i, j$ ilustrado na figura 1.1, é

$$a_{i,j}x_{i,j} + a_{i+1,j}x_{i+1,j} + a_{i-1,j}x_{i-1,j} + a_{i,j+1}x_{i,j+1} + a_{i,j-1}x_{i,j-1} = s_{i,j}. \tag{1.1}$$

Em que por exemplo $x_{i,j+1}$ representa a variável (pressão, velocidade, etc.) no ponto de coordenadas $i, j+1$ e $a_{i,j+1}$ o coeficiente relativo à contribuição desse ponto em relação ao ponto de coordenadas $i, j$; o termo $s_{i,j}$ é o termo independente ou termo fonte. Os pontos do domínio resultantes da discretização são também designados por nós da malha.

As condições de fronteira são definidas sobre os pontos situados no limite com o exterior do domínio.

Quanto mais densa for a malha mais informações serão obtidas sobre a distribuição da variável no interior do domínio. No caso tridimensional são incluídos dois novos termos na equação (1.1); um relativo à contribuição do nó superior e outro relativo ao nó inferior.

Escrevendo o conjunto das equações relativas a todos os nós do domínio obtemos um sistema de equações algébricas lineares

$$Ax = b. \tag{1.2}$$

A matriz dos coeficientes $A$ possui apenas um pequeno número de termos não nulos (esparsa), situados sobre algumas diagonais. Por este motivo é também designada por matriz banda. No caso de um domínio uni-dimensional existem apenas três diagonais com termos não-nulos, no caso bidimensional, como o aqui abordado e ilustrado na figura 1.2, existem cinco e no caso tridimensional existem sete diagonais.

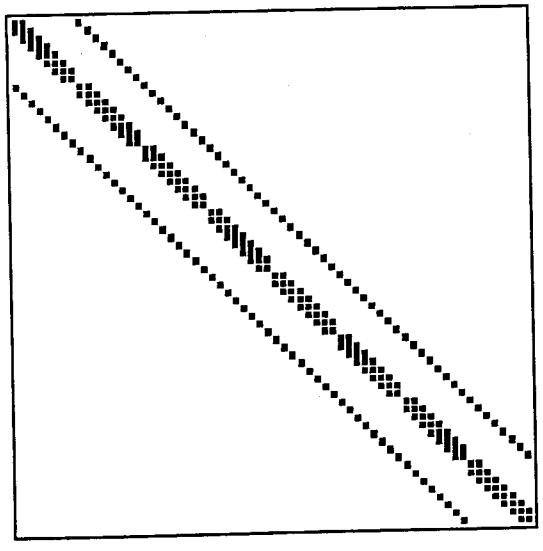


Figura 1.2: Distribuição padrão dos termos não-nulos de matrizes provenientes da discretização de PDEs num domínio plano, através de uma malha regular.

A necessidade de resolver sistemas de grande dimensão e esparsos, associada aos grandes avanços tecnológicos, é responsável pela explosão verificada no campo dos métodos iterativos, nos últimos cinquenta anos (Saad and Vorst, 1999). A opção por métodos iterativos, na resolução de sistema esparsos, em detrimento dos métodos directos prende-se com os requisitos de memória necessários. Os métodos directos requerem que todos os coeficientes

da matriz $A$ sejam guardados, enquanto que os iterativos podem ser aplicados guardando apenas os termos não nulos. Por este motivo as dimensões do problema impossibilitam por vezes a utilização de métodos directos (Dongarra and Eijkhout, 1999). Os métodos iterativos dividem-se ainda em duas grandes classes: estacionários e não-estacionários, sendo geralmente os não-estacionários mais potentes do que os estacionários (ver capítulo 2).

## 1.1 Objectivos e contribuição do presente trabalho

A resolução de sistemas de equações algébricas lineares tem uma importância vital em muitos problemas de computação científica, porque consome grande parte do tempo de execução dos algoritmos. A escolha adequada do método de resolução pode melhorar substancialmente o desempenho global do programa de cálculo.

Pretende-se com este trabalho testar o desempenho computacional de diversos métodos iterativos não-estacionários na resolução de sistemas de equações provenientes de um problema de CFD (*Computational Fluid Dynamics*). Para este efeito, usou-se como caso teste a simulação do escoamento de um fluido, em regime laminar, no interior de uma caixa com tampa deslizante, utilizando o algoritmo SIMPLE (*Semi Implicit Pressure Linked Equation*) (Patankar and Spalding, 1972). O módulo relativo à resolução de sistemas de equações (*solver*), utilizado por este programa, consiste numa implementação sequencial do método iterativo designado por TDMA (ver secção 2.2). Neste trabalho testamos o desempenho de outros métodos iterativos alternativos para a resolução destas equações, com o objectivo de identificar qual ou quais aqueles com menor tempo de execução. Essa tarefa é de particular importância, se tivermos em conta que no mesmo problema o *solver* (TDMA) ocupou 60% do tempo de execução do algoritmo SIMPLE (Areal, 1999, pág. 2).

A escolha do método iterativo mais adequado para utilização em programas de simulação de CFD é uma tarefa dificultada por factores diversos. Por exemplo, alguns dos métodos foram concebidos e funcionam apenas para matrizes cujo formato é diferente daquele que se encontra nos programas de CFD. Acresce a isto ainda, o facto das matrizes de um determinado programa de CFD dependerem da geometria e das condições de fronteira do problema que se pretende simular. Por estas razões, a escolha do método iterativo mais adequado passa pela experimentação numérica: a via seguida neste trabalho.

## 1.2 Origem dos sistemas estudados

Os sistemas lineares que servem de teste para o estudo dos *solvers*, efectuado neste trabalho, foram retirados à décima iteração de um programa de cálculo baseado no algoritmo SIMPLE. Em cada iteração deste algoritmo são resolvidos três sistemas de equações lineares distintos. Um relativo ao campo da pressão e dois outros relativos aos campos da velocidade horizontal e vertical, dum fluido no interior do domínio (caixa). Neste trabalho consideramos apenas os sistemas da pressão e da velocidade horizontal.

A geometria do domínio consiste num quadrado cujos lados medem 1 metro de comprimento. Utilizaram-se malhas regulares com os seguintes números de nós: $40 \times 40$, $80 \times 80$, $120 \times 80$ e $120 \times 120$. O programa de simulação efectua a discretização do domínio em todos os nós da malha, utilizando um esquema híbrido: a montante se $Re_m > 2$ e a jusante se $Re_m < 2$. O parâmetro $Re$ representa o número de Reynolds da malha e é dado por $Re_m = \varrho u_m h/\mu$, sendo $\varrho$ a massa específica ($\varrho = 1$), $\mu$ a viscosidade dinâmica ($\mu = 0.01$), $u_m$ a velocidade horizontal na malha e $h$ o afastamento entre nós da malha. O número de Reynolds do domínio, considerado na simulação, é $Re = 100$. As condições iniciais consideram a velocidade vertical $v$ e a pressão $p$ nulas em todo o domínio, enquanto que a velocidade horizontal $u$ é dada por $u = u_{tampa} = 1m/s$.

Com o objectivo de facilitar a designação destes sistemas será utilizada a seguinte nomenclatura, daqui para a frente: $nixnj$, em que $ni$ designa o número de pontos na direcção horizontal e $nj$ na direcção vertical e $x$ designa a variável física. Utilizaremos a letra $p$ para a pressão e $u$ para a velocidade horizontal. Por exemplo o sistema das pressões num domínio discretizado em $80 \times 80$ pontos será designado por $80p80$.

## 1.3 Breve revisão bibliográfica

Os métodos iterativos são em grande número e muito deles foram desenvolvidos nos últimos XX anos. Uma perspectiva da evolução dos métodos iterativos ao longo do século vinte poderá ser encontrada em (Saad and Vorst, 1999). Em língua portuguesa, dos poucos manuais que abordam este tema, destacamos (Pina, 1995) onde são abordados alguns métodos iterativos não-estacionários da família do Gradiente Conjugado, não incluindo contudo métodos como por exemplo o GMRES ou o BICG (ver secção 2.3).

Efectuou-se uma pesquisa bibliográfica às principais bases de dados disponibilizadas pela FEUP. Estas são em grande número e abrangem vários ramos do conhecimento relacionados com as ciências e engenharias (*Dialog, Ei Compendex, Pascal, INSPEC*, etc.). Os resultados apresentados na tabela 1.1 foram obtidos na base de dados *INSPEC*.

| Palavra Chave | Registos |
|---|---|
| *Parallel Computation* | 3372 |
| *Iterative Methods* | 26868 |
| *Iterative Methods* and *Parallel Computation* | 117 |
| *Iterative Methods* and *Parallel Solvers* | 29 |
| *Sparse Algebra* | 28 |
| *Iterative Stationary Methods* | 2 |
| *Pre-Conditioner* | 6 |
| *Iterative Non-Stationary Methods* | 1 |
| *Iterative Stationary Methods* | 17 |

Tabela 1.1: Resultados da pesquisa bibliográfica efectuada na base de dados INSPECC

A tabela 1.1 mostra o número de registos encontrados em função de 9 palavras-chave relacionadas com o presente trabalho. As três, especialmente as duas primeiras, por serem demasiado gerais conduzem a um número exageradamente elevado de registos, tornando impossível a sua consulta. O agrupamento das palavras-chave *Iterative Methods* e *Parallel Computation* permitiu concluir que dos 26868 registos sob a palavra *Iterative Methods*, apenas 117 estão relacionados com computação paralela. Contudo, como no âmbito do presente trabalho apenas os métodos de resolução de sistemas de equações algébricas interessam, especificou-se ainda mais a procura com as palavras-chaves *Iterative Methods* e *Parallel Solvers*. Os vinte e nove artigos registados abrangem todos eles a resolução de sistemas em vários contextos. O método dos elementos finitos é referido em quatro artigos, a análise e descrição de software standard são referidas em sete artigos, os pré-condicionadores em quatro, sendo os restantes artigos relacionados com a análise de métodos na resolução de problemas específicos.

Destacamos o artigo (Amodio and Mazzia, 1997) que se refere à aplicação do método iterativo designado por SOR (ver secção 2.2), paralelizado através da técnica de decomposição do domínio. Noutro artigo (Cela et al., 1995) é efectuado um estudo comparativo de *solvers*, paralelizados através da mesma técnica, na resolução de sistemas bandeados por blocos.

A pesquisa realizada com a palavra chave *Sparse Algebra* resultou essencialmente em artigos relacionados com *software* destinado a problemas com matrizes esparsas, inclusive a resolução de sistemas. Destes destacamos o artigo (Krommer et al., 1997), onde é apresentada a biblioteca NAG, constituída por *solvers* paralelizados, destinados à resolução de sistemas densos e esparsos, e o artigo (Eijkhout, 1998) onde se efectua uma descrição e uma análise das principais bibliotecas de *solvers* sequenciais e paralelos.

Para a elaboração desta tese foram utilizados essencialmente quatro livros. Todos eles abordam o tema dos métodos iterativos na resolução de sistemas de equações algébricas, derivados de problemas de engenharia ou computação científica. Em *Iterative Methods for Sparse Linear Systems* (Saad, 1996) é efectuada uma descrição muito completa dos métodos iterativos principais sob o ponto de vista matemático, incluindo também capítulos com temas directamente relacionados, como a discretização de equações diferenciais e o pré-condicionamento. O livro *Numerical Linear Algebra for High-Performance Computers* (Dongarra et al., 1998) aborda as várias questões relacionadas com a resolução de sistemas lineares em computadores vectorizados e paralelos. O livro *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (Barrett et al., 1994) faz uma descrição dos principais métodos iterativos e dos aspectos relacionados com a sua implementação na biblioteca *Templates*. Finalmente, *Matrix Computation* (Golub and Loan, 1989) é um manual de computação científica com matrizes que contém as bases matemáticas e os algoritmos necessários para o desenvolvimento de *software* numérico.

A utilização destes 4 livros como as fontes de conhecimento essenciais para a realização do nosso trabalho resultou da análise de um conjunto alargado de outras obras (incluindo também artigos científicos) de cuja listagem nos dispensamos aqui. Uma obra fundamental que seria de juntar às quatro identificadas acima seria o livro de (Greenbaum, 1997), intitulado *Iterative Methods for Solving Linear Systems*; no entanto, só tivemos acesso a

esta obra depois de concluído o trabalho e a redacção da presente tese.

## 1.4 Considerações gerais sobre a resolução de sistemas esparsos em CFD

### 1.4.1 Estruturas de dados para sistemas esparsos

A estrutura particular da matriz $A$, figura 1.2, pode ser aproveitada de forma a tornar a resolução mais eficiente, guardando em memória apenas os termos não nulos. Existem vários esquemas de armazenamento de dados para matrizes esparsas. Todos eles consistem em guardar apenas os termos não nulos juntamente com alguma informação adicional que permita localizar as suas posições na matriz $A$. No problema em estudo o método utilizado é designado por *Regular Grid Stencil storage* (RGS) (Eijkhout, 1998) que consiste em guardar cada diagonal de $A$, com termos não nulos, numa matriz diferente. Para um domínio bidimensional, os dados da matriz $A$ são guardados em cinco matrizes com as dimensões do domínio, onde a indexação utilizada corresponde à posição do nó no domínio físico. Este esquema é adequado para sistemas em que os termos não nulos estão situados apenas em algumas diagonais bem definidas como o exemplo ilustrado na figura 1.2. Contudo para problemas de outras áreas como por exemplo a Engenharia Química, Simulação de Circuitos, etc., em que o padrão de distribuição dos termos não nulos é completamente diferente, a sua utilização é inviável.

Existem vários esquemas alternativos. Um dos mais conhecidos é o *Compressed Sparse Row Format* (CSR), consiste em guardar os dados em três vectores. O primeiro contém os termos $a_{i,j}$ não nulos, o segundo os índices das colunas dos $a_{i,j}$ e o terceiro vector contém apontadores para o início de cada linha no primeiro vector. Outro esquema de estrutura de dados, também muito utilizado, é o *Banded format* (BND), em que todos os termos situados sobre uma banda compreendida entre duas diagonais são guardados. Para além destes dois esquemas existem outros que não são utilizados neste trabalho e cuja descrição poderá ser encontrada em (Barrett et al., 1994). Todos estes formatos têm a vantagem de exigirem menos esforço computacional do que o formato denso (DNS), em que todas as entradas de $A$ são memorizadas, mas também implicam que os métodos de resolução sejam implementados de acordo com essa estrutura de dados. Este aspecto dificulta o desenvolvimento de *solvers standard* para todos os problemas esparsos.

### 1.4.2 Bibliotecas de *software* para sistemas esparsos

Existe um conjunto de bibliotecas ou *packages* de rotinas implementadas em linguagens de programação como o FORTRAN ou C, para a resolução de sistemas de equações algébricas. Neste trabalho optámos por não escrever as rotinas com os *solvers*, mas sim utilizar bibliotecas do domínio público e adaptá-las ao nosso problema. Pretende-se desta forma que o estudo comparativo dependa o mínimo possível da implementação numa determinada

linguagem de programação, uma vez que estas rotinas já foram testadas e optimizadas. A descrição e a comparação de algumas das principais bibliotecas de *solvers*, baseados em métodos iterativos, podem ser encontradas em (Eijkhout, 1998).

A biblioteca de *solvers* utilizada para este trabalho foi a *Sparskit* (Saad, 1990). Esta disponibiliza rotinas de álgebra linear implementadas em FORTRAN77 para sistemas esparsos, incluindo um conjunto de *solvers* implementáveis de acordo com o *Reverse Comunication Interface* (Dongarra et al., 1995), cuja definição é apresentada abaixo. A razão principal para a utilização desta biblioteca prende-se com o facto de ser complementada por uma outra chamada *Psparslib* (Saad and Malevsky, 1995) que contém os mesmos *solvers* implementados em paralelo. Esta segunda biblioteca poderá ser utilizada em trabalhos futuros de forma a comparar os resultados com os deste trabalho. A *Sparskit* contém igualmente algumas rotinas de análise, utilizadas neste trabalho para caracterizar os sistemas, e pré-condicionadores derivados da factorização incompleta LU (ver secção 2.5). Todas estas rotinas estão desenvolvidas para aceitar os dados estruturados na forma CSR. A *Psparslib* contém apenas *solvers* e pré-condicionadores para plataformas paralelas.

Para além da *Sparskit* utilizamos também algumas rotinas da biblioteca *Lapack* (Anderson et al., 1992) para análise dos sistemas com estimativa dos valores próprios das matrizes. A biblioteca *Lapack* contém uma colecção completa de programas de álgebra linear escritos em FORTRAN77 mas que apenas aceitam dados na forma densa (matriz completa, inclui os zeros) e BND. Embora a estrutura BND permita uma grande economia de memória em relação à densa, obriga a guardar todos os elementos nulos situados entre as duas diagonais mais afastadas da principal. Este aspecto penaliza os *solvers* da *Lapack* relativamente a outras bibliotecas que utilizam estruturas de dados mais compactas e consequentemente mais eficientes na resolução de sistemas esparsos.

No sentido de ultrapassar as dificuldades inerentes às várias estruturas de dados possíveis, na implementação dos métodos de resolução em FORTRAN, foi criado um estilo de programação designado por *Reverse Comunication Interface*, (Saad, 1996, pág. 333). Os *solvers* implementados de acordo com este estilo não dependem da estrutura de dados. Sempre que o programa principal necessita de efectuar uma operação que implica ordenação dos dados, como por exemplo o produto interno, multiplicação matriz-vector, etc., é chamada uma rotina externa para o efeito. Essas rotinas são fornecidas pelo utilizador e podem ser desenvolvidas de acordo com qualquer estrutura de dados.

## 1.5 Parâmetros caracterizadores de sistemas esparsos

Descreve-se de seguida alguns dos parâmetros e propriedades mais importantes utilizados na caracterização de sistemas de equações algébricas esparsos. Estas características são utilizadas na descrição sumária dos principais métodos iterativos, efectuada no Capítulo 2, e na caracterização dos sistemas de teste apresentada na secção 3.2.

### 1.5.1 Dimensão, elementos não nulos e simetria

A dimensão do sistema depende das dimensões da malha utilizada. Por exemplo, se esta for constituída por `ni` × `nj` pontos, a dimensão do sistema será `N = (ni-2) × (nj-2)`, porque os pontos extremos contêm as condições de fronteira. O número de elementos não-nulos na matriz $A$ é definido neste trabalho pelo parâmetro `NNZ`. A simetria refere-se ao facto de $A = A^T$. As rotinas utilizadas para obter os valores destes parâmetros foram retiradas da biblioteca *SparsKit*.

### 1.5.2 Normas e número de condição

As normas mais utilizadas para a caracterização de matrizes são as seguintes:

1. Norma-1:
$$\|A\|_1 = \max_{j=1,\ldots,m} \sum_{i=1}^{n} |a_{ij}| \,. \tag{1.3}$$

2. Norma infinita:
$$\|A\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{m} |a_{ij}| \,. \tag{1.4}$$

3. Norma Frobenius:
$$\|A\|_F = \left( \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij}^2 \right)^{1/2} . \tag{1.5}$$

O número de condição é definido em função de uma das normas. Por exemplo o número de condição baseado na norma infinita é definido pela expressão

$$\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty. \tag{1.6}$$

Este parâmetro é uma medida da sensibilidade aos cálculos em aritmética discreta. Se $\kappa(A)$ estiver próximo de 1 significa que o sistema está bem condicionado e que pouco ou nada será afectado por pequenas perturbações em $A$. Caso contrário, se $\kappa(A)$ for muito elevado, o sistema é mal-condicionado e é muito sensível a pequenas perturbações nos coeficientes de $A$ (Golub and Loan, 1989, pág. 79). Na prática o cálculo de $A^{-1}$ é demasiado dispendioso, i.e., implica demasiadas operações, pelo que é apenas feita uma estimativa do número de condição. A rotina que utilizámos, extraída da biblioteca *Lapack*, baseia-se na exploração da relação:

$$Ay = d \quad \Longrightarrow \quad \|A^{-1}\|_\infty \geq \|y\|_\infty / \|d\|_\infty.$$

A ideia consiste em escolher $d$ de forma a que a solução $y$ seja elevada em norma, fazendo com que $\|y\|_\infty / \|d\|_\infty$ esteja próximo do seu valor máximo $\|A^{-1}\|_\infty$. O estimador para o número de condição é então dado por

$$\hat{\kappa}_\infty(A) = \|A\|_\infty \|y\|_\infty / \|d\|_\infty. \tag{1.7}$$

Uma descrição completa desta técnica encontra-se em (Golub and Loan, 1989, pág. 128).

### 1.5.3 Colunas e linhas diagonalmente dominantes

Uma linha é diagonalmente dominante se

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ i \neq j}}^{j=n} |a_{ij}| . \tag{1.8}$$

Por sua vez, uma coluna é diagonalmente dominante se

$$|a_{jj}| \geq \sum_{\substack{i=1 \\ j \neq i}}^{i=n} |a_{ij}| . \tag{1.9}$$

Os parâmetros $LRD$ e $LCD$, utilizados na secção 3.2, referem-se, respectivamente, à percentagem de linhas e de colunas diagonalmente dominantes. Quando $LCD = 100\%$ a matriz $A$ é diagonalmente dominante (Saad, 1996, pág. 108). Esta propriedade é muito importante pois em operações como a factorização $LU$, as matrizes diagonalmente dominantes não necessitam de pivotagem.

As matrizes definidas positivas, embora não sejam diagonalmente dominantes, admitem factorização $LU$ sem necessidade de recorrer à pivotagem (Golub and Loan, 1989, pág. 139). Uma matriz é definida positiva se $x^T Ax > 0$ para todo o $x$ não nulo. Em termos práticos significa que a maior entrada de $A$ está sobre a diagonal principal e é positiva, i.e., a diagonal principal tem um peso superior às restantes. Esta propriedade garante também que a matriz $A$ não é singular, i.e., $\det(A) \neq 0$. Quando $x^T Ax \geq 0$, para todo o $x$ não nulo, a matriz $A$ é definida semi-positiva.

### 1.5.4 Maior elemento de $A$ e norma-2 do vector $b$

A título comparativo incluímos na secção 3.2 dois parâmetros adicionais, para caracterização dos sistemas de teste. O maior elemento de $A$ em valor absoluto ($\max|a_{i,j}|$) é utilizado como um indicador da ordem de grandeza das entradas da matriz $A$. A norma-2 de $b$

$$\|b\|_2 = \sqrt{\sum_{i=1}^{n} b_i^2}, \tag{1.10}$$

ou norma euclidiana, indica o resíduo inicial, i.e., a distância entre a primeira aproximação à solução $x^{(0)}$ e a solução exacta $x$, considerando o vector nulo $x^{(0)} = 0$ como a aproximação inicial à solução (ver secção 3.1.3).

### 1.5.5 Análise espectral

O conjunto dos valores próprios de uma matriz $A$ é designado por espectro de $A$ e é denotado por $\sigma(A)$. O maior dos valores próprios de $A$, em valor absoluto, é designado por

raio espectral, sendo representado por $\rho(A)$:

$$\rho(A) = \max\{|\lambda| : \lambda \in \sigma(A)\}. \tag{1.11}$$

Os valores próprios de uma matriz real são todos reais se essa matriz for simétrica. Se a matriz não for simétrica os valores próprios poderão ser complexos, ocorrendo então aos pares conjugados (Pina, 1995, pág. 139).

O espectro da matriz dos coeficientes de um sistema de equações lineares revela algumas propriedades úteis para o estudo da facilidade com que o sistema poderá ser resolvido. Um dos aspectos consiste na relação existente entre normas e a distribuição dos valores próprios no plano complexo. O raio espectral é limitado superiormente por qualquer norma:

$$\rho(A) \leq \|A\|.$$

Em particular se $A$ for simétrica $\|A\|_2 = \rho(A)$. A norma Frobenius por sua vez também está relacionada com os valores próprios através da seguinte relação:

$$\|A\|_F^2 = |\lambda_1|^2 + |\lambda_2|^2 + \ldots + |\lambda_n|^2. \tag{1.12}$$

Outro aspecto importante revelado pelos valores próprios refere-se ao número de condição $\kappa_2$, designado também por número de condição espectral, cálculado através da relação

$$\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \frac{\sigma_1(A)}{\sigma_n(A)}, \tag{1.13}$$

sendo $\sigma_1(A) = \rho(A)$ e $\sigma_n(A)$ o menor dos valores próprios em valor absoluto (Golub and Loan, 1989, pág. 80). Esta expressão mostra que o condicionamento da matriz depende da dispersão dos valores próprios. Quanto menor for a dispersão no plano complexo, menor será o afastamento entre $\sigma_1(A)$ e $\sigma_n(A)$ e consequentemente a razão entre estes dois parâmetros se aproximará de 1. Outro aspecto importante evidenciado pela equação (1.13) consite no facto do número de condição $\kappa_2(A)$ ser também um indicador da singularidade da matriz $A$. Este poderá ser muito elevado se $\sigma_n(A)$ for muito pequeno, i.e., muito próximo de 0. No caso de tal se verificar o determinante da matriz, calculado pelo produto de todos os valores próprios de $A$, aproximar-se-á também de 0. Quanto mais próxima da singularidade estiver a matriz $A$ mais difícil será a resolução do sistema correspondente por este se aproximar da indeterminação ou da inexistência de solução.

Os valores próprios indicam também se uma matriz é definida positiva ou não. Uma matriz simétrica $A$ é definida semi-positiva se todos os seus valores próprios forem não-negativos e é definida positiva se todos os seus valores próprios forem positivos (Pina, 1995, pág. 293). Também para as matrizes não-simétricas é importante que o conjunto dos valores próprios estejam distribuídos sobre a parte do plano complexo correspondente ao eixo real positivo. Se tal acontecer haverá uma maior facilidade de resolução através de métodos iterativos (Dongarra et al., 1998, pág. 191).

Na secção 3.2 são apresentados os resultados do cálculo dos valores próprios de todos os sistemas de teste. Os valores próprios foram calculados através de rotinas da biblioteca

*Lapack*. Estas rotinas fazem estimativas dos valores próprios através de métodos iterativos. A ideia geral subjacente a estes métodos consiste em reduzir a matriz dos coeficientes $A$ a uma matriz triangular por blocos através de transformações semelhantes (Golub and Loan, 1989, pág. 334).

# Capítulo 2

# Métodos Iterativos Principais

Neste capítulo descrevem-se de forma breve os métodos iterativos mais utilizados na resolução de sistemas de equações algébricas lineares e faz-se também uma pequena descrição das principais técnicas de pré-condicionamento utilizadas na resolução desses sistemas. Este capítulo baseia-se essencialmente em (Saad, 1996) e (Dongarra et al., 1998). A tradução para a língua portuguesa do nome da maior parte dos métodos é da nossa responsabilidade. A razão prende-se com o facto da literatura sobre o assunto, publicada em Portugal, referir apenas um pequeno número de métodos iterativos.

O capítulo está dividido em seis secções. Na secção 2.1 incluem-se as ideias principais sobre as quais se baseiam os métodos iterativos, na secção 2.2 apresentam-se alguns dos métodos estacionários, na secção 2.3 os não-estacionários derivados de processos de projecção ortogonal, na secção 2.4 os métodos derivados de projecções oblíquas e na secção 2.5 faz-se uma descrição das técnicas principais de pré-condicionamento baseadas na factorização incompleta LU. O capítulo é concluído na secção 2.5, com um resumo das ideias principais apresentadas.

## 2.1 Princípios básicos

Os métodos iterativos para resolução de sistemas de equações lineares

$$Ax = b$$

podem ser divididos em dois grupos: estacionários e não-estacionários. Esta designação deriva do método utilizar ou não uma matriz de iteração constante.

Nos métodos estacionários, a aproximação à solução é efectuada através da fórmula recorrente

$$x^{(k)} = M^{-1}Nx^{(k-1)} + M^{-1}b, \tag{2.1}$$

onde $M^{-1}N$ é a matriz de iteração e $M^{-1}b$ um vector. As matrizes $M$ e $N$ são constantes e resultam da decomposição $A = M - N$. Os métodos deste tipo (Jacobi, Gauss-Seidel,

SOR , etc.) diferem entre eles pela maneira como é efectuada a decomposição $A = M - N$. Em (Saad, 1996, pág. 104-108) são apresentadas as seguintes considerações gerais sobre a convergência dos métodos deste tipo:

- Sempre que o raio espectral da matriz de iteração $G = (M^{-1}N)$ for menor do que um, $\rho(G) < 1$, e que a matriz $I - G$ não seja singular, a equação (2.1) converge para a solução, qualquer que seja a aproximação inicial $x^{(0)}$. Como $\rho(G) < \|G\|$, para qualquer norma $\|.\|$., uma condição suficiente para a convergência é que $\|G\| < 1$, com $I - G$ não-singular.
- Se $M$ e $N$ resultarem de uma decomposição regular de $A$, então $\rho(M^{-1}N) < 1$ se e só se $A$ não for singular e $A^{-1}$ não for negativa. Entende-se por matriz não-negativa uma matriz cujas entradas são todas maiores ou iguais a zero. Uma decomposição regular de uma matriz $A$ é da forma $A = M - N$, com $A$ não singular e $M^{-1}$ e $N$ matrizes não-negativas.

Os métodos estacionários são de implementação simples, contudo são limitados em termos de eficiência. Os métodos não-estacionários procuram melhorar a aproximação à solução de forma dinâmica, em cada iteração. Muitos dos métodos não-estacionários utilizam técnicas de projecção para aproximar a solução, onde a aproximação $x^{(k)}$ é extraída de um sub-espaço $\mathcal{K} \in \mathcal{R}^n$, designado por sub-espaço de procura. Se este for de dimensão $m$, são normalmente impostas $m$ restrições, independentes, de ortogonalidade ao resíduo $r^{(k)} = b - Ax^{(k)}$. Pelo que os $m$ vectores restrição, linearmente independentes, definem um outro sub-espaço, designado por sub-espaço das restrições $\mathcal{L} \in \mathcal{R}^n$. Este esquema é conhecido por condições de *Petrov-Galerkin* (Saad, 1996).

Os métodos não-estacionários baseados nesta aproximação são designados métodos do sub-espaço de Krylov e dividem-se em duas classes: métodos baseados em projecções ortogonais ($\mathcal{K} = \mathcal{L}$) e métodos baseados em projecções oblíquas ($\mathcal{K} \neq \mathcal{L}$). O sub-espaço de procura é

$$\mathcal{K}_m(A, r^{(0)}) \equiv \{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \ldots, A^{m-1}r^{(0)}\}. \tag{2.2}$$

Em que $\mathcal{K}_m(A, r^{(0)})$ representa o sub-espaço de Krylov de dimensão $m$, gerado por $A$ e $r^{(0)}$. Estes métodos consistem em definir uma aproximação à solução da forma $x^{(k)} = V_k y$. Sendo $V_k$ a matriz cujos vectores coluna constituem a base do sub-espaço de Krylov e $y$ o vector dos coeficientes da combinação linear dessa base.

Se $W_k$ for a matriz que contém a base do sub-espaço $\mathcal{L}_k$, a restrição $r^{(k)} \perp \mathcal{L}_k$ é equivalente a $W_k^T(b - AV_k y) = 0$ ou a

$$W_k{}^T A V_k y = W_k{}^T b. \tag{2.3}$$

A equação (2.3) representa o sistema auxiliar, resolvido pelos métodos do sub-espaço de Krylov, baseados na aproximação de *Petrov-Galerkin*. Os métodos deste tipo como o BICG, CGS ou TFQMR possibilitam a resolução de sistemas não-simétricos.

Quando $W = V$ estamos na situação da projecção ortogonal, também designada por aproximação de *Ritz-Galerkin*. Os métodos deste género, como o Lanczos ou CG foram

desenvolvidos para resolver $V_k{}^T A V_k y = V_k{}^T b$, tirando partido da estrutura simétrica da matriz $A$.

Uma variante destes métodos baseia-se na ideia de procura $x^{(k)} = V_k y$, de forma a minimizar o resíduo $\|b - AV_k y\|_2$. Este processo conduz à determinação de $y$ no sentido dos mínimos quadrados, pelo que também é conhecido por aproximação dos mínimos quadrados. São métodos deste tipo o GMRES e o MINRES.

Os métodos iterativos apesar de poderosos nem sempre conseguem resolver o sistema pretendido. As razões principais são a exigência de recursos computacionais, que podem não estar disponíveis, ou o sistema de equações ser muito mal condicionado, exigindo demasiadas iterações. A solução passa por transformar o sistema num outro, com a mesma solução, mas melhor condicionado, i.e., mais facilmente resolúvel. O sistema pré-condicionado será

$$M^{-1}Ax = M^{-1}b. \tag{2.4}$$

A matriz pré-condicionadora $M$ deverá ser, num determinado sentido, uma boa aproximação de A, ter um custo de construção não impeditivo e, finalmente, conduzir a um sistema $My = c$ que seja mais facilmente resolvido do que o original (Dongarra et al., 1998, pág. 195). Para além do pré-condicionamento à esquerda, ilustrado na equação (2.4), o sistema pode ser condicionado à direita $AM^{-1}y = b$, com $x = M^{-1}y$, ou de ambos os lados $M_1{}^{-1}AM_2{}^{-1}z = M_1{}^{-1}b$, com $M = M_1M_2$ e $x = M_2{}^{-1}z$.

## 2.2 Métodos estacionários

Estes métodos caracterizam-se pela recorrência (2.1) que se mantém constante ao longo de todo o processo iterativo. A sua eficiência é inferior à dos métodos não-estacionários (secção seguinte) e condicionada pela decomposição da matriz $A = M - N$.

### 2.2.1 Jacobi

No método de Jacobi a actualização das variáveis é efectuada em todos os pontos em simultâneo, a partir dos valores obtidos na iteração anterior $x^{(k-1)}$. Na forma matricial, as aproximações são geradas pela recorrência

$$x^{(k)} = D^{-1}(L+U)x^{(k-1)} + D^{-1}b \tag{2.5}$$

onde $D$, $-L$ e $-U$ representam as partes diagonal, estritamente, inferior e superior de $A$, respectivamente. A decomposição efectuada neste método é $M = D$ e $N = L + U$.

### 2.2.2 Gauss Seidel

O método de Gauss-Seidel difere do de Jacobi apenas no facto de, em cada equação, se utilizar os valores actualizados, logo que estes estejam disponíveis. A actualização das

variáveis em $x^{(k)}$ não é efectuada em simultâneo e as aproximações dependem da ordem pela qual são efectuadas as actualizações. A versão *forward*, na forma matricial, é definida por

$$x^{(k)} = (D - L)^{-1}(Ux^{(k-1)} + b), \tag{2.6}$$

onde a decomposição é $M = D - L$. Na versão *backward* temos $M = D - U$.

A convergência dos métodos de Jacobi e Gauss-Seidel, como já foi referido na secção anterior, depende do raio espectral das matrizes de iteração. Este é da ordem de $1 - O(h^2)$, sendo $h$ o espaçamento da malha, i.e., $h = n^{-d}$ onde $n$ é o número de variáveis e $d$ a dimensão do espaço associado ($d = 2$ num domínio plano) (Barrett et al., 1994, pág. 9).

Em termos matriciais, exige-se que $A$ seja diagonalmente dominante para que os métodos de Jacobi e Gauss-Seidel convirjam para qualquer valor inicial $x^{(0)}$ (Saad, 1996, pág. 111).

### 2.2.3 Sobre-Relaxações Sucessivas (Successive Over-Relaxation, SOR)

Em cada iteração deste método é efectuada uma média, ponderada para cada variável, entre o valor obtido pela iteração anterior, ponderado com $(1 - \omega)$, e o valor obtido pela iteração de Gauss-Seidel, ponderado com $\omega$. O parâmetro $\omega$ é designado por factor de extrapolação e uma das dificuldades deste método é a escolha do valor óptimo de $\omega$. Na sua forma matricial a recorrência é da forma

$$x^{(k)} = (D - \omega L)^{-1}[\omega U + (1 - \omega)D]x^{(k-1)} + \omega(D - \omega L)^{-1}b \tag{2.7}$$

Neste método a decomposição caracteriza-se por $M = 1/\omega(D - \omega L)$. Quando $\omega = 1$ o método simplifica-se no Gauss-Seidel (*forward* neste caso). Em (Barrett et al., 1994, pág. 13) são apresentadas algumas conclusões sobre a convergência deste método, directamente relacionada com $\omega$. Se $\omega$ não estiver no intervalo $[0, 2]$ a convergência deixa de se verificar. Se a matriz dos coeficientes $A$ for simétrica definida positiva, a convergência é garantida para qualquer $\omega \in [0, 2]$. Para uma classe especial de matrizes dos coeficientes designadas por consistentemente ordenadas com propriedade-A, ver definição completa em (Saad, 1996, pág. 112), é possível efectuar uma aproximação teórica ao valor óptimo de $\omega$:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho}}, \tag{2.8}$$

sendo $\rho$ o maior valor próprio da matriz de iteração do Jacobi. Na prática, o cálculo de um valor óptimo para $\omega$ é demasiado dispendioso para ser efectuado, pelo que é muitas vezes utilizada uma estimativa heurística da forma $\omega = 2 - O(h)$, que depende de $h$, o espaçamento da malha utilizada para discretizar o domínio (Barrett et al., 1994, pág. 11).

### 2.2.4 Sobre-Relaxações Sucessivas Simétricas (Symmetric Successive Over-Relaxation, SSOR)

Consiste em duas etapas do SOR, uma que utiliza o Gauss-Seidel *forward*, seguida de outra em *backward*. O facto da matriz de iteração ser parecida com $A$, quando esta é simétrica, faz com que a matriz $M$ do SSOR seja utilizada como pré-condicionador de sistemas simétricos, quando resolvidos por métodos não-estacionários. Este método é sobretudo utilizado para esse efeito, uma vez que em termos de convergência apresenta um pior desempenho que o SOR (Barrett et al., 1994, pág. 12). Na forma matricial a iteração é definida por

$$x^{(k)} = BFx^{(k-1)} + \omega(2-\omega)(D-\omega U)^{-1}D(D-\omega L)^{-1}b, \tag{2.9}$$

onde

$$\begin{aligned} B &= (D\omega - U)^{-1}(\omega L + (1-\omega)D), \\ F &= (D-\omega L)^{-1}(\omega U + (1-\omega)D). \end{aligned}$$

As matrizes $B$ e $F$ são as matrizes das versões (*bakward*) e (*forward*) do SOR, respectivamente. A decomposição efectuada neste método consiste em

$$M = \frac{1}{\omega(2-\omega)}(D-\omega L)D^{-1}(D-\omega U). \tag{2.10}$$

A matriz $M$ é utilizada como pré-condicionador de métodos como o Gradiente Conjugado aplicado à equação normal, originando esquemas iterativos como o CGNE/SSOR ou o CGNR/SSOR (Saad, 1996, pág. 312).

### 2.2.5 Algoritmo para Matrizes Tridiagonais (Tridiagonal Matrix Algorithm, TDMA)

Na discretização de equações elípticas num domínio rectangular, sujeito a determinadas condições de fronteira, o sistema resultante pode ser expresso como

$$Hx + Vu = b, \tag{2.11}$$

em que $H$ (de horizontal) contém os termos resultantes da discretização do operador $\partial^2/\partial x_1^2$ e $V$ (de vertical) contém os termos resultantes da discretização do operador $\partial^2/\partial x_2^2$. O método TDMA consiste em resolver o sistema (2.11) sucessivamente na direcção $x_1$ e na direcção $x_2$. Um algoritmo possível consiste em resolver alternadamente, até se verificar a convergência, as equações

$$\begin{aligned} (H+\rho_k I)x^{(k+\frac{1}{2})} &= (\rho_k I - V)x^{(k)} + b, \\ (V+\rho_k I)x^{(k+1)} &= (\rho_k I - H)x^{(k+\frac{1}{2})} + b. \end{aligned} \tag{2.12}$$

O nome deste método prende-se com a estrutura tridiagonal das matrizes $H$ e $V$. Contudo também aparece na literatura designado por *Alternating Direction Implicit* (ADI).

Apresentam-se de seguida algumas considerações sobre a convergência deste método, retiradas de (Saad, 1996, pág. 118). São conhecidos alguns resultados sobre a convergência, no caso de sistemas definidos positivos. Nesta situação $H$ e $V$ são simétricas definidas positivas e a iteração estacionária ($\rho_k = \rho > 0$, para todo o $k$) converge. Verifica-se também que a taxa de convergência, utilizando o $\rho$ óptimo, é próxima da obtida pelo SSOR, com o $\omega$ óptimo. Contudo as iterações do TDMA são mais dispendiosas do que as do SSOR. Um dos resultados mais importantes consiste na possibilidade dum aumento considerável da taxa de convergência, conseguido através da utilização de uma sequência cíclica de parâmetros $\rho_k$.

Em (Ferziger, 1998, pág. 266) são apresentados também alguns resultados sobre a forma de acelerar este método. Para uma grelha uniforme com o mesmo número de pontos $n$ em ambas as direcções, o valor óptimo é $\rho \approx n/2$. No caso de uma sequência cíclica, uma boa escolha é

$$\rho_i = d\left(\frac{\rho_1}{\rho_n}\right)^{(i-1)/(k-1)}, \qquad i = 1, 2, \ldots, j \tag{2.13}$$

em que $\rho_1$ e $\rho_n$ representam, respectivamente, o menor e o maior valor próprio das matrizes $H$ e $V$, $k$ é o número da iteração e $j$ a dimensão do sequência cíclica. Utilizando um ciclo com $j$ parâmetros definidos por esta sequência, a taxa de convergência poderá ser superior à do SOR.

## 2.3 Métodos não-estacionários: métodos de projecção ortogonal

Os métodos apresentados nesta secção utilizam o processo de ortogonalização de Arnoldi, ou derivam de métodos que o utilizam. Este processo é utilizado para determinar uma base ortonormal, comum ao sub-espaço de procura $\mathcal{K}_m$ e das restrições $\mathcal{L}_m$. Efectivamente a base óbvia do sub-espaço de Krylov

$$\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \ldots, A^{m-1}r^{(0)}\},$$

não apresenta boas características do ponto de vista numérico (Dongarra et al., 1998, pág. 153). Partindo do princípio de que temos uma base ortonormal $v_1, v_2, \ldots, v_j$ de $\mathcal{K}_m(A, v_1)$, esta é expandida calculando $t = Av_j$ e ortonormalizando-o relativamente a $v_1, v_2, \ldots, v_j$. O processo é iniciado com $v_1 = r^{(0)}/\|r^{(0)}\|_2$. A ortonormalização pode ser obtida de várias maneiras, como seja pelos processos de Gram-Shmidt, Gram-Shmidt modificados ou Householder. Contudo o Gram-Shmidt modificado é o mais utilizado (Golub and Loan, 1989, pág. 218). O processo de Arnoldi pode ser descrito sob a seguinte forma matricial

$$AV_{m-1} = V_m H_{m,m-1}, \tag{2.14}$$

onde $H_{m,m-1}$ é uma matriz de Hessenberg superior ($h_{i,j} = 0$ se $i > j + 1$).

### 2.3.1 Ortogonalização Completa (Full Ortogonalization Method, FOM)

Procuramos $x^{(m)} = x^{(0)} + \mathcal{K}_m$ de forma a termos $r^{(m)} \perp \mathcal{L}_m$. Supondo que $x^{(0)} = 0$, i.e., $b = r^{(0)}$, e que os vectores coluna de $V_m$ são gerados pelo processo de Arnoldi, podemos reescrever a equação (2.3) como

$$V_m{}^T A V_m y = \|r^{(0)}\|_2 e_1. \tag{2.15}$$

A ortogonalidade de $V_m$ reduz a matriz $H_{m,m} = V_m{}^T A V_m$ numa matriz de Hessemberg superior (Golub and Loan, 1989, pág. 501). O FOM consiste em resolver $H_{m,m}y = \|r^{(0)}\|_2 e_1$ e posteriormente determinar $x = V_m y$. A eliminação dos termos não nulos, inferiores à diagonal de $H_{m,m}$, pode ser efectuada pela rotação de Givens (Golub and Loan, 1989, pág. 202).

Este método é muito oneroso tanto em número de operações, $2\texttt{NNZ}+2mn$ em média por iteração, como em memória, $(m+3)n + m^2/n$ por iteração (Saad, 1996, pág. 153). Estes números aumentam linearmente com o número de iterações, tornando-o impraticável. Existem duas alternativas para contornar este aspecto: reiniciar ou truncar a ortogonalização no processo de Arnoldi. A primeira deu origem ao FOM Reiniciado, que consiste em fazer $v_1 = v_m$ ao fim de um período $m$ de iterações, ajustável em função do problema. A segunda deu origem ao IOM (*Incomplet Ortogonalisation Method*), que consiste em limitar as direcções, às quais $v_j$ é ortogonal, a um conjunto de $k$ vectores, cuja extensão depende da capacidade de memória disponível.

### 2.3.2 Resíduo Mínimo Generalizado (Generalized Minimal Residual, GMRES)

A aproximação deste método consiste em determinar $x = x^{(0)} + V_m y_m$ através da minimização de $\|b - Ax\|_2$. Utilizando a relação (2.14)

$$\begin{aligned} \|b - Ax_m\|_2 &= \|b - A(x^{(0)} + V_m y_m)\|_2 \\ &= \|r^{(0)} - AV_m y_m\|_2 \\ &= \|r^{(0)} - V_{m+1} H_{m+1,m} y_m\|_2 \\ &= \|V_{m+1}(\|r^{(0)}\|_2 e_1 - H_{m+1,m} y_m)\|_2 \end{aligned}$$

Como $V_{m+1}$ é ortogonal,

$$\|b - Ax_m\|_2 = \|\|r^{(0)}\|_2 e_1 - H_{m+1,m} y_m\|_2.$$

A determinação de $y_m$ no sentido dos mínimos quadrados conduz ao sistema

$$H_{m+1,m} y_m = \|r^{(0)}\|_2 e_1. \tag{2.16}$$

Tal como no FOM, $H_{m+1,m}$ é reduzida à forma triangular superior através da rotação de Givens. Contudo, existe aqui mais uma linha de termos não nulos que é necessário

### 2.3.1 Ortogonalização Completa (Full Ortogonalization Method, FOM)

Procuramos $x^{(m)} = x^{(0)} + \mathcal{K}_m$ de forma a termos $r^{(m)} \perp \mathcal{L}_m$. Supondo que $x^{(0)} = 0$, i.e., $b = r^{(0)}$, e que os vectores coluna de $V_m$ são gerados pelo processo de Arnoldi, podemos reescrever a equação (2.3) como

$$V_m{}^T A V_m y = \|r^{(0)}\|_2 e_1. \tag{2.15}$$

A ortogonalidade de $V_m$ reduz a matriz $H_{m,m} = V_m{}^T A V_m$ numa matriz de Hessemberg superior (Golub and Loan, 1989, pág. 501). O FOM consiste em resolver $H_{m,m}y = \|r^{(0)}\|_2 e_1$ e posteriormente determinar $x = V_m y$. A eliminação dos termos não nulos, inferiores à diagonal de $H_{m,m}$, pode ser efectuada pela rotação de Givens (Golub and Loan, 1989, pág. 202).

Este método é muito oneroso tanto em número de operações, 2NNZ+$2mn$ em média por iteração, como em memória, $(m+3)n + m^2/n$ por iteração (Saad, 1996, pág. 153). Estes números aumentam linearmente com o número de iterações, tornando-o impraticável. Existem duas alternativas para contornar este aspecto: reiniciar ou truncar a ortogonalização no processo de Arnoldi. A primeira deu origem ao FOM Reiniciado, que consiste em fazer $v_1 = v_m$ ao fim de um período $m$ de iterações, ajustável em função do problema. A segunda deu origem ao IOM (*Incomplet Ortogonalisation Method*), que consiste em limitar as direcções, às quais $v_j$ é ortogonal, a um conjunto de $k$ vectores, cuja extensão depende da capacidade de memória disponível.

### 2.3.2 Resíduo Mínimo Generalizado (Generalized Minimal Residual, GMRES)

A aproximação deste método consiste em determinar $x = x^{(0)} + V_m y_m$ através da minimização de $\|b - Ax\|_2$. Utilizando a relação (2.14)

$$\begin{aligned} \|b - Ax_m\|_2 &= \|b - A(x^{(0)} + V_m y_m)\|_2 \\ &= \|r^{(0)} - AV_m y_m\|_2 \\ &= \|r^{(0)} - V_{m+1} H_{m+1,m} y_m\|_2 \\ &= \|V_{m+1}(\|r^{(0)}\|_2 e_1 - H_{m+1,m} y_m)\|_2 \end{aligned}$$

Como $V_{m+1}$ é ortogonal,

$$\|b - Ax_m\|_2 = \|\|r^{(0)}\|_2 e_1 - H_{m+1,m} y_m\|_2.$$

A determinação de $y_m$ no sentido dos mínimos quadrados conduz ao sistema

$$H_{m+1,m} y_m = \|r^{(0)}\|_2 e_1. \tag{2.16}$$

Tal como no FOM, $H_{m+1,m}$ é reduzida à forma triangular superior através da rotação de Givens. Contudo, existe aqui mais uma linha de termos não nulos que é necessário

eliminar. O armazenamento em memória de todos os vectores $v_j$, previamente calculados, é, à semelhança do FOM, muito penalizador em termos de eficiência. Pelo que as alternativas existentes, foram desenvolvidas no mesmo sentido do que as do FOM: reinicialização ou truncatura.

No GMRES Reiniciado ou GMRES($m$) o processo é reiniciado ao fim de $m$ iterações, definindo $x^{(0)} = x^{(m)}$. Um dos problemas que pode surgir com o reinício periódico, de $m$ em $m$ iterações, é o da estagnação, i.e., quando a convergência do GMRES só se verifica para valores de $m$ muito grandes. Este fenómeno surge nas matrizes que não são definidas positivas (Saad, 1996, pág. 167), e a solução passa pela utilização de um pré-condicionador.

A alternativa ao reinício é a truncatura no processo de Arnoldi. O QGMRES (Quasi-GMRES) consiste, tal como o IOM, em fazer uma ortogonalização incompleta. O novo $v_j$ é ortogonalizado apenas em relação aos últimos $k$ vectores calculados. Embora esta solução poupe numerosos cálculos, não evita a necessidade de guardar todos os $v_j$ para poder calcular $x = x^{(0)} + V_m y_m$.

### 2.3.3 Resíduo Mínimo Generalizado Pré-condicionado (Preconditioned Generalized Minimal Residual, PGMRES)

O método do Resíduo Mínimo Generalizado Pré-condicionado (PGMRES) é uma versão do GMRES em que o sistema original é pré-condicionado. O pré-condicionamento pode ser realizado à esquerda ou à direita. No entanto quando aplicado à direita tem a vantagem de, em cada iteração, fornecer uma aproximação do resíduo em relação ao sistema original, enquanto que o pré-condicionamento à esquerda dá uma estimativa do resíduo em relação ao sistema pré-condicionado (Saad, 1996, pág. 252).

O GMRES pré-condicionado à direita consiste em resolver o sistema

$$AM^{-1}u = b, \qquad u = Mx. \tag{2.17}$$

A variável $u$ não precisa de ser calculada explicitamente, pois os vectores da base do subespaço de Krylov são obtidos a partir do resíduo inicial $r^{(0)}$. O algoritmo do PGMRES (Saad, 1996, pág. 252) difere do algoritmo do GMRES em dois aspectos. O primeiro deriva do facto da base calculada pelo ciclo de Arnoldi ser igual a

$$\{r^{(0)}, AM^{-1}r^{(0)}, \ldots, (AM^{-1})^{m-1}r^{(0)}\}$$

em vez de

$$\{r^{(0)}, Ar^{(0)}, \ldots, A^{m-1}r^{(0)}\}.$$

O segundo consiste na aproximação à solução ser obtida através de $x^{(m)} = x^{(0)} + M^{-1}V_m y_m$ em vez de $x^{(m)} = x^{(0)} + V_m y_m$.

### 2.3.4 Quase GMRES Directo (Direct Quasi-GMRES, DQGMRES)

O Direct Quasi-GMRES (DQGMRES) é uma alternativa ao GMRES obtida a partir do QGMRES. Num processo de ortogonalização incompleta, como o utilizado no QGMRES, a estrutura da matriz $H_{m+1,m}$ é bandeada. Os termos não nulos estão situados sobre $k+1$ diagonais ($k$ é o número de vectores da base). Multiplicando os dois membros do sistema (2.16), pelas matrizes de rotação de Givens $\Omega_i$, de forma a anular os termos situados abaixo da diagonal principal de $H_{m+1,m}$, obtém-se o sistema $R_{m+1,m}y_m = g_m$. Desprezando a última linha de $R_{m+1,m}$, obtém-se $y_m = {R_m}^{-1}g_m$ e consequentemente

$$x^{(m)} = x^{(0)} + V_m {R_m}^{-1} g_m.$$

Definindo $P_m = V_m R_m^{-1}$,

$$x^{(m)} = x^{(0)} + P_m g_m.$$

Esta aproximação dá origem à seguinte fórmula de recorrência

$$x^{(m)} = x^{(m-1)} + p_m \gamma_m, \tag{2.18}$$

onde $\gamma_m$ é a componente $m$ de $g_m$ e $p_m$ é o vector coluna $m$ de $P_m$. Este método tem a vantagem de fazer a aproximação de $x^{(m)}$ através de $x^{(m-1)}$ e de um pequeno número de actualizações de vectores.

Existe uma relação entre a norma residual deste método e a do GMRES,

$$\|{r_m}^Q\|_2 \leq \kappa(V_{m+1})\|{r_m}^G\|_2, \tag{2.19}$$

onde $V_{m+1}$ é a base ortonormal associada com o DQGMRES, ${r_m}^Q$ e ${r_m}^G$ são os resíduos do DQGMRES e do GMRES, respectivamente. A demonstração deste resultado assim como outros aspectos deste método encontram-se em (Saad, 1996, pág. 168).

### 2.3.5 Lanczos

No caso da matriz dos coeficientes de um sistema ser real e simétrica, a matriz de Hessemberg $H_m$, resultante do processo de Arnoldi é tridiagonal e simétrica: $T_m$. O algoritmo de Lanczos, uma variante do Arnoldi para estes tipo de sistemas, tem como diferença principal a geração de uma base ortonormal através de uma recorrência de três termos, i.e., cada novo vector da base é calculado com o auxílio dos dois anteriores apenas. Na prática, verifica-se a perda de ortogonalidade entre os vectores da base a partir de um certo número de iterações (Saad, 1996, pág. 173).

A utilização deste processo para resolver um sistema de equações lineares dá origem ao método de Lanczos, que tal como os restantes métodos de projecção ortogonal em $\mathcal{K}_m$, procura uma aproximação à solução

$$x^{(m)} = x^{(0)} + V_m y_m, \qquad y_m = T_m^{-1}(\beta e_1).$$

A resolução do sistema $T_m y_m = \beta e_1$, com $\beta = \|r^{(0)}\|_2$, é conseguida através da decomposição $LU$ do sistema ($T_m = L_m U_m$). Um dos problemas que podem surgir neste ponto é a instabilidade numérica relacionada com a eliminação gaussiana. Não havendo pivotagem parcial, é essencial que $T_m$ seja definida positiva, para garantir a decomposição. Isto acontecerá se $A$ for também simétrica definida positiva, (Golub and Loan, 1989, pág. 495). A aproximação da solução será então dada por

$$x^{(m)} = x^{(0)} + V_m U_m^{-1} L_m^{-1}(\beta e_1).$$

Definindo $P_m = V_m U_m^{-1}$ e $z_m = L_m^{-1}(\beta e_1)$ obtemos $x^{(m)} = x^{(0)} + P_m z_m$. Tal como no DQGMRES esta igualdade pode ser reduzida a uma simples fórmula de actualização

$$x^{(m)} = x^{(m-1)} + \zeta_m p_m. \tag{2.20}$$

Em termos de memória, no algoritmo deste método, cada actualização implica que se guarde, para além da matriz $A$, os 4 vectores, $v_m, v_{m-1}, p, x$ e um vector auxiliar de trabalho $w$. Tendo em conta as necessidades de memórias doutros métodos, como o GMRES, este aspecto é vantajoso.

### 2.3.6 Gradiente Conjugado (Conjugate Gradient, CG)

Em (Saad, 1996, pág. 173) são apresentados dois resultados importantes, relativamente ao método de Lanczos:

- Os resíduos $r^{(i)} = b - Ax^{(i)}$ produzidos pelo método de Lanczos, são ortogonais.
- Os vectores auxiliares $p_i$ formam um conjunto A-conjugado, i.e., o produto interno $(Ap_i, p_j) = 0$ se $i \neq j$.

O método do Gradiente Conjugado pode ser visto como uma versão do método de Lanczos, derivado de forma a impor os dois resultados anteriores. Neste sentido $x^{(m+1)}$ pode ser expresso como

$$x^{(m+1)} = x^{(m)} + \alpha_m p_m.$$

Consequentemente os resíduos devem estar de acordo com a recorrência

$$r^{(m+1)} = r^{(m)} - \alpha_m A p_m.$$

Da condição de ortogonalidade dos resíduos resulta

$$\alpha_m = \frac{(r^{(m)}, r^{(m)})}{(Ap_m, r^{(m)})}. \tag{2.21}$$

Como a próxima direcção de procura $p_{m+1}$ é uma combinação linear de $r^{(m+1)}$ e de $p_m$ segue-se

$$p_{m+1} = r^{(m+1)} + \beta_m p_m. \tag{2.22}$$

Da ortogonalidade de $Ap_m$ em relação a $Ap_{m-1}$ resulta

$$\beta_m = \frac{(r^{(m+1)}, r^{(m+1)})}{(r^{(m)}, r^{(m)})}. \tag{2.23}$$

Tal como no Lanczos, este método recorre implicitamente a uma decomposição LU de $T_m$. Para garantir a sua existência é necessário que $A$ seja definida positiva, para além de ser simétrica. Esta é uma condição suficiente para garantir o funcionamento deste método, contudo ele poderá também funcionar sem que ela se verifique, desde que $T_m$ admita decomposição LU. Em termos de memória, o algoritmo pode ser implementado, guardando em cada iteração, para além da matriz $A$, 4 vectores $x, p, Ap$ e $r$, em vez dos 5 vectores, no caso do método de Lanczos.

Relativamente à convergência, se $A = I - B$ for simétrica definida positiva e se $B$ tiver $r$ vectores coluna linearmente independentes, então o Gradiente Conjugado converge, no máximo, em $r + 1$ iterações. Outro resultado importante, também apresentado em (Golub and Loan, 1989, pág. 525), relativamente ao majorante do erro da aproximação é o seguinte

$$\|x - x^{(m)}\|_A \leq 2\|x - x^{(0)}\|_A \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^m, \tag{2.24}$$

onde $x$ é a solução exacta, $\kappa$ o número de condição baseado na norma-2 $\kappa_2(A)$, sendo a norma-A de um vector $w$ definida como $\|w\|_A = \sqrt{w^T A w}$. Este resultado mostra que o CG converge rapidamente se $\kappa_2(A)$ estiver próximo de 1.

### 2.3.7 Gradiente Conjugado para Resíduos Normais (Conjugate Gradient Normal Residual, CGNR)

Uma técnica particularmente útil na resolução de sistema sobre-determinados ($A$ de dimensões $n \times m$ com $n > m$) consiste em transformar o sistema num outro equivalente, representado pela equação $A^T Ax = A^T b$. Esta equação é designada por equação normal. A resolução deste sistema conduz à minimização de $\|b - Ax\|_2$. Por esta razão, os métodos iterativos, quando aplicados a esta equação, são completados por NR (*Normal Residual*). Por exemplo CGNR designa o Gradiente Conjugado aplicado à equação Normal que minimiza o Resíduo.

No caso de sistemas indeterminados ($n < m$), recorre-se a outra versão da equação normal: $AA^T u = b$, com $x = A^T u$. A resolução desta equação conduz à minimização de $\|x - A^T u\|_2$, i.e., do erro $\|x - x^{(k)}\|_2$, onde $x$ é a solução exacta de $Ax = b$ . A designação dos métodos iterativos, quando estes são aplicados a esta equação, é completada por NE (*Normal Error*).

As equações normais podem também ser utilizadas para converter um sistema não-simétrico num sistema simétrico definido positivo. Efectivamente $A^T A$ é uma matriz simétrica positiva definida se $A$ for quadrada e não singular. Esta técnica possibilita a aplicação de métodos iterativos que necessitem desta premissa, como é o caso do CG. Em contrapartida a nova matriz dos coeficientes $A^T A$ tem um número de condição igual ao quadrado

do número de condição da matriz dos coeficientes original, i.e., $\kappa_2(A^TA) = [\kappa_2(A)]^2$ (Saad, 1996, pág. 231). Este aspecto torna-se particularmente desvantajoso quando a matriz $A$ é mal condicionada.

### 2.3.8 Resíduos Conjugados (Conjugate Residuals, CR)

O método dos Resíduos Conjugados (CR) é deduzido do GMRES, no caso particular de $A$ ser Hermitiana. O algoritmo deste método (Saad, 1996, pág. 182) é muito semelhante ao do CG mas exige mais actualizações de vectores. Para além da matriz A, cada actualização necessita de 5 vectores $x, p, Ap, r$ e $Ar$.

No caso não-simétrico, métodos como o CG ou CR, baseados em diferentes conceitos de ortogonalidade, também podem ser derivados do GMRES. Contudo, os métodos resultantes exigem grande quantidade memória. No GCR (*Generalized CR*) é necessário guardar ambos os conjuntos $p_i$ e $Ap_i$, duplicando a memória e o número de cálculos exigidos, relativamente ao GMRES. As alternativas habituais estão patentes em métodos como o GCG($m$), versão reiniciada, ou ORTHODIR($k$), versão truncada.

## 2.4 Métodos não-estacionários: métodos de projecção oblíqua

Os métodos abordados nesta secção são métodos de projecção em $\mathcal{K}_m = (A, v_1)$, ortogonalmente a $\mathcal{L}_m$, com

$$\mathcal{L}_m = \mathcal{K}_m(A^T, w_1) \equiv \{w_1, A^T w_1, \ldots, (A^T)^{m-1} w_1\}. \tag{2.25}$$

O processo de determinação das duas bases consiste numa extensão do algoritmo de Lanczos, designado por bi-ortogonalização. As duas bases geradas $\{v_i\}_{i=1,\ldots,m}$ e $\{w_i\}_{i=1,\ldots,m}$, formam um sistema bi-ortogonal, i.e., $(v_i, w_i) = \delta_{ij}$. Neste processo não é imposta qualquer condição de ortogonalidade entre vectores da mesma base. A particularidade deste processo está no facto da matriz definida por ${W_m}^T A V_m$ ser tridiagonal (Saad, 1996, pág. 205).

O facto de ${W_m}^T A V_m = T_m$ ser tridiagonal sem que $A$ seja simétrica, possibilita a resolução do sitema em duas etapas

$$T_m y = \beta e_1, \qquad \text{com} \quad x^{(m)} = V_m y \tag{2.26}$$

por métodos que não exigem grandes recursos de memória, tal como os métodos derivados do Lanczos normal.

### 2.4.1 Gradiente BiConjugado (BiConjugate Gradient, BICG)

O método do Gradiente BiConjugado (BICG) é deduzido de (2.26), tal como o CG é deduzido do Lanczos. O algoritmo define $v_1 = r^{(0)}/\|r^{(0)}\|_2$, $w_1$ é arbitrário desde que

$(v_1, w_1) \neq 0$. Efectua a decomposição $T_m = L_m U_m$ e determina

$$x^{(m)} = x^{(0)} + P_m {L_m}^{-1} \beta e_1, \quad \text{com} \quad P_m = V_m {U_m}^{-1}. \tag{2.27}$$

Implicitamente, este algoritmo resolve outro sistema $A^T x_* = b_*$, designado por sistema dual. No caso em que se pretende também a solução do sistema dual começa-se por definir $w_1 = b_* - A^T x_*^{(0)}$. Ao todo utiliza 8 vectores $p, p_*, r, r_*, x, x_*, Ap, Ap_*$ e 2 matrizes $A, A^T$. No caso de não se pretender resolver o sistema dual, este aspecto é desvantajoso pois é efectuado um grande número de operações desnecessárias, em particular uma multiplicação matriz-vector adicional em cada iteração. Outro aspecto que pode ser penalizante é a necessidade da matriz $A^T$, que pode não estar disponível explicitamente.

### 2.4.2 Resíduo Quase Mínimo (Quasi-Minimal Residual, QMR)

O método Quasi-Minimal Residual (QMR) baseia-se na aproximação dos resíduos mínimos, tal como o GMRES. Partindo da relação de recorrência na forma matricial para $v_j$,

$$AV_i = V_{i+1} T_{i+1,i}$$

e efectuando um desenvolvimento semelhante ao utilizado para obter (2.16) resulta

$$\|b - Ax_m\|_2 = \|V_{m+1}(\beta e_1 - T_{m+1,m} y_m)\|_2.$$

Como $V_{m+1}$ não é ortogonal não é permitida a simplificação $\|V_{m+1}\|_2 = 1$, utilizada para a dedução do GMRES, da qual resulta:

$$\|b - Ax_m\|_2 = \|\beta e_1 - T_{m+1,m} y_m\|_2$$

A aproximação do QMR consiste em pressupor que os vectores $v_j$ mantêm apesar de tudo algum tipo de ortogonalidade e admitir a validade da simplificação $\|V_{m+1}\|_2 \approx 1$, determinando $x^{(m)} = x^{(0)} + V_m y$ a partir da minimização de $\|\beta e_1 - T_{m+1,m} y_m\|_2$. Devido à estrutura de $T_m$ o algoritmo para este método (Saad, 1996, pág. 212) é semelhante ao do DQGMRES.

O QMR tem menos possibilidades de falhar do que o BICG, por não recorrer à factorização $LU$, além de convergir de forma mais regular. Por estas razões é mais robusto que o BICG. Apesar de não minimizar o resíduo, converge por vezes tão rapidamente como GMRES (Dongarra et al., 1998, pág. 182) em relação ao qual tem a vantagem, de actualizar os vectores das bases com poucos cálculos e exigir menos espaço de memória. Em contrapartida, precisa de dois produtos matriz-vector por iteração, um com $A$ e outro com $A^T$.

### 2.4.3 Gradiente Conjugado Quadrado (Conjugate Gradient-Squared, CGS)

O método do Gradiente Conjugado Quadrado (CGS) deriva do BICG, no sentido de evitar a utilização da transposta de $A$ e obter uma convergência mais rápida, mantendo os mesmos

custos computacionais. No algoritmo do BICG, os coeficientes utilizados em cada iteração $\alpha_i$ e $\beta_i$, são calculados através dos resíduos gerados pelas duas sequências bi-ortogonais $r^{(i)}$ e $r_*^{(i)}$. Estes podem ser expressos como $r^{(i)} = \phi_i(A)r^{(0)}$ e $r_*^{(i)} = \pi_i(A)r^{(0)}$, com $\phi_i$ e $\pi_i$ dois polinómios de grau $i$. Fazendo uso de relações de ortogonalidade verifica-se que a actualização pode ser efectuada através do quadrado de um único polinómio $\phi_i{}^2(A)r^{(i)}$. Definindo uma forma recorrente para determinar

$$r^{(i)} = \phi_i{}^2(A)r^{(0)} \tag{2.28}$$

é possível calcular todos os parâmetros necessários do BICG, sem utilizar os vectores relacionados com o sistema dual $r_*^{(i)}$ e $p_*^{(i)}$. Esta é a ideia na qual assenta o CGS, cuja derivação completa e o algoritmo podem ser encontrados em (Saad, 1996, pág. 214).

Comparativamente ao BICG, o CGS converge mais rapidamente e necessita de menos memória. Não necessita também da matriz $A^T$. Contudo porque deriva do BICG, todas as irregularidades de convergência do BICG, aparecem no CGS elevadas ao quadrado. Este aspecto pode conduzir a comportamentos muito irregulares. No caso do sistema ser simétrico definido positivo o BICG dá a mesma solução que o CG e, obviamente, o CGS também converge. Relativamente ao caso não-simétrico, quando correctamente pré-condicionado, parece convergir sem problemas (Saad, 1996).

### 2.4.4 Gradiente BiConjugado Estabilizado (BiConjugate Gradient Stabilized, BICGSTAB)

Devido ao facto de potenciar as irregularidades da convergência do BICG, tais como erros de arredondamento, o CGS apresenta por vezes grandes irregularidades na convergência. O BICGSTAB foi desenvolvido no sentido de corrigir este aspecto (Saad, 1996, pág. 216). A ideia consiste em actualizar os parâmetros da iteração através do produto de dois polinómios em vez do quadrado de um só

$$r^{(i)} = \psi_i(A)\phi_i(A)r^{(0)}. \tag{2.29}$$

O polinómio $\psi_i(A)$, novo relativamente ao BICG, é definido recursivamente em cada iteração com o objectivo de estabilizar ou regularizar a convergência do algoritmo original

$$\psi_i(t) = (1 - \omega_j t)\phi_i(t). \tag{2.30}$$

O parâmetro $\omega_j$ é determinado em cada iteração de forma a minimizar os resíduos $r^{(i)}$.

O algoritmo do BICGSTAB comporta dois produtos internos a mais do que o CGS. Converge mais regularmente do que o CGS e em certos casos converge mesmo mais rápido; contudo, existe a possibilidade do algoritmo falhar se $\omega_j$ for nulo ou aproximadamente nulo (Dongarra et al., 1998, pág. 186).

### 2.4.5 Resíduo Quase Mínimo Sem Transposta (Tanspose-Free Quasi-Minimal Residual, TFQMR)

O Tanspose-Free QMR (TFQMR), tal como o CGS e o BICGSTAB, são variantes do BICG desenvolvidas para evitar a utilização da matriz $A^T$. O TFQMR pode ser deduzido do algoritmo do CGS, actualizando o vector $x^{(k)}$ em duas etapas, i.e., desdobrando cada iteração do CGS. O desdobramento só é possível porque a actualização de $x^{(k)}$ no algoritmo do CGS envolve duas multiplicações matriz-vector.

A redefinição do algoritmo do CGS conduz a uma nova determinação de $x^{(k)}$, baseada na minimização dos resíduos $r^{(k)}$ no sentido dos mínimos quadrados. Esta aproximação baseia-se num artificialismo semelhante ao utilizado no algoritmo do QMR, de forma a ultrapassar a falta de ortogonalidade entre os vectores da base do sub-espaço de Krylov (Saad, 1996, pág. 221). O TFQMR pode também ser visto como uma versão quadrada do QMR, tal como o CGS em relação ao BICG (Dongarra et al., 1998, pág. 183).

O TFQMR e o BICGSTAB convergem em geral no mesmo número de iterações. Comparativamente com o BICG, o TFQMR e o BICGSTAB têm a vantagem de exigir apenas uma multiplicação matriz-vector por iteração. O GMRES converge em menos iterações do que estes dois métodos se o sistema estiver bem condicionado. No caso contrário, se forem necessárias muitas iterações para que se verifique a convergência, o TFQMR e o BICGSTAB são mais eficientes (Saad, 1996, pág. 225).

## 2.5 Pré-condicionamento

Como já foi referido o pré-condicionador desempenha um papel muito importante nos métodos iterativos. A sua utilização pode levar a uma redução drástica do número de iterações. Por outro lado a sua construção pode prejudicar o desempenho global se o número de iterações não for suficiente para amortizar este custo adicional. Quanto mais próximo estiver o pré-condicionador $M$ de $A$, mais eficiente se torna o pré-condicionamento, mas em contrapartida, maiores são os recursos computacionais necessários à sua elaboração.

São utilizadas várias técnicas para a obtenção do pré-condicionador. Uma destas, já referida, consiste em utilizar as matrizes resultantes da decomposição subjacente aos métodos estacionários, tais como o SSOR. Um outro tipo de processos baseiam-se em factorizações incompletas de $A$, tais como a factorização LU ou de Cholesky, ver (Golub and Loan, 1989, pág. 141). Nesta secção abordaremos apenas as técnicas derivadas da factorização LU.

### 2.5.1 Factorização Incompleta LU: ILU

A Factorização Incompleta LU é uma técnica simples que consiste em decompor a matriz esparsa $A$ da seguinte forma $A = LU - R$, onde $L$ é uma matriz triangular inferior, $U$ uma

matriz triangular superior e $R$ a matriz resíduo ou erro da factorização. A decomposição é efectuada impondo determinadas restrições à matriz $R$, tais como a existência de zeros em determinadas posições. O algoritmo utilizado para a obtenção de $LU$ baseia-se na eliminação gaussiana seguida da substituição por zero de certos termos, não pertencentes à diagonal principal, de acordo com determinadas regras. A existência da factorização incompleta $LU$ é garantida para M-matrizes, qualquer que seja a estrutura esparsa de $A$ (Dongarra et al., 1998, pág. 200).

### 2.5.2 Enchimento com zeros: ILU(0)

Nesta versão da factorização ILU, a matriz $L$ tem a mesma estrutura da parte triangular inferior de $A$ e $U$ tem a mesma estrutura da parte triangular superior de $A$. Em consequência, a matriz $R = A - LU$ tem termos nulos nas posições correspondentes a termos não-nulos de $A$. Na eliminação gaussiana, os termos não-nulos, fora do padrão definido por $A$, são substituídos por zeros. Se $A$ for uma matriz com cinco diagonais não-nulas, a matriz $LU$ resultante de ILU(0) terá uma diagonal adicional situada imediatamente após cada uma das diagonais mais afastadas.

### 2.5.3 Nível de enchimento $\ell$: ILU($\ell$)

A imprecisão da factorização ILU(0) pode conduzir a taxas de convergência insuficientes. Factorizações com mais precisão poderão conduzir a melhores resultados. A factorização ILU($\ell$) não descarta tantos elementos como a ILU(0). Esta estratégia consiste em definir um nível de enchimento `lfil(ij)` associado a cada elemento processado pela eliminação gaussiana. Sempre que `lfil(ij)`$> \ell$ o elemento $l_{ij}$ ou $u_{ij}$ é substituído por zero. Os níveis de enchimento iniciais são `lfil(ij)`$= 0$ se $a_{ij} \neq 0$ e `lfil(ij)`$= \infty$ se $a_{ij} = 0$. Ao longo da eliminação gaussiana os elementos são actualizados da seguinte forma

$$\texttt{lfil(ij)} = min\{\texttt{lfil(ij)}, \texttt{lfil(ik)} + \texttt{lfil(kj)} + 1\}. \tag{2.31}$$

A situação $\ell = 0$ corresponde à factorização ILU(0). Se $A$ for uma matriz com cinco diagonais não nulas, a matriz $L_1U_1$ resultante de ILU(1) corresponde à matriz $L_0U_0$ resultante de ILU(0), acrescida de uma diagonal a seguir a cada uma das diagonais mais afastadas.

### 2.5.4 Factorização ILU Modificada: MILU

Nas estratégias anteriores, os termos eliminados na factorização, são simplesmente descartados. A factorização MILU compensa esse efeito através da correcção dos termos situados na diagonal. A cada um dos termos situados sobre a diagonal é subtraída a soma dos termos eliminados na linha ou coluna correspondente. Esta estratégia pode ser combinada com qualquer uma das anteriores. Por exemplo MILU(0) consiste na ILU(0) com compensação.

Existem variantes, como a ILUD, que consistem em compensar apenas uma fracção $\alpha$ da soma dos termos eliminados em cada linha. Nesta abordagem um termo é apagado se for inferior à média da linha, multiplicada por `tol`, tudo em valores absolutos. Este pré-condicionamento depende de dois parâmetros previamente definidos pelo que é designado por ILUD(`tol`,$\alpha$).

### 2.5.5 Estratégias limitadoras: ILUT

Esta estratégia visa determinar, de uma forma dinâmica, os termos que deverão ser eliminados. Os dois parâmetros utilizados, `tol` e `lfil`, determinam a eliminação dos termos mais pequenos. Em cada linha são mantidos, no máximo, os `lfil` maiores termos não nulos, não incluindo o termo sobre a diagonal. O parâmetro `tol` tem o mesmo significado que anteriormente. Neste pré-condicionador não é feita qualquer compensação.

### 2.5.6 ILU com Pivotagem parcial: ILUP

As aproximações ILU podem falhar em matrizes derivadas de aplicações reais por várias razões, como por exemplo na presença de zeros na posição do *pivot*, *overflow* ou *underflow* devido ao crescimento exponencial das entradas de $L$ ou $U$. Para além destas razões, pode suceder que o pré-condicionador seja instável, i.e., $M^{-1} = L^{-1}U^{-1}$ tenha uma norma elevada, conduzindo a uma convergência lenta ou mesmo divergência.

Alguns destes problemas podem ser ultrapassados através da pivotagem parcial. As colunas são preferencialmente pivotadas devido à estrutura de dados utilizada (CSR). Neste processo, existem dois parâmetros, `permtol` e `nbloc`, que definem em que circunstâncias se deve efectuar a pivotagem. A permutação das colunas $i$ e $j$ deverá ocorrer sempre que `permtol`$\times|a_{ij}| > |a_{ii}|$. O parâmetro `nbloc` visa limitar o número de colunas entre as quais é possível permutar. Se `nbloc`$\geq n$ não há restrições de permutação.

O pré-condicionador ILUD com pivotagem parcial é designado por ILUDP. Da mesma forma o ILUTP resulta do ILUT com pivotagem parcial das colunas.

## 2.6 Resumo

Embora a descrição sumária, exposta nas secções anteriores, mencione apenas alguns métodos iterativos, é possível verificar que estes existem em grande número, dificultando a escolha. O primeiro aspecto a ter em conta na escolha de um método iterativo prende-se com as características do sistema a resolver. A figura 2.1 apresenta uma árvore de decisão, baseada em (Barrett et al., 1994, pág. 11), que poderá servir de ponto de partida. Se a matriz $A$ for simétrica definida positiva a escolha recairá naturalmente sobre o CG. É um método que tem a vantagem de exigir menos recursos de memória e menos operações por iteração do que os restantes. Contudo em termos de convergência depende muito do

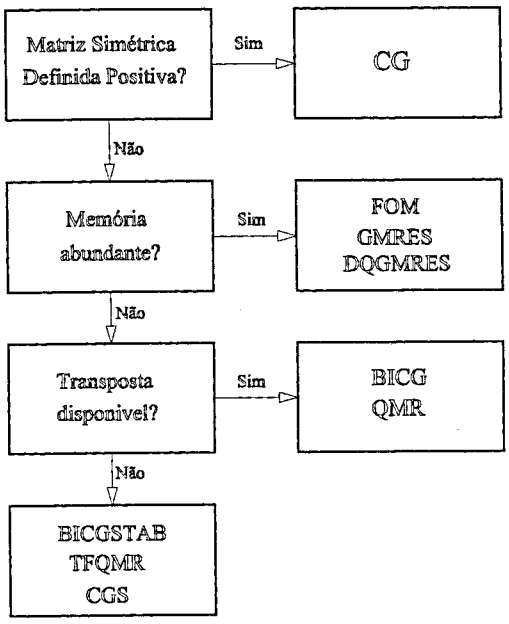



Figura 2.1: Critérios na escolha de um método iterativo

condicionamento da matriz $A$. Tal como o Lanczos este método está também sujeito à progressiva perda de ortogonalidade dos resíduos em relação aos previamente calculados.

No caso da matriz $A$ não ser simétrica definida positiva, deverá analisar-se os recursos computacionais disponíveis. O FOM e o GMRES são métodos robustos mas que exigem memória ilimitada. Na prática estes poderão ser utilizados nas suas versões reiniciadas, utilizando um período $k$ (dimensão do sub-espaço de Krylov) o mais elevado possível. O DQGMRES é um alternativa a estes dois métodos, que, em vez de reíniciar o processo de ortonormalização, limita-o aos últimos $k$ vectores calculados.

No caso de não se dispor de recursos computacionais suficientes, a escolha deverá ter em conta a utilização da matriz $A^T$. A sua utilização implica operações adicionais que tornam o processo iterativo mais demorado. Os métodos BICG e QMR são por isso penalizados relativamente a métodos como o BICGSTAB, CGS ou TFQMR, que não requerem a existência da matriz transposta.

As premissas relativas aos métodos iterativos atrás expostas poderão ser alteradas se o sistema for pré-condicionado. A sua utilização transforma o sistema original noutro equivalente, mas com características diferentes. Quando o novo sistema é mais facilmente resolúvel, i.e., são necessárias menos iterações, a acção do pré-condicionador é eficiente. Contudo quanto mais eficiente este for mais dispendiosa se torna a sua construção. É necessário analisar a eficiência do pré-condicionamento em termos globais. O parâmetro a ter em conta, na análise, deve incluir o tempo de pré-condicionamento juntamente com o tempo de convergência utilizado pelo método iterativo.

Na elaboração dos pré-condicionadores derivados da factorização incompleta $LU$ são utilizadas várias estratégias. Estas consistem em descartar termos ao longo do processo de eliminação gaussiana. Em alguns pré-condicionadores como o ILUT, estes termos são simplesmente descartados. Noutros como o ILUD, é feita uma compensação na diagonal pelos termos apagados em cada linha. Todas elas visam obter uma aproximação da matriz $A$, com o número mínimo de entradas, de forma a que $(LU)^{-1}A$ seja mais próxima da matriz identidade. Outro aspecto que se reverte de particular importância é a pivotagem parcial das colunas. A existência da factorização incompleta $LU$ é garantida apenas para M-matrizes (Dongarra et al., 1998, pág. 204). Caso a matriz $A$ não tenha esta propriedade, o processo de elaboração do pré-condicionador poderá sofrer de instabilidade numérica ou conduzir a uma convergência deficiente. Utilizando a pivotagem parcial é possível remediar este aspecto.

# Capítulo 3

# Análise dos resultados

Neste capítulo são apresentados os resultados da resolução dos sistemas de teste, obtidos a partir da discretização da equação de correcção da pressão e da equação de transporte da quantidade de movimento na direcção $x$ (velocidade horizontal), conforme descrito na subsecção 1.2. Para cada variável usaram-se matrizes com dimensões: $40 \times 40$, $80 \times 80$, $120 \times 80$ e $120 \times 120$.

Este capítulo está dividido em três partes. Na primeira faz-se uma descrição da metodologia seguida nos testes. Na segunda parte (secção 3.2) efectua-se a análise das características dos sistemas, incluíndo uma análise baseada na estimativa dos valores próprios das matrizes dos coeficientes. Na terceira parte (secções 3.3 a 3.6) são apresentados os resultados da resolução destes sistemas por dez métodos iterativos descritos no capítulo anterior. Os resultados estão agrupados de acordo com as dimensões das matrizes, ocupando cada matriz uma secção diferente.

## 3.1 Especificação dos testes

A anteceder a apresentação dos resultados, nesta secção apresentamos as condições em que os diferentes métodos iterativos foram testados.

### 3.1.1 Métodos utilizados

O estudo incidiu sobre 10 métodos iterativos diferentes, dos quais 1 é um método estacionário (TDMA) e os restantes 9 são não-estacionários, 3 de projecção oblíqua (BICG, BICGSTAB e TFQMR) e 6 de projecção ortogonal (CG, CGNR, FOM, GMRES, PGMRES e DQGMRES). O TDMA foi utilizado neste trabalho a título de comparação com os métodos não-estacionários. A versão utilizada foi retirada do mesmo programa de simulação de fluidos baseado no algoritmo SIMPLE, utilizado para produzir as matrizes usadas neste trabalho. Esta versão do TDMA trata-se de uma implementação não optimizada,

com $\rho = 0$ em todas as iterações da equação (2.12).

Utilizaram-se versões do FOM, GMRES e PGMRES reiniciados de $k$ em $k$ iterações. Da mesma maneira limitou-se o número de direcções, em relação às quais o DQGMRES efectua a ortogonalização dos resíduos, ao mesmo número $k$. O valor de $k$, dimensão da base do sub-espaço de Krylov, é sempre comum a estes quatro métodos. Nos sistemas derivados do campo da pressão testou-se o efeito da variação de $k$ desde $k = 1$ até $k = 50$. Utilizou-se apenas $k = 10$, para efeito de comparação com os restantes métodos pré-condicionados, com excepção do DQGMRES em que se utilizou também $k = 2$.

Os pré-condicionadores utilizados para os sistemas das pressões foram o ILU($\ell$) e o ILUDP(`tol`,$\alpha$). Para além destes recorreu-se por vezes a outros de forma a comparar os resultados. Em todos os testes com os sistemas da velocidade utilizou-se o pré-condicionador ILU(0), recorrendo-se pontualmente a outros de forma a comparar os resultados.

### 3.1.2 Parâmetros analisados

Os resultados da resolução dos vários sistemas pelos métodos atrás referidos foram sempre registados através dos mesmos parâmetros. O parâmetro Iterações refere-se ao número de multiplicações matriz-vector efectuadas até se verificar a convergência. Para a generalidade dos métodos estes termos são equivalentes, i.e., em cada iteração é efectuada apenas uma multiplicação matriz-vector. Dos métodos aqui utilizados a única excepção é o BICG que efectua duas multiplicações por iteração. O parâmetro It/Itp, quando apresentado, consiste na razão entre o número de iterações, sem pré-condicionador (It), e o número de iterações com pré-condicionador (Itp). O parâmetro Tempo refere-se ao tempo de execução do *solver*, em segundos medidos pelo CPU. Este tempo adicionado do tempo de pré-condicionamento é designado pelo parâmetro Tempo Total. Finalmente o parâmetro Resíduo representa o valor de $||Ax^{(m)} - b||_2$ correspondente à Iteração $m$.

### 3.1.3 Critério de paragem

O critério de paragem é sempre definido da mesma forma, de acordo com a expressão

$$||r^{(m)}|| \leq \texttt{rtol} \times ||r^{(0)}|| + \texttt{atol}. \tag{3.1}$$

Sempre que o método não consegue cumprir este critério, são registados o Tempo e o Resíduo relativos ao número máximo de iterações permitidas, i.e., 1000 iterações.

Em todos os testes a aproximação inicial à solução é $x^{(0)} = 0$. Em consequência o vector resíduo inicial $r^{(0)} = b - Ax^{(0)}$ pode ser simplificado como $r^{(0)} = b$. O parâmetro `rtol`, tolerância relativa, é definido para todos os sistema como `rtol`$= 10^{-6}$. Da mesma maneira, a tolerância absoluta é definida por `atol`$= 10^{-10}$. A opção por este critério tem a vantagem de exigir o mesmo esforço aos *solvers*, qualquer que seja o sistema a resolver. Nos casos em que se opta por um critério de paragem fixo, corre-se o risco de exigir um esforço demasiado grande ou demasiado pequeno, de acordo com o valor de $b$.

### 3.1.4 Computador utilizado

Os testes apresentados neste capítulo foram todos realizados num computador pessoal com processador Intel MMX Pentium (400 MHz) com 128 Mbytes de RAM. O sistema operativo utilizado foi o Red Hat LINUX 6.2. O compilador FORTRAN é o GNU FORTRAN versão 0.2.24. Os tempos apresentados foram medidos pelo processador, através da função: `ETIME`.

## 3.2 Caracterização dos sistemas de teste

A caracterização dos sistemas de teste aqui apresentada é efectuada de acordo com alguns dos parâmetros descritos na secção 1.5.

### 3.2.1 Análise directa dos sistemas de teste

Apresentam-se os valores de alguns parâmetros calculados directamente a partir dos coeficientes e dos termos independentes dos sistemas resultantes da discretização da equação da pressão (tabela 3.1) e da velocidade (tabela 3.2).

| Sistema | $40p40$ | $80p80$ | $120p80$ | $120p120$ |
|---|---|---|---|---|
| N | 1444 | 6084 | 9204 | 13924 |
| NNZ | 7068 | 30108 | 45628 | 69148 |
| NNZ/(N×N) | 3.390E-03 | 8.134E-04 | 5.386E-04 | 3.335E-04 |
| Simetria | Sim | Sim | Sim | Sim |
| $LCD(\%)$ | 58.5 | 73.1 | 29.7 | 89.5 |
| $LRD(\%)$ | 58.5 | 73.1 | 29.7 | 89.5 |
| $\max\|a_{ij}\|$ | 0.485E-01 | 0.115E-01 | 0.761E-02 | 0.503E-02 |
| $\|A\|_\infty$ | 0.970E-01 | 0.230E-01 | 0.152E-01 | 0.101E-01 |
| $\|A\|_F$ | 0.199E+01 | 0.987E+00 | 0.818E+00 | 0.656E+00 |
| $\kappa_\infty(A)$ | 0.812E+01 | 0.793E+01 | 0.876E+01 | 0.790E+01 |
| $\|b\|_2$ | 1.441E-03 | 1.358E-03 | 1.050E-03 | 9.110E-04 |

Tabela 3.1: Caracterização dos sistemas do campo da pressão

Nos sistemas do campo da pressão (tabela 3.1) verifica-se a tendência para que o número de linhas e colunas diagonalmente dominantes ($LCD$ e $LRD$) aumente proporcionalmente a N. Esta tendência é no entanto quebrada pelo sistema $120p80$ que resulta de uma malha não-simétrica com $120 \times 80$ pontos.

Os resultados apresentados nas tabelas 3.1 e 3.2 mostram que os sistemas da velocidade possuem normas mais elevadas e menores números de condição do que os sistema da pressão. Além disso, todas as matrizes dos coeficientes são diagonalmente dominantes, contrariamente aos sistemas da pressão onde tal nunca ocorre. De acordo com estes resultados

| Sistema | $40u40$ | $80u80$ | $120u80$ | $120u120$ |
|---|---|---|---|---|
| N | 1444 | 6084 | 9204 | 13924 |
| NNZ | 7041 | 30108 | 45628 | 69148 |
| NNZ/(N×N) | 3.377E-03 | 8.134E-04 | 5.386E-04 | 3.335E-04 |
| Simetria | Não | Sim | Sim | Sim |
| $LCD(\%)$ | 100 | 100 | 100 | 100 |
| $LRD(\%)$ | 100 | 100 | 100 | 100 |
| $\max\|a_{ij}\|$ | 0.857E-01 | 0.857E-01 | 0.932E-01 | 0.857E-01 |
| $\|A\|_\infty$ | 0.106E+00 | 0.106E+00 | 0.115E+00 | 0.106E+00 |
| $\|A\|_F$ | 0.236E+01 | 0.478E+01 | 0.642E+01 | 0.720E+01 |
| $\kappa_\infty(A)$ | 0.344E+01 | 0.319E+01 | 0.314E+01 | 0.312E+01 |
| $\|b\|_2$ | 6.040E-02 | 1.015E-01 | 1.076E-01 | 1.336E-01 |

Tabela 3.2: Caracterização dos sistemas do campo da velocidade

e com a definição apresentada em 1.5.3 constatamos que todos os sistemas do campo da velocidade são definidos positivos. Verifica-se também que no conjunto de todos os sistemas apenas o sistema $40u40$ não é simétrico.

Como se pode observar pela diminuição do parâmetro NNZ/(N×N) à medida que a dimensão das matrizes aumenta (tabelas 3.1 e 3.2), as maiores matrizes são mais esparsas, tornando ainda mais vantajoso a utilização de métodos iterativos no contexto de problemas de CFD.

### 3.2.2 Análise espectral dos sistemas de teste

Apresentamos aqui uma análise dos sistemas de teste baseada nos valores próprios das matrizes dos coeficientes. Apresentam-se os gráficos de dispersão dos valores próprios para o único sistema que não é simétrico ($40u40$) e as curvas de distribuição características de cada um dos conjuntos de sistemas.

Todos os valores próprios dos sistemas da pressão se localizam sobre o eixo real porque as matrizes dos coeficientes destes sistemas são simétricas. A dispersão não varia muito de sistema para sistema pelo que incluímos apenas a curva representativa da distribuição desses valores para o sistema $120p120$ (figura 3.1) e incluem-se os parâmetros principais para cada matriz na tabela 3.3. Como é possível observar a curva de distribuição encontra-se encostada à origem devido ao facto de muitos dos valores próprios serem de pequena dimensão (tabela 3.3). Por exemplo no sistema $120p120$, o valor mínimo é $\lambda_{min} = -6.965E-12$.

Da análise dos resultados da tabela 3.3 observa-se que os valores da norma Frobenius calculados através dos valores próprios estão de acordo com os valores desta norma calculados através dos coeficientes da matriz (tabela 3.1). Verifica-se também que $\lambda_{max}$ é inferior a qualquer uma das normas calculadas. Estes aspectos são indiciadores da boa aproximação entre os valores próprios reais e os valores próprios estimados pelas rotinas da biblioteca

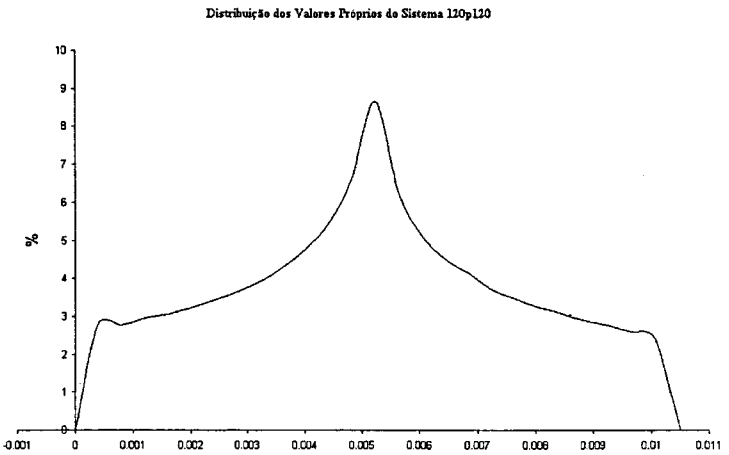



Figura 3.1: Densidade de probabilidade dos valores próprios do sistema 120$p$120

| Sistema | 40$p$40 | 80$p$80 | 120$p$80 | 120$p$120 |
|---|---|---|---|---|
| $\|A\|_F$ | 1.991E+00 | 9.871E-01 | 8.181E-01 | 6.561E-01 |
| $\lambda_{max}$ | 9.679E-02 | 2.300E-02 | 1.521E-02 | 1.005E-02 |
| $\lambda_{min}$ | 3.220E-11 | -5.434E-12 | -1.237E-11 | -6.965E-12 |
| $\kappa_2(A)$ | 3.006E+09 | 4.233E+09 | 1.230E+09 | 1.443E+09 |

Tabela 3.3: Caracterização espectral dos sistemas do campo da pressão

*Lapack.*

Verifica-se igualmente que os valores de $\lambda_{min}$ em todos os sistemas da pressão estão muito próximo de 0. Pela definição enunciada em 1.5.5, apenas o sistema 40$p$40 poderá ser considerado definido positivo. Contudo como os valores próprios aqui apresentados são estimativas e não valores exactos, dado que $\lambda_{min} \approx 0$ consideramos que todos estes sistemas são aproximadamente definidos semi-positivos.

O facto de muitos dos valores próprios dos sistemas derivados da pressão estarem próximos de 0 implica que as matrizes dos coeficientes apresentem um elevado grau de singularidade. Este aspecto traduz-se nos elevados valores dos números de condição $\kappa_2(A)$. Comparando estes valores com os obtidos a partir da norma infinita (tabela 3.1) verifica-se que $\kappa_2(A) \gg \kappa_\infty(A)$ para todos os sistemas.

Os valores próprios do sistema 40$u$40 têm componente imaginária porque $A$ não é simétrica e distribuem-se de forma simétrica em relação ao eixo real porque ocorrem aos pares conjugados (figura 3.2). É possível verificar igualmente que a distribuição da componente real se encontra afastada da origem, ao contrário do que acontecia com as matrizes da pressão.

Os valores próprios dos sistemas da velocidade, com excepção do sistema 40$u$40 têm apenas componente real, uma vez que são simétricos. A curva de distribuição destes valores varia pouco de sistema para sistema. Apresentamos a título representativo a curva

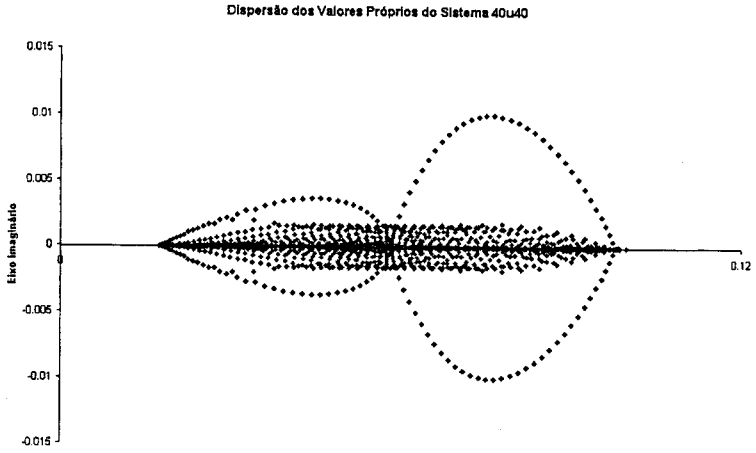



Figura 3.2: Dipersão dos valores próprios do sistema $40u40$

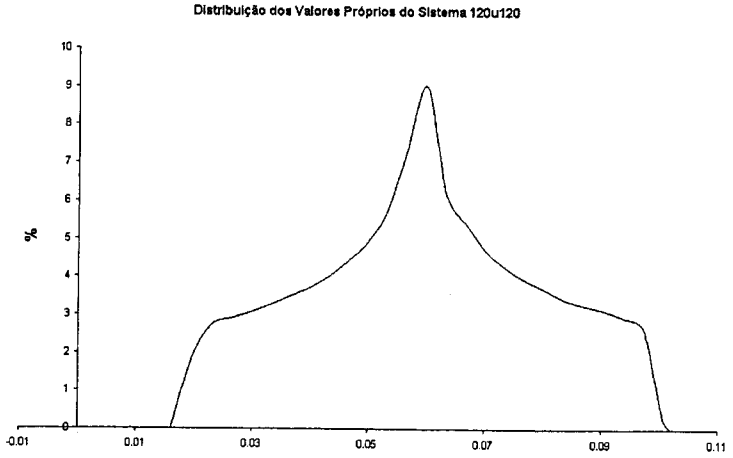



Figura 3.3: Densidade de probabilidade dos valores próprios do sistema $120u120$

relativa ao sistema $120u120$ na figura 3.3. É notório que a dispersão dos valores próprios se encontra mais afastada em relação à origem comparativamente com os sistemas da pressão (figura 3.1). O valor de $\lambda_{min}$ para o sistema $120u120$ é superior ao valor de $\lambda_{max}$ para o sistema $120p120$, o que ilustra bem as diferenças entre as matrizes dos coeficientes dos dois conjuntos de sistemas.

| Sistema | $40u40$ | $80u80$ | $120u80$ | $120u120$ |
|---|---|---|---|---|
| $\|A\|_F$ | 2.357E+00 | 4.777E+00 | 6.420E+00 | 7.199E+00 |
| $\|\lambda_{max}\|$ | 9.973E-02 | 1.015E-01 | 1.093E-01 | 1.006E-01 |
| $\|\lambda_{min}\|$ | 1.732E-02 | 1.518E-02 | 1.742E-02 | 1.618E-02 |
| $\kappa_2(A)$ | 5.772E-00 | 6.686E+00 | 6.274E+00 | 6.218E+00 |

Tabela 3.4: Caracterização espectral dos sistemas do campo da velocidade

Todos os parâmetros apresentados na tabela 3.4 estão de acordo com os parâmetros da

tabela 3.2, tal como se tinha verificado para os sistemas da variável pressão. As normas Frobenius calculadas a partir dos coeficientes da matriz $A$ e dos seus valores próprios são iguais. Os valores de $\lambda_{max}$ são todos inferiores às normas Frobenius e infinita. A razão pela qual apresentamos na tabela 3.4 os valores máximos e mínimos dos valores próprios em valor absoluto, está relacionado com o sistema $40u40$ onde existe uma componente imaginária. Assim para este sistema $|\lambda_{max}|$ refere-se à distância máxima entre a origem e um valor próprio ao longo do plano complexo e $|\lambda_{min}|$ à distância mínima. Para os restantes sistemas, dado que os valores próprios são todos positivos, o módulo não altera o significado destes parâmetros.

Da análise da tabela 3.4 verifica-se que todas as matrizes dos coeficientes dos sistemas das velocidades possuem unicamente valores próprios positivos e consequentemente são definidas positivas. Os valores de $\lambda_{min}$ são muito mais elevados do que para os sistemas da pressão (tabela 3.3). Este aspecto significa que as matrizes dos coeficientes destes sistemas são muito menos singulares do que as matrizes pertencentes aos sistemas da variável pressão. Em consequência os números de condição são também muito menores.

Desta análise sumária das características dos sistemas de equações é de prever a maior dificuldade de resolução dos sistemas de equações de pressão, do que na resolução dos sistemas de equações da velocidade.

## 3.3 Sistemas resultantes da discretização do domínio em 40×40 pontos

Apresentam-se de seguida os resultados obtidos na resolução dos sistemas da pressão ($40p40$) e da velocidade horizontal ($40u40$) provenientes da discretização do domínio em $40 \times 40$ pontos.

### 3.3.1 Resolução do sistema $40p40$

Da discretização homogénea do domínio em $40 \times 40$ pontos de que resultaram $38 \times 38$ equações. Esta foi de todas as malhas utilizadas aquela com menor número de pontos. A sua pequena dimensão é revelada pela rapidez com que o sistema é resolvido (tempo inferior a 1 segundo na maior parte dos casos, tabela 3.5), impossibilitando uma avaliação rigorosa do desempenho dos métodos utilizados. No entanto este mesmo motivo facilitou o conhecimento e familiarização mais rápida com a biblioteca *Sparskit*. Este sistema foi resolvido por todos os *solvers*, já referidos (subsecção 3.1.1), com e sem pré-condicionamento. Os pré-condicionadores utilizados foram: ILUD(0.01,0.75), ILUDP(0.01,0.75), ILUTP(0.001,0.15) e ILU(4).

Os resultados são agrupados em tabelas onde se ilustra o desempenho de todos os métodos iterativos utilizados e a influência da dimensão do sub-espaço de Krylov no caso dos métodos FOM, GMRES e DQGMRES (tabela 3.6), e o efeito dos pré-condicionadores

ILUD, ILUDP, ILUTP e ILU(4), nas tabelas 3.7, 3.8, 3.9 e 3.10 respectivamente.

| Método | Iterações | Tempo (s) | Resíduo |
|---|---|---|---|
| TDMA | 1000 | 4.13 | 9.698E-09 |
| CG | 152 | 0.09 | 1.349E-09 |
| BICG | 303 | 0.17 | 1.349E-09 |
| CGNR | 1000 | 0.52 | 3.638E-04 |
| BICGSTAB | 203 | 0.11 | 1.366E-09 |
| TFQMR | 236 | 0.15 | 3.471E-10 |
| FOM(10) | 893 | 0.84 | 1.545E-09 |
| GMRES(10) | 827 | 0.77 | 1.569E-09 |
| DQGMRES(10) | 133 | 0.32 | 1.566E-09 |

Tabela 3.5: Resolução do sistema $40p40$ sem pré-condicionamento

O TDMA não converge em menos de 1000 iterações, apesar de apresentar um resíduo próximo do desejado, mas com tempo elevado (tabela 3.5). O TDMA tem um desempenho pior do que os restantes métodos não pré-condicionados, com excepção do CGNR. Como já foi referido no capítulo anterior (secção 2.3), o número de condição da matriz dos coeficientes da equação normal $A^TAx = A^Tb$, é igual ao quadrado do número de condição da matriz original $A$. Quando $A$ é mal condicionada, a matriz $A^TA$ é ainda pior condicionada. Esta é a razão pela qual o CGNR não converge na resolução do sistema $40p40$ sem pré-condicionamento. Utilizando um pré-condicionador, o CGNR converge em relativamente poucas iterações. Os tempos e o número de iterações obtidos pelo CGNR para os vários pré-condicionadores estão próximos dos conseguidos pelo BICG nas mesma condições (ver tabelas 3.7 a 3.10), apesar do BICG convergir sem pré-condicionamento.

Verifica-se também que o BICG efectua o dobro das multiplicações matriz-vector (Iterações) efectuadas pelo CG, sempre que ambos convergem. Como já foi referido na secção 2.4, isto deve-se ao facto do BICG efectuar em cada iteração uma multiplicação com $A$ e outra com $A^T$. Tendo em conta o tempo, o CG é obviamente 2 vezes mais rápido. Em contrapartida o BICG é menos sensível ao método de pré-condicionamento, pois converge em todas as situações analisadas. O CG não converge quando são aplicados os pré-condicionadores ILUD(0.01,0.75) (ver tabela 3.7) e ILUTP(0.0001,15)(ver tabela 3.9). Mas se utilizarmos o mesmo pré-condicionador ILUD, mas com pivotagem parcial (tabela 3.8), o CG converge novamente. Como a convergência deste método depende muito do condicionamento do sistema e de este ser ou não simétrico definido positivo, observamos que estes dois pré-condicionadores não garantem a permanência destas características quando aplicados ao sistema $40p40$. A pivotagem parcial na construção do ILUD consegue reverter a situação (tabela 3.8), enquanto que para o ILUT esta não tem efeito nenhum (tabela 3.9). Sem pré-condicionamento, o CG é o método que converge mais rapidamente (tabela 3.5).

O BICGSTAB converge em todas as situações. É sempre mais rápido do que o BICG. Quando pré-condicionado é ligeiramente mais rápido do que o CG. É também um pouco mais rápido do que o TFQMR, com excepção do pré-condicionamento ILU(4) (tabela 3.10),

| Método | k | Iterações | Tempo | Resíduo |
|---|---|---|---|---|
| FOM | 1 | 596 | 0.70 | 1.551E-09 |
| | 2 | 1000 | 0.59 | 4.991E-05 |
| | 3 | 1000 | 0.63 | 1.634E-05 |
| | 5 | 1000 | 0.69 | 6.243E-07 |
| | 10 | 893 | 0.84 | 1.545E-09 |
| | 15 | 596 | 0.71 | 1.551E-09 |
| | 20 | 485 | 0.77 | 1.521E-09 |
| | 30 | 343 | 0.68 | 1.541E-09 |
| GMRES | 1 | 573 | 0.67 | 1.539E-09 |
| | 2 | 1000 | 0.58 | 3.963E-05 |
| | 3 | 1000 | 0.62 | 7.789E-06 |
| | 5 | 1000 | 0.70 | 2.815E-07 |
| | 10 | 827 | 0.77 | 1.569E-09 |
| | 15 | 573 | 0.78 | 1.539E-09 |
| | 20 | 452 | 0.69 | 1.562E-09 |
| | 30 | 357 | 0.79 | 1.565E-09 |
| DQGMRES | 1 | 133 | 0.47 | 1.566E-09 |
| | 2 | 133 | 0.13 | 1.566E-09 |
| | 3 | 133 | 0.14 | 1.566E-09 |
| | 5 | 133 | 0.17 | 1.566E-09 |
| | 10 | 133 | 0.32 | 1.566E-09 |
| | 15 | 133 | 0.47 | 1.566E-09 |
| | 20 | 133 | 0.61 | 1.566E-09 |
| | 30 | 133 | 0.91 | 1.566E-09 |

Tabela 3.6: Efeito da dimensão do sub-espaço de Krylov $k$ na resolução do sistema $40p40$ pelos métodos FOM, GMRES e DQGMRES sem pré-condicionamento.

onde este último utiliza menos duas iterações e menos 0.01 segundos. O TFQMR tem assim um desempenho semelhante ao do BICGSTAB, sendo também mais rápido do que o CG, quando pré-condicionado.

Analisando o efeito da dimensão do sub-espaço de Krylov sobre o FOM($k$) e GMRES($k$) não pré-condicionados (tabela 3.6) verifica-se que estes têm um comportamento semelhante. Não convergem para valores de $k$ baixos, com excepção de $k = 1$, melhorando o seu desempenho para $k \geq 10$. Quanto maior for o valor de $k$ menos iterações são necessárias para convergir, mas a partir de determinada ordem de grandeza, o elevado número de vectores, em relação aos quais é feita a ortogonalização, faz com que o tempo necessário a cada iteração seja também ele elevado. Nesta situação a redução do número de iterações provocada pelo aumento de $k$ não compensa o aumento de tempo necessário a cada uma destas iterações. Por exemplo na tabela 3.6 pode observar-se que o número de iterações do GMRES($k$) baixa de 827 para 573 ao aumentar de $k = 10$ para $k = 15$, contudo o tempo de execução aumenta de 0.77 para 0.78 segundos. O mesmo se passa com o FOM($k$) quando

| Método | Iterações | It/Itp | Tempo | Tempo Total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0 | 2.01 | 2.03 | 1.888E-06 |
| BICG | 54 | 6 | 0.11 | 0.13 | 9.120E-10 |
| CGNR | 55 | 18 | 0.11 | 0.13 | 6.536E-10 |
| BICGSTAB | 21 | 10 | 0.04 | 0.06 | 1.710E-10 |
| TFQMR | 21 | 11 | 0.05 | 0.07 | 4.909E-10 |
| FOM(10) | 19 | 47 | 0.05 | 0.07 | 8.036E-10 |
| GMRES(10) | 19 | 44 | 0.04 | 0.06 | 6.505E-10 |
| DQGMRES(10) | 16 | 8 | 0.05 | 0.07 | 6.881E-10 |
| PGMRES(10) | 17 | - | 0.06 | 0.08 | 6.505E-10 |

Tabela 3.7: Resolução do sistema 40$p$40 pré-condicionado com ILUD(0.01,0.75). O tempo de construção do pré-condicionador foi 0.02 segundos.

se varia de $k = 10$ para $k = 15$. Outra consequência importante é a quantidade de recursos de memória adicional exigida à medida que $k$ aumenta.

O FOM($k$) e o GMRES($k$) melhoram os seus desempenhos quando o sistema é pré-condicionado. Nesses casos convergem em tempos e iterações iguais, ou mesmo inferiores, aos obtidos pelo BICGSTAB (tabela 3.9). Os resultados mostram que o pré-condicionamento é um aspecto essencial para o bom funcionamento destes dois métodos.

| Método | Iterações | It/Itp | Tempo | Tempo Total | Resíduo |
|---|---|---|---|---|---|
| CG | 20 | 8 | 0.06 | 0.09 | 1.425E-09 |
| BICG | 42 | 7 | 0.12 | 0.15 | 2.549E-10 |
| CGNR | 37 | 27 | 0.11 | 0.14 | 1.011E-09 |
| BICGSTAB | 13 | 16 | 0.04 | 0.07 | 1.092E-09 |
| TFQMR | 17 | 14 | 0.05 | 0.08 | 1.958E-10 |
| FOM(10) | 13 | 69 | 0.04 | 0.07 | 5.145E-10 |
| GMRES(10) | 13 | 64 | 0.05 | 0.08 | 5.009E-10 |
| DQGMRES(10) | 12 | 11 | 0.05 | 0.08 | 3.308E-10 |
| PGMRES(10) | 11 | - | 0.05 | 0.08 | 0.501E-09 |

Tabela 3.8: Resolução do sistema 40$p$40 pré-condicionado com ILUDP(0.01,0.75). O tempo de construção do pré-condicionador foi 0.03 segundos.

Contrariamente ao FOM($k$) e ao GMRE($k$), o DQGMRES($k$) converge sempre no mesmo número de iterações, qualquer que seja o valor de $k$ (tabela 3.6). Em termos de tempo utilizado, o melhor resultado surge para $k = 2$, onde converge em 0.13 segundos. Estes resultados obtidos sem pré-condicionador colocam o DQGMRES(2) na posição imediatamente a seguir ao CG que converge em 0.09 segundos (tabela 3.5). Contudo o DQGMRES(2) converge em menos iterações do que o CG. Quando pré-condicionado, o DQGMRES(10) obtém resultados semelhantes aos obtidos pelo GMRES(10), convergindo também qualquer

que seja o pré-condicionador utilizado.

| Método | Iterações | It/Itp | Tempo | Tempo Total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0 | 4.38 | 4.44 | 2.030E-04 |
| BICG | 32 | 9 | 0.12 | 0.18 | 2.635E-10 |
| CGNR | 29 | 34 | 0.11 | 0.17 | 3.950E-10 |
| BICGSTAB | 11 | 18 | 0.05 | 0.11 | 2.109E-10 |
| TFQMR | 13 | 18 | 0.07 | 0.13 | 2.267E-10 |
| FOM(10) | 9 | 99 | 0.05 | 0.11 | 1.456E-09 |
| GMRES(10) | 9 | 92 | 0.04 | 0.10 | 1.412E-09 |
| DQGMRES(10) | 9 | 15 | 0.05 | 0.11 | 1.412E-09 |
| PGMRES(10) | 8 | - | 0.06 | 0.11 | 1.412E-09 |

Tabela 3.9: Resolução do sistema $40p40$ pré-condicionado com ILUTP(0.0001,15). O tempo de construção do pré-condicionador foi 0.06 segundos.

Nos casos analisados o PGMRES(10) tem um comportamento semelhante ao do GMRES(10). Utiliza uma ou duas iterações a menos do que o GMRES(10) com excepção do pré-condicionador ILU(4), em que utiliza mais uma (tabela 3.10). É, no entanto, ligeiramente mais lento.

| Método | Iterações | It/Itp | Tempo | Tempo Total | Resíduo |
|---|---|---|---|---|---|
| CG | 28 | 5 | 0.06 | 0.08 | 1.489E-09 |
| BICG | 50 | 6 | 0.12 | 0.14 | 8.704E-10 |
| CGNR | 55 | 18 | 0.12 | 0.14 | 1.530E-09 |
| BICGSTAB | 23 | 9 | 0.06 | 0.08 | 4.162E-10 |
| TFQMR | 21 | 11 | 0.05 | 0.07 | 9.995E-10 |
| FOM(10) | 15 | 60 | 0.05 | 0.07 | 1.460E-09 |
| GMRES(10) | 15 | 55 | 0.05 | 0.07 | 1.405E-09 |
| DQGMRES(10) | 15 | 9 | 0.06 | 0.08 | 1.405E-09 |
| PGMRES(10) | 16 | - | 0.07 | 0.09 | 0.735E-09 |

Tabela 3.10: Resolução do sistema $40p40$ pré-condicionado com ILU(4). O tempo de construção do pré-condicionador foi 0.02 segundos.

Todos os pré-condicionadores aqui utilizados têm o efeito de reduzir o número de iterações dos métodos iterativos, com excepção do CG. O ILUTP(0.0001,15) consegue uma redução generalizada (tabela 3.9), fazendo com que o FOM($k$) e GMRES($k$) realizem aproximadamente 100 vezes menos iterações do que sem pré-condicionador. Contudo o tempo de construção deste pré-condicionador (0.06 segundos) é superior ao dos restantes. Este aspecto penaliza o rendimento global do método iterativo como BICG e BICGSTAB que baixam os tempos de convergência que apresentam um maior tempo total.

O tempo total mais baixo (0.06 segundo) obtido na resolução do sistema $40p40$ é devido ao BICGSTAB e GMRES(10) associados ao ILUD(0.01,75) (tabela 3.7). Utilizando a pivotagem parcial na elaboração deste pré-condicionador, consegue-se a convergência do CG, em detrimento dum ligeiro aumento do tempo total dos restantes métodos (tabela 3.8). O pré-condicionador ILU(4) é responsável pela convergência de todos os métodos, em tempos totais muito próximos dos melhores (tabela 3.10).

### 3.3.2 Resolução do sistema $40u40$

Apresentam-se nesta secção os resultados relativos à resolução do sistema proveniente do campo da velocidade, num domínio com 40 × 40 pontos. Este sistema é resolvido pelos mesmos métodos iterativos do que o sistema anterior, com e sem pré-condicionamento. Os pré-condicionadores utilizados são o ILUD(0.01,0.75), ILU(4) e ILU(0). O critério de paragem, definido de acordo com os parâmetros previamente estipulados, consiste em $\|r^{(k)}\|_2 \leq 1.126 \times 10^{-7}$.

| Método | Iterações | Tempo (s) | Resíduo |
|---|---|---|---|
| TDMA | 7 | 0.03 | 3.923E-08 |
| CG | 1000 | 0.61 | 5.097E-04 |
| BICG | 40 | 0.02 | 8.173E-08 |
| CGNR | 81 | 0.04 | 8.276E-08 |
| BICGSTAB | 21 | 0.01 | 8.619E-08 |
| TFQMR | 27 | 0.01 | 7.338E-08 |
| FOM(10) | 21 | 0.02 | 6.342E-08 |
| GMRES(10) | 21 | 0.02 | 5.582E-08 |
| DQGMRES(10) | 19 | 0.03 | 1.029E-07 |

Tabela 3.11: Resolução do sistema $40u40$ sem pré-condicionamento.

Da análise da globalidade dos resultados obtidos na resolução deste sistema e em particular dos obtidos sem pré-condicionamento (tabela 3.11), verifica-se que este sistema é de muito mais fácil resolução do que o anterior. Todos os métodos que permitem a resolução de sistemas não-simétricos, convergem com um número reduzido de iterações, inclusive o TDMA cujos tempos de execução variam entre 0.01 e 0.04 segundos. A razão deve-se ao bom condicionamento deste sistema, analisado na secção 3.2. O seu pré-condicionamento tem o efeito de reduzir a matriz dos coeficientes a uma matriz próxima da identidade. Em consequência dos tempos de resolução sem pré-condicionador serem muito baixos os resultados com pré-condicionamento nem sempre conseguem tempos totais mais baixos (tabelas 3.12 a 3.14).
O CG não converge sem pré-condicionamento porque o sistema $40u40$ não é simétrico. Mas converge quando associado a qualquer um dos três pré-condicionadores utilizados na resolução deste sistema. O melhor resultado deste método ocorre quando aplicado em conjunto com o ILU(4) (tabelas 3.13).

| Método | Iterações | It/Itp | Tempo (s) | Resíduo |
|---|---|---|---|---|
| CG | 5 | 200 | 0.01 | 2.790E-09 |
| BICG | 8 | 5 | 0.01 | 3.135E-09 |
| CGNR | 9 | 9 | 0.01 | 3.234E-08 |
| BICGSTAB | 5 | 4 | 0.01 | 2.347E-09 |
| TFQMR | 5 | 5 | 0.01 | 1.750E-08 |
| FOM(10) | 5 | 4 | 0.01 | 2.280E-09 |
| GMRES(10) | 5 | 4 | 0.01 | 2.280E-09 |
| DQGMRES(10) | 5 | 4 | 0.01 | 2.280E-09 |
| PGMRES(10) | 4 | - | 0.01 | 0.228E-08 |

Tabela 3.12: Resolução do sistema $40u40$ pré-condicionado com ILUD(0.01,0.75). O tempo de construção do pré-condicionador foi inferior a 0.001 segundos.

O BCGSTAB e o TFQMR apresentam tempos muito próximos, diferindo apenas no número de Iterações quando não é utilizado um pré-condicionador (tabelas 3.11).

| Método | Iterações | It/Itp | Tempo | Tmp Total | Resíduo |
|---|---|---|---|---|---|
| CG | 3 | 333 | 0.00 | 0.02 | 1.583E-08 |
| BICG | 4 | 10 | 0.01 | 0.03 | 2.194E-08 |
| CGNR | 5 | 16 | 0.01 | 0.03 | 5.614E-08 |
| BICGSTAB | 3 | 7 | 0.01 | 0.03 | 1.519E-08 |
| TFQMR | 3 | 9 | 0.01 | 0.03 | 1.860E-08 |
| FOM(10) | 3 | 7 | 0.01 | 0.03 | 1.503E-08 |
| GMRES(10) | 3 | 7 | 0.01 | 0.03 | 1.503E-08 |
| DQGMRES(10) | 3 | 6 | 0.01 | 0.03 | 1.503E-08 |
| PGMRES(10) | 2 | - | 0.01 | 0.03 | 1.503E-08 |

Tabela 3.13: Resolução do sistema $40u40$ pré-condicionado com ILU(4). O tempo de construção do pré-condicionador foi 0.02 segundos.

O FOM(10) e o GMRES(10) apresentam os mesmos valores para todas as situações e fazem parte dos métodos mais rápidos, tanto em Iterações como em Tempo.

Todos os métodos convergem em tempos inferiores à centésima de segundos, quando é utilizado o pré-condicionador ILU(4) (tabela 3.13). No entanto o tempo de construção deste é superior aos tempos utilizados pelos *solvers* (0.02 s), pelo que a sua aplicação não se justifica. Baixando o nível de enchimento deste pré-condicionador para o seu nível mais baixo ($l = 0$), o tempo de construção deste pré-condicionador baixa para valores inferiores às centésimas de segundos. Apesar da sua simplicidade o pré-condicionador ILU(0) é muito eficiente pois baixa os Tempos Totais do CG e do BICG para 0.02 segundos e os dos restantes métodos para 0.01 segundos (tabela 3.14).

| Método | Iterações | It/Itp | Tempo (s) | Resíduo |
|---|---|---|---|---|
| CG | 5 | 200 | 0.02 | 2.790E-09 |
| BICG | 8 | 4 | 0.02 | 3.135E-09 |
| CGNR | 9 | 9 | 0.01 | 3.234E-08 |
| BICGSTAB | 5 | 4 | 0.01 | 2.347E-09 |
| TFQMR | 5 | 5 | 0.01 | 1.750E-08 |
| FOM(10) | 5 | 4 | 0.01 | 2.280E-09 |
| GMRES(10) | 5 | 4 | 0.01 | 2.280E-09 |
| DQGMRES(10) | 5 | 4 | 0.01 | 2.280E-09 |
| PGMRES(10) | 4 | - | 0.01 | 2.280E-09 |

Tabela 3.14: Resolução do sistema $40u40$ pré-condicionado com ILU(0). O tempo de construção do pré-condicionador foi 0.00 segundos.

## 3.4 Sistemas resultantes da discretização do domínio em 80×80 pontos

Apresentamos nesta secção (subsecções 3.4.1 e 3.4.2) os resultados obtidos na resolução dos sistemas da pressão ($80p80$) e da velocidade horizontal ($80u80$) provenientes da discretização do domínio em $80 \times 80$ pontos.

### 3.4.1 Resolução do sistema $80p80$

Os pré-condicionadores utilizados foram ILU(4), ILU(6) e ILUDP(0.01,0.75). É também feita uma análise da variação do parâmetro $\ell$ no pré-condicionador ILU($\ell$) e do parâmetro $\alpha$ no pré-condicionador ILUDP(0.01,$\alpha$). O critério de paragem é $\|r^{(k)}\|_2 \leq 1.37 \times 10^{-9}$, de acordo com o procedimento habitual.

Para além dos *solvers* utilizados na resolução dos sistemas anteriores, o DQGMRES(2) é também utilizado para resolver este sistema com pré-condicionador, devido ao seu bom desempenho na resolução do sistema $40p40$ (ver tabela 3.6), confirmado na resolução do sistema $80p80$ (tabela 3.15). Verifica-se novamente que o número de iterações necessárias para convergir (263), se mantém constante qualquer que seja o valor de $k$ e que o tempo mínimo é conseguido para $k = 2$ (1.48 segundos).

Da análise dos resultados obtidos sem pré-condicionador (tabela 3.16) verifica-se que a resolução deste sistema apresenta um grau de dificuldade maior do que o anterior sistema das pressões (tabela 3.5). Efectivamente o TDMA, CGNR, FOM(10) e GMRES(10) não convergem dentro do limite permitido e os métodos que convergem fazem-no em mais iterações.

O método mais rápido sem pré-condicionador é novamente o CG que converge em 0.97 segundos (tabela 3.16). Este é também o menor de todos os tempos registados na

| Método | $k$ | Iterações | Tempo | Resíduo |
|---|---|---|---|---|
| FOM | 1 | 1000 | 6.79 | 8.392E-06 |
| | 2 | 1000 | 3.49 | 0.694E-03 |
| | 3 | 1000 | 3.71 | 0.697E-03 |
| | 5 | 1000 | 4.18 | 0.286E-03 |
| | 10 | 1000 | 5.38 | 4.154E-05 |
| | 15 | 1000 | 6.79 | 8.392E-06 |
| | 20 | 1000 | 7.92 | 8.935E-07 |
| | 30 | 1000 | 10.22 | 2.263E-08 |
| | 50 | 855 | 12.59 | 1.448E-09 |
| GMRES | 1 | 1000 | 6.79 | 3.099E-06 |
| | 2 | 1000 | 3.49 | 0.547E-03 |
| | 3 | 1000 | 3.70 | 0.351E-03 |
| | 5 | 1000 | 4.24 | 0.127E-03 |
| | 10 | 1000 | 5.37 | 1.771E-05 |
| | 15 | 1000 | 6.71 | 3.098E-06 |
| | 20 | 1000 | 7.93 | 3.803E-07 |
| | 30 | 1000 | 10.12 | 1.807E-08 |
| | 50 | 784 | 11.42 | 1.462E-09 |
| DQGMRES | 1 | 263 | 4.72 | 1.407E-09 |
| | 2 | 263 | 1.48 | 1.407E-09 |
| | 3 | 263 | 1.75 | 1.407E-09 |
| | 5 | 263 | 2.30 | 1.407E-09 |
| | 10 | 263 | 3.46 | 1.407E-09 |
| | 15 | 263 | 4.71 | 1.407E-09 |
| | 20 | 263 | 5.84 | 1.407E-09 |
| | 30 | 263 | 8.08 | 1.407E-09 |
| | 50 | 263 | 12.16 | 1.407E-09 |

Tabela 3.15: Efeito da dimensão do sub-espaço de Krylov $k$ na resolução do sistema $80p80$ pelos métodos FOM, GMRES e DQGMRES sem pré-condicionamento.

resolução do sistema $80p80$ com e sem pré-condicionador. Os resultados obtidos por este método pioram com a utilização de pré-condicionadores. Em todas as situações analisadas, a utilização de pré-condicionador reduziu o número de iterações, mas aumentou o tempo de convergência do CG. Por esta razão o pré-condicionamento deste sistema não se justifica quando resolvido pelo CG. A redução do número de iterações não é suficiente para compensar os tempos adicionais devidos à construção do pré-condicionador e às multiplicações de $A$ por este.

O BICG, tal como para o sistema $40p40$, converge utilizando aproximadamente o dobro das iterações e do tempo do que o CG. O BICGSTAB em todas as situações analisadas converge mais rapidamente do que o BICG. Comparativamente com o CG, o BICGSTAB é mais rápido apenas quando associado ao pré-condicionador ILUDP(0.01,0.75) em que

| Método | Iterações | Tempo | Resíduo |
|---|---|---|---|
| TDMA | 1000 | 17.75 | 3.069E-05 |
| CG | 271 | 0.97 | 1.412E-09 |
| BICG | 540 | 2.11 | 1.412E-09 |
| CGNR | 1000 | 33.33 | 1.101E-03 |
| BICGSTAB | 413 | 1.57 | 7.821E-10 |
| TFQMR | 517 | 2.38 | 3.071E-10 |
| FOM(10) | 1000 | 5.38 | 4.154E-05 |
| GMRES(10) | 1000 | 5.37 | 1.771E-05 |
| DQGMRES(10) | 263 | 3.46 | 1.407E-09 |
| DQGMRES(2) | 263 | 1.61 | 1.407E-09 |

Tabela 3.16: Resolução do sistema 80$p$80 sem pré-condicionamento

| Método | Iterações | It/Itp | Tempo | Tmp Total | Resíduo |
|---|---|---|---|---|---|
| CG | 40 | 7 | 1.10 | 1.41 | 1.015E-09 |
| BICG | 94 | 5 | 2.72 | 3.04 | 1.146E-09 |
| CGNR | 159 | 6 | 5.45 | 5.76 | 8.512E-10 |
| BICGSTAB | 43 | 10 | 1.19 | 1.50 | 1.404E-09 |
| TFQMR | 41 | 13 | 1.16 | 1.47 | 6.340E-10 |
| FOM(10) | 42 | 24 | 1.22 | 1.53 | 1.051E-09 |
| GMRES(10) | 47 | 21 | 1.37 | 1.68 | 1.138E-09 |
| DQGMRES(10) | 31 | 8 | 1.06 | 1.37 | 1.385E-09 |
| DQGMRES(2) | 40 | 7 | 1.13 | 1.34 | 1.278E-09 |
| PGMRES(10) | 42 | - | 1.40 | 1.71 | 1.138E-09 |

Tabela 3.17: Resolução do sistema 80$p$80 pré-condicionado com ILU(4). O tempo de construção do pré-condicionador foi 0.31 segundos.

converge em 1.27 segundos e o CG em 1.45 (tabela 3.20). Este é também o tempo mínimo apresentado pelo BICGSTAB de todas as situações analisadas para os sistema 80$p$80.

O TFQMR demora mais tempo para convergir do que o BICGSTAB se não for utilizado um pré-condicionador, 2.38 segundos para 1.57 segundos (tabela 3.16), caso contrário é mais rápido. O melhor resultado obtido pelo TFQMR (1.22 segundos) acontece quando associado ao ILUDP(0.01,0.75) (tabela 3.20).

O DQGMRES(10) converge em menos tempo do que DQGMRES(2) se for utilizado um pré-condicionador. Sem pré-condicionador o DQGMRES(2) demora menos de metade do tempo que o DQGMRES(10), convergindo, respectivamente, em 1.61 segundos e 3.46 segundos (tabela 3.16). O melhor resultado de ambos é conseguido com ILUDP(0.01,0.75), em que o DQGMRES(2) converge em 1.21 segundos e o DQGMRES(10) em 1.19 segundos (tabela 3.20).

Mais uma vez o desempenho do FOM(10) e do GMRES($k$) melhora consideravelmente com a utilização de pré-condicionadores. Como se pode observar na tabela 3.15, apenas convergem para $k = 50$, contrariamente ao que sucedia no sistema $40p40$ em que convergiam para $k = 10$. Conjugando a acção do FOM(10) e do GMRES(10) com um pré-condicionador obtêm-se resultados próximos dos melhores. O pré-condicionador com o qual estes dois métodos apresentam menores tempos e menor número de iterações é o ILUDP(0.01,0.75). Nesta situação convergem em menos iterações do que os outros métodos, respectivamente, 24 e 22, sendo o tempo do GMRES(10) (1.15 segundos) o menor de todos (tabela 3.20).

O PGMRES(10) apresenta também o mesmo comportamento já evidenciado na resolução do sistema $40p40$. Exige menos iterações do que o GMRES(10) mas o tempo de cálculo é superior, sendo contudo estas diferenças pouco significativas. Por exemplo quando é utilizado o pré-condicionador ILU(4), o GMRES(10) converge em 47 iterações e 1.68 segundos, enquanto o PGMRES(10) converge em 42 iterações e 1.71 segundos (tabela 3.17). Verifica-se assim que as iterações do PGMRES(10) são mais demoradas do que as do GMRES(10).

| Método | Iterações | It/Itp | Tempo | Tmp Total | Resíduo |
|---|---|---|---|---|---|
| CG | 32 | 8 | 1.10 | 1.67 | 8.188E-10 |
| BICG | 66 | 8 | 2.40 | 2.97 | 3.406E-10 |
| CGNR | 89 | 11 | 3.18 | 3.37 | 6.650E-10 |
| BICGSTAB | 33 | 13 | 1.13 | 1.70 | 3.661E-10 |
| TFQMR | 27 | 19 | 0.96 | 1.53 | 1.619E-09 |
| FOM(10) | 22 | 45 | 0.80 | 1.37 | 1.230E-09 |
| GMRES(10) | 24 | 42 | 0.87 | 1.44 | 1.171E-09 |
| DQGMRES(10) | 21 | 13 | 0.84 | 1.41 | 4.616E-10 |
| DQGMRES(2) | 28 | 9 | 0.96 | 1.53 | 1.481E-09 |
| PGMRES(10) | 21 | - | 0.89 | 1.46 | 1.171E-09 |

Tabela 3.18: Resolução do sistema $80p80$ pré-condicionado com ILU(6). O tempo de construção do pré-condicionador foi 0.57 segundos.

O pré-condicionamento do sistema $80p80$ com ILU(4) revela-se muito eficiente, como se pode observar na tabela 3.17, com redução do número de iterações de 5 a 24 vezes em relação ao caso sem pré-condicionador (tabela 3.16). Mas como o tempo de construção do pré-condicionador permanece inferior aos tempos de execução dos *solvers*, ainda é possível baixar os tempos totais aumentando a precisão do pré-condicionador ILU($\ell$). A figura 3.4, com os tempos de execução dos vários métodos em função do nível de enchimento $\ell$ utilizado por este pré-condicionador, mostra que o CGNR converge mais rapidamente para $\ell = 7$ e que para valores superiores a este, todos os métodos convergem mais lentamente. Isto acontece à medida que o tempo de pré-condicionamento se torna superior ao tempo de execução do *solver*, dominando o processo iterativo. Verifica-se também que o CG e o BICG obtêm melhores resultados sem pré-condicionador. Para estes dois métodos, a redução do número de iterações não é suficiente para justificar este pré-condicionamento. Outro

aspecto que se verifica com o aumento de $\ell$ é a alteração do desempenho relativo dos métodos. Por exemplo com o ILU(4) o método mais rápido é o DQGMRES(10) mas com o ILU(6) (tabela 3.18) é o FOM(10). Em consequência destes resultados verifica-se que a optimização do pré-condicionador ILU($\ell$) não pode ser efectuada em conjunto para todos os métodos pois o valor do nível de enchimento $\ell$ óptimo depende do método utilizado.

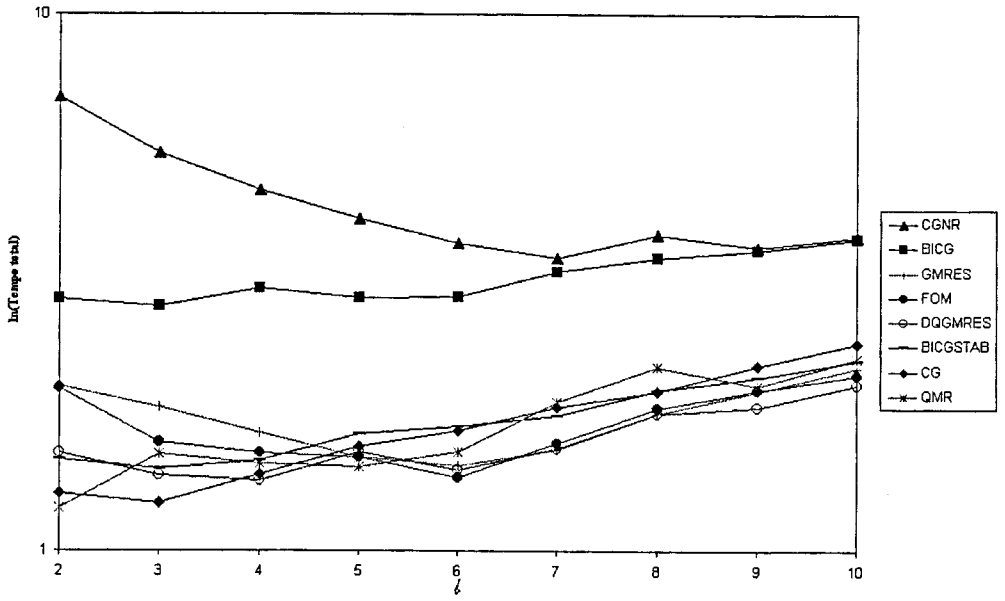



Figura 3.4: Efeito do nível de enchimento do pré-condicionador ILU($\ell$) na resolução do sistema 80$p$80.

A análise efectuada para o ILU($\ell$) foi repetida para o caso do ILUDP(0.01,$\alpha$) relativamente ao parâmetro, $\alpha$. A tabela 3.19 apresenta os tempos totais obtidos pelos vários *solvers*, quando pré-condicionados pelo ILUDP(0.01,$\alpha$), fazendo variar o coeficiente de compensação $\alpha$ e mantendo `atol`=0.01. Verifica-se que 8 dos 10 métodos analisados convergem em menos tempo ocorrem quando $\alpha \simeq 0.75$. As excepções são o BICG, que melhora à medida que $\alpha$ aumenta, e o BICGSTAB que apresenta um tempo ligeiramente mais baixo para $\alpha = 0.5$ (tabela 3.19).

O tempo de pré-condicionamento ILUDP(0.01,0.75), 0.42 segundos, é maior do que o ILU(4), 0.31 segundos, e menor do que o ILU(6), 0.57 segundos. Todos eles possibilitam a convergência de todos os métodos. Mas comparando os tempos totais, verifica-se que os resultados obtidos com o ILUDP(0.01,0.75) são melhores do que os obtidos com o ILU(6). Portanto, apesar do tempo adicional de pré-condicionamento do ILUDP(0.01,0.75) os *solvers* são mais eficazes e o tempo total de todos os métodos (tabela 3.20) é inferior ao obtido com com o ILU(4), na tabela 3.18. A utilização do pré-condicionador ILUDP(0.01,0.75) conduz a reduções do número de iterações entre 8 a 45 vezes (tabela 3.20) em relação ao caso sem pré-condicionador. Em relação ao ILU(4) também são melhores, apesar de não existir tanta diferença e de se verificar uma excepção com o CG. Este apresenta um tempo total de convergência com ILUDP(0.01,0.75) maior do que o obtido com ILU(4).

| $\alpha$ | 0 | 0.5 | 0.75 | 1 |
|---|---|---|---|---|
| CG | 1.71 | 1.51 | 1.45 | 3.77 |
| BICG | 2.97 | 2.62 | 2.45 | 2.15 |
| CGNR | 4.75 | 3.75 | 2.97 | 6.01 |
| BICGSTAB | 1.44 | 1.26 | 1.27 | 1.54 |
| TFQMR | 1.53 | 1.23 | 1.22 | 1.51 |
| FOM(10) | 1.57 | 1.22 | 1.21 | 1.56 |
| GMRES(10) | 1.59 | 1.29 | 1.15 | 1.57 |
| DQGMRES(10) | 1.37 | 1.22 | 1.19 | 1.56 |
| DQGMRES(2) | 1.43 | 1.32 | 1.21 | 1.84 |
| PGMRES(10) | 3.54 | 2.76 | 2.42 | 3.39 |

Tabela 3.19: Efeito do factor de compensação $\alpha$ sobre o tempo total de resolução do sistema $80p80$ pré-condicionado com ILUDP(0.01,$\alpha$).

| Método | Iterações | It/Itp | Tempo | Tmp Total | Resíduo |
|---|---|---|---|---|---|
| CG | 33 | 8 | 1.03 | 1.45 | 1.249E-09 |
| BICG | 62 | 9 | 2.03 | 2.45 | 1.075E-09 |
| CGNR | 79 | 13 | 2.55 | 2.97 | 1.330E-09 |
| BICGSTAB | 27 | 15 | 0.85 | 1.27 | 2.519E-10 |
| TFQMR | 25 | 21 | 0.80 | 1.22 | 1.254E-09 |
| FOM(10) | 24 | 42 | 0.79 | 1.21 | 1.211E-09 |
| GMRES(10) | 22 | 45 | 0.73 | 1.15 | 1.381E-09 |
| DQGMRES(10) | 21 | 13 | 0.77 | 1.19 | 1.037E-09 |
| DQGMRES(2) | 25 | 11 | 0.78 | 1.21 | 6.954E-10 |
| PGMRES(10) | 20 | - | 0.74 | 1.16 | 1.382E-09 |

Tabela 3.20: Resolução do sistema $80p80$ pré-condicionado com ILUDP(0.01,0.75). O tempo de construção do pré-condicionador foi 0.42 segundos.

### 3.4.2 Resolução do sistema $80u80$

O sistema $80u80$ foi resolvido utilizando os métodos habituais juntamente com o pré-condicionador ILU(0). O critério de paragem utilizado por todos os métodos na resolução deste sistema é $\|r^{(k)}\|_2 \leq 1.812 \times 10^{-7}$.

Tal como no sistema $40u40$ (secção 3.3.2), este sistema é rapidamente resolvido por todos os métodos. Sem pré-condicionamento (tabela 3.21), o CG e o BICGSTAB são os métodos mais rápidos, convergindo ambos em 19 iterações e 0.07 segundos. O mais lento é o CGNR que converge em 79 iterações e 0.70 segundos. O TDMA converge em menos de metade do número de iterações utilizadas pelos CG e BICGSTAB mas demora mais tempo (0.11 segundos).

A aplicação do pré-condicionador ILU(0) a este sistema reduz o número de iterações

| Método | Iterações | Tempo (s) | Resíduo |
|---|---|---|---|
| TDMA | 7 | 0.11 | 3.277E-08 |
| CG | 19 | 0.07 | 1.663E-07 |
| BICG | 34 | 0.13 | 1.191E-07 |
| CGNR | 79 | 0.70 | 1.197E-07 |
| BICGSTAB | 19 | 0.07 | 1.197E-07 |
| TFQMR | 23 | 0.10 | 2.370E-08 |
| FOM(10) | 19 | 0.10 | 8.771E-08 |
| GMRES(10) | 18 | 0.10 | 1.744E-07 |
| DQGMRES(10) | 17 | 0.18 | 1.370E-07 |
| DQGMRES(2) | 18 | 0.12 | 1.050E-07 |

Tabela 3.21: Resolução do sistema $80u80$ sem pré-condicionamento

| Método | Iterações | It/Itp | Tempo | Tmp Total | Resíduo |
|---|---|---|---|---|---|
| CG | 7 | 2.71 | 0.08 | 0.14 | 7.588E-08 |
| BICG | 12 | 2.83 | 0.14 | 0.20 | 2.502E-08 |
| CGNR | 15 | 5.27 | 0.17 | 0.23 | 1.166E-07 |
| BICGSTAB | 7 | 2.71 | 0.08 | 0.14 | 7.871E-08 |
| TFQMR | 7 | 3.29 | 0.09 | 0.15 | 4.788E-08 |
| FOM(10) | 7 | 2.71 | 0.09 | 0.15 | 2.247E-08 |
| GMRES(10) | 7 | 2.57 | 0.09 | 0.15 | 2.242E-08 |
| DQGMRES(10) | 7 | 4.43 | 0.10 | 0.16 | 2.242E-08 |
| DQGMRES(2) | 7 | 2.57 | 0.08 | 0.14 | 2.381E-08 |
| PGMRES(10) | 6 | - | 0.09 | 0.15 | 2.242E-08 |

Tabela 3.22: Resolução do sistema $80u80$ pré-condicionado com ILU(0). O tempo de construção do pré-condicionador foi 0.06 segundos.

(tabela 3.22), contudo o tempo total baixa apenas para 2 métodos: CGNR e DQGMRES(2). Se analisarmos apenas os tempos de resolução, não incluindo o tempo de pré-condicionamento, verifica-se que 3 dos métodos aumentam os tempos de convergência (CG, BICG e BICGSTAB) e que os restantes baixam, mas são penalizados pelo tempo necessário à resolução do ILU(0). O sistema $80u80$ é bem condicionado com características que o tornam facilmente resolúvel por todos os métodos iterativos aqui testados. Dos 9 métodos testados com e sem pré-condicionador, o seu pré-condicionamento apenas se justifica para 2: CGNR e DQGMRES(10).

O aumento de dificuldade na resolução do sistema $40u40$ para o $80u80$ reflecte-se apenas no aumento do tempo de convergência (tabelas 3.11 e 3.22). O número de iterações baixa para todos os métodos com excepção do CG que não convergia na resolução do sistema $40u40$. O aumento do tempo de convergência é devido unicamente ao aumento da dimensão do sistema e consequente maior número de operações. A diminuição do número de iterações

será porque o sistema $80u80$ é ligeiramente melhor condicionado do que o sistema $40u40$. Este aspecto está de acordo com os números de condição $\kappa_\infty$ apresentados na secção3.2.1, onde $\kappa_\infty = 0.344E+01$ para o sistema $40u40$ e $\kappa_\infty$ =0.319E+01 para o sistema $80u80$.

## 3.5 Sistemas resultantes da discretização do domínio em 120×80 pontos

Nesta secção incluem-se os resultados obtidos na resolução dos sistemas da pressão ($120p80$) e da velocidade ($120u80$), obtidos a partir da discretização do domínio em $120 \times 80$ pontos.

### 3.5.1 Resolução do sistema $120p80$

Utilizaram-se os métodos habituais sem pré-condicionador e com os pré-condicionadores ILU(4), ILU(6) e ILUDP(0.01,0.75). Analisou-se igualmente o efeito do aumento de $k$ para os métodos FOM($k$), GMRE($k$)S e DQGMRES($k$). O critério de paragem utilizado foi $\|r^{(k)}\|_2 \leq 1.155 \times 10^{-9}$.

Este sistema é resolvido com maior grau de dificuldade, como é possível observar pelos dados da tabela 3.24. Métodos como o FOM(10), GMRES(10), BICG e CGNR não convergem em menos de 1000 iterações, e os restantes métodos convergem em tempos maiores do que os obtidos para os sistemas anteriores.

O DQGMRES($k$), tal como nos anteriores sistemas da pressão, converge mais rapidamente para $k = 2$. Para todos os valores testados de $k$ o DQGMRES converge sempre em 342 iterações, mas em tempos mais elevados do que aquele verificado para $k = 2$ (tabela 3.23). O DQGMRES(2) é o método que converge em menos iterações e em menos tempo (3.11 segundos) quando não é utilizado pré-condicionador (tabela 3.24). Os menores tempos de convergência conseguidos por DQGMRES(2) e DQGMRES(10) foram respectivamente 1.88 e 1.82 segundos, obtidos quando associados ao pré-condicionador ILUDP(0.01,0.75) (tabela 3.27).

A razão pela qual o DQGMRES(2) é mais rápido sem pré-condicionador, contrariamente a todos os sistemas simétricos anteriores em que era o CG, deve-se ao pior condicionamento do sistema $120p80$ (ver secção 3.2), facto ao qual o CG é muito sensível, como está evidenciado na equação (2.24).

O CG não converge também quando associado ao ILU(4) e ILUDP(0.01,0.75). Quando associado ao ILU(6) converge em 20 vezes menos iterações (2.56 segundos) do que sem pré-condicionador (tabela 3.26). O ILU(6) é igualmente responsável pela convergência de todos os restantes métodos.

O BICG e o CGNR convergem apenas quando o sistema é pré-condicionado. Os menores tempos obtidos por estes métodos verificam-se quando associados ao ILUDP(0.01,0.75), respectivamente, 3.77 e 5.30 (tabela 3.27).

| Método | k | Iterações | Tempo | Resíduo |
|---|---|---|---|---|
| FOM | 1 | 1000 | 10.40 | 5.707E-05 |
| | 2 | 1000 | 6.30 | 6.898E-04 |
| | 3 | 1000 | 6.39 | 8.471E-04 |
| | 5 | 1000 | 7.54 | 4.435E-04 |
| | 10 | 1000 | 8.86 | 1.350E-04 |
| | 15 | 1000 | 10.62 | 5.707E-05 |
| | 20 | 1000 | 12.40 | 1.662E-05 |
| | 30 | 1000 | 15.50 | 1.048E-06 |
| | 50 | 1000 | 22.01 | 4.972E-08 |
| GMRES | 1 | 1000 | 10.41 | 1.910E-05 |
| | 2 | 1000 | 6.28 | 5.330E-04 |
| | 3 | 1000 | 6.33 | 3.930E-04 |
| | 5 | 1000 | 7.41 | 2.109E-04 |
| | 10 | 1000 | 8.77 | 5.331E-05 |
| | 15 | 1000 | 10.55 | 1.910E-05 |
| | 20 | 1000 | 12.31 | 5.218E-06 |
| | 30 | 1000 | 15.52 | 5.505E-07 |
| | 50 | 1000 | 22.00 | 1.458E-08 |
| DQGMRES | 1 | 342 | 9.28 | 1.129E-09 |
| | 2 | 342 | 3.16 | 1.129E-09 |
| | 3 | 342 | 3.65 | 1.129E-09 |
| | 5 | 342 | 4.74 | 1.129E-09 |
| | 10 | 342 | 7.02 | 1.129E-09 |
| | 15 | 342 | 9.32 | 1.129E-09 |
| | 20 | 342 | 11.60 | 1.129E-09 |
| | 30 | 342 | 16.00 | 1.129E-09 |
| | 50 | 342 | 24.41 | 1.129E-09 |

Tabela 3.23: Efeito da dimensão do sub-espaço de Krylov $k$ na resolução do sistema $120p80$ pelos métodos FOM, GMRES e DQGMRES sem pré-condicionamento.

Para além do DQGMRES(2), também o BICGSTAB é mais rápido do que o CG sem pré-condicionamento, convergindo em 3.46 segundos e o CG em 3.70 segundos (tabela 3.24). O tempo de convergência mínimo obtido pelo BICGSTAB é de 1.95 segundos e surge quando associado ao ILUDP(0.01,0.75) (tabela 3.27). O TFQMR converge mais rápido do que o BICGSTAB quando são utilizados os pré-condicionadores ILU(4) e ILU(6), mas o seu tempo mínimo (2.00 segundos) é obtido quando associado ao ILUDP(0.01,0.75).

Os métodos FOM($k$) e GMRES($k$) não convergem novamente sem pré-condicionador para qualquer dos valores de $k$ testados (ver tabela 3.23). Para $k = 10$ convergem sempre que o sistema é pré-condicionado. O comportamento destes dois métodos difere do verificado anteriormente, em que apresentaram sempre desempenhos muito semelhantes. O GMRES(10) associado ao ILUDP(0.01,0.75) é o método que converge em menos tempo

| Método | Iterações | Tempo | Resíduo |
|---|---|---|---|
| TDMA | 1000 | 25.58 | 5.614E-05 |
| CG | 640 | 3.70 | 1.118E-09 |
| BICG | 1000 | 6.74 | 3.731E-07 |
| CGNR | 1000 | 5.57 | 8.813E-04 |
| BICGSTAB | 541 | 3.46 | 1.013E-09 |
| TFQMR | 665 | 5.18 | 5.562E-10 |
| FOM(10) | 1000 | 8.86 | 1.351E-04 |
| GMRES(10) | 1000 | 8.86 | 5.331E-05 |
| DQGMRES(10) | 342 | 7.04 | 1.129E-09 |
| DQGMRES(2) | 342 | 3.11 | 1.129E-09 |

Tabela 3.24: Resolução do sistema 120$p$80 sem pré-condicionamento

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0.64 | 42.02 | 42.50 | 1.368E-02 |
| BICG | 92 | 10.87 | 4.12 | 4.60 | 5.922E-10 |
| CGNR | 165 | 6.06 | 7.22 | 7.70 | 1.061E-09 |
| BICGSTAB | 45 | 10.02 | 1.90 | 2.38 | 9.367E-10 |
| TFQMR | 39 | 17.05 | 1.70 | 2.18 | 6.854E-10 |
| FOM(10) | 43 | 23.26 | 1.93 | 2.41 | 1.140E-09 |
| GMRES(10) | 53 | 18.87 | 2.38 | 2.86 | 9.445E-10 |
| DQGMRES(10) | 33 | 10.36 | 1.72 | 2.20 | 1.103E-09 |
| DQGMRES(2) | 42 | 8.14 | 1.84 | 2.32 | 1.148E-09 |
| PGMRES(10) | 48 | - | 2.43 | 2.91 | 9.445E-10 |

Tabela 3.25: Resolução do sistema 120$p$80 pré-condicionado com ILU(4). O tempo de construção do pré-condicionador foi 0.48 segundos.

(1.77 segundos) enquanto que o FOM(10), nessa mesma situação, demora mais do dobro do tempo (3.91 segundos) para convergir (tabela 3.27). O FOM(10) e o CG são os únicos métodos que apresentam melhores resultados quando associados ao ILU(6).

O PGMRES(10) não apresenta novidades em relação ao comportamento evidenciado anteriormente. Converge num número de iterações menor ou igual ao GMRES(10), mas demora mais tempo.

Dos pré-condicionadores utilizados para este sistema apenas o ILU(6) consegue que todos os métodos convirjam dentro do limite de iterações fixado (tabela 3.26). Mas é com o ILUDP(0.01,0.75) que 80% dos métodos convergem em tempo mínimo (tabela 3.27). Verifica-se assim que embora o ILU(6) seja um pré-condicionador com maior precisão, pois demora 0.88 segundos a ser construído enquanto que o ILUDP(0.01,0.75) demora 0.49 segundos, a sua eficiência não é superior em 80% dos métodos. Este aspecto já evidenciado anteriormente deve-se ao facto de uma maior precisão se traduzir num pré-condicionador

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 32 | 20.00 | 1.68 | 2.56 | 9.372E-10 |
| BICG | 64 | 15.63 | 3.56 | 4.44 | 4.531E-10 |
| CGNR | 93 | 10.75 | 5.18 | 6.06 | 8.523E-10 |
| BICGSTAB | 31 | 17.54 | 1.62 | 2.54 | 5.996E-10 |
| TFQMR | 27 | 24.39 | 1.46 | 2.34 | 1.215E-09 |
| FOM(10) | 26 | 38.46 | 1.42 | 2.30 | 1.039E-09 |
| GMRES(10) | 26 | 38.46 | 1.42 | 2.30 | 7.903E-10 |
| DQGMRES(10) | 23 | 14.93 | 1.40 | 2.28 | 8.150E-10 |
| DQGMRES(2) | 29 | 11.76 | 1.53 | 2.41 | 1.116E-09 |
| PGMRES(10) | 23 | - | 1.53 | 2.41 | 0.790E-09 |

Tabela 3.26: Resolução do sistema 120$p$80 pré-condicionado com ILU(6). O tempo de construção do pré-condicionador foi 0.88 segundos.

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0.64 | 41.67 | 42.16 | 3.228E-05 |
| BICG | 74 | 13.51 | 3.28 | 3.77 | 4.032E-10 |
| CGNR | 111 | 90.09 | 4.81 | 5.30 | 9.267E-10 |
| BICGSTAB | 35 | 15.38 | 1.46 | 1.95 | 6.667E-10 |
| TFQMR | 35 | 18.87 | 1.51 | 2.00 | 5.796E-10 |
| FOM(10) | 77 | 12.99 | 3.42 | 3.91 | 9.971E-10 |
| GMRES(10) | 29 | 34.48 | 1.28 | 1.77 | 1.077E-09 |
| DQGMRES(10) | 26 | 13.16 | 1.33 | 1.82 | 9.735E-10 |
| DQGMRES(2) | 31 | 10.99 | 1.39 | 1.88 | 1.030E-09 |
| PGMRES(10) | 26 | - | 1.40 | 1.89 | 0.104E-08 |

Tabela 3.27: Resolução do sistema 120$p$80 pré-condicionado com ILUDP(0.01,0.75). O tempo de construção do pré-condicionador foi 0.49 segundos.

menos esparso e consequentemente obrigar a mais operações, para além de um maior tempo de construção.

### 3.5.2 Resolução do sistema 120$u$80

Os resultados aqui apresentados referem-se à resolução do sistema da velocidade 120 × 80. Foram utilizados os métodos habituais juntamente com o pré-condicionador ILU(0). O critério de paragem utilizado por todos os métodos foi $\|r^{(k)}\|_2 \leq 2.078 \times 10^{-7}$.

Este sistema é muito melhor condicionado do que o 120$p$80, apesar de resultar da mesma discretização não homogénea. Os resultados obtidos sem pré-condicionador (tabela 3.28) demonstram-no claramente. Comparativamente ao sistema 120$p$80, todos os métodos con-

| Método | Iterações | Tempo (s) | Resíduo |
|---|---|---|---|
| TDMA | 6 | 0.15 | 1.619E-07 |
| CG | 19 | 0.11 | 1.584E-07 |
| BICG | 34 | 0.24 | 9.028E-08 |
| CGNR | 79 | 0.46 | 1.694E-07 |
| BICGSTAB | 19 | 0.13 | 1.784E-07 |
| TFQMR | 21 | 0.18 | 4.916E-08 |
| FOM(10) | 18 | 0.15 | 1.736E-07 |
| GMRES(10) | 18 | 0.16 | 1.484E-07 |
| DQGMRES(10) | 17 | 0.28 | 1.177E-07 |
| DQGMRES(2) | 17 | 0.18 | 1.709E-07 |

Tabela 3.28: Resolução do sistema $120u80$ sem pré-condicionamento

vergem em número de iterações e tempos muito reduzidos.

O número de iterações utilizadas pelos vários métodos é igual ou inferior ao obtido pelos mesmos métodos na resolução do sistema $80u80$ (tabela 3.21). O aumento de dimensão provocou apenas um ligeiro aumento do tempo de convergência de 9 dos 10 métodos e uma diminuição no CGNR. Estes resultados demonstram que o sistema $120u80$ é melhor condicionado do que o sistema $80u80$.

O método mais rápido sem pré-condicionador é o CG, que converge em 0.11 segundos, seguido pelo BICGSTAB, com 0.13 segundos. O TDMA converge em 6 iterações enquanto que o CG e BICGSTAB convergem ambos em 19. Como o TDMA converge em mais tempo (15 segundos) verificamos que as iterações deste método são mais demoradas do que as iterações da maior parte dos métodos não-estacionários (tabela 3.28).

| Método | Iterações | It/Itp. | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 7 | 3.16 | 0.13 | 0.22 | 1.857E-08 |
| BICG | 12 | 2.83 | 0.24 | 0.33 | 1.466E-08 |
| CGNR | 15 | 5.27 | 0.28 | 0.37 | 9.214E-08 |
| BICGSTAB | 7 | 2.71 | 0.14 | 0.23 | 5.703E-08 |
| TFQMR | 7 | 3.00 | 0.15 | 0.24 | 3.329E-08 |
| FOM(10) | 7 | 2.57 | 0.15 | 0.24 | 1.436E-08 |
| GMRES(10) | 7 | 2.57 | 0.14 | 0.23 | 1.433E-08 |
| DQGMRES(10) | 7 | 2.42 | 0.15 | 0.24 | 1.433E-08 |
| DQGMRES(2) | 7 | 2.42 | 0.13 | 0.22 | 1.453E-08 |
| PGMRES(10) | 6 | - | 0.15 | 0.24 | 1.433E-08 |

Tabela 3.29: Resolução do sistema $120u80$ pré-condicionado com ILU(0). O tempo de construção do pré-condicionador foi 0.09 segundos.

O pré-condicionamento deste sistema com ILU(0) (ver tabela 3.29) reduz o número de

iterações para todos os métodos. Em contrapartida os tempos totais aumentam também para a generalidade dos métodos, com excepção de CGNR e do DQGMRES(10). Mais uma vez, dada a ordem de grandeza dos tempos de convergência, o pré-condicionamento do sistema 120$u$80 é ineficiente, qualquer que seja o pré-condicionador.

## 3.6 Sistemas resultantes da discretização do domínio em 120×120 pontos

Os maiores sistemas testados resultaram da discretização do domínio com uma malha regular de 120 × 120 pontos. Os resultados da resolução dos sistemas da pressão (120$p$120) e da velocidade horizontal (120$u$120) provenientes da discretização desse domínio são o motivo desta secção.

Os testes com matrizes desta dimensão representam uma exigência maior sobre os métodos iterativos utilizados, porque para além das suas características de convergência trata-se de analisar a sua capacidade para atingir elevados níveis de precisão, i.e., a sua capacidade para não propagar erros de arredondamento, resultantes das operações aritméticas com precisão finita (Chaitin-Chatelin and Frayssé, 1996). Por esta razão e para ilustrar o comportamento de cada um dos métodos, representa-se na figura 3.5 a evolução dos resíduos.

### 3.6.1 Resolução do sistema 120$p$120

O sistema da pressão 120$p$120 foi resolvido pelos métodos habituais juntamente com os pré-condicionador ILU(6) e ILUDP(0.01,0.75) e ILUDP(0.001,0.75). Efectuou-se, tal como nos anteriores sistemas da pressão uma análise do desempenho dos métodos FOM($k$), GMRES($k$) e DQGMRES($k$) em função do valor de $k$. O critério de paragem utilizado por todos os métodos na resolução deste sistema foi $\|r^{(k)}\|_2 \leq 1.015 \times 10^{-9}$.

A resolução deste sistema apresenta um grau de dificuldade superior a todos os anteriores, como se pode verificar a partir da comparação dos resultados na tabela 3.30, com tabelas análogas referentes a sistemas de menor dimensão. Apenas 2 métodos convergem sem pré-condicionamento: o CG e o DQGMRES($k$). O mais rápido é o CG que converge em 6.98 segundos. O BICGSTAB e o TFQMR apesar de não convergirem em menos de 1000 iterações encontram-se muito próximos, como é possível observar pelos valores dos respectivos resíduos (tabela 3.30).

O FOM($k$) e GMRES($k$) não convergem sem pré-condicionador para nenhum dos valores de $k$ analisados (tabela 3.31), sendo os resíduos ligeiramente superiores aos verificados para o sistema 120$p$80 (tabela 3.23). Das quatro situações analisadas para o sistema 120$p$120, FOM(10) e GMRES(10) convergem apenas quando associados ao pré-condicionador ILUDP(0.001,0.75), com tempos muito superiores aos melhores (tabela 3.34).

| Método | Iterações | Tempo (s) | Resíduo |
|---|---|---|---|
| TDMA | 1000 | 41.74 | 1.128E-04 |
| CG | 750 | 6.98 | 8.738E-10 |
| BICG | 1000 | 10.1 | 1.802E-06 |
| CGNR | 1000 | 9.11 | 7.646E-04 |
| BICGSTAB | 1000 | 9.74 | 1.821E-09 |
| TFQMR | 1000 | 11.75 | 2.144E-09 |
| FOM(10) | 1000 | 14.50 | 2.183E-04 |
| GMRES(10) | 1000 | 14.42 | 8.521E-05 |
| DQGMRES(10) | 723 | 22.73 | 9.756E-10 |
| DQGMRES(2) | 723 | 10.35 | 9.749E-10 |

Tabela 3.30: Resolução do sistema 120$p$120 sem pré-condicionamento

O DQGMRES($k$) sem pré-condicionamento converge novamente no mesmo número de iterações, qualquer que seja o valor de $k$, sendo o tempo mínimo de 10.35 segundos, obtido para $k = 2$ (tabela 3.31). O menor tempo obtido pelo DQGMRES($k$), para $k = 2$ e $k = 10$, é respectivamente, 6.36 e 8.22 segundos, quando associado ao ILUDP(0.01,0.75) (tabela 3.33).

O BICG e o CGNR convergem nas três situações analisadas em que foi utilizado um pré-condicionador. O menor tempo total de convergência destes dois métodos é, respectivamente, 7.45 e 18.38 segundos, sendo obtido quando associados ao pré-condicionador ILUDP(0.01,0.75) (tabela 3.33).

O pré-condicionamento do sistema 120$p$120 com ILU(6) melhora sobretudo a eficiência do TFQMR, BICGSTAB e BICG que passam a convergir em, respectivamente, 4.74, 6.91 e 9.23 segundos (tabela 3.32). O tempo obtido pelo TFQMR é inferior a todos os outros, obtidos na resolução do sistema 120$p$120. O tempo mais próximo (5.50 segundos) é obtido pelo BICGSTAB associado ao ILUDP(0.01,0.75) (tabela 3.33).

Apesar do TFQMR apresentar, com o pré-condicionador ILU(6), o tempo total mais baixo das situações analisadas, a eficiência deste pré-condicionador não atinge todos os métodos. Os métodos CG, FOM(10), GMRES(10) e PGMRES(10) não atingem o resíduo mínimo pretendido (tabela 3.33). O CG tem uma primeira fase em que os resíduos diminuem rapidamente aproximando-se do resíduo mínimo, sem no entanto o atingir. Numa segunda fase, que começa a partir de um número de iterações próximo daquele em que deveria convergir, o resíduo aumenta de forma muito acentuada (figura 3.5). O pré-condicionador ILU(6) associado ao CG, eficiente na resolução do sistema 120$p$80 (tabela 3.26), deixa de ser eficiente na resolução do sistema 120$p$120. Concluímos que este sistema é pior condicionado e que o pré-condicionamento com ILU(6) tem um efeito contrário ao desejado, piorando ainda mais o condicionamento.

O CG, apesar de ser um dos 3 em 10 métodos que convergiu sem pré-condicionamento (tabela 3.30), após o pré-condicionamento com qualquer dos pré-condicionadores utilizados

| Método | k | Iterações | Tempo | Resíduo |
|---|---|---|---|---|
| FOM | 1 | 1000 | 15.92 | 1.212E-04 |
| | 2 | 1000 | 10.11 | 6.639E-04 |
| | 3 | 1000 | 10.33 | 8.204E-04 |
| | 5 | 1000 | 11.58 | 5.013E-04 |
| | 10 | 1000 | 14.50 | 2.183E-04 |
| | 15 | 1000 | 16.55 | 1.212E-04 |
| | 20 | 1000 | 18.84 | 4.199E-05 |
| | 30 | 1000 | 23.83 | 8.561E-06 |
| | 50 | 1000 | 33.69 | 5.494E-07 |
| GMRES | 1 | 1000 | 15.97 | 3.782E-05 |
| | 2 | 1000 | 10.01 | 4.984E-04 |
| | 3 | 1000 | 10.30 | 3.894E-04 |
| | 5 | 1000 | 11.41 | 2.258E-04 |
| | 10 | 1000 | 14.42 | 8.521E-05 |
| | 15 | 1000 | 16.43 | 3.782E-05 |
| | 20 | 1000 | 18.80 | 1.338E-05 |
| | 30 | 1000 | 23.75 | 3.130E-06 |
| | 50 | 1000 | 33.62 | 1.726E-07 |
| DQGMRES | 1 | 723 | 30.05 | 9.749E-10 |
| | 2 | 723 | 10.66 | 9.749E-10 |
| | 3 | 723 | 11.98 | 9.755E-10 |
| | 5 | 723 | 15.44 | 9.751E-10 |
| | 10 | 723 | 23.59 | 9.756E-10 |
| | 15 | 723 | 30.49 | 9.749E-10 |
| | 20 | 723 | 37.77 | 9.761E-10 |
| | 30 | 723 | 52.33 | 9.766E-10 |
| | 50 | 723 | 81.37 | 9.758E-10 |

Tabela 3.31: Efeito da dimensão do sub-espaço de Krylov $k$ na resolução do sistema 120$p$120 pelos métodos FOM, GMRES e DQGMRES sem pré-condicionamento.

não convergiu.

A figura 3.5 mostra que o DQGMRES constitui uma melhoria do GMRES, porque aumenta a taxa de convergencia e permite reduzir o erro para valores inferiores ao do critério de convergência; ao contrário do GMRES(10) que apesar de ter boa taxa de convergência mostra-se incapaz de reduzir o erro para valores inferiores a 1.077E-09, atingido após cerca de 55 iterações.

A figura 3.5 mostra igualmente as características superiores do BICG em relação ao CG. Ambos os métodos depois de atingirem um valor mínimo do erro após cerca de 40 iterações, invertem a tendência de redução do erro; o qual se mantém para o CG, que após 1000 iterações indica um resíduo de 8.9E+01, ao contrario do BICG que retoma a

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0.75 | 80.00 | 81.37 | 8.900E+01 |
| BICG | 92 | 10.87 | 7.86 | 9.23 | 9.636E-10 |
| CGNR | 245 | 3.94 | 20.59 | 21.96 | 1.003E-09 |
| BICGSTAB | 69 | 14.49 | 5.54 | 6.91 | 2.206E-10 |
| TFQMR | 41 | 24.39 | 3.37 | 4.74 | 1.953E-09 |
| FOM(10) | 1000 | 1.00 | 84.00 | 85.37 | 4.210E-09 |
| GMRES(10) | 1000 | 1.00 | 84.08 | 85.45 | 1.883E-09 |
| DQGMRES(10) | 83 | 8.70 | 8.10 | 9.47 | 1.913E-09 |
| DQGMRES(2) | 70 | 10.31 | 5.87 | 7.24 | 1.921E-09 |
| PGMRES(10) | 1000 | - | 94.38 | 95.75 | 1.883E-09 |

Tabela 3.32: Resolução do sistema 120$p$120 pré-condicionado com ILU(6). O tempo de construção do pré-condicionador foi 1.37 segundos.

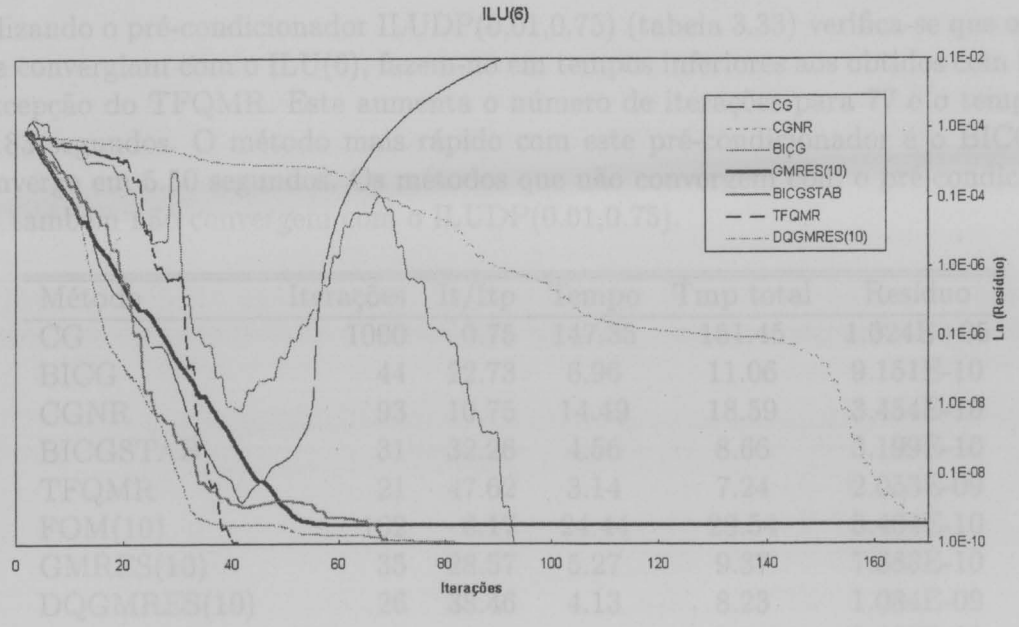



Figura 3.5: Evolução dos resíduos na resolução do sistema 120$p$120 pré-condicionado com ILU(6)

tendência de redução do erro após a iteração 70 e vem a satisfazer o critério de paragem à iteração 92.

O FOM(10), GMRES(10) e PGMRES(10) associados ao ILU(6), após uma primeira fase de diminuição dos resíduos, acabam por estabilizar num patamar próximo do mínimo desejado (figura 3.5). Este fenómeno é designado por estagnação e pode ser ultrapassado com recurso a uma maior dimensão da base do sub-espaço de Krylov ou utilizando um pré-condicionador mais eficiente do que o ILU(6).

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0.75 | 73.47 | 74.47 | 6.366E+08 |
| BICG | 82 | 12.20 | 6.45 | 7.45 | 9.325E-10 |
| CGNR | 225 | 4.44 | 17.38 | 18.38 | 5.528E-10 |
| BICGSTAB | 61 | 16.39 | 4.50 | 5.50 | 4.323E-10 |
| TFQMR | 77 | 12.98 | 5.83 | 6.83 | 2.797E-10 |
| FOM(10) | 1000 | 1.00 | 77.28 | 78.28 | 1.426E-08 |
| GMRES(10) | 1000 | 1.00 | 77.34 | 78.28 | 1.894E-09 |
| DQGMRES(10) | 79 | 9.15 | 7.22 | 8.22 | 1.915E-09 |
| DQGMRES(2) | 70 | 10.33 | 5.36 | 6.36 | 1.939E-09 |
| PGMRES(10) | 1000 | - | 86.00 | 87.00 | 1.894E-09 |

Tabela 3.33: Resolução do sistema 120$p$120 pré-condicionado com ILUDP(0.01,0.75). O tempo de construção do pré-condicionador foi 1.00 segundos.

Utilizando o pré-condicionador ILUDP(0.01,0.75) (tabela 3.33) verifica-se que os métodos que convergiam com o ILU(6), fazem-no em tempos inferiores aos obtidos com ILU(6), com excepção do TFQMR. Este aumenta o número de iterações para 77 e o tempo total para 6.83 segundos. O método mais rápido com este pré-condicionador é o BICGSTAB que converge em 5.50 segundos. Os métodos que não convergem com o pré-condicionador ILU(6) também não convergem com o ILUDP(0.01,0.75).

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 1000 | 0.75 | 147.35 | 151.45 | 1.024E+05 |
| BICG | 44 | 22.73 | 6.96 | 11.06 | 9.151E-10 |
| CGNR | 93 | 10.75 | 14.49 | 18.59 | 3.454E-10 |
| BICGSTAB | 31 | 32.26 | 4.56 | 8.66 | 3.199E-10 |
| TFQMR | 21 | 47.62 | 3.14 | 7.24 | 2.053E-09 |
| FOM(10) | 162 | 6.17 | 24.44 | 28.54 | 8.464E-10 |
| GMRES(10) | 35 | 28.57 | 5.27 | 9.37 | 7.883E-10 |
| DQGMRES(10) | 26 | 38.46 | 4.13 | 8.23 | 1.084E-09 |
| DQGMRES(2) | 37 | 19.54 | 5.41 | 9.51 | 2.007E-09 |
| PGMRES(10) | 31 | - | 5.33 | 9.43 | 7.882E-10 |

Tabela 3.34: Resolução do sistema 120$p$120 pré-condicionado com ILUDP(0.001,0.75). O tempo de construção do pré-condicionador foi 4.10 segundos.

Utilizando o pré-condicionador ILUDP(`tol`,0.75) com `tol` = 0.001, FOM(10), GMRES(10) e PGMRES(10) conseguem finalmente ultrapassar o patamar de estagnação dos resíduos e convergir (tabela 3.34). O FOM(10) necessita 3 vezes mais tempo do que o GMRES(10) para convergir. O PGMRES(10) converge num tempo próximo do GMRES(10), respectivamente, 9.43 e 9.37 segundos. O método mais rápido é novamente o TFQMR com menos iterações do que com o ILU(4). O tempo de convergência deste método (3.14 segun-

dos) é também inferior, contudo em termos de tempo total (7.24 segundos) tem um pior desempenho. Os restantes métodos, com excepção do CGNR, apresentam piores resultados com este pré-condicionador.

A maior precisão do pré-condicionador traduz-se numa diminuição do tempo de convergência do método e num aumento do tempo de construção do pré-condicionador. Podemos considerar que o tempo total, soma dos dois, diminui enquanto o segundo for inferior ao primeiro. Para o TFQMR esse ponto já está ultrapassado, i.e., o tempo de pré-condicionamento é superior ao tempo de convergência, respectivamente 4.10 e 3.14 segundos (tabela 3.34). Por esta razão não se espera um melhor rendimento do TFQMR com o aumento da precisão do pré-condicionador ILUDP(`tol`,0.75).

Em relação ao GMRES(10) ainda existe um pequena margem de progressão dado que o tempo de convergência (5.27 segundos) é superior aos 4.10 segundos necessários para a construção do ILUDP(0.001,0.75) (tabela 3.34). Mas também é necessário ter em conta que um pré-condicionador menos esparso conduz a mais operações em cada iteração e a um aumento dos recursos de memória necessários. Estes dois aspectos aumentam o tempo de convergência, e consequentemente a diminuição da margem de progressão. Por esta razão o aumento da precisão do pré-condicionador faz sentido sobretudo para os métodos que apresentam tempos de convergência muito superiores ao tempo de pré-condicionamento.

### 3.6.2 Resolução do sistema $120u120$

Para a resolução do sistema $120u120$, foram utilizados os métodos habituais juntamente com o pré-condicionado ILU(0). O critério de paragem utilizado por todos os métodos foi $\|r^{(k)}\|_2 \leq 2.326 \times 10^{-7}$.

| Método | Iterações | Tempo (s) | Resíduo |
|---|---|---|---|
| TDMA | 7 | 0.29 | 4.813E-08 |
| CG | 17 | 0.17 | 2.283E-07 |
| BICG | 31 | 0.35 | 2.272E-07 |
| CGNR | 79 | 0.74 | 2.158E-07 |
| BICGSTAB | 21 | 0.22 | 7.560E-08 |
| TFQMR | 21 | 0.26 | 7.263E-08 |
| FOM(10) | 18 | 0.24 | 1.920E-07 |
| GMRES(10) | 18 | 0.25 | 1.810E-07 |
| DQGMRES(10) | 17 | 0.44 | 1.354E-07 |
| DQGMRES(2) | 17 | 0.22 | 1.663E-07 |

Tabela 3.35: Resolução do sistema $120u120$ sem pré-condicionamento

Apesar do elevado número de incógnitas deste sistema, a sua resolução não apresenta qualquer dificuldade aos métodos utilizados, inclusive o TDMA. Todos os métodos convergem num número reduzido de iterações e em tempos inferiores a 1 s.

Comparando estes resultados com os obtidos na resolução do sistema $120u80$ (tabela 3.28), verifica-se apenas um pequeno aumento do tempo de convergência de todos os métodos. O número de iterações aumenta apenas para um método, mantém-se igual para 6, e diminui para os restantes.

O método que converge em menos tempo (0.17 segundos) é o CG sem pré-condicionamento (tabela 3.35). O método que converge em mais tempo é o CGNR (0.74 segundos), também sem pré-condicionamento.

| Método | Iterações | It/Itp | Tempo | Tmp total | Resíduo |
|---|---|---|---|---|---|
| CG | 7 | 2.43 | 0.21 | 0.34 | 4.130E-08 |
| BICG | 12 | 2.58 | 0.37 | 0.50 | 2.628E-08 |
| CGNR | 15 | 5.26 | 0.44 | 0.57 | 1.556E-07 |
| BICGSTAB | 9 | 2.33 | 0.21 | 0.34 | 9.413E-08 |
| TFQMR | 7 | 3.00 | 0.22 | 0.35 | 5.161E-08 |
| FOM(10) | 7 | 2.57 | 0.23 | 0.36 | 2.525E-08 |
| GMRES(10) | 7 | 2.57 | 0.23 | 0.36 | 2.519E-08 |
| DQGMRES(10) | 7 | 2.43 | 0.24 | 0.37 | 2.519E-08 |
| DQGMRES(2) | 7 | 2.43 | 0.21 | 0.34 | 2.586E-08 |
| PGMRES(10) | 6 | - | 0.26 | 0.39 | 0.501E-07 |

Tabela 3.36: Resolução do sistema $120u120$ pré-condicionado com ILU(0). O tempo de construção do pré-condicionador foi 0.13 segundos.

Como já foi observado para outros sistemas das velocidades, a aplicação de um pré-condicionador a este sistema é desnecessária. Contudo para ilustrar o seu bom condicionamento apresenta-se na tabela 3.36 os resultados da aplicação do ILU(0). Verifica-se aqui também que embora o número de iterações seja menor, para todos os métodos, o tempo total apenas baixa para o CGNR, aumentando para os restantes.

# Capítulo 4

# Conclusões e sugestões de trabalho futuro

Neste capítulo reúnem-se as conclusões principais e algumas sugestões possíveis para trabalho futuro. Cada um destes pontos ocupa uma secção diferente no capítulo.

## 4.1 Conclusões

No sentido de procurar alternativas ao *solver* TDMA, muito utilizado em programas de CFD, estudou-se a convergência de 10 métodos iterativos não-estacionários na resolução de dois conjuntos de sistemas de equações algébricas. Um relativo à variável pressão e outro relativo à variável velocidade horizontal. Para cada um destes casos foram analisados 4 matrizes resultantes da discretização do domínio com os seguintes números de nós: $40 \times 40$, $80 \times 80$, $120 \times 80$ e $120 \times 120$.

### 4.1.1 As dificuldades de resolução dos sistemas da pressão e dos sistemas da velocidade

Os sistemas derivados do cálculo das pressões são de resolução mais difícil do que os derivados do cálculo da velocidade. Todos os sistemas do campo da velocidade são resolvidos em tempos reduzidos e num pequeno número de iterações por todos os métodos iterativos testados, contrariamente aos sistemas do campo da pressão. As diferenças de dificuldade podem ser compreendidas através da análise da distribuição dos valores próprios. Os sistemas da velocidade têm todos os seus valores próprios situados claramente do lado direito do plano complexo, enquanto que os valores próprios dos sistemas da pressão situam-se próximos da origem, havendo mesmo alguns do lado esquerdo (secção 3.2.2). Em consequência as quatro matrizes $A$, para a variável velocidade, são todas definidas positivas e não-singulares. Para a variável pressão todas as quatro matrizes estão próximas da singularidade, sendo apenas

uma delas definida positiva.

### 4.1.2 O desempenho relativo de cada um dos métodos

Em todos os sistemas da pressão o TDMA não cumpre o critério de convergência, em menos de 1000 iterações. Nos sistemas de maior dimensão (malha 120 × 120) apenas alguns métodos não-estacionários, como o CG e DQGMRES($k$), o conseguem fazer sem pré-condicionamento. Nos sistemas do campo da velocidade, o grau de dificuldade aumenta ligeiramente com o aumento da dimensão. Em todos eles, os métodos testados convergem num número reduzido de iterações, inclusive o TDMA. Estes resultados mostram que o TDMA tem um desempenho próximo ao dos métodos não-estacionários na resolução de sistemas definidos positivos e não-singulares, i.e., bem condicionados. Por exemplo na resolução do sistema da velocidade com maior dimensão ($120u120$) o TDMA converge num tempo igual a 0.29 segundos, muito próximo do melhor tempo 0.17 segundos, realizado pelo CG, e inferior aos tempos realizados pelos CGNR, BICG e DQGMRES(10) (tabela 3.35). Contudo nos sistemas de resolução mais difícil como os da pressão, os métodos não-estacionários são muito mais eficientes. Para ilustrar este facto tomamos como exemplo a resolução do sistema da pressão com menor dimensão ($40p40$). Ao fim de 4.13 segundos o TDMA não cumpre o critério de convergência enquanto que o método mais rápido cumpre-o ao fim de 0.09 segundos (tabela 3.5).

O CG converge sem pré-condicionamento em todos os sistemas simétricos. Em todos esses casos, o CG é o método mais rápido, com a excepção do sistema das pressões $120p80$, em que o método mais rápido sem pré-condicionamento é o DQGMRES(2)(tabela 3.24). Verifica-se assim que embora faça parte das suas premissas que o sistema seja simétrico e definido positivo, o CG também converge em situações onde esta propriedade não se verifica. Por exemplo, resolve o sistema $120p80$ que não é definido positivo, pois tem uma pequena percentagem de linhas e colunas diagonalmente dominantes (29.7%), assim como um valor próprio negativo (tabela 3.24). Como foi referido na secção 2.3, o método do CG faz implicitamente um factorização LU de uma matriz tridiagonal. A existência dessa factorização é garantida se $A$ for definida positiva. Verifica-se neste trabalho que essa factorização também é possível em certas matrizes apesar de não serem definidas positivas.

O CG tem a desvantagem de não se poder aplicar aos sistemas não-simétricos. Esta situação pode ser no entanto remediada com a utilização de um pré-condicionador que aproxime razoavelmente a matriz $A$. Por exemplo o sistema não-simétrico $40u40$ é resolvido pelo CG quando associado ao pré-condicionador ILUD(0.01, 0.75) (tabela 3.12). Contudo o pré-condicionamento pode ter um efeito contrário e fazer com que o CG deixe de convergir. Esta situação sucede na resolução do sistema $120p120$ (secção 3.6.1) em que o CG converge em 750 iterações sem pré-condicionamento, mas diverge se o sistema for pré-condicionado. Isto foi atribuído ao facto do pré-condicionador ser instável e piorar o condicionamento do sistema.

O CGNR não é um método iterativo autónomo mas sim uma técnica que consiste em aplicar o CG à equação normal do sistema. Esta abordagem permite aplicar o CG

a sistemas que não sejam simétricos definidos positivos. Como exemplo pode observar-se que o sistema não-simétrico $40u40$ não é resolvido pelo CG mas é resolvido pelo CGNR (tabela 3.11). O facto de multiplicar $A$ pela sua transposta obriga a um maior número de operações por iteração, fazendo do CGNR um método de convergência lenta. Por exemplo, apesar de convergir em todos os sistemas da velocidade é sempre o método mais lento. Por outro lado, o CGNR tem a desvantagem de fazer aumentar o número de condição da matriz dos coeficientes para o seu quadrado, fazendo com que os sistemas, originalmente mal condicionados, aumentem de forma pronunciada as suas dificuldades de resolução. Este aspecto está bem patente na resolução dos sistemas da pressão sem pré-condicionamento em que o CGNR não converge em nenhum deles. A aplicação de um pré-condicionador melhora muito o desempenho deste método. Como se pode observar na resolução do sistema $120p120$ pré-condicionado com ILU(0.001,0.75), o CGNR converge em cerca de 11 vezes menos iterações do que sem pré-condicionador (tabela 3.34).

O BICG tem um comportamento semelhante ao CG quando o sistema é simétrico mas convergindo duas vezes mais lentamente do que este, devido ao facto de efectuar duas multiplicações matriz-vector por iteração. Em relação ao CG tem a vantagem de solucionar sistemas não-simétricos e de ser menos sensível ao pré-condicionamento como se pode observar na resolução do sistema $120p120$ em que o CG não converge com pré-condicionamento e o BICG converge sempre (secção 3.6.1). Comparativamente com o CGNR, o BICG converge em menos tempo para todos os sistemas com excepção do $40u40$ e do $40p40$ pré-condicionados.

O BICGSTAB e o TFQMR têm comportamentos semelhantes, alternando ligeiramente os seus desempenhos relativos. De todas as situações analisadas sem pré-condicionador só não convergiram na resolução do sistema $120p120$. Convergem sempre mais rapidamente do que o BICG e do que o CGNR. Nos sistemas sem pré-condicionamento utilizam mais iterações e demoram mais tempo do que o CG. As excepções acontecem para o sistema não simétrico $40u40$ em que o CG não converge e para o sistema $120p80$. Nesta situação o BICGSTAB converge em 3.46 segundos, o CG em 3.70 e o TFQMR em 5.18 (tabela 3.24). O pré-condicionamento destes dois métodos tem sempre o efeito de diminuir os seus tempos de convergência. Por exemplo na resolução do sistema $120p120$ o tempo de convergência mais baixo (4.74 segundos) é obtido pelo TFQMR, associado ao pré-condicionador ILU(6), e o tempo imediatamente a seguir é conseguido pelo BICGSTAB associado ao ILUDP(0.01,0.75) (tabelas 3.32 e 3.33).

O FOM(10) e o GMRES(10) têm desempenhos muito próximos para os sistemas de menor dimensão. Para os sistemas de maior dimensão estes dois métodos não convergem sem pré-condicionador, verificando-se então que o GMRES apresenta melhores resultados. O caso mais significativo ocorre na resolução do sistema $120p120$ pré-condicionado por ILUDP(0.001,0.75) o GMRES(10) converge em 9.37 segundos enquanto que o FOM(10) converge em 28.95 segundos (tabela 3.34). O sucesso destes métodos depende muito do pré-condicionamento do sistema. Quando este é efectuado apresentam uma melhoria considerável do seu desempenho, convergindo para todos os sistemas estudados. Por exemplo o GMRES(10) associado ao ILUDP(0.01,0.75) é o método que converge em menos tempo na resolução do sistema $120p80$ (tabela 3.27). O desempenho do GMRES($k$) associado

a um pré-condicionador pode ainda ser melhorado com o aumento do valor de $k$. O aumento deste parâmetro tem por consequência diminuir o número de iterações necessárias para se verificar a convergência (ver por exemplo a tabela 3.15). Mas, em contrapartida é responsável pelo aumento do tempo de cálculo. Este aspecto pode ser compensado pela utilização de um pré-condicionador adequado.

O PGMRES(10) não apresenta vantagens relativamente ao GMRES(10) com pré-condicionamento convencional. Em todos os casos analisados o PGMRES(10) não converge em menos tempo do que o GMRES(10).

O DQGMRES($k$) converge sempre no mesmo número de iterações, qualquer que seja a dimensão da base do sub-espaço de Krylov $k$, contrariamente ao FOM($k$) e ao GMRES($k$). Converge em 133 iterações para a resolução do sistema $40p40$, 263 para o sistema $80p80$, 342 para o sistema $120p80$ e 723 iterações para o sistema $120p120$. Em todos eles verifica-se também que os tempos mínimos conseguidos pelo DQGMRES($k$) são conseguidos para $k = 2$. O DQGMRES($k$), com $k = 10$ ou $k = 2$ é o único método que converge sem pré-condicionamento em todos os casos analisados. Sem pré-condicionamento o DQGMRES(2) utiliza menos iterações mas demora mais tempo do que o CG quando este converge. A única excepção verifica-se para o sistema das pressões $120p80$, em que o DQGMRES(2) demora 3.11 segundos e utiliza 342 iterações e o CG demora 3.70 segundos e utiliza 640 iterações (tabela 3.24).

O DQGMRES($k$) é uma alternativa mais eficiente ao GMRES($k$). Não depende tanto do pré-condicionamento e da capacidade de memória como o GMRES($k$), pois converge em todas as situações analisadas sem pré-condicionamento, obtendo o tempo mais baixo para $k = 2$. Além disso não está sujeito aos fenómenos de estagnação do resíduo, evidenciado pelo GMRES(10) na resolução do sistema $120p120$ (ver figura 3.5).

### 4.1.3 A eficiência dos pré-condicionadores

De todos os pré-condicionadores utilizados o ILU($\ell$) e o ILUDP(`tol`,$\alpha$) são os que conseguem melhorar mais a eficiência dos métodos iterativos testados. Todos os parâmetros destes pré-condicionadores necessitam de uma optimização prévia em função do método a utilizar e do sistema a resolver. Devido ao elevado número de cálculos que seria necessário realizar para o efeito, testaram-se apenas alguns destes parâmetros em situações pontuais. Os resultados obtidos mostram que o pré-condicionador ILU($\ell$), aplicado ao sistema das pressões $80p80$, deixa de ser eficiente sobre todos os métodos para $\ell \geq 7$, i.e., a partir de valores de $\ell$ superiores a 7 o tempo de convergência aumenta para todos os métodos (figura 3.4). Para valores de $\ell$ inferiores a 7, a eficiência deste pré-condicionador varia de método para método.

No caso do pré-condicionador ILUDP(0.01,$\alpha$), na resolução do mesmo sistema $80p80$ e para `tol`= 0.01, verifica-se que 8 dos 10 métodos testados obtêm melhores resultados para $\alpha = 0.75$ (tabela 3.19). Em relação ao parâmetro `tol`, cuja diminuição corresponde a uma melhor aproximação do pré-condicionador à matriz dos coeficientes, verifica-se que ao baixar de 0.01 para 0.001, na resolução do sistema $120p120$, o tempo total de convergência

de 3 métodos diminui enquanto que para 7 métodos esse tempo aumenta (tabela 3.34). Os métodos que melhoram os seus desempenhos (FOM, GMRES e PGMRES) são métodos que não convergem dentro do limite fixado sem a utilização de pré-condicionador. Este resultado mostra que a utilização do pré-condicionador ILUDP(tol,$\alpha$) com maior precisão apenas se justifica para métodos que necessitem de muitas iterações para convergir sem pré-condicionamento, de forma a amortizar o tempo suplementar necessário à elaboração de um pré-condicionador com maior precisão.

### 4.1.4 Qual o método escolhido

A opção por um dos métodos em detrimento de outros com base nos resultados obtidos terá que ter em conta a utilização ou não de um pré-condicionador. No caso de não se utilizar pré-condicionamento, o método mais adequado é o DQGMRES(2). O DQGMRES($k$) é o único método que converge em todas as situações, dentro do limite de iterações pré-definido. A opção por $k = 2$ minimiza os tempos de convergência. Por exemplo na resolução do sistema 120$p$120 sem pré-condicionamento os únicos métodos a convergir são CG, DQGMRES(2) e DQGMRES(10), verificando-se os seguintes tempos de convergência, respectivamente: 6.98, 10.35 e 22.73 segundos (tabela 3.30). Se não existissem sistemas não-simétricos para solucionar, a nossa escolha recaíria obviamente no CG.

No caso de se utilizar pré-condicionamento, a nossa escolha recai sobre o TFQMR associado ao pré-condicionador ILU(6) ou, em alternativa, sobre o BICGSTAB associado ao pré-condicionador ILUDP(0.01,0.75). Estes dois métodos com os respectivos pré-condicionadores convergem em todas as situações analisadas, em tempos próximos ou iguais aos melhores. Na resolução do sistema 120$p$120 o melhor tempo de convergência é conseguido pelo TFQMR com pré-condicionamento ILU(6), seguido do BICGSTAB com pré-condicionamento ILUDP(0.01,0.75): 4.74 e 5.50 segundos (tabelas 3.32 e 3.33).

O ganho resultante da substituição do TDMA por um dos métodos escolhidos não é directamente quantificável pois este método não convergiu em qualquer um dos sistemas da pressão, em menos de 1000 iterações. No entanto, comparando o tempo de convergência destes métodos com o tempo utilizado pelo TDMA para realizar 1000 iterações verifica-se uma diferença considerável. Por exemplo na resolução do sistema 120$p$120 o TFQMR associado ao ILU(6) converge num tempo 10 vezes inferior ao utilizado pelo TDMA para realizar 1000 iterações, sendo o resíduo correspondente à milésima iteração: 1.128E-04 (tabelas 3.30 e 3.32). Na resolução do mesmo sistema o DQGMRES(2), sem pré-condicionador, converge em cerca de 4 vezes menos tempo do que o TDMA demora a realizar 1000 iterações (tabela 3.30). Todos estes resultados mostram que a utilização de um dos métodos escolhidos em vez do TDMA, no programa de CFD, se traduz numa grande redução do tempo de resolução dos sistemas derivados do campo da pressão.

## 4.2 Trabalhos futuros

Com uma ordem que nos parece ser também a mais correcta para a sua implementação inclui-se uma lista de tarefas que poderão dar continuidade a esta tese.

1. O trabalho de resolução dos sistemas aqui apresentado, foi realizado de forma independente, pelo que será necessário incluir o *solver* seleccionado no programa de CFD. Para tal, será necessário estudar a melhor forma de compatibilizar a estrutura de dados entre os dois módulos. Uma possibilidade poderá consistir em incluir uma unidade adicional de transferência de dados da forma RGS para a forma CSR. Neste caso será necessário analisar o efeito dos tempos adicionais, devidos a esta unidade, sobre o desempenho global do *solver*.

2. Os resultados aqui apresentados assentam na resolução de sistemas com características específicas. Estas poderão variar em função da geometria do domínio ou do número de iterações do programa de CFD. Por estas razões será necessário estudar as características das matrizes em função destas duas condicionantes e verificar se os resultados aqui estabelecidos se podem generalizar.

3. A resolução de sistemas com grandes dimensões passa pela utilização de plataformas com múltiplos processadores e métodos iterativos paralelizados. Os testes efectuados neste trabalho deverão também ser efectuados em paralelo de maneira a confrontar os resultados. Para o efeito poderá ser utilizada a biblioteca *Psparslib* que contém os métodos iterativos, utilizados neste trabalho, implementados em paralelo.

# Bibliografia


P. Amodio and F. Mazzia. Parallel iterative solvers for banded linear systems. In *Numerical Analysis and Its Applications. First International Workshop*, Berlin, 1997. Springer-Verlag.

A. Anderson, Z. Bai, C. Bishof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK: A portable linear algebra library for high-performance computers.* SIAM, Philadelphia, PA, 1992. http://www.netlib.org/LAPACK.html.

P. Areal. Paralelização do algoritmo simple para dinâmica dos fluidos computacional utilizando sub-divisão do domínio. Master's thesis, FEUP, 1999.

R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, Philadelphia, PA, 1994.

J. Cela, J. Labarta, A. Perez, and J. Navarro. Performance on distributed memory multicomputers of domain decomposition solvers. In *Seventh SIAM Conference on Parallel Processing for Scientific Computing*, Philadelphia, 1995. SIAM.

F. Chaitin-Chatelin and V. Frayssé. *Lectures on Finite precision Computations.* SIAM, Philadelphia, PA, 1996.

J. Dongarra, I. Duff, D. Sorensen, and H. Van der Vorst. *Numerical Linear Algebra for High-Performance Computers.* SIAM, Philadelphia, PA, 1998.

J. Dongarra and V. Eijkhout. Standardized numerical linear algebra software. *Encyclopedia of Computers Science and Technology*, vol. 41, 1999.

J. Dongarra, V. Eijkhout, and A. Kalhan. Reverse comunication interface for linear algebra templates for iterative methods. Technical report, University of Tennessee, 1995. CS-95-291.

V. Eijkhout. Overview of iterative linear solver packages. Technical report, University of Tennessee, 1998. CS-98-411.

J. Ferziger. *Numerical Methods for Engineering Application.* John Wiley & Sons, Inc, New York, 1998.

G. Golub and C. Van Loan. *Matrix Computation.* The Johns Hopkins University Press, Baltimore and London, 1989.

A. Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, Philadelphia, 1997.

A. Krommer, M. Derakhshan, and Hammarling. Solving pde problems on parallel and distributed computer systems using the nag parallel library. In *High-Performance Computing and Networking. International Conference and Exhibition*, Berlin, 1997. Springer-Verlag.

S. V. Patankar and B. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flaws. *International Jornal of Heat and Mass Transfer*, vol. 15, pp. 1787-1806, 1972.

H. Pina. *Métodos Numéricos.* McGraw-Hill, Lisboa, 1995.

Y. Saad. Sparskit: a basic tool kit sparse matrix computations. Technical report, Research Institute for Advanced Computer Science, NASA Ames Reasearch Center, 1990. http://www.cs.umn.edu/Research/arpa/SPARSKIT/sparskit.html.

Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co., Boston, 1996.

Y. Saad and A. Malevsky. A portable library of distributed memory sparse iterative solvers. In *Parallel Computing Technologies*, St. Petersburg, Russia, 1995. V. E. M. et al., ed. http://www.cs.umn.edu/Research/arpa/psparslib/psp-abs.html.

Y. Saad and H. Van Der Vorst. Iterative solution of linear sistems in the 20-th century. *JCAM*, 1999.