SOLUÇÕES IMPLÍCITAS DAS EQUAÇÕES DE EULER EMPREGANDO ESTRUTURA DE DADOS POR ARESTAS

Lucia Catabriga

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

Prof. Alvaro L. G. de A. Coutinho, D.Sc.

Prof. José L. D. Alves, D.Sc.

Prof. Luiz Landau, D.Sc.

Prof. Abimael F. D. Loula, D.Sc.

Dr. Paulo A. B. Sampaio, Ph.D.

Prof. João L. F. Azevedo, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
MAIO DE 2000

CATABRIGA, LUCIA

Soluções Implícitas das Equações de Euler empregando Estrutura de Dados por Arestas [Rio deJaneiro] 2000

XIII, 103 p., 29.7 cm, (COPPE/UFRJ, D.Sc., Engenharia Civil, 2000)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1 - Elementos Finitos

2 - Formulações Estabilizadas

3 - Estrutura de Dados por Arestas

I. COPPE/UFRJ II. Título (Série)

Às minhas estrelinhas Lucas e Clara

Agradecimentos

Gostaria de agradecer a todos que, direta ou indiretamente, contribuiram para que este trabalho se realizasse. Em especial, agradeço:

Ao Prof. Alvaro Luiz Gayoso de Azeredo Coutinho, pelo apoio, amizade e dedicação demonstrado durante todo o período de orientação.

Ao PICD/CAPES pelo apoio financeiro através da bolsa de doutorado, indispensável à realização deste trabalho.

Aos professores, funcionários e alunos do Programa de Engenharia Civil da COPPE-UFRJ, além dos funcionários do Laboratório de Métodos Computacionais em Engenharia (LAMCE) pelo apoio oferecido de várias formas e nas várias etapas da realização desta pesquisa.

Ao Prof. Fernando L. B. Ribeiro pela disponibilidade do Visualizador *View3d*.

À Marcos André Duarte Martins pelos códigos computacionais para formação das arestas que iniciaram a pesquisa e pelas discussões ao longo do trabalho.

À Regina Célia de Almeida pelas discussões que muito ajudaram nos acertos da formulação e implementação das equações de Euler por elemento.

Ao Prof. Tayfun E. Tezduyar pelas discussões realizadas em várias etapas do trabalho.

Às minhas queridas amigas Maria Cristina Rangel, Sandra Mara Cardoso Malta e Maria Claudia Silva Boeres pelas discussões não tão acadêmicas, mas que muito contribuíram para encarar com tranquilidade e bom humor tantas horas de trabalho.

Aos meus familiares, especialmente ao meu querido companheiro Antônio Rocha Neto, pelo carinho e apoio constante demonstrados em todos os momentos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

SOLUÇÕES IMPLÍCITAS DAS EQUAÇÕES DE EULER EMPREGANDO ESTRUTURA DE DADOS POR ARESTAS

Lucia Catabriga

Abril/2000

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho

Programa: Engenharia Civil

As equações de Euler compressíveis tem sido amplamante estudadas nos últimos anos devido a sua grande aplicabilidade. Formulações explícitas tipo *upwind* de volumes finitos e diferenças finitas combinadas com estruturas de dados baseadas nas arestas possuem estratégia de solução mais rápida, particularmente para aplicações tridimensionais. Recentemente tem sido mostrado que métodos implícitos, tipo *upwind* livres de matriz baseados nas arestas, podem ser mais rápidos que métodos explícitos, sem carregamentos adicionais à memória.

Neste trabalho é apresentada uma implementação baseada nas arestas da formulação SUPG semi-discreta com captura de choque para as equações de Euler em variáveis conservativas. Desmembrando as matrizes dos elementos finitos resultantes em suas contribuições das arestas, os coeficientes das arestas, os resíduos e o produto matriz-vetor necessários nas técnicas de atualização de Krylov são calculados usando estrutura de dados por aresta. O método de solução resultante é mais rápido e requer menos memória que implementações baseadas nos elementos. Além disso, é introduzida uma estratégia de passo local combinada com um processo de congelamento do termo de captura de descontinuidade para acelerar a convergência para o regime permanente. Nos experimentos numéricos mostra-se que os esquemas implícitos com estrutura de dados por aresta podem convergir até a precisão da máquina.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

IMPLICIT SUPG SOLUTION OF EULER EQUATIONS USING EDGE-BASED DATA STRUCTURES

Lucia Catabriga

April/2000

Advisor: Alvaro Luiz Gayoso de Azeredo Coutinho
Department: Civil Engineering

The compressible Euler equations have been studied extensively in the past years because of their broad range of applicability. Explicit upwind finite volume and finite element formulations combined with edge-based data structures provide a fast solution strategy, particularly for 3D applications. Recently it has been shown that an implicit upwind, matrix-free edge-based solution method can be faster than explicit methods, without a heavy memory burden.

In this work we present an implicit, edge-based implementation of the semi-discrete SUPG formulation with shock capturing for the Euler equations in conservative variables. By disassembling the resulting finite element matrices into their edge contributions, edge coefficients, residuals and matrix-vector products needed in Krylov-update techniques are computed based on edge data structures. The resulting solution method is faster and requires less memory than the standard element-based implementation. Further, we introduce a local time-stepping strategy combined with a procedure to freeze the shock-capturing term when convergence stagnates. We show in our numerical experiments that our implicit, edge-based frozen-jacobian scheme converges towards machine precision.

"É preciso sonhar, mas sob a condição de se crer seriamente em nosso sonho, de se examinar com atenção a vida real, de confrontar nossas observações com nosso sonho, de realizar escrupulosamente nossa fantasia."

(J. Cortázar)

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Aplicações do método dos elementos finitos a problemas de mecânica dos fluidos se tornaram populares no final da década de 60. Porém, bons resultados usando a formulação padrão de Galerkin só eram encontrados para fluidos potenciais. Soluções precisas para problemas predominantemente convectivos ou escoamentos compressíveis só eram obtidas por diferenças finitas ou volumes finitos com a inclusão de alguma técnica de viscosidade artificial [2].

Soluções usando a formulação de Galerkin em problemas compressíveis são completamente deterioradas por oscilações espúrias que se propagam por todo o domínio. As formulações Petrov-Galerkin, que modificam o espaço de funções peso, surgiram na tentativa de controlar essas oscilações . O uso de funções descontínuas, primeiramente proposto por HUGHES e BROOKS [3] com a formulação SUPG (*Streamline Upwind Petrov-Galerkin*) abriu um leque de possibilidades no controle dessas instabilidades. TEZDUYAR e HUGHES [6] usam a formulação SUPG para solucionar a equação de Euler compressível. Atenção especial foi dada ao tratamento de grandes velocidades e ocorrência de choques usando variáveis conservativas. Os resultados encontrados são satisfatórios, porém oscilações espúrias ainda permaneceram na vizinhança das regiões de gradientes elevados.

A utilização de variáveis de entropia em elementos finitos foi apresentada por HUGHES et al. [8, 9], permitindo simetrizar as matrizes jacobianas da forma quasi-linear da equação de Euler (ou de Navier-Stokes). Demonstrou-se que as soluções obtidas a partir desta formulação satisfazem automaticamente a lei termodinâmica de conservação de entropia. SHAKIB [20] usou variáveis de entropia e acrescentou à formulação de Galerkin espaço-tempo com Mínimos Quadrados (GLS - *Galerkin Least Squares*) um termo para evitar as oscilações nas regiões de gradientes elevados, denominado "captura de descontinuidades". ALMEIDA e GALEÃO [7] acrescentaram à formulação espaço-tempo SUPG o operador de captura de choque CAU (*Consistent Approximate Upwind Method*), proporcional ao resíduo no interior dos elementos. Resultados numéricos evidenciam o excelente desempenho da formulação SUPG+CAU.

Comparações entre variáveis conservativas e de entropia têm sido feitas nos últimos anos. Usando uma formulação SUPG semi-discreta com captura de descontinuidades, LE BEAU et al. [11] e ALIABADI et al. [12] mostram que formulações baseadas em variáveis conservativas produzem soluções similares àquelas obtidas com variáveis de entropia. Nessas formulações o termo SUPG é ponderado através de uma matriz diagonal, simplificando a implementação. Usando uma formulação espaço-tempo baseada em variáveis de entropia, HAUKE e HUGHES [17] fizeram um estudo comparativo de vários tipos de variáveis (entropia, conservativa, primitiva com relação à pressão e primitiva com relação à densidade), definindo tensores de transformação de variáveis a partir das variáveis de entropia. O estudo conclui que existe semelhança entre as soluções obtidas no limite compressível e que somente variáveis de entropia e primitivas (em função da pressão) conduzem a métodos conservativos no limite incompressível. Todos os resultados são satisfatórios, mas as variáveis de entropia e primitivas produzem resultados melhores que variáveis de conservação.

ALIABADI e TEZDUYAR [13] empregam formulações SUPG semi-discretas e espaço-tempo com captura de choque na equações de Euler e Navier-Stokes usando variáveis conservativas. As variáveis conservativas foram escolhidas pela simplicidade, já que em trabalhos anteriores foi observada sua eficiência. As implementações desenvolvidas são aplicadas a problemas de grande porte, sendo essencialmente livres de matrizes e paralelas. O algoritmo de avanço no tempo é do tipo implícito preditor/multicorretor. ALIABADI e TEZDUYAR [10] concluem que para malhas fixas soluções muito boas são encontradas pela formulação semi-discreta, não sendo portanto necessário usar a formulação espaço-tempo.

Por outro lado, estruturas de dados baseadas nas arestas tem sido empregadas para soluções explícitas de elementos finitos para fluidos compressíveis em malhas não estruturadas compostas de triângulos e tetraedros [22, 23]. Estas estruturas de dados foram desenvolvidas a partir do método dos volumes finitos aplicado a escoamentos compressíveis em malhas não-estruturadas [24, 27]. As vantagens dos esquemas baseados nas arestas em relação aos esquemas convencionais são uma considerável redução nas operações de endereçamentos indiretos e na memória utilizada [28]. Permitem ainda implementações diretas dos esquemas de estabilização baseado nas linhas de corrente (*upwind*), no contexto do método dos elementos finitos. Para reduzir ainda mais o número de endereçamentos indiretos baseados nas arestas, LOHNER [29] propõe novas estruturas de dados, conhecidas por: *stars*, *superedges* e *chains*. Tais estruturas podem ser consideradas como agrupamento de arestas simples. Dentre elas, as super-arestas (*superedges*) são as mais apropriadas para implementação em computadores paralelos. Porém, em uma implementação paralela recente da solução da equação de Euler bidimensional foi observada uma melhora insignificante do tempo de CPU ao se usar super-arestas em relação aos tempos obtidos com arestas simples [30].

A estrutura de dados baseada nas arestas também pode ser vista como uma representação do grafo nodal da malha composta por triângulos e tetraedros. Pode-se dizer então que a representação por arestas é uma estrutura de dados alternativa para calcular o produto matriz-vetor global necessário nas técnicas iterativas baseadas nos espaços de Krylov. O desempenho dos primeiros experimentos efetuados por VENKATAKRISHNAN e MAVRIPLIS [31] da equação de Navier-Stokes foram promissores. Além disso, VENKATAKRISHNAN [32] demonstra que o produto matriz-vetor baseado nas arestas usando a matriz transposta é mais eficiente que o produto usando a matriz esparsa com a estrutura convencional. Também foi verificado que em soluções paralelas por elementos finitos de vários problemas tridimensionais de escoamento potencial para configurações industriais, as estruturas de dados por aresta são mais eficientes que a estrutura de dados baseada nos elementos [21]. Recentemente LUO et al. [34] empregaram um esquema implícito livre de matrizes com formulação *upwind* para problemas compressíveis com grande sucesso.

Os custos computacionais de cada passo de tempo na solução de problemas de mecânica dos fluidos é, reconhecidamente, muito elevado. Métodos de aceleração de convergência devem estar prioritariamente baseados na difícil tarefa de acelerar a velocidade de convergência, levando-se em consideração tais custos. Os esforços na tentativa de acelerar a convergência podem estar concentrados na própria definição do modelo, ou seja, um modelo que retrata com mais detalhes o fenômeno físico tende a convergir mais rápidamente. Por outro lado, aumentando a complexidade das equações, em muitos casos o processo de solução acaba ficando muito difícil, ou mesmo impossível. Assim, os maiores esforços para acelerar a convergência acabam se concentrando nos algoritmos de solução. MAVRIPLIS [25] relata que estudos realizados pela *National Science Foundation* concluem que avanços nos algoritmos de solução de problemas de mecânica dos fluidos crescem na mesma proporção que avanços de *hardware*. Portanto, a aceleração de convergência está intimamente ligada a qualidade dos algoritmos de solução desenvolvidos, que por sua vez são responsáveis majoritariamente pelos custos efetivos de *hardware*.

A velocidade dos processadores e a disponibilidade de memória tem crescido bastante nas duas últimas décadas, porém os procedimentos de acesso à memória não têm acompanhado esse crescimento [24]. O desenvolvimento de algoritmos que buscam otimizar o acesso aos processadores é um importante campo de pesquisa na atualidade. A reordenação dos nós de malhas não estruturadas é um exemplo de estratégia que visa otimizar o acesso à memória *cache*, aumentando a localidade dos dados [41]. Em muitos casos esses procedimentos tem dobrado, ou mesmo triplicado o desempenho computacional.

Este trabalho emprega uma formulação SUPG semi-discreta em variáveis de conservação para a equação de Euler, similar à de ALIABADI e TEZDUYAR [13]. Discute-se a incorporação de um operador tipo CAU à esta formulação. Considera-se uma particularização da formulação para malhas não estruturadas compostas de triângulos com aproximação linear onde, para aumentar a eficiência da implementação, evitou-se o uso de integração numérica. É introduzida nessa formulação uma estrutura de dados baseada nas arestas dos elementos. Tais estruturas são definidas a partir da malha de elementos finitos formadas por triângulos, sendo feito um desmembramento das contribuições de cada aresta. Comparações entre as implementações por elemento e aresta mostram a maior eficiência do esquema por aresta em relação ao esquema por elemento, levando-se em conta o tempo de processamento e a memória utilizada. Os resultados numéricos da formulação proposta são validados através da comparação com soluções exatas de problemas existentes na literatura. Também são desenvolvidas técnicas de aceleração de convergência dos processos de avanço no tempo. Mostra-se que com a combinação de algumas técnicas é possível atingir a precisão da máquina no algoritmo de avanço no tempo com um número reduzido de passos.

O restante deste texto é organizado como se segue. No próximo capítulo apresenta-se as equações de Euler em duas dimensões. No capítulo 3 descreve-se a formulação de elementos finitos adotada, onde são apresentadas as matrizes dos elementos e as componentes do resíduo para a estrutura de dados por elemento. O capítulo seguinte apresenta a mesma formulação usando a estrutura de dados por aresta. É feito um estudo de estratégias de armazenamento mostrando as vantagens das arestas sobre os elementos. No final dos capítulos 3 e 4 são apresentados exemplos de validação para todas as formulações e estratégias adotadas. O capítulo 5 trata de técnicas que visam acelerar a convergência do processo de avanço no tempo. É apresentada uma técnica de escolha do passo no algoritmo de avanço no tempo e uma estratégia de congelamento do operador de captura quando o resíduo de densidade estagna. Finalmente, no último capítulo são apresentadas as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Equações de Euler

A definição das equações que governam a Mecânica dos Fluidos está baseada nos conceitos presentes na Mecânica do Contínuo [4]. Em muitos casos na Mecânica dos Fluidos, forças que governam o movimento possuem denominações próprias, outras acabam carregando denominações clássicas da Mecânica dos Sólidos. Neste texto são usadas essas denominações sem a preocupação de definir cada propriedade. Maiores detalhes podem ser encontrados em [1] e [2].

2.1 Propriedades Físicas de Fluidos

2.1.1 Tensões em Fluidos

A característica essencial de um fluido é sua inabilidade de suportar tensões cisalhantes em repouso. Um fluido só suporta pressões, ou seja, "tensões" hidrostáticas. Portanto, a variável essencial na Mecânica dos Fluidos é a velocidade:

$$\mathbf{u} = \{u_1, u_2, u_3\}^T \quad \text{ou} \quad u_i \quad \text{em notação indicial} \tag{2.1}$$

A taxa de deformação, que é a causa primária das tensões σ_{ij}, é definida de forma análoga as deformações infinitesimais existentes na Mecânica dos Sólidos, ou seja:

$$\dot{\varepsilon}_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \tag{2.2}$$

A relação tensão-deformação de um fluido linear (ou Newtoniano) isotrópico requer a definição de duas constantes:

- viscosidade, que relaciona as tensões desviadoras τ_{ij} às taxas de deformações desviadoras:

$$\tau_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{ii} = 2\mu(\dot{\varepsilon}_{ij} - \frac{1}{3}\delta_{ij}\dot{\varepsilon}_{ii}) \tag{2.3}$$

onde δ_{ij} é o delta de Kroneker; $\sigma_{ii} = \sigma_{11} + \sigma_{22} + \sigma_{33}$ e $\dot{\varepsilon}_{ii} = \dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}$; e μ é a viscosidade cisalhante, ou simplesmente a viscosidade, sendo análoga ao módulo de cisalhamento G em elasticidade, definido como, $G = \frac{E}{2(1+\nu)}$, sendo E o módulo de Young e ν o coeficiente de Poisson.

- coeficiente de viscosidade volumétrica, que estabelece uma relação entre as variações médias de tensão e as taxas de deformações volumétricas, o que permite definir a pressão como:

$$p = -\frac{\sigma_{ii}}{3} = \kappa_{vol}\dot{\varepsilon}_{ii} - p_0 \tag{2.4}$$

onde κ_{vol} é o coeficiente de viscosidade volumétrica, análogo ao módulo de deformação volumétrica B em elasticidade linear ($B = \frac{E}{3(1-2\nu)}$); p_0 é a pressão hidrostática inicial, que independe da taxa de deformação. É importante lembrar que, tanto p como p_0 são definidas como positivas quando as tensões forem de compressão.

Sendo assim, pode-se escrever as relações constitutivas para fluidos Newtonianos isotrópicos por:

$$\sigma_{ij} = \underbrace{2\mu(\dot{\varepsilon}_{ij} - \frac{\delta_{ij}\dot{\varepsilon}_{ii}}{3})}_{\tau_{ij}} + \underbrace{\delta_{ij}\kappa_{vol}\dot{\varepsilon}_{ii} - \delta_{ij}p_0}_{\delta_{ij}p} \tag{2.5}$$

ou

$$\sigma_{ij} = \tau_{ij} + \delta_{ij}p \tag{2.6}$$

ou, ainda

$$\sigma_{ij} = 2\mu\dot{\varepsilon}_{ij} + \delta_{ij}(\kappa_{vol} - 2/3\mu)\dot{\varepsilon}_{ii} - \delta_{ij}p_0 \tag{2.7}$$

usando a notação de Lamé, $\kappa_{vol} - 2/3\mu = \lambda$, tem-se:

$$\sigma_{ij} = 2\mu\dot{\varepsilon}_{ij} + \delta_{ij}\lambda\dot{\varepsilon}_{ii} - \delta_{ij}p_0 \tag{2.8}$$

assumindo que não há viscosidade volumétrica, ou seja,

$$\kappa_{vol} \approx 0 \quad \Rightarrow \quad \lambda = 2/3\mu \tag{2.9}$$

a relação constitutiva pode ser reescrita por,

$$\sigma_{ij} = 2\mu(\dot{\varepsilon}_{ij} + \frac{\delta_{ij}\dot{\varepsilon}_{ii}}{3}) - \delta_{ij}p \equiv \tau_{ij} - \delta_{ij}p \tag{2.10}$$

e reescrevendo-se a expressão de τ_{ij}, tem-se:

$$\tau_{ij} = \mu\left[(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) - \delta_{ij}\frac{2}{3}\frac{\partial u_i}{\partial x_i}\right] \tag{2.11}$$

ou

$$\tau_{ij} = \mu(u_{i,j} + u_{j,i}) + \lambda\delta_{ij}u_{k,k} \tag{2.12}$$

2.1.2 Conservação de Massa

Considera-se um infinitésimo de fluido, com densidade ρ. A quantidade de massa que passa em uma área infinitesimal dA por unidade de tempo é denominada fluxo de massa, dada por ρu_i. Logo, se não há perda de massa, o balanço de fluido de massa entrando e saindo em um volume unitário estabelece que:

$$\frac{\partial\rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0 \tag{2.13}$$

ou

$$\frac{\partial\rho}{\partial t} + \nabla^T(\rho\mathbf{u}) = 0 \tag{2.14}$$

2.1.3 Conservação de Momentum

O momentum na direção j é dado por $(\rho u_j)u_i$. A diferença entre o momentum que entra e sai em um volume unitário deve estar em equilíbrio com as tensões σ_{ij} e as forças de corpo ρb_j, ou seja,

$$\frac{\partial(\rho u_j)}{\partial t} + \frac{\partial\left[(\rho u_j)u_i\right]}{\partial x_i} - \frac{\partial(\sigma_{ij})}{\partial x_i} - \rho b_j = 0 \tag{2.15}$$

ou

$$\frac{\partial(\rho u_j)}{\partial t} + \frac{\partial\left[(\rho u_j)u_i\right]}{\partial x_i} - \frac{\partial(\tau_{ij})}{\partial x_i} + \frac{\partial p}{\partial x_j} - \rho b_j = 0 \tag{2.16}$$

2.1.4 Conservação de Energia e Equação de Estado

Nas equações (2.14) e (2.16) as variáveis independentes são as velociadaes u_i, a pressão p e a densidade ρ. As tensões σ_{ij} são definidas em termos das velocidades e portanto não são independentes. Entretanto, se for possível assumir que a densidade é constante (caso dos fluidos incompressíveis) ou que exista uma relação entre pressão e densidade (fluidos isotérmicos com pequena compressibilidade), o sistema se torna completo e solucionável.

De forma geral, a pressão p, densidade ρ e a temperatura θ estão relacionados por uma equação de estado da forma,

$$\rho = \rho(p, \theta) \tag{2.17}$$

Por exemplo, para um gás ideal a equação (2.17) é dada por:

$$\rho = \frac{p}{R\theta} \tag{2.18}$$

onde R é a constante universal dos gases perfeitos. Ou ainda,

$$p = (\gamma - 1)\rho c_v \theta = \left(\frac{c_p}{c_v} - 1\right)\rho c_v \theta \tag{2.19}$$

onde c_p e c_v são os calores específicos a pressão e a volume constantes, respectivamente.

É necessário adicionar às equações governantes uma equação de conservação de energia. Para tal, faz-se necessário definir algumas grandezas:

- A energia intrínseca por unidade de massa, também denominada de "densidade de energia", depende do estado do fluido, ou seja, depende da pressão e da temperatura:

$$\imath = \imath(\theta, p) \tag{2.20}$$

- A energia total por unidade de massa inclui a energia cinética por unidade de massa. Também é denominada de "densidade total de energia" e possui a expressão:

$$e = \imath + \frac{|\mathbf{u}|^{\mathbf{2}}}{2} = \imath + \frac{u_i u_i}{2} \tag{2.21}$$

- A entalpia, dada por:

$$h = \imath + \frac{p}{\rho} \tag{2.22}$$

ou

$$H = h + \frac{|\mathbf{u}|^{\mathbf{2}}}{2} = e + \frac{p}{\rho} \tag{2.23}$$

É importante lembrar que a transferência de energia pode se dar por convecção ou por difusão. O fluxo de calor difusivo é definido pela lei de Fourier,

$$q_i = -\kappa \frac{\partial \theta}{\partial x_i} = -\kappa \theta_{,i} \tag{2.24}$$

onde κ é o coeficiente de condutividade térmica. Para completar a definição da equação de balanço de energia é necessário especificar os termos de fonte de calor. Estes devem incluir termos devido a reações químicas, r, e dissipação de energia devido as tensões, isto é,

$$\frac{\partial(\sigma_{ij}u_j)}{\partial x_i} = \frac{\partial(\tau_{ij}u_j)}{\partial x_i} - \frac{\partial(pu_j)}{\partial x_j} \tag{2.25}$$

Logo, a conservação de energia em um volume unitário é dada por,

$$\frac{\partial(\rho e)}{\partial t} + \frac{\partial(\rho u_i e)}{\partial x_i} + \frac{\partial q_i}{\partial x_i} + \frac{\partial(pu_i)}{\partial x_i} - \frac{\partial(\tau_{ji}u_i)}{\partial x_i} - \rho b_i u_i - r = 0 \tag{2.26}$$

ou sabendo que $\rho u_i H = \rho u_i e + u_i p$, tem-se:

$$\frac{\partial(\rho e)}{\partial t} + \frac{\partial(\rho u_i H)}{\partial x_i} + \frac{\partial q_i}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\kappa\frac{\partial T}{\partial x_i}\right) - \frac{\partial(\tau_{ji}u_i)}{\partial x_i} - \rho b_i u_i - r = 0 \tag{2.27}$$

onde o termo $-\rho b_i u_i$ representa o trabalho realizado pelas forças de corpo. As expressões (2.14), (2.16) e (2.27) representam as equações gerais da Mecânica dos Fluidos, sendo denominadas de equações de Navier-Stokes.

2.2 Equações Governantes da Mecânica dos Fluidos

As equações de Navier-Stokes (N-S), dadas pelas expressões (2.14), (2.16) e (2.27), podem ser escritas de forma mais compacta, ou de forma conservativa, como:

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} = \mathbf{F}^d_{i,i} + \mathcal{F} \quad \text{em} \quad \Omega \times [0, T] \tag{2.28}$$

onde para $\Omega \subset \Re^3$ e $t \in [0, T]$, tem-se,

$$\mathbf{U} = \underbrace{\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix}}_{\text{variáveis conservativas}} = \underbrace{\rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{Bmatrix}}_{\text{variáveis primitivas}}$$

$$\mathbf{F}_i = \rho u_i \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{Bmatrix} + p \begin{Bmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{Bmatrix} \quad \text{fluxo de Euler}$$

$$\mathbf{F}_i^d = \begin{Bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij}u_j \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -q_i \end{Bmatrix} \quad \text{fluxo difusivo}$$

$$\mathcal{F} = \rho \begin{Bmatrix} 0 \\ b_1 \\ b_2 \\ b_3 \\ b_i u_i + r \end{Bmatrix} \quad \text{vetor fonte}$$

onde,

ρ é a densidade do fluido,

$\mathbf{u} = \{u_1, u_2, u_3\}^T$ é o vetor de velocidades,

e é a densidade total de energia,

p é a pressão termodinâmica,

δ_{ij} é o delta de Kronecker,

$\tau = [\tau_{ij}]$ é o tensor de tensões viscosas,

$\mathbf{q} = \{q_1, q_2, q_3\}^T$ é o vetor de fluxo de calor,

$\mathbf{b} = \{b_1, b_2, b_3\}^T$ é o vetor de forças de corpo por unidade de massa gravitacional, centrífuga ou de coriolis.

r é o calor por unidade de massa.

As equações presentes em (2.28) representam a conservação de massa, momentum e energia. A elas deve-se adicionar as relações constitutivas e condições de contorno. As equações constitutivas usuais são:

$$\begin{array}{ll} \text{Equações dos gases perfeitos} & \imath = c_v\theta \;\text{ e }\; p = (\gamma - 1)\rho\imath \\ \text{Tensões viscosas} & \tau_{ij} = \lambda u_{k,k}\delta_{i,j} + \mu(u_{i,j} + u_{j,i}) \\ \text{Lei de Fourier} & q_i = -\kappa\theta_{,i} \end{array} \tag{2.29}$$

onde,

c_v é o calor específico à volume constante,

θ é a temperatura absoluta,

$\gamma = \frac{c_p}{c_v}$, sendo c_p o calor específico à pressão constante,

λ e μ são coeficientes de viscosidade,

κ é o coeficiente de condutividade térmica.

As equações (2.28) podem ainda ser escritas na forma quasi-linear:

$$\mathbf{U}_{,t} + \mathbf{A}_i \mathbf{U}_{,i} = (k_{i,j} \mathbf{U}_{,j})_{,i} + \mathcal{F} \tag{2.30}$$

onde,

$\mathbf{A}_i$ é i-ésima matriz Jacobiana de Euler

$\mathbf{K} = [k_{ij}]$ é a matriz de difusividade, tal que $k_{ij}\mathbf{U}_{,j} = \mathbf{F}_i^d$

Considerando Ω o domínio de definição do problema, com contorno $\Gamma = \Gamma_g \cup \Gamma_h$ as equações de N-S admitem condições de contorno dos tipos:

- Condições de Dirichlet: $\mathbf{Q}(\mathbf{U}) = g(t)$ em Γ_g, onde $\mathbf{Q}(\mathbf{U})$ é a função vetorial não-linear das variáveis de conservação $\mathbf{U}$ e $g(t)$ são os valores prescritos de $\mathbf{Q}$ no contorno Γ_g de Ω.

- Condições de Neumann: $\mathbf{F}_i$ e $\mathbf{F}_i^d$ são especificados no contorno Γ_h de Ω. Esta condição é similar a se especificar valores das derivadas primeiras de algumas ou todas as variáveis dependentes do problema.

2.3 Equações de Euler Bidimensionais

As equações de Euler podem ser entendidas como as equações de N-S compressíveis com difusão e termo fonte nulos, sendo representadas por:

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} = \mathbf{0} \quad \text{em} \quad \Omega \times [0, T] \tag{2.31}$$

ou,

$$\mathbf{U}_{,t} + \mathbf{A}_i \mathbf{U}_{,i} = \mathbf{0} \quad \text{em} \quad \Omega \times [0, T] \tag{2.32}$$

para $\Omega \subset \Re^{n_{sd}}$, $n_{sd} = 1, 2$ ou 3 e $t \in [0, T]$. As equações de Euler formam um sistema invíscido de equações de conservação, ou ainda, um conjunto acoplado de equações diferenciais parciais não lineares de primeira ordem. O presente trabalho limita-se a problemas onde $n_{sd} = 2$.

2.3.1 Matrizes Jacobianas dos Fluxos de Euler

As matrizes Jacobianas $\mathbf{A}_x$ e $\mathbf{A}_y$ são definidas por:

$$\mathbf{A}_x = \frac{\partial \mathbf{F}_x}{\partial \mathbf{U}} \quad ; \quad \mathbf{A}_y = \frac{\partial \mathbf{F}_y}{\partial \mathbf{U}} \tag{2.33}$$

Sendo $\mathbf{A}_i$ no caso bidimensional uma matriz de ordem 4×4, ou seja,

$$\mathbf{A}_i = \begin{bmatrix} \frac{\partial \mathbf{F}_i}{\partial U_1} & \frac{\partial \mathbf{F}_i}{\partial U_2} & \frac{\partial \mathbf{F}_i}{\partial U_3} & \frac{\partial \mathbf{F}_i}{\partial U_4} \end{bmatrix} \tag{2.34}$$

onde $\frac{\partial \mathbf{F}_i}{\partial U_j}$ é um vetor de 4 posições.

As componentes $\mathbf{F}_x$ e $\mathbf{F}_y$ são funções homogêneas de grau 1 com relação ao vetor de variáveis de conservação $\mathbf{U}$ para o caso de um gás perfeito. De forma geral, se o fluido satisfizer a relação:

$$p = \rho \hat{p}(\imath) \tag{2.35}$$

Tem-se que $\mathbf{F}(\lambda \mathbf{U}) = \lambda \mathbf{F}(\mathbf{U}), \quad \forall \lambda$, que implica em:

$$\mathbf{F}(\mathbf{U}) = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \mathbf{U} = \mathbf{A}\mathbf{U} \quad \text{(considerando } \lambda = 1\text{)} \tag{2.36}$$

obtido pela diferenciação com relação a λ. Em forma de componentes as matrizes Jacobianas podem ser representados por:

$$\begin{aligned} \mathbf{F}_x &= \mathbf{A}_x \mathbf{U} \\ \mathbf{F}_y &= \mathbf{A}_y \mathbf{U} \end{aligned} \tag{2.37}$$

Os vetores $\mathbf{F}_x$ e $\mathbf{F}_y$ podem ser reescritos em termos das variáveis primitivas ou das variáveis de conservação, obtendo-se:

$$
\mathbf{F}_x = \overbrace{\begin{Bmatrix} \rho u_1 \\ \\ \rho u_1 u_1 + p \\ \\ \rho u_1 u_2 \\ \\ \rho u_1 e + p u_1 \end{Bmatrix}}^{\text{Primitivas}} = \overbrace{\begin{Bmatrix} U_2 \\ \dfrac{U_2^2}{U_1} + p \\ \dfrac{U_2 U_3}{U_1} \\ \dfrac{U_2 U_4}{U_1} + p\dfrac{U_2}{U_1} \end{Bmatrix}}^{\text{Conservativas}} \tag{2.38}
$$

$$
\mathbf{F}_y = \begin{Bmatrix} \rho u_2 \\ \\ \rho u_2 u_1 \\ \\ \rho u_2 u_2 + p \\ \\ \rho u_2 e + p u_2 \end{Bmatrix} = \begin{Bmatrix} U_3 \\ \dfrac{U_3 U_2}{U_1} \\ \dfrac{U_3^2}{U_1} + p \\ \dfrac{U_3 U_4}{U_1} + p\dfrac{U_3}{U_1} \end{Bmatrix} \tag{2.39}
$$

Sendo a pressão p uma grandeza dependente das variáveis conservativas (ou primitivas), é possível reescrevê-la através da expressão:

$$
p = (\gamma - 1)\left[U_4 - \frac{\|U_{23}\|^2}{2U_1}\right] \tag{2.40}
$$

onde $\|U_{23}\|^2 \equiv \{U_2, U_3\}^T\{U_2, U_3\} = \rho^2|\mathbf{u}|^2$.

Para se obter as matrizes Jacobianas é necessário calcular as derivadas da pressão com relação a algumas grandezas conservativas. Assim, através de regras simples de diferenciação chega-se a:

$$
\begin{aligned}
\frac{\partial p}{\partial U_1} &= (\gamma - 1)\frac{\|U_{23}\|^2}{2U_1^2} = \frac{\partial p}{\partial \rho}(\gamma - 1)\frac{|\mathbf{u}|^2}{2} = \frac{\partial p}{\partial \rho} \\
\frac{\partial p}{\partial U_{23}} &= -(\gamma - 1)\rho U_{23}^T = -(\gamma - 1)\rho\mathbf{u}^T = \frac{\partial p}{\partial \mathbf{u}} \\
\frac{\partial p}{\partial U_4} &= (\gamma - 1) = \frac{\partial p}{\partial e}
\end{aligned} \tag{2.41}
$$

Portanto,

$$\frac{\partial \mathbf{F}_x}{\partial U_1} = \begin{Bmatrix} 0 \\ -\dfrac{U_2^2}{U_1^2} + (\gamma-1)\dfrac{\|U_{23}\|^2}{2U_1^2} \\ -\dfrac{U_2U_3}{U_1^2} \\ -\dfrac{U_2}{U_1^2}\left[\gamma U_4 - (\gamma-1)\dfrac{\|U_{23}\|^2}{U_1}\right] \end{Bmatrix} = \begin{Bmatrix} 0 \\ -u_1^2 + \dfrac{(\gamma-1)|\mathbf{u}|^2}{2} \\ -u_1u_2 \\ -u_1(\gamma e - (\gamma-1)|\mathbf{u}|^2) \end{Bmatrix} = \frac{\partial \mathbf{F}_x}{\partial \rho} \tag{2.42}$$

$$\frac{\partial \mathbf{F}_x}{\partial U_2} = \begin{Bmatrix} 1 \\ (3-\gamma)\dfrac{U_2}{U_1} \\ \dfrac{U_3}{U_1} \\ \gamma\dfrac{U_4}{U_1} - \dfrac{(\gamma-1)}{2U_1^2}(\|U_{23}\|^2 + 2U_2^2) \end{Bmatrix} = \begin{Bmatrix} 1 \\ (3-\gamma)u_1 \\ u_2 \\ \gamma e - \dfrac{(\gamma-1)}{2}(|\mathbf{u}|^2 + 2u_1^2) \end{Bmatrix} = \frac{\partial \mathbf{F}_x}{\partial u_1} \tag{2.43}$$

$$\frac{\partial \mathbf{F}_x}{\partial U_3} = \begin{Bmatrix} 0 \\ -(\gamma-1)\dfrac{U_3}{U_1} \\ \dfrac{U_2}{U_1} \\ -(\gamma-1)\dfrac{U_3U_2}{U_1^2} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -(\gamma-1)u_2 \\ u_1 \\ -(\gamma-1)u_1u_2 \end{Bmatrix} = \frac{\partial \mathbf{F}_x}{\partial u_2} \tag{2.44}$$

$$\frac{\partial \mathbf{F}_x}{\partial U_4} = \begin{Bmatrix} 0 \\ (\gamma-1) \\ 0 \\ \gamma\dfrac{U_2}{U_1} \end{Bmatrix} = \begin{Bmatrix} 0 \\ (\gamma-1) \\ 0 \\ \gamma u_1 \end{Bmatrix} = \frac{\partial \mathbf{F}_x}{\partial e} \tag{2.45}$$

A matriz Jacobiana $\mathbf{A}_x$ em variáveis primitivas é dada por:

$$\mathbf{A}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -u_1^2 + \frac{(\gamma-1)|\mathbf{u}|^2}{2} & (3-\gamma)u_1 & -(\gamma-1)u_2 & (\gamma-1) \\ -u_1u_2 & u_2 & u_1 & 0 \\ -u_1(\gamma e - (\gamma-1)|\mathbf{u}|^2) & \gamma e - \frac{(\gamma-1)}{2}(|\mathbf{u}|^2 + 2u_1^2) & -(\gamma-1)u_1u_2 & \gamma u_1 \end{bmatrix} \tag{2.46}$$

e em variáveis conservativas tem-se:

$$\mathbf{A}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{U_2^2}{U_1^2} + (\gamma-1)\frac{\|U_{23}\|^2}{2U_1^2} & (3-\gamma)\frac{U_2}{U_1} & -(\gamma-1)\frac{U_3}{U_1} & (\gamma-1) \\ -\frac{U_2U_3}{U_1^2} & \frac{U_3}{U_1} & \frac{U_2}{U_1} & 0 \\ -\frac{U_2}{U_1^2}\left[\gamma U_4 - (\gamma-1)\frac{\|U_{23}\|^2}{U_1}\right] & \gamma\frac{U_4}{U_1} - \frac{(\gamma-1)}{2U_1^2}(\|U_{23}\|^2 + 2U_2^2) & -(\gamma-1)\frac{U_3U_2}{U_1^2} & \gamma\frac{U_2}{U_1} \end{bmatrix} \tag{2.47}$$

Através de manipulações similares, obtem-se:

$$\mathbf{A}_y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -u_1u_2 & u_2 & u_1 & 0 \\ -u_2^2 + \frac{(\gamma-1)|\mathbf{u}|^2}{2} & -(\gamma-1)u_1 & (3-\gamma)u_2 & (\gamma-1) \\ -u_2(\gamma e - (\gamma-1)|\mathbf{u}|^2) & -(\gamma-1)u_1u_2 & \gamma e - \frac{(\gamma-1)}{2}(|\mathbf{u}|^2 + 2u_2^2) & \gamma u_2 \end{bmatrix} \tag{2.48}$$

$$\mathbf{A}_y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -\frac{U_2U_3}{U_1^2} & \frac{U_3}{U_1} & \frac{U_2}{U_1} & 0 \\ -\frac{U_3^2}{U_1^2} + (\gamma-1)\frac{\|U_{23}\|^2}{2U_1^2} & -(\gamma-1)\frac{U_2}{U_1} & (3-\gamma)\frac{U_3}{U_1} & (\gamma-1) \\ -\frac{U_3}{U_1^2}\left[\gamma U_4 - (\gamma-1)\frac{\|U_{23}\|^2}{U_1}\right] & -(\gamma-1)\frac{U_2U_3}{U_1^2} & \gamma\frac{U_4}{U_1} - \frac{(\gamma-1)}{2U_1^2}(\|U_{23}\|^2 + 2U_3^2) & \gamma\frac{U_3}{U_1} \end{bmatrix} \tag{2.49}$$

2.4 Simetrização das Equações Empregando Variáveis de Entropia

Com o objetivo de melhorar a precisão das soluções numéricas encontradas para as equações de Euler (ou N-S) tem sido investigado o uso de variáveis especiais, denominadas variáveis de entropia, que simetrizam as matrizes Jacobianas, [8], [12], [13], [20]. Para tal, define-se uma função escalar denominada função de entropia generalizada $H = H(\mathbf{U})$. Para

$$H(\mathbf{U}) = \rho s \tag{2.50}$$

onde $s = ln(\frac{P}{\rho^\gamma}) + c^{\underline{te}}$ é a entropia física por unidade de massa, define-se as variáveis de entropia por $\mathbf{V}^t = H_{,\mathbf{U}}$ e o mapeamento $\mathbf{U} \rightarrow \mathbf{V}$ é tal que:

$$\mathbf{V} = \left\{ \begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_4 \end{array} \right\} = \frac{1}{\rho\imath} \left\{ \begin{array}{l} -U_4 + \rho\imath(\gamma + 1 - s) \\ \quad U_2 \\ \quad U_3 \\ -U_1 \end{array} \right\} \tag{2.51}$$

onde $s = ln\left(\frac{(\gamma-1)\rho\iota}{U_1^\gamma}\right)$ e $\rho\iota = U_4 - \frac{\|U_{23}\|^2}{2U_1}$. O mapeamento inverso, $\mathbf{V} \rightarrow \mathbf{U}$ é dado por:

$$\mathbf{U} = \rho\iota \begin{Bmatrix} -V_4 \\ V_2 \\ V_3 \\ 1 - \dfrac{V_2^2 + V_2^3}{2V_4} \end{Bmatrix} \tag{2.52}$$

onde $s = \gamma - V_1 + \frac{V_2^2+V_2^3}{2V_4}$ e $\rho\iota = \left(\frac{\gamma-1}{(-V_4)^\gamma}\right)^{\frac{1}{(\gamma-1)}} exp\left(\frac{-s+s_0}{\gamma-1}\right)$. Fazendo uso da mudança de variável $\mathbf{U} = \mathbf{U}(\mathbf{V})$, definida pelas expressões (2.51) e (2.52), as equações de Euler tornam-se:

$$\tilde{\mathbf{A}}_0\mathbf{V}_{,t} + \tilde{\mathbf{A}}_i\mathbf{V}_{,i} = \mathbf{0} \quad \text{em} \quad \Omega \times [0,T] \tag{2.53}$$

onde $\tilde{\mathbf{A}}_0 = \mathbf{U}_{,V}$ é simétrica positiva-definida e $\tilde{\mathbf{A}}_i = \mathbf{A}_i\tilde{\mathbf{A}}_0$ é simétrica. As transformações $\tilde{\mathbf{A}}_0$ e $\tilde{\mathbf{A}}_i$ são obtidas a partir de diferenciações similares aquelas apresentadas em variáveis conservativas. As matrizes $\tilde{\mathbf{A}}_0$ e $(\tilde{\mathbf{A}}_0)^{-1}$ são dadas por:

$$\tilde{\mathbf{A}}_0 = \mathbf{U}_{,V} = \frac{\rho\iota}{(\gamma-1)V_4}\begin{bmatrix} -V_4^2 & V_2V_4 & V_3V_4 & V_4(1-k_1) \\ & \bar{\gamma}V_4 - V_2^2 & -V_2V_3 & V_2(k_1-\gamma) \\ & & \bar{\gamma}V_4 - V_3^2 & V_3(k_1-\gamma) \\ & sim & & -k_2 \end{bmatrix} \tag{2.54}$$

$$\tilde{\mathbf{A}}_0^{-1} = \mathbf{V}_{,U} = \frac{-1}{\rho\iota V_4}\begin{bmatrix} k_1^2+\gamma & k_1V_2 & k_1V_3 & (k_1+1)V_4 \\ & V_2^2 - V_4 & V_2V_3 & V_2V_4 \\ & & V_3^2 - V_4 & V_3V_4 \\ & sim & & V_4^2 \end{bmatrix} \tag{2.55}$$

onde $k_1 = \frac{V_2^2+V_2^3}{2V_4}$, $k_2 = k_1^2 - 2\gamma k_1 + \gamma$ e $\bar{\gamma} = \gamma - 1$.

A simetria de (2.53) é importante somente do ponto de vista matemático. A forma simétrica para as equações de Navier-Stokes possibilita uma situação global de estabilidade fazendo com que a solução encontrada satisfaça automaticamente a *Inequação de Clausius-Duhem*, que é a condição básica de estabilidade não-linear das equações de Navier-Stokes [20]. Tal condição implica que as soluções de elementos finitos via formulação de Galerkin para as equações de Navier-Stokes devem produzir entropia. Entretanto, uma formulação de Galerkin para as equações de Euler conserva entropia. Isto significa que a formulação de Galerkin é inadequada para a solução das equações de Euler, onde se produz entropia como em problemas com choques. Logo, mecanismos adicionais são necessários para garantir apropriadamente a produção de entropia. Tais mecanismos podem ser resumidos em formulações estabilizadas (SUPG ou GLS) com operadores de captura de descontinuidade que atuam nas regiões de gradientes elevados [8, 7, 10]. Se por um lado as variáveis de entropia representam com maior precisão o comportamento da solução nessas regiões, as variáveis conservativas possuem implementações mais simples apresentando soluções bastante precisas [11, 12, 17]. Aqui adota-se o uso de variáveis conservativas.

Capítulo 3

Formulação das Equações de Euler em Variáveis Conservativas

Neste capítulo descreve-se a formulação em variáveis conservativas para as equações de Euler, primeiramente usada por ALIABADI e TEZDUYAR [10]. Na seção que se segue é apresentada a formulação variacional semi-discreta, enfatizando o cálculo dos parâmetros dos operadores SUPG e captura de descontinuidades. Logo após são descritas as diversas componentes da matriz efetiva do algoritmo de avanço no tempo e o vetor de resíduos. Finalmente são apresentados testes para validação da formulação proposta.

3.1 Formulação Variacional Semi-Discreta

A formulação semi-discreta caracteriza-se pela discretização por elementos finitos no espaço seguida pela discretização por diferenças finitas no tempo. Considera-se o domínio espacial Ω dividido em n_{el} elementos, Ω_e, $e = 1, 2, \ldots, nel$, tal que $\Omega \doteq \bigcup_{e=1}^{nel} \Omega_e$ e $\Omega_i \cap \Omega_j = \emptyset$. Sejam o conjunto das funções teste $\mathcal{S}^h$ e o espaço das variações admissíveis $\mathcal{V}^h$, respectivamente definidos por:

$$\mathcal{S}^h = \{\mathbf{U}^h / \mathbf{U}^h \in [\mathbf{H}^{1h}(\Omega)]^4, \mathbf{U}^h/_{\Omega_e} \in [P^1(\Omega^e)]^4, \mathbf{U}^h.\mathbf{e}_k \doteq g_k \text{ em } \Gamma_{g_k}\} \tag{3.1}$$

$$\mathcal{V}^h = \{\mathbf{W}^h / \mathbf{W}^h \in [\mathbf{H}^{1h}(\Omega)]^4, \mathbf{W}^h/_{\Omega_e} \in [P^1(\Omega^e)]^4, \mathbf{W}^h.\mathbf{e}_k \doteq 0 \text{ em } \Gamma_{g_k}\} \tag{3.2}$$

onde $\mathbf{H}^{1h}(\Omega)$ é um espaço de funções de dimensão finita sobre Ω, $P^1(\Omega^e)$ representa o conjunto de polinômios de 1ª ordem em Ω^e e Γ_{g_k} é o contorno de Ω com condições de Dirichlet prescritas.

A formulação semi-discreta para a equação de Euler é dada por: encontrar $\mathbf{U}^h \in \mathcal{S}^h$ tal que $\forall\ \mathbf{W}^h \in \mathcal{V}^h$, tem-se:

$$
\begin{aligned}
\int_{\Omega} \mathbf{W}^h \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega \quad &+ \\
\sum_{e}^{nel} \int_{\Omega^e} \mathbf{T}_k \left(\frac{\partial \mathbf{W}^h}{\partial x_k} \right) \left[\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right] d\Omega \quad &+ \\
\sum_{e}^{nel} \int_{\Omega^e} \delta \frac{\partial \mathbf{W}^h}{\partial x_i} \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega \quad &= \quad \mathbf{0}
\end{aligned}
\tag{3.3}
$$

A primeira integral na equação (3.3) corresponde ao termo de Galerkin. O primeiro somatório é o termo de estabilização SUPG, enquanto o segundo somatório de integrais no elemento é o operador de captura de descontinuidades. Na equação (3.3) os tensores $\mathbf{T}_k$ podem ser dados por:

$$
\mathbf{T}_k = \begin{cases} \boldsymbol{\tau} \mathbf{A}_k & \text{(formulação } A^2 \text{)} \\ \boldsymbol{\tau} \mathbf{A}_k^t & \text{(formulação } A^t A \text{)} \end{cases}
\tag{3.4}
$$

sendo essas definições propostas inicialmente por HUGHES e TEZDUYAR [6]. Segundo os autores a definição $\mathbf{T}_k = \boldsymbol{\tau} \mathbf{A}_k$ produz os melhores resultados, fato também verificado nos testes realizados no presente trabalho, que serão mostrados adiante na seção 3.6.

3.2 Parâmetros de Estabilização

A matriz de estabilização $\boldsymbol{\tau}$ que aparece na expressão (3.4) será definida através de várias matrizes diagonalizadas, simplificando a implementação da formulação. Esta forma de estabilização foi adotada inicialmente por HUGHES e TEZDUYAR [6] e aperfeiçoada em [13]. Em muitos trabalhos são encontradas várias definições não diagonalizadas para a matriz $\boldsymbol{\tau}$, aumentando a complexidade da implementação. São utilizados dois operadores de captura de descontinuidades. Um operador introduzido por ALIABADI e TEZDUYAR [10], e outro desenvolvido por ALMEIDA e GALEÃO [7].

3.2.1 Matriz de Estabilização SUPG

A matriz $\boldsymbol{\tau} = \tau \mathbf{I}$ depende do parâmetro τ, que é definido por:

$$\tau = max[0, \tau_t + \zeta(\tau_a - \tau_\delta)] \tag{3.5}$$

onde:

- τ_t é o parâmetro de estabilização correspondente aos termos dependentes do tempo

$$\tau_t = \frac{2}{3(1 + 2\alpha CFL)}\tau_a \tag{3.6}$$

- τ_a é o parâmetro de estabilização correspondente aos termos advectivos

$$\tau_a = \frac{h}{2(c + |\mathbf{u}\boldsymbol{\beta}|)} \tag{3.7}$$

- τ_δ é o parâmetro de estabilização para descontar os efeitos do termo de captura de descontinuidades

$$\tau_\delta = \frac{\delta}{(c + |\mathbf{u}\boldsymbol{\beta}|)^2} \tag{3.8}$$

- O parâmetro ζ é um coeficiente usado no algoritmo do tempo

$$\zeta = \frac{2\alpha CFL}{1 + 2\alpha CFL} \tag{3.9}$$

- CFL é número de *Courant-Friedrichs-Lewy*

$$CFL = \frac{(c + |\mathbf{u}\boldsymbol{\beta}|)\Delta t}{h} \tag{3.10}$$

- $\boldsymbol{\beta}$ é um vetor unitário arbitrário dado por

$$\boldsymbol{\beta} = \frac{\nabla\|\mathbf{U}\|_*^2}{\|\nabla\|\mathbf{U}\|_*^2\|_2} \tag{3.11}$$

sendo $\|.\|_* = \|.\|_2$ ou $\|.\|_{\mathbf{A}_0^{-1}}$. A velocidade acústica c é dada por $c^2 = \gamma(\gamma - 1)(e - \frac{|\mathbf{u}|^2}{2})$ e h é um parâmetro de malha. No caso de triângulos lineares, que serão usados neste trabalho, adota-se $h = \sqrt{2A}$, sendo A a área do triângulo.

3.2.2 Operadores de Captura de Descontinuidades

Primeiramente descreve-se o operador de captura de descontinuidades desenvolvido inicialmente por SHAKIB [20], em variáveis de entropia. Posteriormente ALIABADI e TEZDUYAR [13], através de transformações matriciais, reescrevem este operador de captura de descontinuidades em variáveis conservativas. O parâmetro deste operador, denominado por δ_{CD}, depende das matrizes $\mathbf{A}_x$, $\mathbf{A}_y$ e $\widetilde{\mathbf{A}}_0^{-1}$ e das derivadas globais e locais de $\mathbf{U}^h$, podendo ser representado por:

$$\delta_{CD} = \left[\frac{\| \mathbf{A}_x \frac{\partial \mathbf{U}^h}{\partial x} + \mathbf{A}_y \frac{\partial \mathbf{U}^h}{\partial y} \|^2_{\widetilde{\mathbf{A}}_0^{-1}}}{\| \nabla_\xi \mathbf{U}^h \|^2_{\widetilde{\mathbf{A}}_0^{-1}}} \right]^{\frac{1}{2}} \tag{3.12}$$

onde

$$\| \nabla_\xi \mathbf{U}^h \|_{\widetilde{\mathbf{A}}_0^{-1}} = \| \frac{\partial x}{\partial \xi} \frac{\partial \mathbf{U}^h}{\partial x} + \frac{\partial y}{\partial \xi} \frac{\partial \mathbf{U}^h}{\partial y} \|_{\widetilde{\mathbf{A}}_0^{-1}} + \| \frac{\partial x}{\partial \eta} \frac{\partial \mathbf{U}^h}{\partial x} + \frac{\partial y}{\partial \eta} \frac{\partial \mathbf{U}^h}{\partial y} \|_{\widetilde{\mathbf{A}}_0^{-1}} \tag{3.13}$$

Os componentes $\frac{\partial x_j}{\partial \xi_l}$ são os termos da matriz de transformação das coordenadas físicas para o sistema de referência das coordenadas locais (ξ_l) nos elementos.

Outro operador de captura de descontinuidades baseado na formulação de *Upwind* aproximado consistentemente (CAU, em inglês) de ALMEIDA e GALEÃO [7] será utilizado no presente trabalho. O parâmetro deste operador, denominado por δ_{CAU}, pode ser representado por:

$$\delta_{CAU} = \begin{cases} \frac{\| \mathcal{L}\mathbf{V}^h \|}{\| \nabla_\xi \mathbf{V}^h \|_{\widetilde{\mathbf{A}}_0^{-1}}} & \text{se} \quad |\nabla \mathbf{V}^h|_{\mathbf{A}_0} \neq 0 \\ 0 & \text{se} \quad |\nabla \mathbf{V}^h|_{\mathbf{A}_0} = 0 \end{cases} \tag{3.14}$$

onde o resíduo no interior do elemento é dado por,

$$\mathcal{L}\mathbf{V}^h = \mathbf{A}_0 \mathbf{V}^h_{,t} + \tilde{\mathbf{A}}_i \frac{\partial \mathbf{V}^h}{\partial x_i} \tag{3.15}$$

$$\| \nabla_\xi \mathbf{V}^h \|_{\widetilde{\mathbf{A}}_0^{-1}} = \| \frac{\partial x}{\partial \xi} \frac{\partial \mathbf{V}^h}{\partial x} + \frac{\partial y}{\partial \xi} \frac{\partial \mathbf{V}^h}{\partial y} \|_{\widetilde{\mathbf{A}}_0^{-1}} + \| \frac{\partial x}{\partial \eta} \frac{\partial \mathbf{V}^h}{\partial x} + \frac{\partial y}{\partial \eta} \frac{\partial \mathbf{V}^h}{\partial y} \|_{\widetilde{\mathbf{A}}_0^{-1}} \tag{3.16}$$

Lembrando o mapeamento das variáveis de entropia para as variáveis de conservação definido em (2.51) e (2.52), tem-se $\mathbf{V}^h_i = \mathbf{V}_{\mathbf{U}^h} \mathbf{U}^h_{,i} = \mathbf{A}_0^{-1} \mathbf{U}^h_{,i}$. Através desta definição é possível obter as igualdades:

$$
\begin{aligned}
|\mathcal{L}\mathbf{V}^h|_{\mathbf{A}_0^{-1}} &= |\mathcal{L}\mathbf{U}^h|_{\mathbf{A}_0^{-1}} \\
|\nabla_\xi \mathbf{V}^h|_{\mathbf{A}_0} &= |\nabla_\xi \mathbf{U}^h|_{\mathbf{A}_0^{-1}} \\
|\nabla \mathbf{V}^h|_{\mathbf{A}_0} &= |\nabla \mathbf{U}^h|_{\mathbf{A}_0^{-1}}
\end{aligned}
\tag{3.17}
$$

Assim, o parâmetro do operador δ_{CAU} pode ser reescrito em variáveis conservativas por:

$$
\delta_{CAU} = \begin{cases} \frac{\| \mathcal{L}\mathbf{U}^h \|}{\| \nabla_\xi \mathbf{U}^h \|_{\tilde{\mathbf{A}}_0^{-1}}} & \text{se} \quad |\nabla \mathbf{U}^h|_{\mathbf{A}_0} \neq 0 \\ 0 & \text{se} \quad |\nabla \mathbf{U}^h|_{\mathbf{A}_0} = 0 \end{cases}
\tag{3.18}
$$

onde

$$
\mathcal{L}\mathbf{U}^h = \mathbf{U}^h_{,t} + \mathbf{A}_i \frac{\partial \mathbf{U}^h}{\partial x_i}
\tag{3.19}
$$

3.3 Matrizes dos Elementos

Sejam as aproximações usuais de elementos finitos para um elemento triangular linear:

$$
\begin{aligned}
\mathbf{U}^h &= \mathbf{N}\mathbf{v} \\
\mathbf{W}^h &= \mathbf{N}\mathbf{c} \\
\mathbf{U}^h_{,t} &= \mathbf{N}\dot{\mathbf{v}} = \mathbf{N}\mathbf{a}
\end{aligned}
\tag{3.20}
$$

onde $\mathbf{v}$ é o vetor (dependente somente do tempo) de valores nodais de $\mathbf{U}$, $\mathbf{c}$ é um vetor de contantes arbitrárias, $\mathbf{a}$ é a derivada no tempo de $\mathbf{v}$ ($\mathbf{a} = \frac{d\mathbf{v}}{dt}$) e $\mathbf{N}$ é a matriz de interpolação (dependente somente de $\mathbf{x}$) podendo ser representada por:

$$
\mathbf{N} = \begin{bmatrix} N_1 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & N_3 & 0 & 0 & 0 \\ 0 & N_1 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & N_3 & 0 & 0 \\ 0 & 0 & N_1 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & N_3 & 0 \\ 0 & 0 & 0 & N_1 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & N_3 \end{bmatrix}_{4\times 12}
\tag{3.21}
$$

ou ainda de forma mais compacta:

$$
\mathbf{N} = \begin{bmatrix} N_1\mathbf{I} & N_2\mathbf{I} & N_3\mathbf{I} \end{bmatrix}
\tag{3.22}
$$

onde N_1, N_2 e N_3 são as funções de interpolação no interior do elemento [5] e $\mathbf{I}$ é a matriz identidade de ordem 4. As funções de interpolação para elementos triangulares lineares (Figura 3.1), são dadas por:

$$N_1 = \xi \quad N_2 = \eta \quad N_3 = 1 - \xi - \eta \tag{3.23}$$

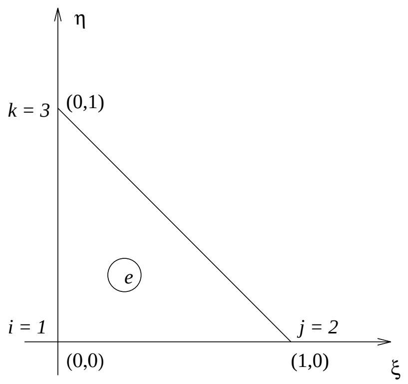


Figura 3.1: Elemento Triangular Linear

A matriz Jacobiana que define a transformação das coordenadas globais para as coordenadas locais é dada por:

$$\mathbf{J} = \frac{\partial x_i}{\partial \xi_j} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_{13} & -y_{31} \\ -x_{32} & y_{23} \end{bmatrix} \tag{3.24}$$

sendo x_i e y_j coordenadas globais do elemento e, e $x_{ij} = x_i - x_j$, $i, j = 1, 2, 3$. A transformação inversa é dada por:

$$\mathbf{J}^{-1} = \begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \xi}{\partial y} \\ \dfrac{\partial \eta}{\partial x} & \dfrac{\partial \eta}{\partial y} \end{bmatrix} = \frac{\partial \xi_i}{\partial x_j} = \frac{1}{2A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \end{bmatrix} \tag{3.25}$$

onde A^e é a área do elemento e. É comum definir o operador gradiente discreto das funções de interpolação por:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{N}}{\partial x} \\ \dfrac{\partial \mathbf{N}}{\partial y} \end{bmatrix} \tag{3.26}$$

Pode-se, então, reescrever $\mathbf{B}$ por:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} = \frac{1}{2A^e} \begin{bmatrix} y_{23} & 0 & 0 & 0 & y_{31} & 0 & 0 & 0 & y_{12} & 0 & 0 & 0 \\ 0 & y_{23} & 0 & 0 & 0 & y_{31} & 0 & 0 & 0 & y_{12} & 0 & 0 \\ 0 & 0 & y_{23} & 0 & 0 & 0 & y_{31} & 0 & 0 & 0 & y_{12} & 0 \\ 0 & 0 & 0 & y_{23} & 0 & 0 & 0 & y_{31} & 0 & 0 & 0 & y_{12} \\ x_{32} & 0 & 0 & 0 & x_{13} & 0 & 0 & 0 & x_{21} & 0 & 0 & 0 \\ 0 & x_{32} & 0 & 0 & 0 & x_{13} & 0 & 0 & 0 & x_{21} & 0 & 0 \\ 0 & 0 & x_{32} & 0 & 0 & 0 & x_{13} & 0 & 0 & 0 & x_{21} & 0 \\ 0 & 0 & 0 & x_{32} & 0 & 0 & 0 & x_{13} & 0 & 0 & 0 & x_{21} \end{bmatrix} \tag{3.27}$$

Ou ainda as matrizes $\mathbf{B}_x$ e $\mathbf{B}_y$ podem ser representadas por:

$$\mathbf{B}_x = \frac{1}{2A^e} \begin{bmatrix} y_{23}\mathbf{I} & y_{31}\mathbf{I} & y_{12} \end{bmatrix} \tag{3.28}$$

$$\mathbf{B}_y = \frac{1}{2A^e} \begin{bmatrix} x_{32}\mathbf{I} & x_{13}\mathbf{I} & x_{21}\mathbf{I} \end{bmatrix} \tag{3.29}$$

Portanto as derivadas de $\mathbf{U}^h$ com relação a x e a y são:

$$\begin{aligned} \mathbf{U}^h_{,x} &= \mathbf{B}_x\mathbf{v} \\ \mathbf{U}^h_{,y} &= \mathbf{B}_y\mathbf{v} \end{aligned} \tag{3.30}$$

As aproximações definidas em (3.20) levam a formulação (3.3) a um conjunto acoplado de equações diferenciais ordinárias não lineares que pode ser escrito como:

$$\mathbf{Ma} + \mathbf{Cv} = \mathbf{0} \tag{3.31}$$

onde $\mathbf{M}$ é a matriz de "massa" generalizada e $\mathbf{C}$ é um operador não-linear de $\mathbf{v}$ representando os termos das equações de estado. É conveniente representar as parcelas de (3.31) em termos das contribuições de cada elemento. Assim, agrupando-se os termos dependentes de $\mathbf{a}$ em (3.3), obtem-se:

$$\begin{aligned} \mathbf{M} &= \mathop{\mathbf{A}}_{e=1}^{nel}(\mathbf{m}^e) \\ \mathbf{m}^e &= [\mathbf{m}^e_{ab}], \quad a,b = 1,2,3 \\ \mathbf{m}^e_{ab} &= \int_{\Omega^e} \left(N_a\mathbf{I} + N_{a,x}\boldsymbol{\tau}\mathbf{T}_x + N_{a,y}\boldsymbol{\tau}\mathbf{T}_y\right) N_b d\Omega^e \end{aligned} \tag{3.32}$$

Desta forma, a matriz do elemento e é dada por:

$$\mathbf{m}^e = \int_{\Omega e} \mathbf{N}^T \mathbf{N} d\Omega^e + \tau \int_{\Omega e} (\mathbf{B}_x^T \mathbf{T}_x \mathbf{N} + \mathbf{B}_y^T \mathbf{T}_y \mathbf{N}) d\Omega^e \tag{3.33}$$

sendo a primeira integral a matriz de "massa" e a segunda os termos da correção SUPG. De forma similar, agrupando os termos dependentes de $\mathbf{v}$, em (3.3), obtem-se:

$$\begin{aligned}
\mathbf{C} &= \mathop{\mathsf{A}}_{e=1}^{nel} (\mathbf{c}^e) \\
\mathbf{c}^e &= [\mathbf{c}^e_{ab}], \quad a, b = 1, 2, 3 \\
\mathbf{c}^e_{ab} &= \int_{\Omega^e} [(N_a \mathbf{A}_x N_{b,x} + N_a \mathbf{A}_y N_{b,y}) \\
&\quad + \tau \left(N_{a,x} \mathbf{T}_x \mathbf{A}_x N_{b,x} + N_{a,x} \mathbf{T}_x \mathbf{A}_y N_{b,y}\right. \\
&\quad + \quad \left. N_{a,y} \mathbf{T}_y \mathbf{A}_x N_{b,x} + N_{a,y} \mathbf{T}_y \mathbf{A}_y N_{b,y}\right) \\
&\quad + \delta \left(N_{a,x} N_{b,x} + N_{a,x} N_{b,y} + N_{a,y} N_{b,x} + N_{a,y} N_{b,y}\right)] \, d\Omega^e
\end{aligned} \tag{3.34}$$

ou ainda, a matriz do elemento e pode ser representada por:

$$\begin{aligned}
\mathbf{c}^e &= \int_{\Omega e} \left(\mathbf{N}^T \mathbf{A}_x \mathbf{B}_x + \mathbf{N}^T \mathbf{A}_y \mathbf{B}_y\right) d\Omega^e \\
&\quad + \tau \int_{\Omega e} \mathbf{B}^T \begin{bmatrix} \mathbf{T}_x \mathbf{A}_x & \mathbf{T}_x \mathbf{A}_y \\ \mathbf{T}_y \mathbf{A}_x & \mathbf{T}_y \mathbf{A}_y \end{bmatrix} \mathbf{B} d\Omega^e \\
&\quad + \delta \int_{\Omega e} \mathbf{B}^T \mathbf{B} d\Omega^e
\end{aligned} \tag{3.35}$$

sendo a primeira integral a matriz de convecção, a segunda a correção SUPG dos termos convectivos e a terceira o operador de captura de choque. As matrizes $\mathbf{A}_x$, $\mathbf{A}_y$ e $\mathbf{A}_0^{-1}$, definidas respectivamente em (2.47),(2.49) e (2.55), deveriam ser avaliadas nos pontos de integração do elemento. Entretanto SOULAÏMANI e FORTIN observaram [15] que os resultados não são significativamente alterados se as matrizes forem avaliadas somente no baricentro do elemento. Assim, com o objetivo de simplificar a formulação as matrizes $\mathbf{A}_x$, $\mathbf{A}_y$ e $\mathbf{A}_0^{-1}$ serão avaliadas somente no baricentro de cada elemento, o que significa que está sendo considerado apenas um ponto de integração. Para tal, observa-se que:

$$\mathbf{v}^T = \{\underbrace{v_1, v_2, v_3, v_4}_{\text{nó 1}}, \underbrace{v_5, v_6, v_7, v_8}_{\text{nó 2}}, \underbrace{v_9, v_{10}, v_{11}, v_{12}}_{\text{nó 3}}\} \tag{3.36}$$

Sendo $\mathbf{v}$ no baricentro do triângulo definido por:

$$\mathbf{v}_b = \left\{ \begin{array}{c} U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \right\} = \frac{1}{3} \left\{ \begin{array}{c} v_1 + v_5 + v_9 \\ v_2 + v_6 + v_{10} \\ v_3 + v_7 + v_{11} \\ v_4 + v_8 + v_{12} \end{array} \right\} \tag{3.37}$$

De forma similar pode-se definir o valor nodal da derivada temporal $\mathbf{a}$ por

$$\mathbf{a}^T = \{\underbrace{a_1, a_2, a_3, a_4}_{\text{nó } 1}, \underbrace{a_5, a_6, a_7, a_8}_{\text{nó } 2}, \underbrace{a_9, a_{10}, a_{11}, a_{12}}_{\text{nó } 3}\} \tag{3.38}$$

e seu valor no baricentro é dado por:

$$\mathbf{a}_b = \frac{1}{3} \left\{ \begin{array}{c} a_1 + a_5 + a_9 \\ a_2 + a_6 + a_{10} \\ a_3 + a_7 + a_{11} \\ a_4 + a_8 + a_{12} \end{array} \right\} \tag{3.39}$$

Portanto, $\mathbf{A}_x = \mathbf{A}_x(\mathbf{v}_b)$, $\mathbf{A}_y = \mathbf{A}_y(\mathbf{v}_b)$ e $\mathbf{A}_0^{-1} = \mathbf{A}_0^{-1}(\mathbf{v}_b)$

3.3.1 Matriz de Massa

Na expressão (3.33) a contribuição referente a matriz de massa é dada por $\int_{\Omega e} \mathbf{N}^T \mathbf{N} d\Omega^e$. Considerando a definição (3.22) para a função de interpolação $\mathbf{N}$, tem-se:

$$\mathbf{m}_g^e = \int_{\Omega e} \left[\begin{array}{c} N_1\mathbf{I} \\ N_2\mathbf{I} \\ N_3\mathbf{I} \end{array} \right] \left[\begin{array}{ccc} N_1\mathbf{I} & N_2\mathbf{I} & N_3\mathbf{I} \end{array} \right] d\Omega^e \tag{3.40}$$

Resolvendo cada integral trivial envolvida, obtem-se:

$$\mathbf{m}_g^e = \frac{A^e}{12} \left[\begin{array}{cccccccccccc} 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \end{array} \right] \tag{3.41}$$

$\mathbf{m}_g^e$ pode ainda ser reescrita de forma mais compacta por:

$$\mathbf{m}_g^e = \frac{A^e}{12} \left[\begin{array}{ccc} 2\mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{I} & 2\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & 2\mathbf{I} \end{array} \right] \tag{3.42}$$

3.3.2 Correção SUPG da matriz de massa

Reescrevendo o termo referente à correção $SUPG$ da matriz de massa contido na expressão (3.33), tem-se:

$$\mathbf{m}_{pg}^{e} = \tau \int_{\Omega e} (\mathbf{B}_x^T \mathbf{T}_x \mathbf{N} + \mathbf{B}_y^T \mathbf{T}_y \mathbf{N}) d\Omega^e \tag{3.43}$$

Avaliando-se também $\mathbf{T}_x$ e $\mathbf{T}_y$ em função das grandezas no baricentro do elemento tem-se,

$$\mathbf{m}_{pg}^{e} = \tau \mathbf{B}_x^T \mathbf{T}_x (\int_{\Omega e} \mathbf{N} d\Omega^e) + \tau \mathbf{B}_y^T \mathbf{T}_y (\int_{\Omega e} \mathbf{N} d\Omega^e) \tag{3.44}$$

A integral das funções de interpolação N_i, definidas em (3.23), pode facilmente ser calculada. Tem-se que $\int_{\Omega^e} N_i d\Omega^e = \frac{2A^e}{6}$. Assim, obtem-se:

$$\int_{\Omega^e} \mathbf{N} d\Omega^e = \frac{2A^e}{6} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.45}$$

ou de forma mais compacta pode-se representar (3.45) por:

$$\int_{\Omega^e} \mathbf{N} d\Omega^e = \frac{2A^e}{6} \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} \tag{3.46}$$

Considerando os resultados contidos em (3.46) a matriz $\mathbf{m}_{pg}^{e}$ pode ser representada por:

$$\mathbf{m}_{pg}^{e} = \frac{\tau}{6} \begin{bmatrix} y_{23}\mathbf{I} \\ y_{31}\mathbf{I} \\ y_{12}\mathbf{I} \end{bmatrix} \mathbf{T}_x \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} + \frac{\tau}{6} \begin{bmatrix} x_{32}\mathbf{I} \\ x_{13}\mathbf{I} \\ x_{21}\mathbf{I} \end{bmatrix} \mathbf{T}_y \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} \tag{3.47}$$

Portanto, tem-se:

$$\mathbf{m}_{pg}^{e} = \frac{\tau}{6} \begin{bmatrix} y_{23}\mathbf{T}_x + x_{32}\mathbf{T}_y & y_{23}\mathbf{T}_x + x_{32}\mathbf{T}_y & y_{23}\mathbf{T}_x + x_{32}\mathbf{T}_y \\ y_{31}\mathbf{T}_x + x_{13}\mathbf{T}_y & y_{31}\mathbf{T}_x + x_{13}\mathbf{T}_y & y_{31}\mathbf{T}_x + x_{13}\mathbf{T}_y \\ y_{12}\mathbf{T}_x + x_{21}\mathbf{T}_y & y_{12}\mathbf{T}_x + x_{21}\mathbf{T}_y & y_{12}\mathbf{T}_x + x_{21}\mathbf{T}_y \end{bmatrix} \tag{3.48}$$

A matriz descrita em (3.48) pode ser representada a partir de 3 submatrizes de ordem 4, ou seja,

$$\mathbf{m}_{pg}^{e} = \frac{\tau}{6}\begin{bmatrix} \mathbf{m}_{pg}^{1} & \mathbf{m}_{pg}^{1} & \mathbf{m}_{pg}^{1} \\ \mathbf{m}_{pg}^{2} & \mathbf{m}_{pg}^{2} & \mathbf{m}_{pg}^{2} \\ \mathbf{m}_{pg}^{3} & \mathbf{m}_{pg}^{3} & \mathbf{m}_{pg}^{3} \end{bmatrix} \tag{3.49}$$

onde

$$\begin{aligned} \mathbf{m}_{pg}^{1} &= y_{23}\mathbf{T}_x + x_{32}\mathbf{T}_y \\ \mathbf{m}_{pg}^{2} &= y_{31}\mathbf{T}_x + x_{13}\mathbf{T}_y \\ \mathbf{m}_{pg}^{3} &= y_{12}\mathbf{T}_x + x_{21}\mathbf{T}_y \end{aligned} \tag{3.50}$$

3.3.3 Matriz de Convecção

A matriz que contém os termos convectivos devido a formulação de Galerkin, presente em (3.35), pode ser reescrita por:

$$\mathbf{c}_g^e = \int_{\Omega e} (\mathbf{N}^T\mathbf{A}_x\mathbf{B}_x + \mathbf{N}^T\mathbf{A}_y\mathbf{B}_y)d\Omega^e \tag{3.51}$$

Portanto, a matriz de convecção $\mathbf{c}_g^e$ pode então ser representada por:

$$\mathbf{c}_g^e = \left(\int_{\Omega e} \mathbf{N}^T d\Omega^e\right)\mathbf{A}_x\mathbf{B}_x + \left(\int_{\Omega e} \mathbf{N}^T d\Omega^e\right)\mathbf{A}_y\mathbf{B}_y \tag{3.52}$$

De forma similar à expressão (3.46) pode-se dizer que:

$$\int_{\Omega^e} \mathbf{N}^T d\Omega^e = \frac{2A^e}{6}\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{bmatrix} \tag{3.53}$$

A matriz de convecção pode ser representada por:

$$\mathbf{c}_g^e = \frac{2A^e}{6}\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{bmatrix}\mathbf{A}_x\frac{1}{2A^e}\begin{bmatrix} y_{23}\mathbf{I} & y_{31}\mathbf{I} & y_{12}\mathbf{I} \end{bmatrix} + \frac{2A^e}{6}\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{bmatrix}\mathbf{A}_y\frac{1}{2A^e}\begin{bmatrix} x_{32}\mathbf{I} & x_{13}\mathbf{I} & x_{21}\mathbf{I} \end{bmatrix} \tag{3.54}$$

Ou, em forma matricial, tem-se:

$$\mathbf{c}_g^e = \frac{1}{6}\begin{bmatrix} y_{23}\mathbf{A}_x + x_{32}\mathbf{A}_y & y_{31}\mathbf{A}_x + x_{13}\mathbf{A}_y & y_{12}\mathbf{A}_x + x_{21}\mathbf{A}_y \\ y_{23}\mathbf{A}_x + x_{32}\mathbf{A}_y & y_{31}\mathbf{A}_x + x_{13}\mathbf{A}_y & y_{12}\mathbf{A}_x + x_{21}\mathbf{A}_y \\ y_{23}\mathbf{A}_x + x_{32}\mathbf{A}_y & y_{31}\mathbf{A}_x + x_{13}\mathbf{A}_y & y_{12}\mathbf{A}_x + x_{21}\mathbf{A}_y \end{bmatrix} \tag{3.55}$$

A partir da expressão (3.55) observa-se que a matriz de convecção $\mathbf{c}_g^e$ pode ser representada por apenas três submatrizes distintas de ordem 4, ou seja:

$$\mathbf{c}_g^e = \frac{1}{6}\begin{bmatrix} \mathbf{c}_g^1 & \mathbf{c}_g^2 & \mathbf{c}_g^3 \\ \mathbf{c}_g^1 & \mathbf{c}_g^2 & \mathbf{c}_g^3 \\ \mathbf{c}_g^1 & \mathbf{c}_g^2 & \mathbf{c}_g^3 \end{bmatrix} \tag{3.56}$$

onde

$$\begin{aligned} \mathbf{c}_g^1 &= y_{23}\mathbf{A}_x + x_{32}\mathbf{A}_y \\ \mathbf{c}_g^2 &= y_{31}\mathbf{A}_x + x_{13}\mathbf{A}_y \\ \mathbf{c}_g^3 &= y_{12}\mathbf{A}_x + x_{21}\mathbf{A}_y \end{aligned} \tag{3.57}$$

3.3.4 Correção SUPG da matriz de convecção

O termo da correção SUPG da matriz de convecção, definido em (3.35), pode ser reescrito por:

$$\mathbf{c}_{pg}^e = \tau \int_{\Omega^e} \mathbf{B}^T \left[\begin{array}{cc} \mathbf{T}_x\mathbf{A}_x & \mathbf{T}_x\mathbf{A}_y \\ \mathbf{T}_y\mathbf{A}_x & \mathbf{T}_y\mathbf{A}_y \end{array} \right] \mathbf{B} d\Omega^e \tag{3.58}$$

Para facilitar notações, introduz-se as seguintes denominações:

$$\left[\begin{array}{cc} \mathbf{T}_x\mathbf{A}_x & \mathbf{T}_x\mathbf{A}_y \\ \mathbf{T}_y\mathbf{A}_x & \mathbf{T}_y\mathbf{A}_y \end{array} \right] = \mathbf{A} = \left[\begin{array}{cc} \mathbf{A}_{xx} & \mathbf{A}_{xy} \\ \mathbf{A}_{yx} & \mathbf{A}_{yy} \end{array} \right] \tag{3.59}$$

Os coeficientes de $\mathbf{A}$ são obtidos através de operações triviais de matrizes. Além disso, observa-se que os termos do integrando de (3.58) são constantes no elemento e, uma vez que as matrizes $\mathbf{T}_i$ e $\mathbf{A}_i$, $i = x, y$, são avaliadas no baricentro. Portanto, pode-se dizer que:

$$\mathbf{c}_{pg}^e = \frac{\tau}{4A^e} \left[\begin{array}{cc} y_{23}\mathbf{I} & x_{32}\mathbf{I} \\ y_{31}\mathbf{I} & x_{13}\mathbf{I} \\ y_{12}\mathbf{I} & x_{21}\mathbf{I} \end{array} \right] \mathbf{A} \left[\begin{array}{ccc} y_{23}\mathbf{I} & y_{31}\mathbf{I} & y_{12}\mathbf{I} \\ x_{32}\mathbf{I} & x_{13}\mathbf{I} & x_{21}\mathbf{I} \end{array} \right] \tag{3.60}$$

A matriz $\mathbf{c}_{pg}^e$, definida em (3.60), pode ser representa por:

$$\mathbf{c}_{pg}^e = \frac{\tau}{4A^e} \left[\begin{array}{ccc} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} \\ \mathbf{B}_{21} & \mathbf{B}_{22} & \mathbf{B}_{23} \\ \mathbf{B}_{31} & \mathbf{B}_{32} & \mathbf{B}_{33} \end{array} \right] \tag{3.61}$$

Os coeficientes da matriz (3.61) são submatrizes de ordem 4, cujos coeficientes $\mathbf{B}_{ij}$ são:

$$\begin{aligned}
\mathbf{B}_{12} &= y_{23}(y_{31}\mathbf{A}_{xx} + x_{13}\mathbf{A}_{xy}) + x_{32}(y_{31}\mathbf{A}_{yx} + x_{13}\mathbf{A}_{yy}) \\
\mathbf{B}_{13} &= y_{23}(y_{12}\mathbf{A}_{xx} + x_{21}\mathbf{A}_{xy}) + x_{32}(y_{12}\mathbf{A}_{yx} + x_{21}\mathbf{A}_{yy}) \\
\mathbf{B}_{21} &= y_{31}(y_{23}\mathbf{A}_{xx} + x_{32}\mathbf{A}_{xy}) + x_{13}(y_{23}\mathbf{A}_{yx} + x_{32}\mathbf{A}_{yy}) \\
\mathbf{B}_{23} &= y_{31}(y_{12}\mathbf{A}_{xx} + x_{21}\mathbf{A}_{xy}) + x_{13}(y_{12}\mathbf{A}_{yx} + x_{21}\mathbf{A}_{yy}) \\
\mathbf{B}_{31} &= y_{12}(y_{23}\mathbf{A}_{xx} + x_{32}\mathbf{A}_{xy}) + x_{21}(y_{23}\mathbf{A}_{yx} + x_{32}\mathbf{A}_{yy}) \\
\mathbf{B}_{32} &= y_{12}(y_{31}\mathbf{A}_{xx} + x_{13}\mathbf{A}_{xy}) + x_{21}(y_{31}\mathbf{A}_{yx} + x_{13}\mathbf{A}_{yy}) \\
\mathbf{B}_{11} &= -(\mathbf{B}_{12} + \mathbf{B}_{13}) \\
\mathbf{B}_{22} &= -(\mathbf{B}_{21} + \mathbf{B}_{23}) \\
\mathbf{B}_{33} &= -(\mathbf{B}_{31} + \mathbf{B}_{32})
\end{aligned} \tag{3.62}$$

3.3.5 Matriz de Correção do Operador de Descontinuidade

A matriz que contém os termos do operador de captura de descontinuidades, presente em (3.35), pode ser reescrita por:

$$\mathbf{c}^e_{OP} = \int_{\Omega e} \delta \mathbf{B}^T \mathbf{B} d\Omega^e \tag{3.63}$$

sendo δ constante no interior do elemento, a matriz (3.63) pode ser representada por:

$$\mathbf{c}^e_{OP} = \frac{\delta}{4A^e} \begin{bmatrix} y_{23}\mathbf{I} & x_{32}\mathbf{I} \\ y_{31}\mathbf{I} & x_{13}\mathbf{I} \\ y_{12}\mathbf{I} & x_{21}\mathbf{I} \end{bmatrix} \begin{bmatrix} y_{23}\mathbf{I} & y_{31}\mathbf{I} & y_{12}\mathbf{I} \\ x_{32}\mathbf{I} & x_{13}\mathbf{I} & x_{21}\mathbf{I} \end{bmatrix} \tag{3.64}$$

A matriz (3.64) é simétrica e apresenta deficiência de posto. Portanto, tem-se:

$$\mathbf{c}^e_{OP} = \frac{\delta}{4A^e} \begin{bmatrix} -(\bar{\mathbf{B}}_{12} + \bar{\mathbf{B}}_{13}) & \bar{\mathbf{B}}_{12} & \bar{\mathbf{B}}_{13} \\ & -(\bar{\mathbf{B}}_{12} + \bar{\mathbf{B}}_{23}) & \bar{\mathbf{B}}_{23} \\ & & -(\bar{\mathbf{B}}_{13} + \bar{\mathbf{B}}_{23}) \end{bmatrix} \tag{3.65}$$

sendo as três submatrizes distintas dadas por:

$$\begin{aligned}
\bar{\mathbf{B}}_{12} &= (y_{23}y_{31} + x_{32}x_{13})\mathbf{I} \\
\bar{\mathbf{B}}_{13} &= (y_{23}y_{12} + x_{32}x_{21})\mathbf{I} \\
\bar{\mathbf{B}}_{23} &= (y_{31}y_{12} + x_{13}x_{21})\mathbf{I}
\end{aligned} \tag{3.66}$$

3.4 Resíduo

O resíduo da equação diferencial ordinária (3.31) é definido por:

$$\mathbf{R} = -\mathbf{Ma} - \mathbf{Cv} \tag{3.67}$$

Da mesma forma como foram tratadas as matrizes $\mathbf{M}$ e $\mathbf{C}$, na seção anterior, o resíduo pode ser obtido através das respectivas contribuições em cada elemento:

$$\begin{aligned} \mathbf{R} &= \mathop{\mathsf{A}}_{e=1}^{nel}(\mathbf{r}^e) \\ \mathbf{r}^e &= -\mathbf{m}^e\mathbf{a}^e - \mathbf{c}^e\mathbf{v}^e \end{aligned} \tag{3.68}$$

sendo $\mathbf{m}^e$ e $\mathbf{c}^e$ dadas respectivamente por (3.33) e (3.35). Os vetores $\mathbf{a}^e$ e $\mathbf{v}^e$ podem ser representados por:

$$\mathbf{a}^e = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix} \quad \mathbf{v}^e = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} \tag{3.69}$$

onde os índices 1, 2 e 3 representam respectivamente as contribuições dos vetores $\mathbf{a}^e$ e $\mathbf{v}^e$ nos nós 1, 2 e 3. O resíduo $\mathbf{r}^e$ ainda pode ser representado por:

$$\mathbf{r}^e = -(\mathbf{r}^e_{mg} + \mathbf{r}^e_{mpg} + \mathbf{r}^e_{cg} + \mathbf{r}^e_{cpg} + \mathbf{r}^e_{op}) \tag{3.70}$$

Para calcular as componentes do resíduo em cada elemento como descrito em (3.70) é conveniente escrever as equações integrais que definem as matrizes de massa e convecção no elemento e aproveitar algumas simplificações que ocorrem nos produtos $\mathbf{m}^e\mathbf{a}^e$ e $\mathbf{c}^e\mathbf{v}^e$. As componentes do resíduo relativas a matriz de massa são dadas por:

$$\mathbf{r}^e_{mg} = (\int_{\Omega e} \mathbf{N}^t\mathbf{N}d\Omega^e)\mathbf{a}^e = \frac{A^e}{12}\begin{bmatrix} 2\mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{I} & 2\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & 2\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix} = \frac{A^e}{12}\begin{bmatrix} 3\mathbf{a}_b + \mathbf{a}_1 \\ 3\mathbf{a}_b + \mathbf{a}_2 \\ 3\mathbf{a}_b + \mathbf{a}_3 \end{bmatrix} \tag{3.71}$$

onde o vetor $\mathbf{a}_b$ representa a derivada temporal avaliada no baricentro do elemento, definida em (3.39). As componentes do resíduo relativas a $\mathbf{m}_{pg}$ podem ser representadas por:

$$\mathbf{r}_{mg}^e = (\tau \int_{\Omega e} (\mathbf{B}_x^T \mathbf{T}_x \mathbf{N} + \mathbf{B}_y^T \mathbf{T}_y \mathbf{N}) d\Omega^e) \mathbf{a}^e \tag{3.72}$$

De acordo com (3.46) pode-se dizer que:

$$(\int_{\Omega^e} \mathbf{N} d\Omega^e) \mathbf{a}^e = \frac{A^e}{3} \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix} = A^e \mathbf{a}_b \tag{3.73}$$

Assim a contribuição da matriz $\mathbf{m}_{pg}^e$ no vetor de resíduos é dada por:

$$\mathbf{r}_{mpg}^e = \frac{\tau}{2} \begin{bmatrix} y_{23}\mathbf{I} \\ y_{31}\mathbf{I} \\ y_{12}\mathbf{I} \end{bmatrix} \mathbf{T}_x \mathbf{a}_b + \frac{\tau}{2} \begin{bmatrix} x_{32}\mathbf{I} \\ x_{13}\mathbf{I} \\ x_{21}\mathbf{I} \end{bmatrix} \mathbf{T}_y \mathbf{a}_b = \frac{\tau}{2} \begin{bmatrix} y_{23}\mathbf{T}_x \mathbf{a}_b + x_{32}\mathbf{T}_y \mathbf{a}_b \\ y_{31}\mathbf{T}_x \mathbf{a}_b + x_{13}\mathbf{T}_y \mathbf{a}_b \\ y_{12}\mathbf{T}_x \mathbf{a}_b + x_{21}\mathbf{T}_y \mathbf{a}_b \end{bmatrix} \tag{3.74}$$

ou ainda,

$$\mathbf{r}_{mpg}^e = \begin{bmatrix} \mathbf{r}_{mpg}^1 \\ \mathbf{r}_{mpg}^2 \\ \mathbf{r}_{mpg}^3 \end{bmatrix} \tag{3.75}$$

onde:

$$\begin{aligned} \mathbf{r}_{mpg}^1 &= \frac{\tau}{2}(y_{23}\mathbf{T}_x \mathbf{a}_b + x_{32}\mathbf{T}_y \mathbf{a}_b) \\ \mathbf{r}_{mpg}^2 &= \frac{\tau}{2}(y_{31}\mathbf{T}_x \mathbf{a}_b + x_{13}\mathbf{T}_y \mathbf{a}_b) \\ \mathbf{r}_{mpg}^3 &= -(\mathbf{r}_{mpg}^1 + \mathbf{r}_{mpg}^2) \end{aligned} \tag{3.76}$$

As componentes do resíduo relativas à matriz de convecção podem ser representadas por:

$$\mathbf{r}_{cg}^e = \left(\int_{\Omega e} \mathbf{N}^T d\Omega^e\right) \mathbf{A}_x \mathbf{B}_x \mathbf{v}^e + \left(\int_{\Omega e} \mathbf{N}^T d\Omega^e\right) \mathbf{A}_y \mathbf{B}_y \mathbf{v}^e \tag{3.77}$$

Levando-se em consideração as expressões (3.30) e (3.53) tem-se que:

$$\mathbf{r}_{cg}^e = \frac{2A^e}{6} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{bmatrix} (\mathbf{A}_x \mathbf{U}_{,x}^h + \mathbf{A}_y \mathbf{U}_{,y}^h) \tag{3.78}$$

ou seja,

$$\mathbf{r}_{cg}^{e} = \begin{bmatrix} \mathbf{r}_{cg} \\ \mathbf{r}_{cg} \\ \mathbf{r}_{cg} \end{bmatrix} \tag{3.79}$$

onde:

$$\mathbf{r}_{cg} = \frac{A^e}{3}(\mathbf{A}_x\mathbf{U}_{,x}^h + \mathbf{A}_y\mathbf{U}_{,y}^h) \tag{3.80}$$

As contribuições dos termos de Petrov-Galerkin na matriz de convecção para o resíduo podem ser obtidas a partir de (3.58) e (3.59), podendo ser escritas por:

$$\mathbf{r}_{cpg}^{e} = \tau\left(\int_{\Omega^e} \begin{bmatrix}\mathbf{B}_x^t & \mathbf{B}_y^t\end{bmatrix} \begin{bmatrix} \mathbf{A}_{xx} & \mathbf{A}_{xy} \\ \mathbf{A}_{yx} & \mathbf{A}_y y \end{bmatrix} \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} d\Omega^e\right)\mathbf{v}^e \tag{3.81}$$

Considerando as definições (3.29) e (3.30) obtem-se:

$$\mathbf{r}_{cpg}^{e} = \tau\left(\int_{\Omega^e} d\Omega^e\right) \begin{bmatrix}\mathbf{B}_x^t & \mathbf{B}_y^t\end{bmatrix} \begin{bmatrix} \mathbf{A}_{xx} & \mathbf{A}_{xy} \\ \mathbf{A}_{yx} & \mathbf{A}_y y \end{bmatrix} \begin{bmatrix} \mathbf{U}_{,x}^h \\ \mathbf{U}_{,y}^h \end{bmatrix} \tag{3.82}$$

ou

$$\mathbf{r}_{cpg}^{e} = \frac{\tau}{2}\begin{bmatrix} y_{23}\mathbf{A}_{xx}\mathbf{U}_{,x}^h + y_{23}\mathbf{A}_{xy}\mathbf{U}_{,y}^h + x_{32}\mathbf{A}_{yx}\mathbf{U}_{,x}^h + x_{32}\mathbf{A}_{yy}\mathbf{U}_{,y}^h \\ y_{31}\mathbf{A}_{xx}\mathbf{U}_{,x}^h + y_{31}\mathbf{A}_{xy}\mathbf{U}_{,y}^h + x_{13}\mathbf{A}_{yx}\mathbf{U}_{,x}^h + x_{13}\mathbf{A}_{yy}\mathbf{U}_{,y}^h \\ y_{12}\mathbf{A}_{xx}\mathbf{U}_{,x}^h + y_{12}\mathbf{A}_{xy}\mathbf{U}_{,y}^h + x_{21}\mathbf{A}_{yx}\mathbf{U}_{,x}^h + x_{21}\mathbf{A}_{yy}\mathbf{U}_{,y}^h \end{bmatrix} \tag{3.83}$$

ou ainda,

$$\mathbf{r}_{cpg}^{e} = \begin{bmatrix} \mathbf{r}_{cpg}^{1} \\ \mathbf{r}_{cpg}^{2} \\ \mathbf{r}_{cpg}^{3} \end{bmatrix} \tag{3.84}$$

onde

$$\begin{aligned} \mathbf{r}_{cpg}^{1} &= \frac{\tau}{2}(y_{23}\mathbf{A}_{xx}\mathbf{U}_{,x}^h + y_{23}\mathbf{A}_{xy}\mathbf{U}_{,y}^h + x_{32}\mathbf{A}_{yx}\mathbf{U}_{,x}^h + x_{32}\mathbf{A}_{yy}\mathbf{U}_{,y}^h) \\ \mathbf{r}_{cpg}^{2} &= \frac{\tau}{2}(y_{31}\mathbf{A}_{xx}\mathbf{U}_{,x}^h + y_{31}\mathbf{A}_{xy}\mathbf{U}_{,y}^h + x_{13}\mathbf{A}_{yx}\mathbf{U}_{,x}^h + x_{13}\mathbf{A}_{yy}\mathbf{U}_{,y}^h) \\ \mathbf{r}_{cpg}^{3} &= -(\mathbf{r}_{cpg}^{1} + \mathbf{r}_{cpg}^{2}) \end{aligned} \tag{3.85}$$

Finalmente as contribuições do resíduo devido ao termo de captura de descontinuidades são dadas por:

$$\mathbf{r}_{op}^{e} = \tau \int_{\Omega^e} \begin{bmatrix} \mathbf{B}_x^t & \mathbf{B}_y^t \end{bmatrix} \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} d\Omega^e \mathbf{v}^e \tag{3.86}$$

Devido as definições (3.29) e (3.30), pode-se representar $\mathbf{r}_{op}^{e}$ por:

$$\mathbf{r}_{op}^{e} = \delta \int_{\Omega^e} d\Omega^e \begin{bmatrix} \mathbf{B}_x^t & \mathbf{B}_y^t \end{bmatrix} \begin{bmatrix} \mathbf{U}_{,x}^h \\ \mathbf{U}_{,y}^h \end{bmatrix} = \frac{\tau}{2} \begin{bmatrix} y_{23}\mathbf{U}_{,x}^h + x_{32}\mathbf{U}_{,y}^h \\ y_{31}\mathbf{U}_{,x}^h + x_{13}\mathbf{U}_{,y}^h \\ y_{12}\mathbf{U}_{,x}^h + x_{21}\mathbf{U}_{,y}^h \end{bmatrix} \tag{3.87}$$

ou ainda,

$$\mathbf{r}_{op}^{e} = \begin{bmatrix} \mathbf{r}_{op}^{1} \\ \mathbf{r}_{op}^{2} \\ \mathbf{r}_{op}^{3} \end{bmatrix} \tag{3.88}$$

onde

$$\begin{aligned} \mathbf{r}_{op}^{1} &= \frac{\tau}{2}(y_{23}\mathbf{U}_{,x}^h + x_{32}\mathbf{U}_{,y}^h) \\ \mathbf{r}_{op}^{2} &= \frac{\tau}{2}(y_{31}\mathbf{U}_{,x}^h + x_{13}\mathbf{U}_{,y}^h) \\ \mathbf{r}_{op}^{3} &= -(\mathbf{r}_{op}^{1} + \mathbf{r}_{op}^{2}) \end{aligned} \tag{3.89}$$

3.5 Integração no Tempo

Para solucionar o sistema (3.31) é usado o algoritmo Preditor/Multicorretor, encontrado em [13]. Neste algoritmo a equação (3.31) é discretizada por diferenças finitas. Sendo n o contador de passos de tempo, as aproximações para $\mathbf{a}(t_n)$ e $\mathbf{v}(t_n)$, representadas por $\mathbf{a}_n$ e $\mathbf{v}_n$, respectivamente, e Δt o passo de tempo, o algoritmo pode ser resumido nos seguintes passos:

Dados $\mathbf{v}_n$ e $\mathbf{a}_n$,

Predição:

$i = 0$

$\mathbf{v}_n^{(0)} = \bar{\mathbf{v}}_{n+1} = \mathbf{v}_n + (1-\alpha)\Delta t \mathbf{a}_n$

$\mathbf{a}_n^{(0)} = \mathbf{0}$

Correção:

calcule: $\mathbf{R} = -(\mathbf{M}\mathbf{a}_{n+1}^{(i)} + \mathbf{C}\mathbf{v}_{n+1}^{(i)})$

Resolva: $\mathbf{M}^*\Delta\mathbf{a} = \mathbf{R}$

Atualize: $\begin{array}{l} \mathbf{a}_{n+1}^{(i)} = \mathbf{a}_{n+1}^{(i-1)} + \Delta\mathbf{a} \\ \mathbf{v}_{n+1}^{(i)} = \mathbf{v}_{n+1}^{(i-1)} + \alpha\Delta t\Delta\mathbf{a} \end{array}$

Observações:

- $\alpha \in [0,1]$ é um parâmetro que controla a precisão e a estabilidade do método de integração. Nos métodos implícitos alguns valores de α são amplamente conhecidos. Por exemplo, para $\alpha = 1$ tem-se as diferenças atrasadas (conhecidas por *backward*). Para o caso de $\alpha = 0.5$, o procedimento é denominado por regra trapezoidal. Para os métodos explícitos, considerando $\alpha = 0$ tem-se as diferenças adiantadas (conhecidas por *forward*).

- A definição para a matriz $\mathbf{M}^*$ depende do tipo de método a ser escolhido. No caso do método explícito ALIABADI e TEZDUYAR [13] escolhem uma matriz diagonal obtida através da diagonalização da matriz de “massa”. A simplicidade da inversão de matrizes diagonais torna o algoritmo rápido e barato. Porém, a penalidade desta aproximação está no tamanho do passo de tempo, que deve satisfazer a condição de CFL. Em geral, para se garantir a convergência, o passo de tempo deve ser bem pequeno, detalhes são tratados por SHAKIB [20]. No caso do método implícito, que será usado aqui, a matriz $\mathbf{M}^*$ é definida por:

$$\mathbf{M}^* = \mathbf{M} + \alpha\Delta t\mathbf{C} \tag{3.90}$$

onde $\mathbf{M}^*$ é uma matriz não simétrica, porém com perfil simétrico, isto é, para cada m_{ij}^* existe m_{ji}^*, sendo $m_{ij}^* \neq m_{ji}^*$. Os métodos mais usados atualmente para a solução do sistema de equações lineares resultantes são métodos iterativos de Krylov. Em [10, 11, 12, 13, 14] emprega-se variantes do método GMRES com diferentes formas de precondicionamento. No presente trabalho adota-se o método GMRES, descrito no Apêndice A. Nos exemplos da seção 3.6 emprega-se um precondicionador bloco-diagonal nodal e uma implementação elemento-por-elemento. Outras formas de precondicionamento serão discutidas mais adiante.

- No capítulo 5 serão discutidas técnicas de aceleração de convergência para o regime permanente baseadas na introdução de passo local e congelamento do operador de captura.

3.6 Exemplos de Validação

São considerados três exemplos padrão na análise de escoamentos com gradientes elevados regidos pela equação de Euler: Choque Estacionário Unidimensional, Choque Oblíquo Bidimensional e Choque Refletido Bidimensional. Os resultados são comparados com as respectivas soluções analíticas. Para a formulação em variáveis conservativas verifica-se que a forma A^2, definida em (3.4), produz os melhores resultados, sendo adotada em todos os casos. No algoritmo de avanço no tempo foi considerado a regra trapezoidal ($\alpha = 0.5$) e o número de multicorreções foi fixado em 3. Em todos os exemplos serão considerados gases compressíveis com expoente adiabático $\gamma = 1.4$ e $c_v = 716.5$.

3.6.1 Choque Normal Unidimensional

Este problema consiste em observar o escoamento através de duas regiões separadas por um choque normal, ou seja, no instante inicial ($t = 0$) a velocidade de escoamento apresenta um salto, que se mantém invariante. Considera-se uma malha com 39×2 células, sendo 2 elementos triangulares em cada célula, distribuídos no retângulo $0 \leq x \leq 39, -0.5 \leq x \leq 0.5$, conforme descrito na Figura 3.2. O choque ocorre em $x = 20$ e as condições iniciais, em um sistema compatível de unidades, são dadas por:

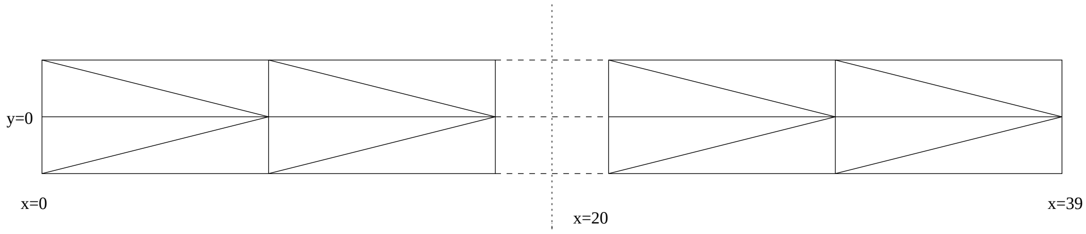
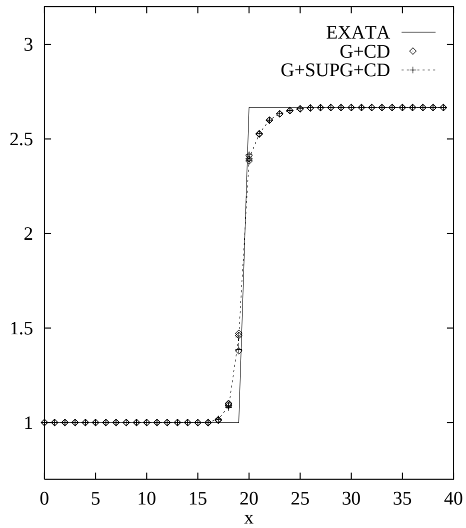


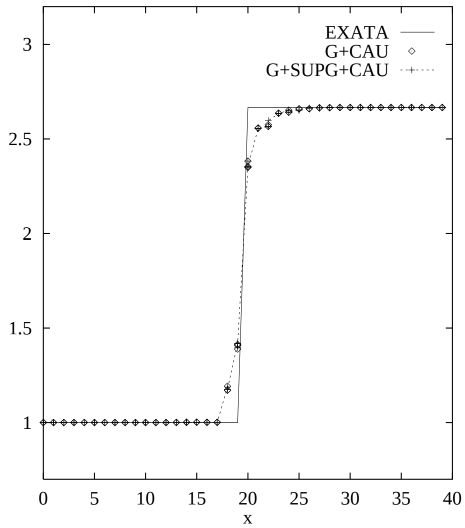
Figura 3.2: Choque Normal (1D) - Esquema do Problema

$$x < 20 \quad \begin{cases} M &= 2 \\ \rho &= 1 \\ u_1 &= 1 \\ p &= 0.17857 \end{cases} \qquad x > 20 \quad \begin{cases} M &= 0.57735 \\ \rho &= 2.66667 \\ u_1 &= 0.37500 \\ p &= 0.80357 \end{cases} \tag{3.91}$$

A componente da velocidade u_2 é nula em toda análise. Densidades, temperaturas e velocidades são consideradas prescritas na entrada do escoamento (parede da esquerda) e a temperatura na saída (parede da direita).

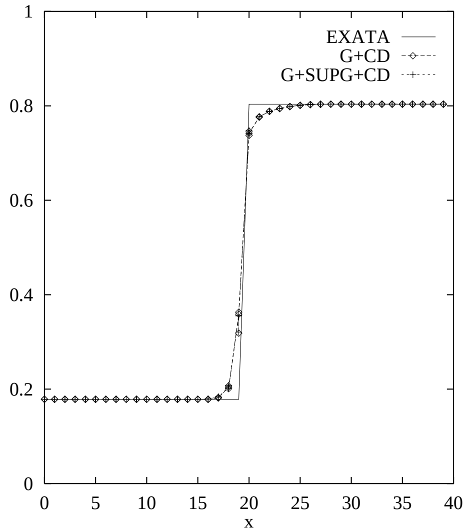


(a) Densidade com Operador CD

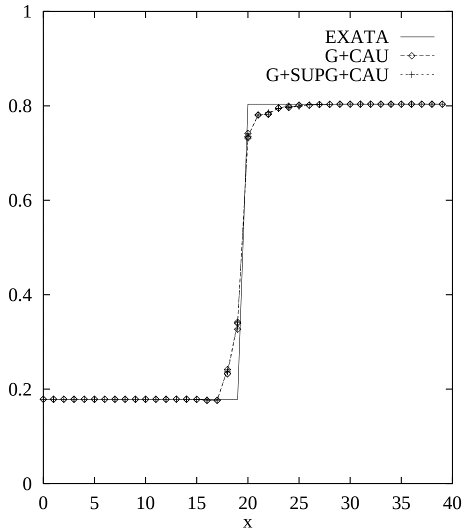


(b) Densidade com Operador CAU

Figura 3.3: Choque Normal (1D) - Perfis de Densidade

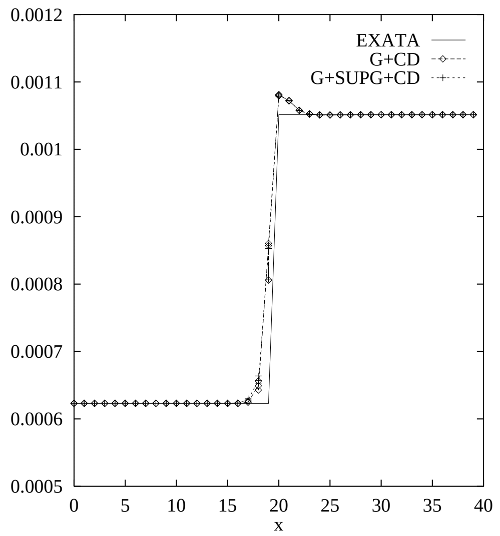


(a) Pressão com Operador CD

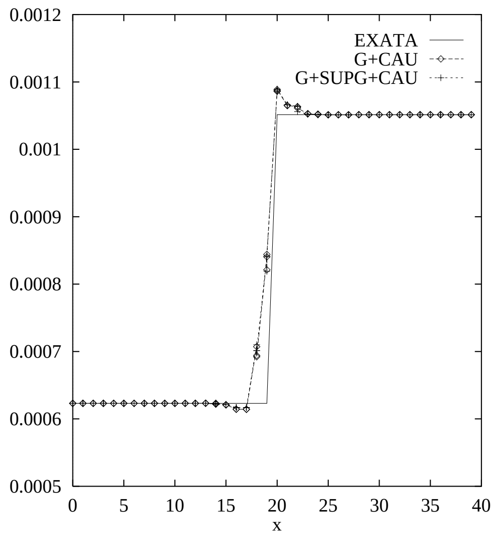


(b) Pressão com Operador CAU

Figura 3.4: Choque Normal (1D) - Perfis de Pressão



(a) Temperatura com Operador CD



(b) Temperatura com Operador CAU

Figura 3.5: Choque Normal (1D) - Perfis de Temperatura

As Figuras 3.3, 3.4 e 3.5 mostram respectivamente a comparação dos perfis de densidade, pressão e temperatura entre a solução exata e as formulações do presente trabalho, após 300 passos. Considerou-se 5 vetores de Krylov para o algoritmo GMRES e tolerância de 0.005. Nessas figuras os resultados obtidos com a formulação dada pela equação (3.3) e os operadores de captura de descontinuidades definidos nas equações (3.12) e (3.18) são identificados respectivamente por G+SUPG+CD e G+SUPG+CAU, e quando o termo SUPG não é considerado, as soluções são identificadas respectivamente por G+CD e G+CAU. Observa-se que as soluções com o operador CD estão mais próximas da solução exata. Nota-se na Figura 3.5 que os perfis de temperatura apresentam pequenas oscilações na região do choque. Este comportamento também foi observado por HAUKE e HUGHES [17], que o associam ao emprego de variáveis de conservação. Outro fato que pode contribuir para a formação dessas pequenas oscilações é a definição da matriz de estabilização SUPG, pois considera a forma diagonalizada com um único coeficiente para densidade, velocidades, pressão e temperatura.

3.6.2 Choque Oblíquo Bidimensional

Este problema consiste em um escoamento bidimensional supersônico (número de Mach M=2) de um fluido invíscido sobre uma cunha, fazendo um ângulo de $-10°$ em relação à malha, conforme mostra a Figura 3.6. Pode-se determinar analiticamente a formação de um choque oblíquo a $29.3°$ com a parede. O domínio computacional

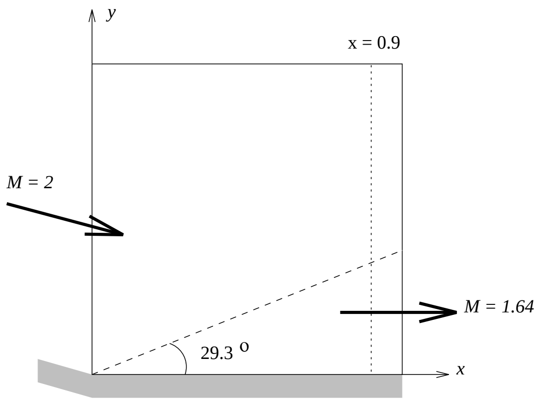


Figura 3.6: Choque Oblíquo (2D) - Esquema do problema

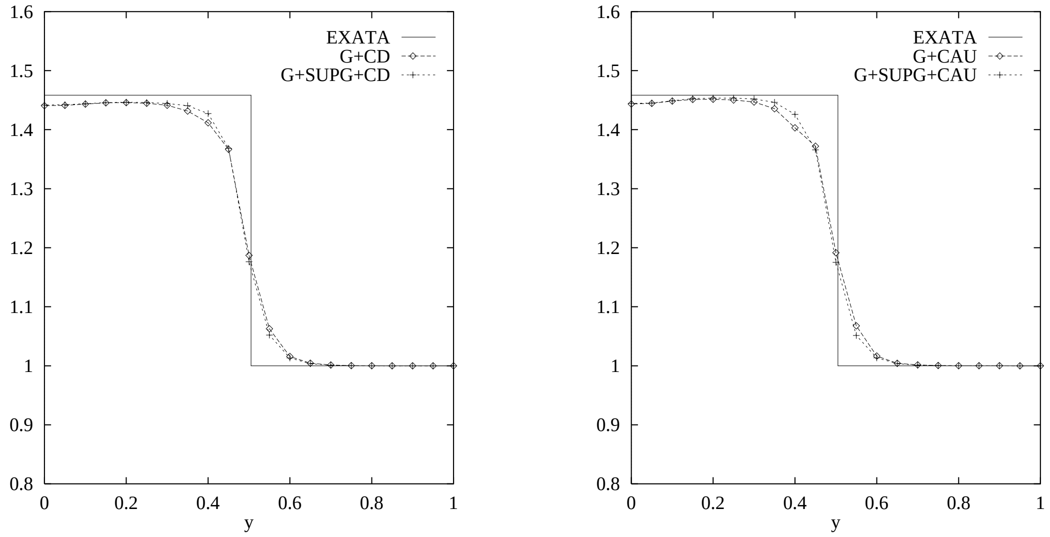
$0 \leq x \leq 1$ e $0 \leq y \leq 1$ é discretizado em 20×20 células com dois elementos triangulares em cada célula. Supondo-se um sistema compatível de unidades, as condições de contorno prescritas na entrada e no topo do domínio são:

$$\begin{array}{rcl} M & = & 2 \\ \rho & = & 1 \\ u_1 & = & cos10^0 \\ u_2 & = & -sen10^0 \\ p & = & 0.17857 \end{array} \tag{3.92}$$

Além disso, a velocidade u_2 é nula na parede inferior, não sendo imposta nenhuma condição na saída. As condições iniciais são consideradas como de escoamento livre. A solução exata na saída abaixo do choque é dada por:

$$\begin{array}{rcl} M & = & 1.64052 \\ \rho & = & 1.45843 \\ u_1 & = & 0.88731 \\ u_2 & = & 0 \\ p & = & 0.30475 \end{array} \tag{3.93}$$

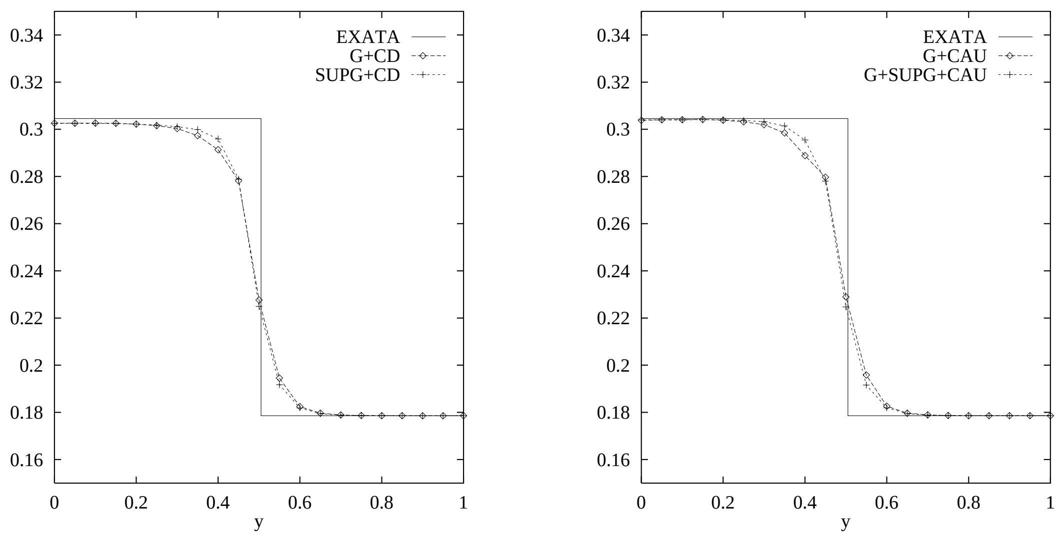
Em todos os testes observa-se o comportamento da solução após 300 passos. As Figuras 3.7, 3.8 e 3.9 comparam respectivamente os perfis de densidade, pressão e temperatura em $x = 0.9$ para as formulações com e sem a parcela de estabilização SUPG considerando os operadores de captura CD e CAU. Observa-se que as soluções com a parcela SUPG representam melhor o choque. Porém as soluções somente com os operadores de captura também são satisfatórias. Os perfis de temperatura também apresentam pequenas oscilações como aquelas observadas no exemplo anterior.



(a) Densidade com operador CD

(b) Densidade com operador CAU

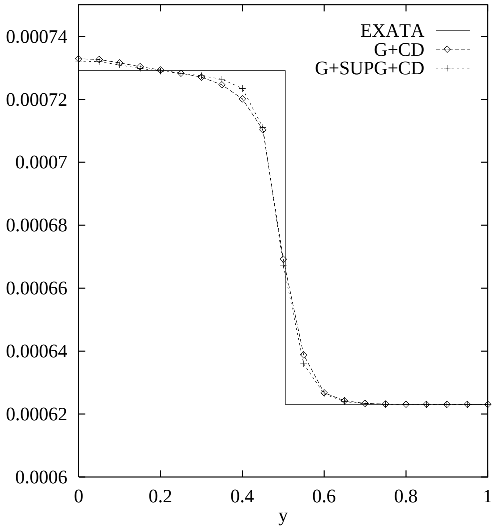
Figura 3.7: Choque Oblíquo (2D) - Comparação de Perfis de Densidade para $x = 0.9$



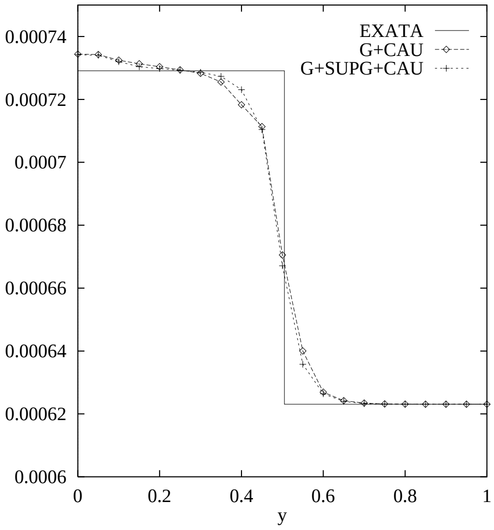
(a) Pressão com operador CD

(b) Pressão com operador CAU

Figura 3.8: Choque Oblíquo (2D) - Comparação de Perfis de Pressão para $x = 0.9$



(a) Temperatura com operador CD



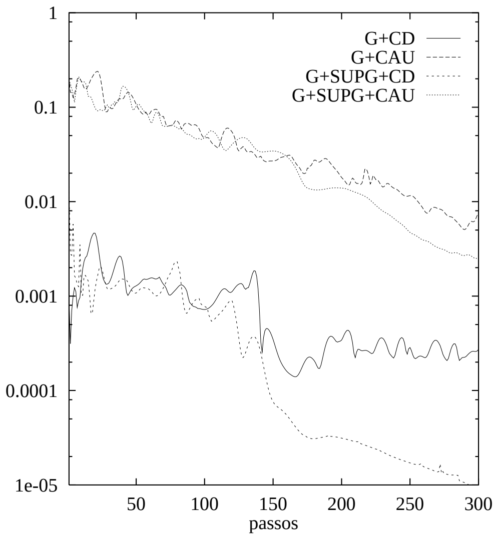
(b) Temperatura com operador CAU

Figura 3.9: Choque Oblíquo (2D) - Comparação de Perfis de Temperatura para $x = 0.9$

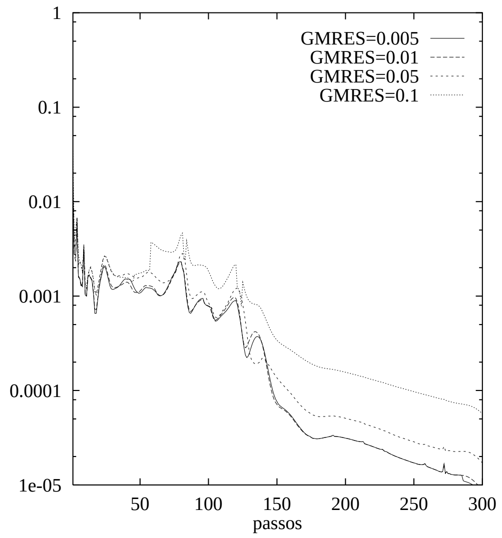
A Figura 3.10(a) mostra o comportamento do resíduo em todas as formulações considerando a tolerância para o GMRES igual a 0.005 e o número de vetores de Krylov igual a 5. Observa-se que nas formulações onde o operador CD foi usado o resíduo diminui com maior rapidez. Uma comparação de tolerâncias para o algoritmo iterativo GMRES usando a formulação com SUPG e operador CD é feita na Figura 3.10(b). A solução para as diversas tolerâncias é a mesma, e os resíduos apresentam comportamentos bastante similares. Portanto conclui-se que não é necessário usar tolerâncias grandes para se obter bons resultados.

Com o objetivo de verificar a importância da orientação da malha para a formulação proposta, considera-se o mesmo exemplo com a orientação da malha invertida, onde as condições de contorno descritas são assumidas na lateral direita do domínio e o ângulo de incidência do escoamento é de $+190°$. Denomina-se condições "A" e "B" os casos de ângulo de escoamentos $-10°$ e $+190°$, respectivamente. As Figuras 3.11(a) e 3.11(b) comparam os perfis de densidade e pressão para as condições "A" e "B". Observa-se que a condição "A" conduz a melhores resultados.

A Figura 12 mostra as isocurvas da densidade obtidas usando as formulações com e sem a parcela de estabilização SUPG, para as condições "A" e "B", usando o operador de captura CD. As isocurvas da densidade com o termo SUPG, mostrada nas Figuras 3.12(a) e 3.12(b), são menos difusivas. Além disso, observa-se que, em geral, as soluções para a condição "B" são mais difusivas, evidenciando um efeito de

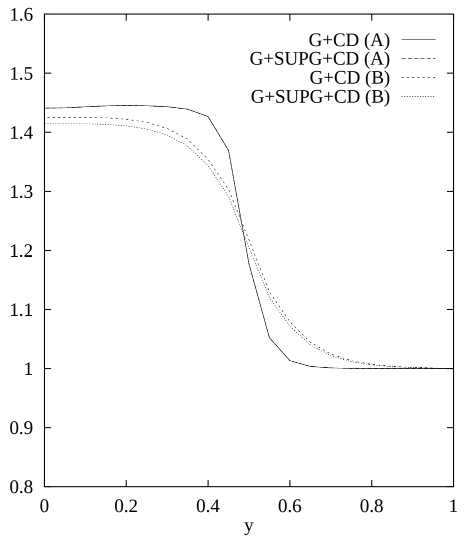


(a) Resíduos das várias formulações implementadas

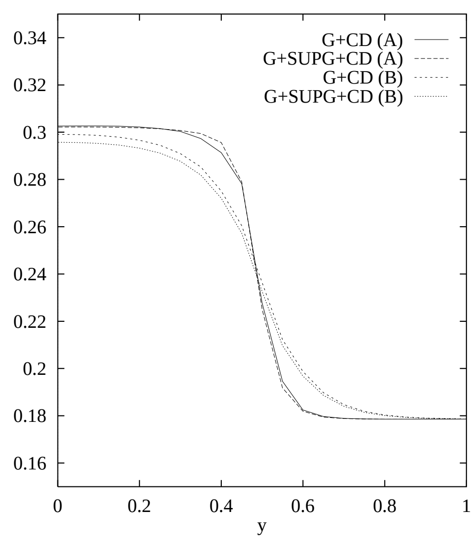


(b) Resíduos da formulação G+SUPG+CD para várias tolerâncias do GMRES, com 5 vetores de Krylov

Figura 3.10: Choque Oblíquo (2D) - Comparações para o resíduo

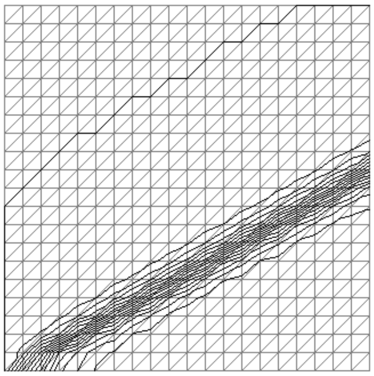


(a) Perfis de Densidade

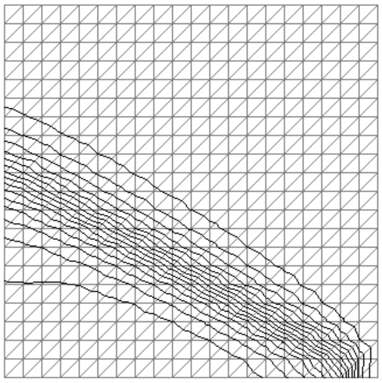


(b) Perfis de Pressão

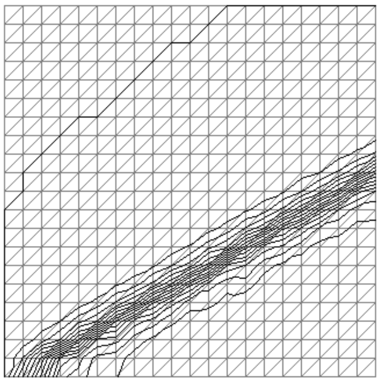
Figura 3.11: Choque Oblíquo (2D) - Comparação de densidade e pressão para os casos “A” e “B”



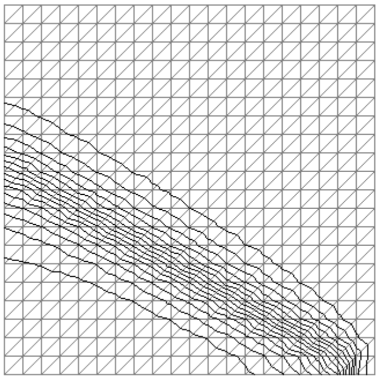
(a) G+SUPG+CD (A)



(b) G+SUPG+CD (B)



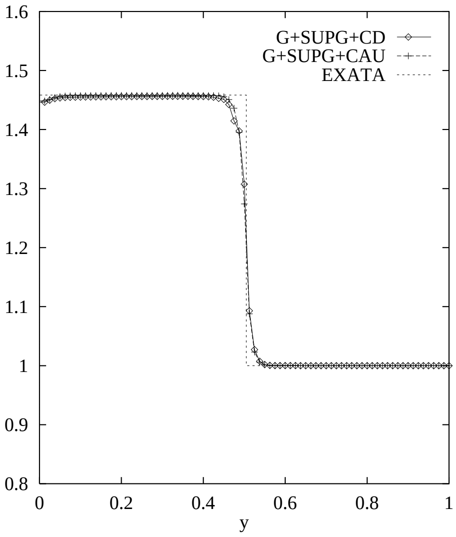
(c) G+CD (A)



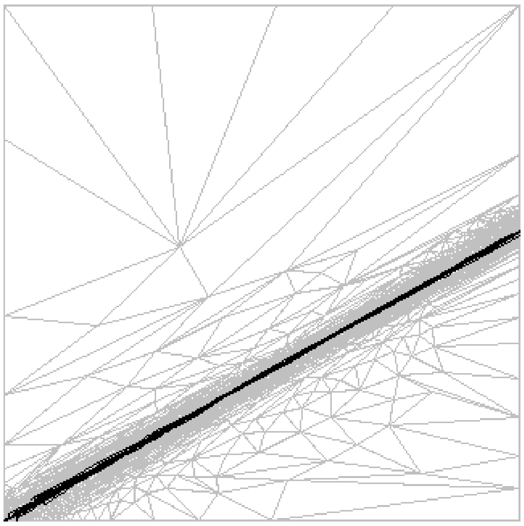
(d) G+CD (B)

Figura 3.12: Choque Oblíquo (2D) - Isocurvas de Densidade para os casos “A” e “B” orientação de malha.

A Figura 3.13(a) mostra os perfis de densidade para uma malha com 80×80 células, com 2 elementos em cada célula. Observa-se nesta figura que os resultados para as formulações G+SUPG+CD e G+SUPG+CAU são praticamente coincidentes. A Figura 3.13(b) apresenta as isocurvas de densidade para a formulação G+SUPG+CD para uma malha não estruturada com 766 nós e 1466 elementos, sendo bastante refinada na região do choque [7]. Pode-se observar a boa precisão da solução obtida.



(a) Perfis de Densidade para malha 80×80



(b) Isocurvas de Densidade para a malha com 766 nós e 1466 elementos

Figura 3.13: Choque Oblíquo (2D) - Resultados para a Densidade

3.6.3 Choque Refletido Bidimensional

Este problema estacionário bidimensional é composto de três regiões de escoamento separadas por um choque oblíquo e sua reflexão ao longo de uma parede, conforme a Figura 3.14 . Para Mach = 2.9 na entrada (parede da esquerda) tem-se os seguintes dados, em um sistema compátivel de unidades:

$$\text{Região 1} \quad \left\{ \begin{array}{lcl} M & = & 2.9 \\ \rho & = & 1.0 \\ u_1 & = & 2.9 \\ u_2 & = & 0.0 \\ p & = & 0.714286 \end{array} \right. \tag{3.94}$$

e considerando que a incidência do choque faz um ângulo de 29°, tem-se a solução exata dada por:

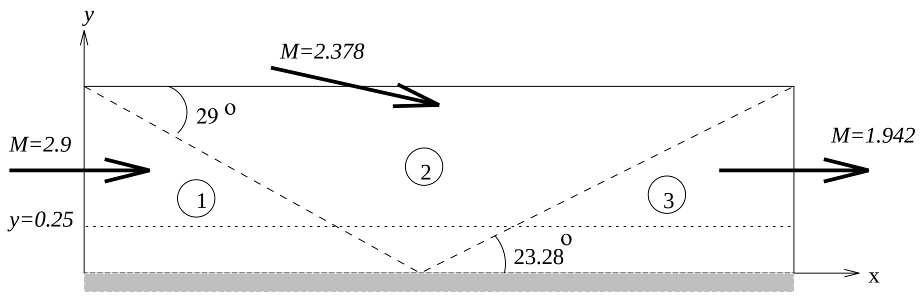


Figura 3.14: Choque Refletido (2D) - Esquema do problema

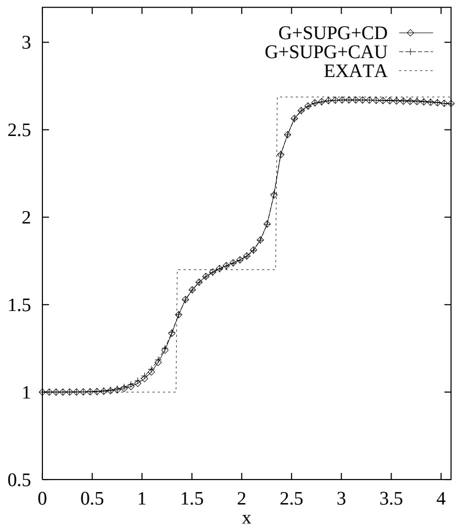
$$\text{Região 2} \quad \begin{cases} M &= \ 2.3781 \\ \rho &= \ 1.7 \\ u_1 &= \ 2.61934 \\ u_2 &= -0.50632 \\ p &= \ 1.52819 \end{cases} \tag{3.95}$$

$$\text{Região 3} \quad \begin{cases} M &= 1.94235 \\ \rho &= 2.68728 \\ u_1 &= 2.40140 \\ u_2 &= 0.0 \\ p &= 2.93407 \end{cases} \tag{3.96}$$

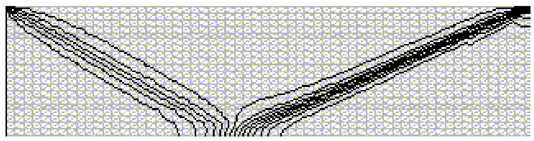
O domínio computacional $0 \leq x \leq 4.1$ e $0 \leq y \leq 1$ é discretizado em 60×20 células com dois elementos triangulares em cada célula. São consideradas prescritas densidade, velocidades e pressão no contorno da esquerda e no topo. Na parede inferior a velocidade u_2 é nula e na saída (parede da direita) não são impostas condições de contorno. São consideradas condições iniciais de escoamento livre.

As Figuras 3.15(a), 3.16(a) e 3.17(a) mostram respectivamente os perfis da densidade, pressão e temperatura após 300 passos, considerando a tolerância do GMRES igual a 0.1, e número de vetores de Krylov igual a 5. Os resultados são satisfatórios, porém observa-se que o choque entre as regiões 1 e 2 é mais difusivo que o choque entre as regiões 2 e 3 para as três grandezas. As Figuras 3.15(b), 3.16(b) e 3.17(b) apresentam respectivamente as isocurvas da densidade, pressão e temperatura. Nota-se que as isocurvas da temperatura também neste exemplo apresentam pequenas oscilações. Contudo testes realizados com malhas mais refinadas demonstram que os procedimentos são estáveis, ou seja, a medida que a malha é refinada a precisão aumenta, como pode ser comprovado na Figura 3.18, que mostra perfis de densidade para uma malha com 60×20 células, considerando 4 elementos em cada célula, onde cada célula foi subdividida ao longo de suas diagonais principais.

A Figura 3.19 mostra isocurvas de densidade, pressão, temperatura, velocidades e número de Mach para uma malha não estruturada com 1837 nós e 3594 elementos, sendo bastante refinada na região do choque. Observa-se também uma boa precisão da solução obtida.

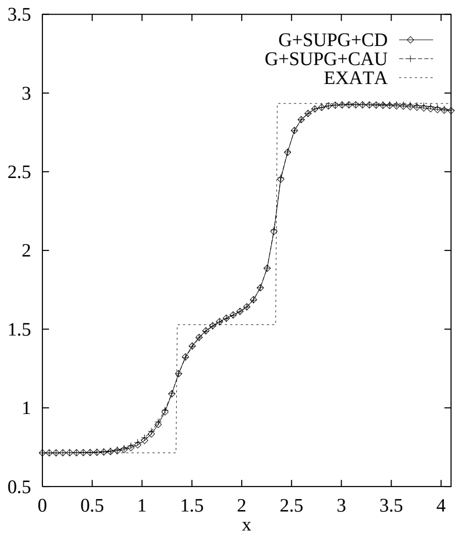


(a) Perfis de Densidade (y=0.25)

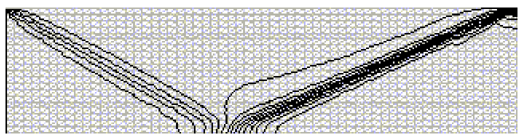


(b) Isocurvas de Densidade (G+SUPG+CD)

Figura 3.15: Choque Refletido (2D) - Malha 60 × 20 - Densidade

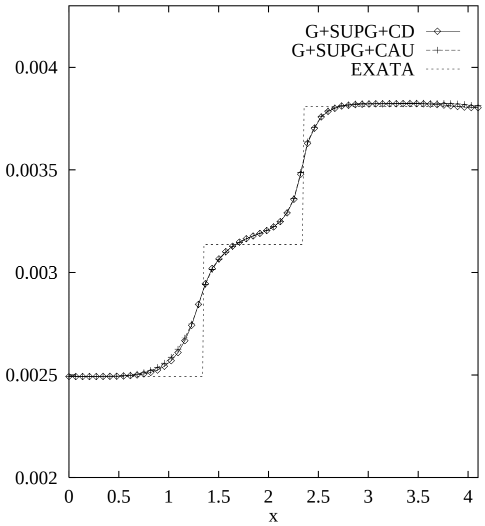


(a) Perfis de Pressão (y=0.25)

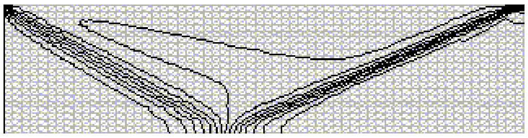


(b) Isocurvas de Pressão (G+SUPG+CD)

Figura 3.16: Choque Refletido (2D) - Malha 60 × 20 - Pressão



(a) Perfis de Temperatura (y=0.25)



(b) Isocurvas de Temperatura (G+SUPG+CD)

Figura 3.17: Choque Refletido (2D) - Malha 60 × 20 - Temperatura

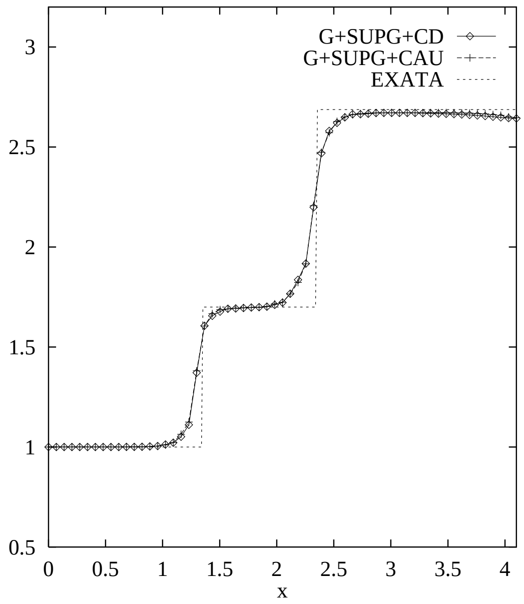
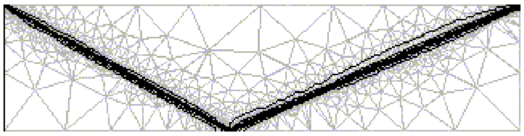
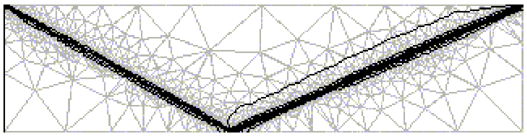


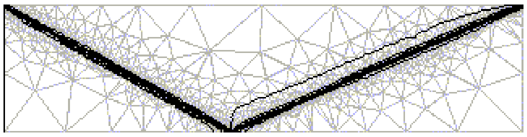
Figura 3.18: Choque Refletido (2D) - Perfis de Densidade para malha 60 × 20, com 4 elementos por célula



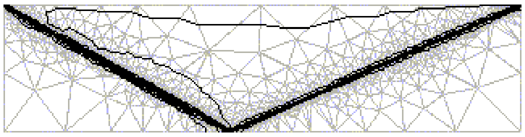
(a) Isocurvas de Densidade



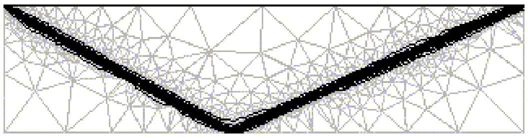
(b) Isocurvas de Pressão



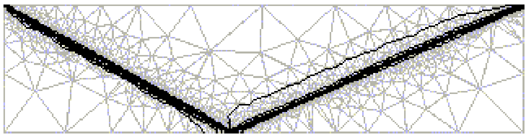
(c) Isocurvas de Temperatura



(d) Isocurvas de V_x



(e) Isocurvas de V_y



(f) Isocurvas de M

Figura 3.19: Choque Refletido (2D) - Isocurvas do Choque Refletido com operador CD - Malha adaptada com 1837 nós e 3594 elementos

Capítulo 4

Implementação da Formulação Estabilizada Semi-Discreta das Equações de Euler por Aresta

Este capítulo é organizado como se segue. Inicialmente apresenta-se um breve estudo sobre estruturas de dados para armazenamento de matrizes esparsas, explorando as relações entre grafos, malhas e as matrizes correspondentes. Procura-se mostrar que as estruturas de dados por arestas são mais vantajosas para o caso de matrizes esparsas oriundas de discretizações de elementos finitos. Em seguida, mostra-se como se constroem as matrizes de arestas a partir do desmembramento das matrizes de elemento provenientes da formulação SUPG para as equações de Euler, vista no Capítulo 3. Da mesma forma que no Capítulo anterior, apresentam-se as expressões resultantes das matrizes de arestas e dos termos do vetor de resíduos. Descreve-se então em detalhe o algoritmo para o produto matriz-vetor esparso necessário no método GMRES empregando-se as estruturas de dados por arestas, procurando compará-lo com algoritmos elemento-por-elemento e métodos esparsos gerais. O Capítulo termina com exemplos numéricos para validação da implementação por arestas.

4.1 Estruturas de Dados para Matrizes Esparsas

Este item tem por objetivo comparar estratégias para representação de matrizes esparsas enfatizando esquemas de armazenamento. Um breve estudo de grafos e representação de malhas por grafos também é apresentado.

4.1.1 Representação de uma matriz esparsa por grafos

Formulações de elementos finitos em geometria complexa geram matrizes esparsas. Neste caso, os elementos não nulos ocorrem em posições arbitrárias, sendo necessário o conhecimento de sua localização sempre que for preciso acessá-los. A eficiência dos métodos iterativos está fortemente ligada a esse fato, pois estes manipulam somente os coeficientes não nulos da matriz. A Teoria dos Grafos é uma importante ferramenta para representar matrizes esparsas, sendo usada, por exemplo, nas técnicas de paralelismo por diversos algoritmos de solução de sistemas lineares [35].

Um grafo não direcionado $G = (V, E)$ poder ser definido pelos conjuntos:

- Um conjunto de vértices:

$$V = \{v_1, v_2, \dots, v_n\}$$

- Um conjunto de arcos E que consiste do par (v_i, v_j), onde v_i e v_j são elementos de V, isto é, $E \subseteq V \times V$.

Uma matriz esparsa de ordem n pode ser representada por um grafo $G = (V, E)$, sendo as n variáveis representadas pelos n vértices em V. Os arcos representam a relação binária estabelecida pelas equações, da seguinte forma:

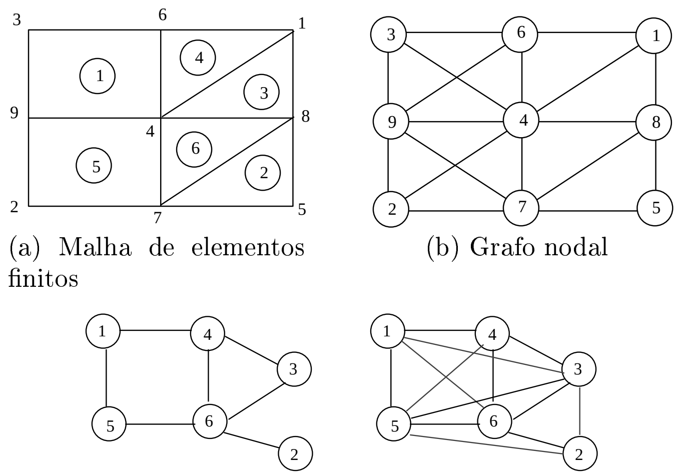
Existe um arco do nó i para o nó j quando $a_{ij} \neq 0$

Este arco representará então as relações binárias:

Equação i envolve a variável j
Equação j envolve a variável i

4.1.2 Representação de grafos para malhas de elementos finitos

Existe uma associação intrínseca entre a matriz esparsa originada pelo método dos elementos finitos e a topologia da malha [35]. Estas propriedades topológicas podem ser exploradas através da associação de grafos, malhas e matrizes esparsas. Três tipos de grafos são mais utilizados: grafo nodal, grafo dual e grafo de comunicação [36]. O grafo nodal tem como vértices os nós da malha e a ligação dos vértices são definidas pelas arestas, enquanto o grafo dual tem por vértices os elementos e a ligação entre vértices ocorre quando há arestas comuns a dois elementos adjacentes. O grafo de comunicação também tem por vértices os elementos, porém a ligação entre eles é definida por todos os elementos que tem pelo menos um nó comum. A Figura 4.1 mostra um exemplo de malha de elementos finitos e os três tipos de grafos correspondentes.



(a) Malha de elementos finitos

(b) Grafo nodal

(c) Grafo Dual

(d) Grafo de Comunicação

Figura 4.1: Diferenças entre tipos de grafos para um malha de elementos finitos

Observa-se ainda que para elementos triangulares o grafo nodal equivale ao grafo da matriz esparsa correspondente à malha de elementos finitos. Assim, percorrer os arcos do grafo nodal equivale a percorrer as arestas geométricas da malha de elementos finitos, ou seja, referenciar os termos envolvendo as equações i e j da matriz esparsa correspondente. O mesmo não ocorre ao se percorrer o grafo dual, isto é, os elementos da malha, uma vez que estes se associam indiretamente a matriz esparsa através das contribuições das matrizes de cada elemento.

4.1.3 Esquemas de armazenamento de matrizes esparsas

Várias técnicas tem sido definidas com o objetivo principal de armazenar somente os elementos diferentes de zero de uma matriz esparsa. Neste item é feito uma comparação entre técnicas que usam estrutura de dados diferentes, visando reduzir a quantidade de memória utilizada. Considera-se uma matriz esparsa A de ordem n com um número total de elementos diferentes de zero igual a nnz.

Estrutura de dados usando o formato CSR (*Compressed Sparse Row*)

Na estrutura CSR são necessários 3 vetores, sendo:

- Um vetor real AA, contendo os valores reais $a_{ij} \neq 0$, armazenados linha por linha, da linha 1 até a linha n. AA tem nnz elementos.

- Um vetor de inteiros JA, contendo o índice da coluna de cada elemento a_{ij} armazenado em AA, com nnz elementos.

- um vetor IA de inteiros, contendo os ponteiros onde começam cada linha em AA e JA. Ou seja, o elemento $IA(i)$ contém a posição de AA e JA onde a i-ésima linha começa. O tamanho de IA é $n+1$, com $IA(n+1)$ contendo o número $IA(1)+nnz$, isto é, o endereço em AA e JA do começo de um número de linha fictício $n+1$.

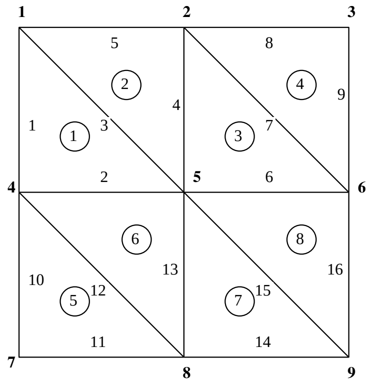


Figura 4.2: Malha de elementos triangulares lineares com 8 elementos, 9 nós e 16 arestas

A Figura 4.2 representa uma malha de elementos triangulares, considerando apenas uma variável em cada nó. A matriz esparsa correspondente possui 41 coeficientes diferentes de zero:

$$A = \begin{bmatrix} \otimes & \times & & \times & \times & & & & \\ \times & \otimes & \times & & \times & \times & & & \\ & \times & \otimes & & & \times & & & \\ \times & & & \otimes & \times & & \times & \times & \\ \times & \times & & \times & \otimes & \times & & \times & \times \\ & \times & \times & & \times & \otimes & & & \times \\ & & & \times & & & \otimes & \times & \\ & & & \times & \times & & \times & \otimes & \times \\ & & & & \times & \times & & \times & \otimes \end{bmatrix}$$

O armazenamento do tipo CSR para esse exemplo pode ser representado por:

$$\begin{array}{llllllllllllll} AA = [& \overbrace{\otimes}^{1} & \times & \times & \times & \overbrace{\times}^{5} & \otimes & \times & \times & \ldots & \overbrace{\times}^{38} & \times & \times & \times] \\ JA = [& 1 & 2 & 4 & 5 & 1 & 2 & 3 & 5 & \ldots & 5 & 6 & 8 & 9] \\ IA = [& 1 & 5\ldots & 38 & 42] \end{array}$$

Para este caso são necessárias 92 posições de memória.

Estrutura de dados usando o formato MSR (***Modified Sparse Row***)

Uma variação comum do esquema CSR explora o fato da diagonal principal de A ser usualmente de elementos diferentes de zero e ser acessada com maior frequência que os demais elementos. O formato MSR utiliza somente 2 vetores:

- Um vetor AA de reais, contendo os elementos não nulos de A. A diagonal de A ocupa as n primeiras posições de AA. A posição $(n+1)$ do vetor AA não é usada para coeficientes de A, mas pode ser usada para alocar alguma informação adicional de A. A partir da posição $(n+2)$ os coeficientes não nulos são armazenados por linha.

- Um vetor de inteiros JA, contendo nas $(n+1)$ primeiras posições os ponteiros de começo de cada linha de AA. Nas posições restantes estão as colunas correspondentes de cada elemento em AA, ou seja, para cada elemento $AA(k)$, o inteiro $JA(k)$ representa o índice da coluna na matriz A. No esquema a seguir encontra-se o armazenamento do tipo MSR para a malha ilustrada na Figura 4.2:

$$\begin{array}{rcccccccccccc} AA = [& \otimes & \otimes & \ldots & \otimes & * & \overbrace{\times}^{11} & \times & \overbrace{\times}^{14} & \ldots & \overbrace{\times}^{40} & \times & \times] \\ JA = [& 11 & 14 & \ldots & 40 & 42 & 4 & 5 & 1 & \ldots & 5 & 6 & 8] \end{array}$$

Neste exemplo são necessárias 84 posições de memória.

Estrutura de Dados Elemento-por-Elemento

Neste caso os coeficientes da matriz esparsa A são armazenados a nível de cada elemento. Considerando elementos triangulares lineares, tem-se em cada elemento uma matriz de ordem 3, ou seja, 9 coeficientes. Para uma malha de elementos finitos com nel elementos todos os coeficientes não nulos da matriz esparsa A podem ser armazenados na matriz de reais AA, com 9 colunas e nel linhas. Além disso, é necessário um arranjo de inteiros, a matriz de localização $lm(nel, 3)$, que contem o número das equações globais associados a cada nó do elemento, conforme definida por HUGHES [5]. Maiores detalhes sobre as estruturas de dados elemento-por-elemento podem ser encontrados em [20, 26]. Em geral a diagonal de A é acessada com frequência, sendo guardada separadamente em um vetor $diagA$ de ordem n. Para o exemplo de malha ilustrado na Figura 4.2, tem-se o seguinte esquema de armazenamento:

$$AA = \begin{bmatrix} \times & \times & \times & \dots & \dots & \times \\ \times & \times & \times & \dots & \dots & \times \\ \times & \times & \times & \dots & \dots & \times \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \times & \times & \times & \dots & \dots & \times \end{bmatrix}, \text{ com 8 linhas e 9 colunas}$$

$$lm = \begin{bmatrix} 1 & 4 & 5 \\ 1 & 5 & 2 \\ 2 & 5 & 6 \\ \vdots & \vdots & \vdots \\ 5 & 9 & 6 \end{bmatrix}, \text{ com 8 linhas e 3 colunas}$$

$$diagA = [\otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes]$$

São necessárias 72 posições de memória para AA, 24 posições de memória para lm e 9 posições para $diagA$, totalizando 105 posições de memória. Observa-se que neste caso são armazenadas informações redundantes, já que cada coeficiente da matriz esparsa global A recebe contribuição dos elementos adjacentes. Além disso, a diagonal principal é armazenada de forma compacta e por elemento. Para aliviar essa redundância pode-se armazenar a diagonal de forma compacta e as matrizes de elemento sem os termos da diagonal. Portanto, neste caso, o arranjo AA ficaria com 8 linhas e 6 colunas resultando em 81 posições de memória.

Estrutura de Dados Aresta-por-Aresta

Os coeficientes da matriz esparsa A são agora armazenados a nível de cada aresta da malha. Tem-se em cada aresta uma matriz de ordem 2, com 4 coeficientes. Para uma malha com $nedges$ arestas os coeficientes de A são armazenados na matriz de inteiros AA com 4 colunas e $nedges$ linhas. Da mesma forma que no esquema por elemento, define-se uma matriz de localização para as arestas, $lm(nedges, 2)$, contendo o número das equações globais associadas aos nós das arestas. Além disso é necessário um vetor de n posições para armazenar $diagA$. Para o exemplo de malha ilustrado na Figura 4.2, tem-se o esquema de armazenamento:

$$AA = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \vdots & \vdots & \vdots & \vdots \\ \times & \times & \times & \times \end{bmatrix}, \text{ com 16 linhas e 4 colunas}$$

$$lm = \begin{bmatrix} 1 & 4 \\ 4 & 5 \\ 5 & 1 \\ \vdots & \vdots \\ 9 & 6 \end{bmatrix}, \text{ com 16 linhas e 2 colunas}$$

$$diagA = [\otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes \quad \otimes]$$

São necessárias 64 posições de memória para AA, 32 posições de memória para lm e 9 posições para $diagA$, totalizando 105 posições de memória. Da mesma forma que no esquema por elemento pode-se armazenar somente os coeficientes fora da diagonal das matrizes das arestas, evitando-se assim redundância de informação no armazenamento da diagonal. Portanto, para esse exemplo o arranjo AA ficaria com 16 linhas e 2 colunas, resultando em 73 posições de memória. É importante ressaltar que os termos de fora da diagonal das matrizes das arestas correspondem aos coeficientes da matriz global A. Dessa forma, pode-se dizer que o esquema de armazenamento por aresta usa os mesmos arranjos do esquema elemento-por-elemento, porém para um elemento de 2 nós, isto é, uma aresta.

Para esse exemplo simples o esquema de armazenamento por aresta é o mais econômico. Em malhas de grande porte, onde o número de arestas é aproximadamente de 2 a 3 vezes maior que o número de triângulos [23, 24, 26], observa-se, ainda assim, que o esquema por aresta é mais vantajoso.

4.2 Estrutura de Dados por Arestas

A partir das matrizes dos elementos oriundas da formulação estabilizada apresentada no capítulo 3, representadas em (3.32) e (3.34), define-se um desmembramento dos coeficientes gerando as contribuições de cada aresta, que podem ser representadas por:

$$\underbrace{\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}}_{\text{elemento } e} = \underbrace{\begin{bmatrix} \times & \times & \mathbf{0} \\ \times & \times & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\text{aresta } ij} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \times & \times \\ \mathbf{0} & \times & \times \end{bmatrix}}_{\text{aresta } jk} = \underbrace{\begin{bmatrix} \times & \mathbf{0} & \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \times & \mathbf{0} & \times \end{bmatrix}}_{\text{aresta } ki} \tag{4.1}$$

onde $\bullet$, $\times$ e $\mathbf{0}$ representam submatrizes de ordem 4.

Porém, sobre a aresta s incidem informações dos elementos adjacentes a ela, como pode ser observado na Figura 4.3. Assim, todas as contribuições referentes à aresta s, estão presentes nos elementos e e f:

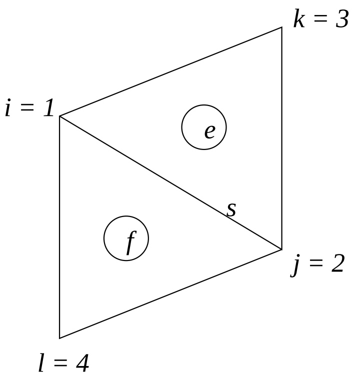


Figura 4.3: Elementos adjacentes a aresta s, formada pelos nós i e j.

$$\underbrace{\begin{bmatrix} \circ & \circ \\ \circ & \circ \end{bmatrix}}_{\text{aresta } s} = \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix}}_{\text{elemento } e} + \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix}}_{\text{elemento } f} \tag{4.2}$$

onde $\circ$ e $\times$ também representam submatrizes de ordem 4.

A estrutura de dados convencional para uma malha triangular linear define a relação entre nós e elementos onde cada elemento e está associado aos nós i, j e k da malha, de acordo com a Figura 3.1. Na estrutura de dados baseada nas arestas cada aresta s está associada aos nós i, j, k e l, conforme a Figura 4.3. A mudança de estrutura de dados é feita considerando que os dados referentes à malha de elementos já existam. As informações geométricas e topológicas da malha são manipuladas de forma a gerar a nova estrutura baseada nas arestas dos elementos. Para tal, é usado uma técnica usual de busca em tabelas conhecida por *Hash Table*, descrita com detalhes para esta aplicação em [33]. Assim, as matrizes definidas em (3.32) e (3.34) , também podem ser escritas por:

$$\begin{aligned} \mathbf{M} &= \mathop{\mathsf{A}}_{s=1}^{nedges} (\mathbf{m}^s) \\ \mathbf{C} &= \mathop{\mathsf{A}}_{s=1}^{nedges} (\mathbf{c}^s) \end{aligned} \tag{4.3}$$

onde *nedges* é o número de arestas da malha. Assim sendo, é claro que a matriz efetiva $\mathbf{M}^*$ também pode ser calculada a partir das contribuições das arestas.

4.3 Matrizes das arestas

A partir das contribuições de Galerkin, correção SUPG e operador de captura de descontinuidade das matrizes dos elementos descritas no capítulo 3, pode-se obter as contribuições das matrizes das arestas. Como visto no esquema descrito em (4.2), cada aresta possui parcelas de contribuições dos elementos adjacentes a ela. A formulação estabilizada baseada nos elementos considera as grandezas $\mathbf{A}_x$, $\mathbf{A}_y$, $\mathbf{A}_0^{-1}$, $\mathbf{T}_x$ e $\mathbf{T}_y$ constantes em cada elemento. Com o objetivo de manter a simplicidade da formulação baseada nas arestas, tais grandezas assumirão valores também constantes em cada aresta. Para isso o vetor de valores nodais em cada aresta é definido por:

$$\mathbf{v}_s^T = \{\underbrace{v_1, v_2, v_3, v_4}_{\text{nó 1}}, \underbrace{v_5, v_6, v_7, v_8}_{\text{nó 2}}\} \tag{4.4}$$

e seu valor no ponto médio da aresta é dado por:

$$\mathbf{v}_m = \left\{ \begin{array}{c} U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \right\} = \frac{1}{2} \left\{ \begin{array}{c} v_1 + v_5 \\ v_2 + v_6 \\ v_3 + v_7 \\ v_4 + v_8 \end{array} \right\} \tag{4.5}$$

De forma similar pode-se definir o valor nodal da derivada temporal $\mathbf{a}$ por:

$$\mathbf{a}_s^T = \{\underbrace{a_1, a_2, a_3, a_4}_{\text{nó 1}}, \underbrace{a_5, a_6, a_7, a_8}_{\text{nó 2}}\} \tag{4.6}$$

e seu valor no ponto médio da aresta é dado por:

$$\mathbf{a}_m = \frac{1}{2} \left\{ \begin{array}{c} a_1 + a_5 \\ a_2 + a_6 \\ a_3 + a_7 \\ a_4 + a_8 \end{array} \right\} \tag{4.7}$$

Portanto, $\mathbf{A}_x = \mathbf{A}_x(\mathbf{v}_m)$, $\mathbf{A}_y = \mathbf{A}_y(\mathbf{v}_m)$, $\mathbf{A}_0^{-1} = \mathbf{A}_0^{-1}(\mathbf{v}_m)$, $\mathbf{T}_x = \mathbf{T}_x(\mathbf{v}_m)$ e $\mathbf{T}_y = \mathbf{T}_y(\mathbf{v}_m)$.

4.3.1 Matriz de massa

Na formulação baseada nos elementos a matriz de massa representada por (3.42), guarda as contribuições do elemento e na matriz da aresta s, conforme a Figura 4.3, e pode ser representada por:

$$(\mathbf{m}_g^s)^e = \frac{A^e}{12}\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \tag{4.8}$$

A matriz de massa total da aresta s é dada por uma soma das contribuições relativas aos elementos e e f:

$$\mathbf{m}_g^s = \frac{(A^e + A^f)}{12}\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \tag{4.9}$$

4.3.2 Correção SUPG da matriz de massa

A correção SUPG da matriz de massa no elemento e, formada pelos nós $i = 1$, $j = 2$ e $k = 3$, é dada por (3.49). Fazendo manipulações algébricas triviais nos seus coeficientes, representados em (3.50), obtem-se:

$$\mathbf{m}_{pg}^e = \frac{\tau^e}{6}\begin{bmatrix} -(\mathbf{m}_{pg}^2 + \mathbf{m}_{pg}^3) & \mathbf{m}_{pg}^1 & \mathbf{m}_{pg}^1 \\ \mathbf{m}_{pg}^2 & -(\mathbf{m}_{pg}^1 + \mathbf{m}_{pg}^3) & \mathbf{m}_{pg}^2 \\ \mathbf{m}_{pg}^3 & \mathbf{m}_{pg}^3 & -(\mathbf{m}_{pg}^1 + \mathbf{m}_{pg}^2) \end{bmatrix} \tag{4.10}$$

Portanto a contribuição do elemento e na matriz da aresta s é dada por:

$$(\mathbf{m}_{pg}^s)^e = \frac{\tau^e}{6}\begin{bmatrix} -\mathbf{m}_{pg}^2 & \mathbf{m}_{pg}^1 \\ \mathbf{m}_{pg}^2 & -\mathbf{m}_{pg}^1 \end{bmatrix} \tag{4.11}$$

O elemento f com nós $j = 2$, $i = 1$ e $l = 4$ possui contribuições similares. Levando-se em consideração as conetividades dos elementos e e f, a matriz total da aresta s é dada por:

$$\mathbf{m}_{pg}^s = \begin{bmatrix} -\mathbf{sm}_{pg}^1 & \mathbf{sm}_{pg}^2 \\ \mathbf{sm}_{pg}^1 & -\mathbf{sm}_{pg}^2 \end{bmatrix} \tag{4.12}$$

onde:

$$\begin{aligned} \mathbf{sm}_{pg}^1 &= \frac{1}{6}\left[\tau^e(y_{31}\mathbf{T}_x + x_{13}\mathbf{T}_y) + \tau^f(y_{14}\mathbf{T}_x + x_{41}\mathbf{T}_y)\right] \\ \mathbf{sm}_{pg}^2 &= \frac{1}{6}\left[\tau^e(y_{23}\mathbf{T}_x + x_{32}\mathbf{T}_y) + \tau^f(y_{42}\mathbf{T}_x + x_{24}\mathbf{T}_y)\right] \end{aligned} \tag{4.13}$$

4.3.3 Matriz de convecção

Fazendo manipulações algébricas triviais nos coeficientes da matriz de convecção do elemento e, dados por (3.57), obtem-se:

$$\mathbf{c}_g^e = \frac{1}{6}\begin{bmatrix} -(\mathbf{c}_g^2+\mathbf{c}_g^3) & \mathbf{c}_g^2 & \mathbf{c}_g^3 \\ \mathbf{c}_g^1 & -(\mathbf{c}_g^1+\mathbf{c}_g^3) & \mathbf{c}_g^3 \\ \mathbf{c}_g^1 & \mathbf{c}_g^2 & -(\mathbf{c}_g^1+\mathbf{c}_g^2) \end{bmatrix} \tag{4.14}$$

Assim a contribuição do elemento e na matriz da aresta s é dada por:

$$(\mathbf{c}_g^s)^e = \frac{1}{6}\begin{bmatrix} -\mathbf{c}_g^2 & \mathbf{c}_g^2 \\ \mathbf{c}_g^1 & -\mathbf{c}_g^1 \end{bmatrix} \tag{4.15}$$

Levando-se em consideração as conectividades dos elementos e e f, a matriz total da aresta s é dada por:

$$\mathbf{c}_{pg}^s = \begin{bmatrix} -\mathbf{sc}_g^1 & \mathbf{sc}_g^1 \\ \mathbf{sc}_g^2 & -\mathbf{sc}_g^2 \end{bmatrix} \tag{4.16}$$

onde:

$$\begin{aligned} \mathbf{sc}_g^1 &= \frac{1}{6}\left[(y_{31}+y_{14})\mathbf{A}_x + (x_{13}+x_{41})\mathbf{A}_y\right] \\ \mathbf{sc}_g^2 &= \frac{1}{6}\left[(y_{23}+y_{42})\mathbf{A}_x + (x_{32}+x_{24})\mathbf{A}_y\right] \end{aligned} \tag{4.17}$$

4.3.4 Correção SUPG da matriz de convecção

A correção SUPG da matriz de convecção, dada pela expressão (3.61) com coeficientes (3.62) para o elemento e, possui as seguintes contribuições na aresta s:

$$(\mathbf{c}_{pg}^s)^e = \frac{\tau^e}{4A^e}\begin{bmatrix} -\mathbf{B}_{12} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & -\mathbf{B}_{21} \end{bmatrix} \tag{4.18}$$

E a matriz total da aresta s, levando-se em conta as contribuições dos elementos e e f, pode ser representada por:

$$\mathbf{c}_{pg}^s = \begin{bmatrix} -\mathbf{sc}_{pg}^1 & \mathbf{sc}_{pg}^1 \\ \mathbf{sc}_{pg}^2 & -\mathbf{sc}_{pg}^2 \end{bmatrix} \tag{4.19}$$

sendo:

$$\begin{aligned} \mathbf{sc}_{pg}^1 &= s_1\mathbf{A}_{xx} + s_2\mathbf{A}_{xy} + s_3\mathbf{A}_{yx} + s_4\mathbf{A}_{yy} \\ \mathbf{sc}_{pg}^2 &= s_1\mathbf{A}_{xx} + s_3\mathbf{A}_{xy} + s_2\mathbf{A}_{yx} + s_4\mathbf{A}_{yy} \end{aligned} \tag{4.20}$$

onde:

$$
\begin{aligned}
s_1 &= \frac{\tau^e}{4A^e} y_{23} y_{31} + \frac{\tau^f}{4A^f} y_{42} y_{14} \\
s_2 &= \frac{\tau^e}{4A^e} y_{23} x_{13} + \frac{\tau^f}{4A^f} y_{42} x_{41} \\
s_3 &= \frac{\tau^e}{4A^e} x_{32} y_{31} + \frac{\tau^f}{4A^f} x_{24} y_{14} \\
s_4 &= \frac{\tau^e}{4A^e} x_{32} x_{13} + \frac{\tau^f}{4A^f} x_{24} x_{41}
\end{aligned}
\tag{4.21}
$$

4.3.5 Matriz de Correção do Operador de Descontinuidade

A matriz com os termos do operador de captura de descontinuidades para o elemento e, representada em (3.65), possui as seguintes contribuições na aresta s:

$$
(\mathbf{c}^s_{cd})^e = \frac{\delta^e}{4A^e} \begin{bmatrix} -\bar{\mathbf{B}}_{12} & \bar{\mathbf{B}}_{12} \\ \bar{\mathbf{B}}_{12} & -\bar{\mathbf{B}}_{12} \end{bmatrix} \tag{4.22}
$$

E a matriz total da aresta s, levando-se em conta as contribuições dos elementos e e f, pode ser representada por:

$$
\mathbf{c}^s_{cd} = \begin{bmatrix} -\mathbf{sc}_{cd} & \mathbf{sc}_{cd} \\ \mathbf{sc}_{cd} & -\mathbf{sc}_{cd} \end{bmatrix} \tag{4.23}
$$

onde:

$$
\mathbf{sc}_{cd} = \left[\frac{\delta^e}{4A^e} (y_{23} y_{31} + x_{32} x_{13}) + \frac{\delta^f}{4A^f} (y_{14} y_{42} + x_{41} x_{24}) \right] \mathbf{I} \tag{4.24}
$$

4.3.6 Resumo das matrizes por aresta

A matriz de massa na aresta s é dada por:

$$
\mathbf{m}^s = \mathbf{m}^s_g + \mathbf{m}^s_{pg} = \frac{A^e}{12} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} + \begin{bmatrix} -\mathbf{sm}^1_{pg} & \mathbf{sm}^2_{pg} \\ \mathbf{sm}^1_{pg} & -\mathbf{sm}^2_{pg} \end{bmatrix} \tag{4.25}
$$

com $\mathbf{sm}^1_{pg}$ e $\mathbf{sm}^2_{pg}$ definidos em (4.13). A matriz $\mathbf{c}^s$ da aresta s é dada por:

$$
\mathbf{c}^s = \mathbf{c}^s_g + \mathbf{c}^s_{pg} + \mathbf{c}^s_{op} = \begin{bmatrix} -\mathbf{sc}^1_g & \mathbf{sc}^1_g \\ \mathbf{sc}^2_g & -\mathbf{sc}^2_g \end{bmatrix} + \begin{bmatrix} -\mathbf{sc}^1_{pg} & \mathbf{sc}^1_{pg} \\ \mathbf{sc}^2_{pg} & -\mathbf{sc}^2_{pg} \end{bmatrix} + \begin{bmatrix} -\mathbf{sc}_{cd} & \mathbf{sc}_{cd} \\ \mathbf{sc}_{cd} & -\mathbf{sc}_{cd} \end{bmatrix} \tag{4.26}
$$

onde $\mathbf{sc}^i_g$, $\mathbf{sc}^i_{pg}$ (para $i = 1, 2$) e $\mathbf{sc}_{cd}$ são respectivamente dados por (4.17), (4.20) e (4.24).

4.4 Resíduo

O resíduo definido pela expressão (3.67), pode ser representado pelas contribuições nos elementos por (3.68) ou pelas contribuições nas arestas por:

$$\mathbf{R} = \mathop{\mathsf{A}}_{s=1}^{nedges} (\mathbf{r}^s) \qquad (4.27)$$
$$\mathbf{r}^s = -\mathbf{m}^s\mathbf{a}^s - \mathbf{c}^s\mathbf{v}^s$$

sendo $\mathbf{m}^s$ e $\mathbf{c}^s$ dadas respectivamente por (4.25) e (4.26). Os vetores $\mathbf{a}^s$ e $\mathbf{v}^s$ podem ser representados por:

$$\mathbf{a}^s = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \quad \mathbf{v}^s = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} \qquad (4.28)$$

onde os índices 1 e 2 representam respectivamente as contribuições dos vetores $\mathbf{a}^s$ e $\mathbf{v}^s$ nos nós 1 e 2 da aresta s. O resíduo $\mathbf{r}^s$ ainda pode ser reescrito pelas contribuições das várias parcelas associadas, ou seja:

$$\mathbf{r}^s = -(\mathbf{r}^s_{mg} + \mathbf{r}^s_{mpg} + \mathbf{r}^s_{cg} + \mathbf{r}^s_{cpg} + \mathbf{r}^s_{cd}) \qquad (4.29)$$

As componentes do resíduo relativas a matriz de massa são dadas por:

$$\mathbf{r}^s_{mg} = \mathbf{m}^s_g\mathbf{a}^s = \frac{A^e + A^f}{12}\begin{bmatrix} \mathbf{a}_1 + \mathbf{a}_1 \\ \mathbf{a}_1 + \mathbf{a}_2 \end{bmatrix} \qquad (4.30)$$

As componentes do resíduo relativas a $\mathbf{m}_{pg}$ podem ser representadas por:

$$\mathbf{r}^s_{mg} = \mathbf{m}^s_{pg}\mathbf{a}^s = \begin{bmatrix} -\mathbf{sm}^1_{pg}\mathbf{a}_1 + \mathbf{sm}^2_{pg}\mathbf{a}_2 \\ \mathbf{sm}^1_{pg}\mathbf{a}_1 - \mathbf{sm}^2_{pg}\mathbf{a}_2 \end{bmatrix} \qquad (4.31)$$

ou ainda pode-se dizer que:

$$\mathbf{r}^s_{mpg} = \begin{bmatrix} \mathbf{r}^1_{mpg} \\ \mathbf{r}^2_{mpg} \end{bmatrix} \qquad (4.32)$$

onde $\mathbf{r}^2_{mpg} = -\mathbf{r}^1_{mpg}$.

As componentes do resíduo relativas a matriz $\mathbf{c}^s$ podem ser representadas por:

$$\mathbf{r}_{cg}^s = \mathbf{c}_g^s \mathbf{v}^s = \begin{bmatrix} \mathbf{sc}_g^1(\mathbf{v}_2 - \mathbf{v}_1) \\ -\mathbf{sc}_g^2(\mathbf{v}_2 - \mathbf{v}_1) \end{bmatrix} \tag{4.33}$$

As componentes do resíduo relativas a correção SUPG da matriz de convecção são dadas por:

$$\mathbf{r}_{cpg}^s = \mathbf{c}_{pg}^s \mathbf{v}^s = \begin{bmatrix} \mathbf{sc}_{pg}^1(\mathbf{v}_2 - \mathbf{v}_1) \\ -\mathbf{sc}_{pg}^2(\mathbf{v}_2 - \mathbf{v}_1) \end{bmatrix} \tag{4.34}$$

As componentes do resíduo relativas ao operador de descontinuidades podem ser representadas por:

$$\mathbf{r}_{cd}^s = \mathbf{c}_{cd}^s \mathbf{v}^s = \begin{bmatrix} \mathbf{sc}_{cd}(\mathbf{v}_2 - \mathbf{v}_1) \\ -\mathbf{sc}_{cd}(\mathbf{v}_2 - \mathbf{v}_1) \end{bmatrix} \tag{4.35}$$

ou ainda pode-se dizer que:

$$\mathbf{r}_{cd}^s = \begin{bmatrix} \mathbf{r}_{cd}^1 \\ \mathbf{r}_{cd}^2 \end{bmatrix} \tag{4.36}$$

onde $\mathbf{r}_{cd}^2 = -\mathbf{r}_{cd}^1$.

4.5 Produto matriz-vetor para as equações de Euler

Em cada iteração do algoritmo GMRES descrito no Apêndice A é necessário executar o produto matriz-vetor esparso. Esta operação pode ser feita usando uma estratégia local, que pode ser executada a nível de cada elemento, denominada elemento-por-elemento, ou a nível de cada aresta, denominada aresta-por-aresta. Assim, tem-se:

$$\mathbf{Ap} = \mathop{\mathbf{A}}_{r=1}^{nn} \mathbf{A}_r \mathbf{p}_r \tag{4.37}$$

onde nn é o número de elementos, ou arestas da malha, $\mathbf{A}_r$ é a matriz efetiva do elemento ou da aresta, descritas respectivamente no capítulo 3 e na seção 4.3, e $\mathbf{p}_r$ são as componentes de $\mathbf{p}$ restritas aos graus de liberdade do elemento ou da aresta.

As matrizes efetivas dos elementos e das arestas para as equações de Euler podem ser representadas esquematicamente por:

$$\mathbf{A}^e = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \tag{4.38}$$

onde $\mathbf{A}_{ij}$, para $i, j = 1, 2, 3$, são submatrizes de ordem 4.

$$\mathbf{A}^s = \begin{bmatrix} \bar{\mathbf{A}}_{11} & \bar{\mathbf{A}}_{12} \\ \bar{\mathbf{A}}_{21} & \bar{\mathbf{A}}_{22} \end{bmatrix} \tag{4.39}$$

onde $\bar{\mathbf{A}}_{ij}$, para $i, j = 1, 2$, são submatrizes de ordem 4.

Observe que cada elemento tem 3 nós e a matriz do elemento possui 3×3 submatrizes de ordem 4. Da mesma forma tem-se que cada aresta possui 2 nós e a matriz da aresta possui 2×2 submatrizes de ordem 4. Outra grandeza definida a partir das matrizes de elementos ou de arestas é o arranjo bloco diagonal nodal **BD**, que guarda as contribuições dos blocos diagonais de cada estrutura local (elemento ou aresta) por nó. Assim, para cada nó o arranjo **BD** possui uma submatriz de ordem 4.

Considerando o armazenamento do arranjo **BD** e das estruturas locais (por elemento e por aresta) é possível otimizar o produto matriz-vetor desconsiderando a redundância de armazenamento dos blocos diagonais de cada estrutura local [37, 38]. Este esquema pode ser entendido como uma extensão do esquema de otimização proposto por GIJZEN [39], que considera apenas a utilização da diagonal principal ao invés dos blocos-diagonais nodais. Sendo assim, o produto matriz-vetor esparso é representado por:

$$\mathbf{Ap} = \mathbf{BD}(\mathbf{A})\mathbf{p} + \mathop{\mathsf{A}}\limits_{r=1}^{nn} (\mathbf{A}_r - \mathbf{BD}(\mathbf{A}_r))\mathbf{p}_r \tag{4.40}$$

Este procedimento diminui o número de operações de ponto flutuante e de endereçamentos indiretos necessários. A implementação da multiplicação matriz-vetor para as estruturas locais, em linhas gerais segue o seguinte algoritmo:

`Para cada estrutura local` r `faça`

    `localizar` $\mathbf{p}_r$ `em` $\mathbf{p}$

    `calcular` $\mathbf{v}_r = (\mathbf{A}_r - \mathbf{BD}(\mathbf{A}_r))\mathbf{p}_r$

    `espalhar + acumular` $\mathbf{v} = \mathbf{v} + \mathbf{v}_r$

`fim do` r

$\mathbf{v} = \mathbf{BD}(\mathbf{A})\mathbf{p} + \mathbf{v}$

É importante notar que este algoritmo apresenta um ótimo potencial de vetorização e paralelização, uma vez que os passos `localizar` e `calcular` podem ser executados independentemente para cada estrutura local. Porém o passo `espalhar + acumular` envolve operação de acesso à escrita de dados globais. Para estruturas locais adjacentes as operações de acesso são executadas em endereços comuns de $\mathbf{v}$. Portanto, operações concorrentes de `espalhar + acumular` são possíveis somente dentro de grupos não adjacentes de elementos ou arestas. Estes grupos são construídos a partir de um pré-processamento de elementos ou arestas por um algoritmo de coloração de malha do tipo guloso [35]. As operações `localizar` e `espalhar` são efetuadas através da matriz de localização $\mathrm{lm}(nn, nd)$, onde nn é o número de elementos ou arestas e nd é o número de graus de liberdade por elementos ou por arestas. Este arranjo define o mapeamento entre os graus de liberdade locais da estrutura local e os graus de liberdade globais do problema [5].

A operação $\mathbf{BD}(\mathbf{A})\mathbf{p}$ é executada pela soma dos valores nodais. Neste caso não há problemas de acessos simultâneos, pois o arranjo $\mathbf{BD}$ guarda em cada nó a soma de todas as contribuições das estruturas locais às quais o nó pertence. Para executá-la são necessárias 28 *flop* (operações de ponto flutuante) e 8 i/a (endereçamentos indiretos) em cada nó da malha.

A seguir, apresenta-se o algoritmo de multiplicação matriz-vetor elemento-por-elemento e aresta-por-aresta.

4.5.1 Multiplicação matriz-vetor elemento-por-elemento

A expressão (4.40), quando a estrutura local é o elemento, pode ser representada por:

$$\mathbf{Ap} = \mathbf{BD}(\mathbf{A})\mathbf{p} + \mathop{\mathbf{A}}_{e=1}^{nel} (\mathbf{A}_e - \mathbf{BD}(\mathbf{A}_e))\mathbf{p}_e \tag{4.41}$$

onde nel é o número de elementos da malha. A operação de multiplicação local pode então ser representada esquematicamente por:

$$(\mathbf{A}_e - \mathbf{BD}(\mathbf{A}_e))\mathbf{p}_e = \begin{bmatrix} \mathbf{O} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{O} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \tag{4.42}$$

Esta operação local é tratada através do seguinte algoritmo:

`para` $e = 1, \ldots, nel$ `faça`

(localizar as componentes globais de $\mathbf{p}^e$)

$\mathbf{neq1} = lm(e, \text{nó } 1)$

$\mathbf{neq2} = lm(e, \text{nó } 2)$

$\mathbf{neq3} = lm(e, \text{nó } 3)$

$\mathbf{p}_1 = \mathbf{p}(\mathbf{neq1})$

$\mathbf{p}_2 = \mathbf{p}(\mathbf{neq2})$

$\mathbf{p}_3 = \mathbf{p}(\mathbf{neq3})$

(multiplicação matriz-vetor elemento-por-elemento)

$\mathbf{ap1} = \mathbf{A}_{12} * \mathbf{p}_2 + \mathbf{A}_{13} * \mathbf{p}_3$

$\mathbf{ap2} = \mathbf{A}_{21} * \mathbf{p}_1 + \mathbf{A}_{23} * \mathbf{p}_3$

$\mathbf{ap3} = \mathbf{A}_{31} * \mathbf{p}_1 + \mathbf{A}_{32} * \mathbf{p}_2$

(espalhar e acumular **ap** global)

$\mathbf{ap}(\mathbf{neq1}) = \mathbf{ap}(\mathbf{neq1}) + \mathbf{ap1}$

$\mathbf{ap}(\mathbf{neq2}) = \mathbf{ap}(\mathbf{neq2}) + \mathbf{ap2}$

$\mathbf{ap}(\mathbf{neq3}) = \mathbf{ap}(\mathbf{neq3}) + \mathbf{ap3}$

`fim do` e

onde $\mathbf{neq}i$, para $i = 1, 2, 3$ é um arranjo de 4 inteiros contendo a posição no vetor global $\mathbf{p}$ das componentes locais $\mathbf{p}i$, que contém as 4 incógnitas do nó i. Observa-se que em cada elemento são necessários 192 *flop* e 36 i/a.

4.5.2 Multiplicação matriz-vetor aresta-por-aresta

A expressão (4.40), quando a estrutura local é a aresta, pode ser representada por:

$$\mathbf{Ap} = \mathbf{BD}(\mathbf{A})\mathbf{p} + \mathop{\mathbf{A}}_{s=1}^{nedges} (\mathbf{A}_s - \mathbf{BD}(\mathbf{A}_s))\mathbf{p}_s \tag{4.43}$$

onde $nedges$ é o número de arestas da malha. A operação de multiplicação local pode então ser representada esquematicamente por:

$$(\mathbf{A}_s - \mathbf{BD}(\mathbf{A}_s))\mathbf{p}_s = \begin{bmatrix} \mathbf{O} & \bar{\mathbf{A}}_{12} \\ \bar{\mathbf{A}}_{21} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} \tag{4.44}$$

Esta operação local é tratada através do seguinte algoritmo:

`para` $s = 1, \ldots, nedges$ `faça`

> (localizar as componentes globais de $\mathbf{p}^s$)
>
> $\mathbf{neq1} = lm(e, \text{nó } 1)$
>
> $\mathbf{neq2} = lm(e, \text{nó } 2)$
>
> $\mathbf{p}_1 = \mathbf{p}(\mathbf{neq1})$
>
> $\mathbf{p}_2 = \mathbf{p}(\mathbf{neq2})$
>
> (multiplicação matriz-vetor elemento-por-elemento)
>
> $\mathbf{ap1} = \bar{\mathbf{A}}_{12} * \mathbf{p}_2$
>
> $\mathbf{ap2} = \bar{\mathbf{A}}_{21} * \mathbf{p}_1$
>
> (espalhar e acumular **ap** global)
>
> $\mathbf{ap}(\mathbf{neq1}) = \mathbf{ap}(\mathbf{neq1}) + \mathbf{ap1}$
>
> $\mathbf{ap}(\mathbf{neq2}) = \mathbf{ap}(\mathbf{neq2}) + \mathbf{ap2}$

`fim do` s

Da mesma forma tem-se que $\mathbf{neq}i$, para $i = 1, 2$, é um arranjo de 4 inteiros. Observa-se que em cada aresta são necessárias 64 *flop* e 24 i/a.

4.5.3 Custos Computationais do Produto Matriz-Vetor Esparso

Devido a importância do produto matriz-vetor no processo de solução implícita das equações de Euler é fundamental escolher a melhor forma para realizá-la. Nas seções 4.5.1 e 4.5.2 foram apresentados respectivamente algoritmos para o produto matriz-vetor utilizando as estratégias elemento-por-elemento e aresta-por-aresta. Além dessas estratégias pode-se citar a operação matriz-vetor considerando o armazenamento da matriz esparsa gerada pelos métodos CSR ou MSR, tratados na seção 4.2. Segundo SAAD [35], para este tipo de armazenamento existe um fator prejudicial na vetorização e paralelização, já que o comprimento efetivo da operação de multiplicação de vetores `ddot` corresponde ao número de coeficientes não nulos das linhas, o que é muito pequeno. PINAR [38] propõe estratégias para minimizar esse problema, porém as estratégias locais ainda possuem vetores bem mais longos, melhorando o desempenho de vetorização e paralelização.

Considerando-se as relações de Euler entre número de vértices (nós), faces (elementos) e arestas em uma triangulação no plano, dados por exemplo por CAREY [26] ou LOHNER [29], pode-se assumir que $nel \approx 2 \times nnos$ e que $nedges \approx 1.5 \times nel \approx 3 \times nnos$. Nessas estimativas supos-se que o número de nós na fronteira e desprezível frente ao número de nós no interior do domínio. Sendo assim, a Tabela 4.1 compara a demanda de memória para se armazenar a matriz esparsa, o número de operações de ponto flutuante ($flop$) e o número de endereçamentos indiretos (i/a) necessários para se efetuar o produto matriz-vetor esparso para as estratégias elemento-por-elemento e aresta-por-aresta do item 4.5. Note que os dados nesta tabela consideram a operação completa conforme as equções (4.42) e (4.43).

Tabela 4.1: Comparação de custos computacionais para o produto matriz-vetor

Estrutura de dados	Memória	*flop*	i/a
Elemento	$208 * nnos$	$412 * nnos$	$84 * nnos$
Aresta	$112 * nnos$	$220 * nnos$	$84 * nnos$

Observa-se nesta tabela que a utilização do esquema de armazenamento dos coeficientes da matriz esparsa por arestas reduz a demanda de memória e o número de operações de ponto flutuante para o cálculo do produto matriz-vetor esparso em praticamente 50%, mantendo-se constante o número de operações de endereçamento indireto. Estas estimativas favorecem bastante o esquema por arestas. Com respeito aos esquemas esparsos gerais, deve-se lembrar que o esquema por aresta, tal como apresentado, armazena somente os termos não nulos da matriz esparsa $\mathbf{A}$, como os esquemas CSR e MSR. Porém, o esquema por aresta necessita da matriz de localização das arestas, o arranjo $lm(nedges, 2)$, o que significa um arranjo de inteiros da ordem de $6 \times nnos$. Já o esquema CSR, por exemplo, necessita de dois vetores, IA e JA, respectivamente de comprimento nnz e $(n + 1)$. Considerando que o número de termos não nulos é da ordem do número de coeficientes a serem armazenados no esquema por aresta listado na Tabela 4.1, isto é, $112 \times nnos$, e que $n \approx 4 \times nnos$, pode-se argumentar que o esquema por aresta implica em uma utilização de ponteiros bem inferior aos esquemas esparsos gerais.

Recentemene LOHNER [40, 41] e LOHNER et al. [42] evidenciaram que para se atingir um desempenho próximo do ideal em máquinas paralelas com memória compartilhada escalável, técnicas de renumeração especiais dos nós e arestas se fazem necessárias. Entre as técnicas de renumeração nodal que procuram minimizar o acesso à memória *cache*, aumentando a localidade dos dados, LOHNER [41] sugere a utilização de algoritmos de minimização de banda. Em todos os exemplos deste capítulo que envolvem malhas não-estruturadas, apresentados adiante, os pontos nodais foram reordenados pelo algoritmo *Reverse Cuthill-McKee* (RCM) [26]. Não foi feito nenhum esforço de se otimizar a contenção de memória ou os efeitos de sobreescrita na memória *cache* utilizando-se das técnicas sugeridas por LOHNER et al. [42].

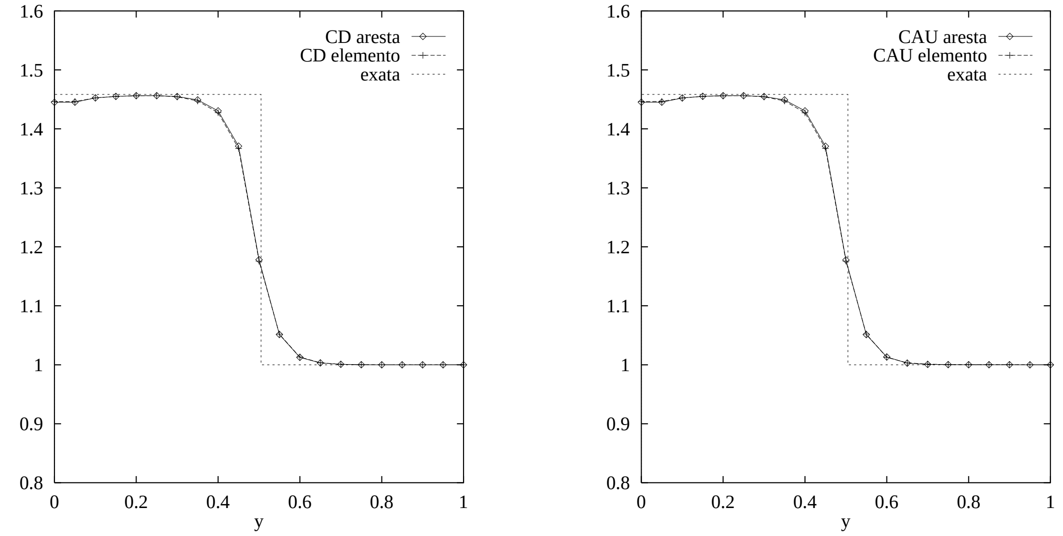
4.6 Exemplos de validação da implementação por aresta

São considerados dois exemplos, já estudados no capítulo 3: Choque Oblíquo Bidimensional e Choque Refletido Bidimensional. Os resultados são comparados com as soluções usando a formulação por elemento. O algoritmo de avanço no tempo é o mesmo descrito na seção 3.5, sendo o número de multicorreções também fixado em 3. Em todos os casos foi considerada a correção SUPG e o operador de captura de descontinuidade. Assim, a notação CD indica que foi usada uma formulação do tipo G+SUPG+CD, e da mesma forma uma notação CAU indica que foi adotada uma formulação do tipo G+SUPG+CAU. Em todos os casos obtem-se a solução considerando 5 vetores de Krylov, tolerância igual a 0.1 e 1000 passos. É importante observar que, em geral, a solução já está convergida após 200 passos. Detalhes serão tratados no próximo capítulo.

4.6.1 Choque Oblíquo Bidimensional

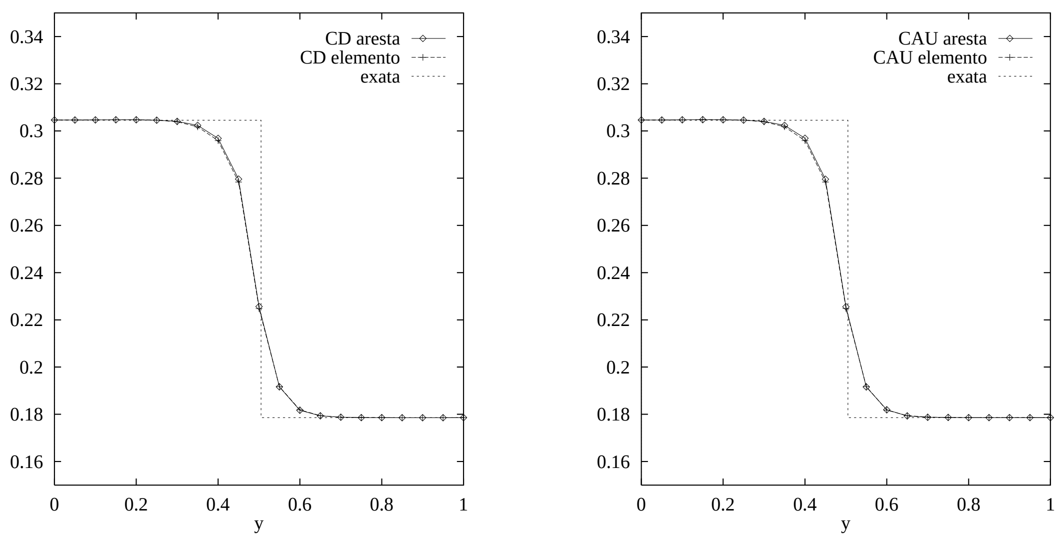
As Figuras 4.4, 4.5 e 4.6 mostram, respectivamente, comparações de perfis de densidade, pressão e temperatura entre as implementações por elemento e aresta para a malha 20×20, com 2 elementos em cada célula. Observa-se que as soluções por aresta são praticamente iguais às soluções por elemento. A Figura 4.7 mostra perfis e isocurvas de densidade para a malha com 80×80 células, considerando dois elementos em cada célula. Observa-se, portanto, que a solução para uma malha mais refinada também apresenta ótimos resultados.

A Figura 4.8 apresenta o comportamento do resíduo da densidade com relação



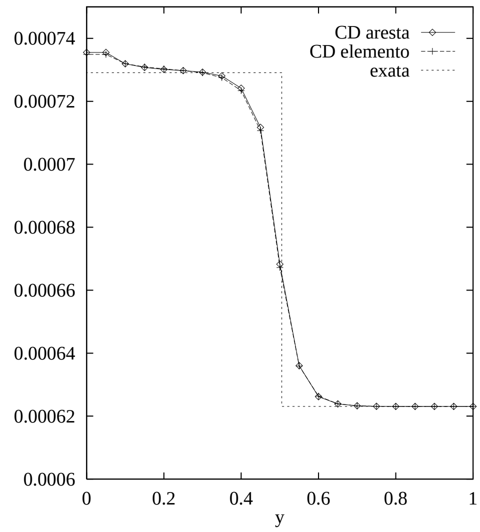
(a) Densidade com operador CD (b) Densidade com operador CAU

Figura 4.4: Choque Oblíquo (2D) - Malha 20×20 com 800 elementos, 441 nós e 1240 arestas - Comparação de Perfis de Densidade para $x = 0.9$ para as implementações por elemento e aresta.

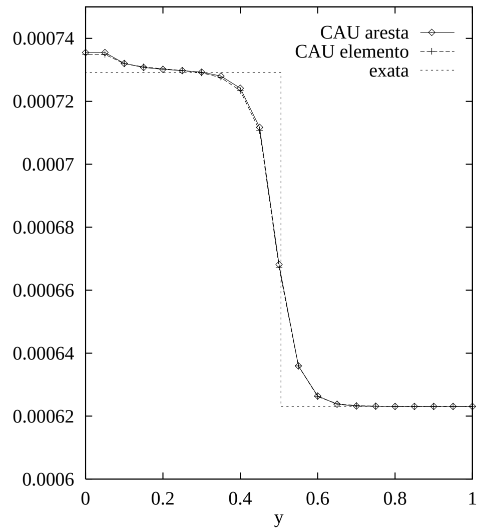


(a) Pressão com operador CD (b) Pressão com operador CAU

Figura 4.5: Choque Oblíquo (2D) - Malha 20×20 com 800 elementos, 441 nós e 1240 arestas - Comparação de Perfis de Pressão para $x = 0.9$ para as implementações por elemento e aresta.

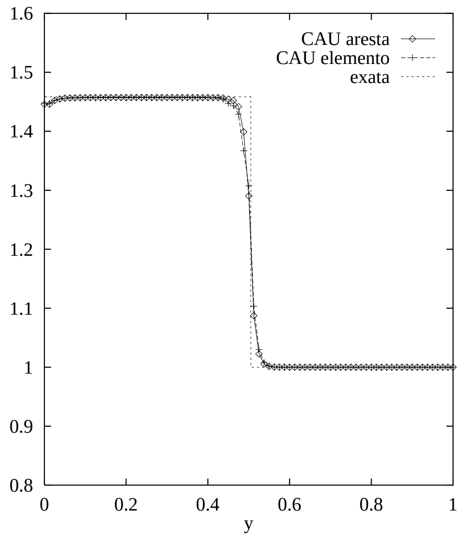


(a) Temperatura com operador CD

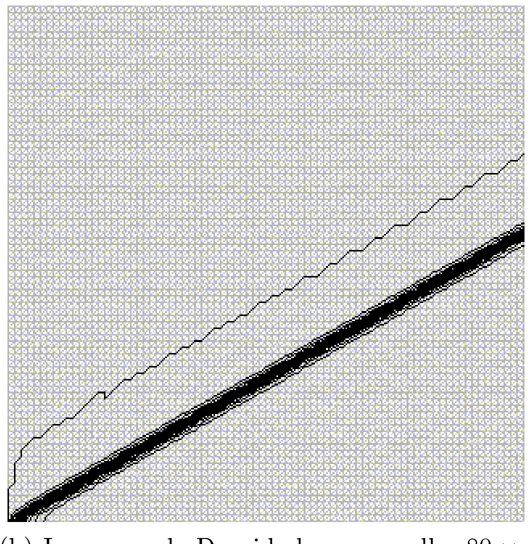


(b) Temperatura com operador CAU

Figura 4.6: Choque Oblíquo (2D) - Malha 20×20 com 800 elementos, 441 nós e 1240 arestas - Comparação de Perfis de Temperatura para $x = 0.9$ para as implementações por elemento e aresta.



(a) Perfis de Densidade para malha 80×80



(b) Isocurvas de Densidade para malha 80×80

Figura 4.7: Choque Oblíquo (2D) - Malha 80×80 com 12800 elementos, 6561 nós e 19360 arestas - Densidade com operador CAU - Formulação por aresta.

ao tempo de CPU, executado na máquina SGI O2, para a malha 80×80 com 12800 elementos, 6561 nós e 19360 arestas, usando o operador CAU. A implementação por aresta é 1.73 vezes mais rápida que a implementação por elemento. Observe que a solução por elemento estagna para um resíduo de densidade um pouco menor que 1×10^{-3}, com um tempo de CPU em torno de 0.3, enquanto que a solução por aresta atinge resíduos de densidade próximos a 1×10^{-4}, com um tempo de CPU menor, em torno de 0.2. Com relação a memória necessária para armazenamento dos coeficientes da matriz efetiva, no esquema por aresta são necessários 2 blocos de 16 componentes para cada aresta para armazenar os coeficientes da matriz efetiva fora dos blocos diagonais e 1 bloco de 16 componentes em cada nó da malha para armazenar os blocos diagonais, totalizando, neste exemplo, 724,496 posições de memória. Em contrapartida, no esquema por elemento são necessários 6 blocos de 16 componentes em cada elemento para armazenar os coeficientes de fora dos blocos diagonais da matriz efetiva, além de 1 bloco de 16 componentes para armazenar os blocos diagonais em cada nó da malha, totalizando, neste exemplo, 1,333,776 posições de memória. Observa-se, portanto, que a memória necessária no esquema por aresta é igual a 54.32% daquela necessária para o esquema por elemento, fato já estimado na Tabela 4.1 para casos gerais.

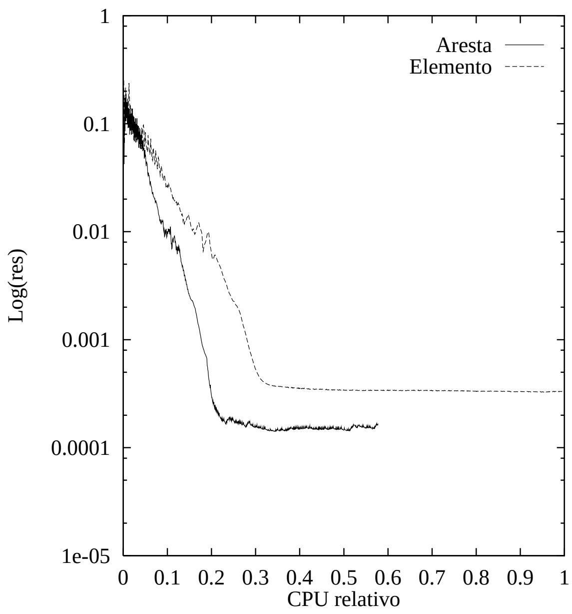


Figura 4.8: Choque Oblíquo (2D) - Malha com com 12800 elementos, 6561 nós e 19360 arestas - 1000 passos do resíduo da densidade com operador CAU para as formulações com aresta e elemento.

O tempo total de execução do processo de solução é praticamente todo preenchido pelas rotinas de montagem das matrizes efetivas locais (`tri2d`), montagem do vetor de resíduos (`resid`) e solução do sistema de equações (`gmres`). A Tabela 4.2 mostra as porcentagens do tempo gasto em cada rotina citada para as estratégias por elemento e por aresta. Observa-se que o tempo gasto na solução do sistema pela estratégia aresta é praticamente a metade daquele gasto no esquema por elemento. Isso se deve ao fato de que o produto matriz-vetor por aresta necessita de menos *flop* que o produto por elemento. Na implementação por arestas o cálculo dos coeficientes da matriz efetiva e do vetor de resíduo dominam os custos computacionais, já que a tolerância do método GMRES é de apenas 0.1 e o produto matriz-vetor está bastante otimizado. O tempo necessário para se calcular os coeficientes da matriz efetiva e as componentes do resíduo é relativamente maior para a implementação por aresta quando comparado com a estratégia elemento. Isto acontece porque em cada aresta os parâmetros de establização da formulação SUPG e dos operadores de captura de choque são calculados para cada elemento adjacente à aresta.

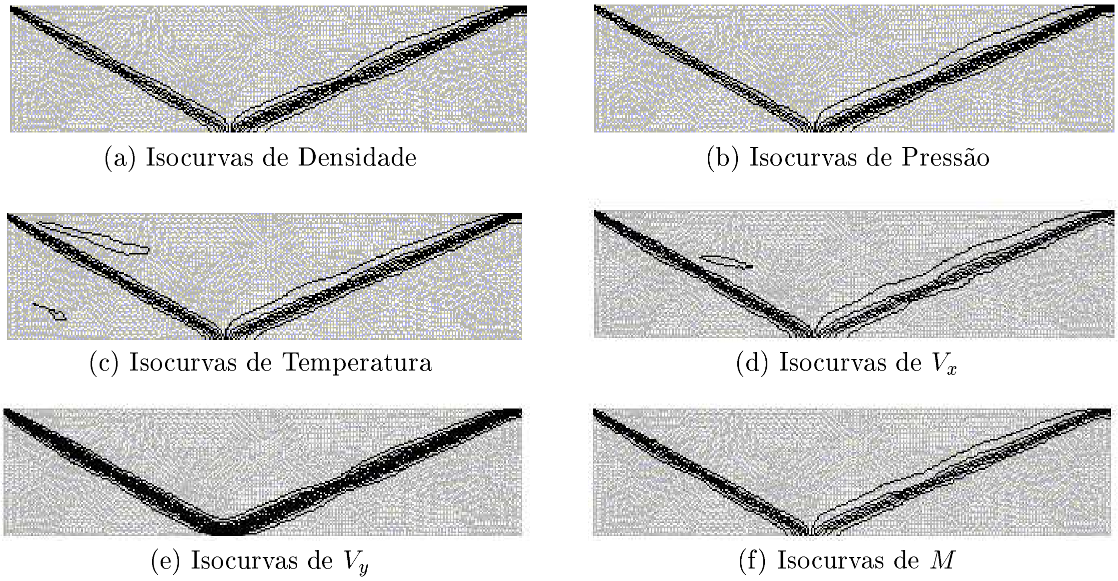
Tabela 4.2: Porcentagens do tempo gasto das rotinas `tri2d`, `resid`, `gmres` para o choque oblíquo bidimensional.

Rotina	elemento	aresta
`tri2d`	33.39%	43.50%
`resid`	6.64%	21.25%
`gmres`	58.55%	32.81%

4.6.2 Choque Refletido Bidimensional

A Figura 4.9 mostra as isocurvas de densidade, pressão, temperatura, velocidade e número de Mach para uma malha não-estruturada com 5440 elementos, 2829 nós e 8268 arestas. Já a Figura 4.10 mostra as mesmas isocurvas para a malha não-estruturada com 3594 elementos, 1837 nós e 5430 arestas, apresentando um refinamento na região do choque. Esta malha foi utilizada no Capítulo 3 na validação da formulação por elemento. Observa-se um ótimo comportamento para as duas malhas, sendo os resultados semelhantes àqueles obtidos com o esquema por elemento.

As Figura 4.11 e 4.12 apresentam o comportamento do resíduo da densidade com relação ao tempo de CPU, também executado na máquina SGI O2, para a malha não-estruturada com 5440 elementos, 2829 nós e 8268 arestas, respectivamente para os operadores CD e CAU. A implementação por aresta é cerca de 1.4 vezes mais rápida que a implementação por elemento para os dois operadores. Observe que para o operador CD a solução por elemento estagna para um resíduo de densidade



(a) Isocurvas de Densidade

(b) Isocurvas de Pressão

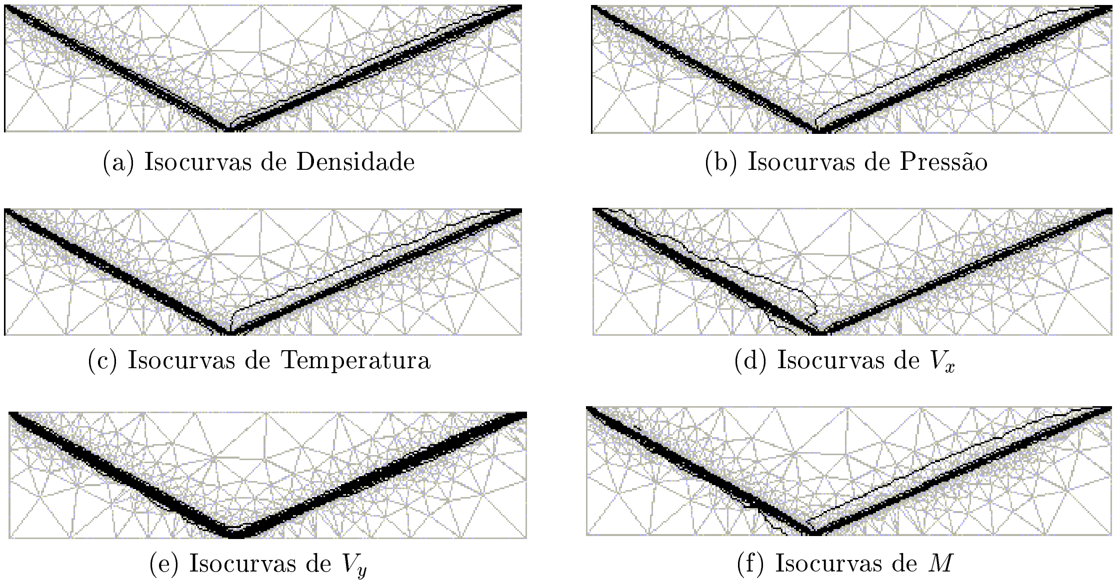
(c) Isocurvas de Temperatura

(d) Isocurvas de V_x

(e) Isocurvas de V_y

(f) Isocurvas de M

Figura 4.9: Choque Refletido (2D) - Isocurvas do Choque Refletido com operador CD - Malha não-estruturada com 5440 elementos, 2829 nós e 8268 arestas - Formulação por arestas.



(a) Isocurvas de Densidade

(b) Isocurvas de Pressão

(c) Isocurvas de Temperatura

(d) Isocurvas de V_x

(e) Isocurvas de V_y

(f) Isocurvas de M

Figura 4.10: Choque Refletido (2D) - Isocurvas do Choque Refletido com operador CD - Malha adaptada com 3594 elementos, 1837 nós e 5430 arestas - Formulação por arestas.

superior a 1×10^{-4}, com um tempo de CPU em torno de 0.3, enquanto que a solução por aresta atinge resíduos de densidade próximos a 1×10^{-5}, com um tempo de CPU menor. Para o operador CAU o comportamento é semelhante, porém os resíduos de densidade estagnam em ordem de grandeza maior. A memória necessária para armazenar todos os coeficientes da matriz efetiva para a estratégia por elemento é de 567,504 posições, e para a estratégia por aresta é de 309,840 posições, o que representa uma economia de 54.59%, mais uma vez confirmando as estimativas mostradas na Tabela 4.1.

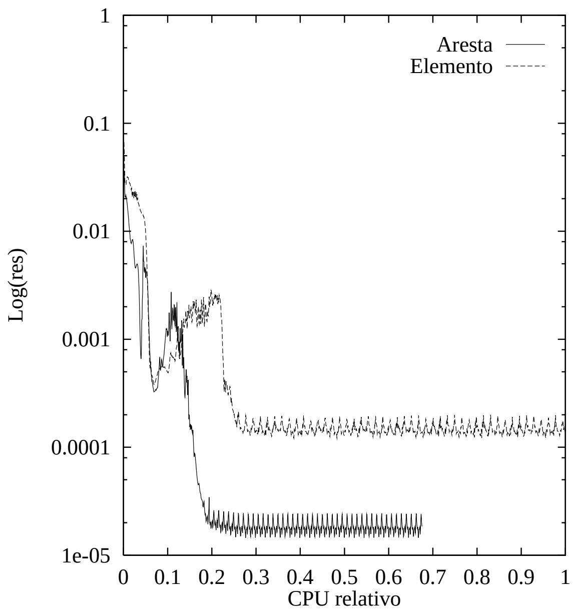


Figura 4.11: Choque Refletido (2D) - Malha não estruturada com 5440 elementos, 2829 nós e 8268 arestas - 1000 passos do resíduo da densidade com operador CD para as formulações com aresta e elemento.

A Tabela 4.3 mostra as porcentagens do tempo de execução das rotinas que dominam o tempo de CPU para as estratégias por elemento e por aresta. As mesmas observações feitas para o choque oblíquo cabem para este caso.

Tabela 4.3: Porcentagens do tempo gasto das rotinas `tria2d`, `resid`, `gmres` para o choque refletido bidimensional.

Rotina	elemento	aresta
`tri2d`	35.04%	43.10%
`resid`	7.88%	20.73%
`gmres`	55.85%	34.41%

As Figuras 4.13 e 4.14 apresentam o comportamento do resíduo com relação ao tempo de CPU para a malha não-estruturada com 3594 elementos, 1837 nós

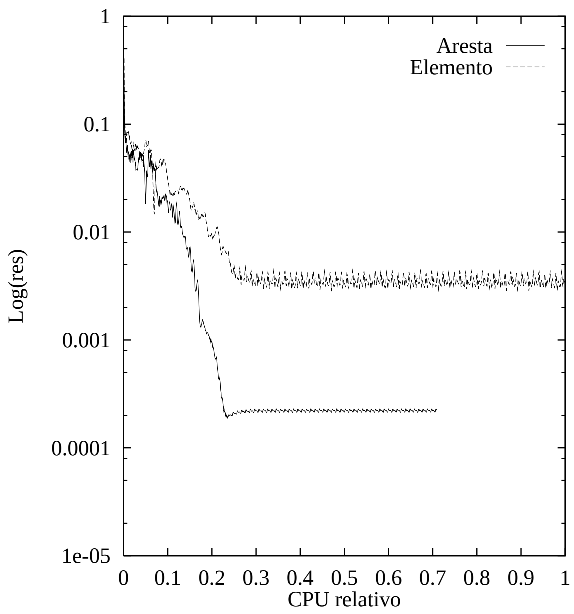


Figura 4.12: Choque Refletido (2D) - Malha não estruturada com 5440 elementos, 2829 nós e 8268 arestas - 1000 passos do resíduo da densidade com operador CAU para as formulações com aresta e elemento.

e 5430 arestas, respectivamente para os operadores CD e CAU. A implementação por aresta também é aproximadamente 30% mais rápida que a implementação por elemento. Para o operador CD também tem-se que o resíduo da densidade para a implementação por aresta é inferior o da implementação por elemento. Para o operador CAU os resíduos estagnam na mesma ordem de grandeza, porém a implementação por aresta necessita de um tempo de CPU menor. A memória necessária para armazenar todos os coeficientes da matriz efetiva para a estratégia por elemento é de 374,416 posições, e para a estratégia por aresta é de 203,152 posições, o que representa novamente uma economia de cerca de 54%.

4.6.3 Análise de desempenho vetorial e paralelo

Neste item é analisado o desempenho vetorial e paralelo da implementação por aresta. O exemplo escolhido foi o choque oblíquo com a malha 80×80, pois é o exemplo que apresenta o maior número de arestas. Os testes foram realizados no CRAY J90, sendo usado as ferramentas disponíveis para medida de desempenho. Para uma CPU utiliza-se o *hardware performance monitor (hpm)* e o *profiling tool (perfview)*, e para 8 CPU's utiliza-se a ferramenta *atexpert*. Os parâmetros básicos para as medidas de desempenho nesta máquina são o número de operações de ponto flutuante por segundo (*flop/s*) e o ganho em tempo de CPU (*speed-up*).

O tempo total de execução com uma CPU foi de aproximadamente 1 hora. A

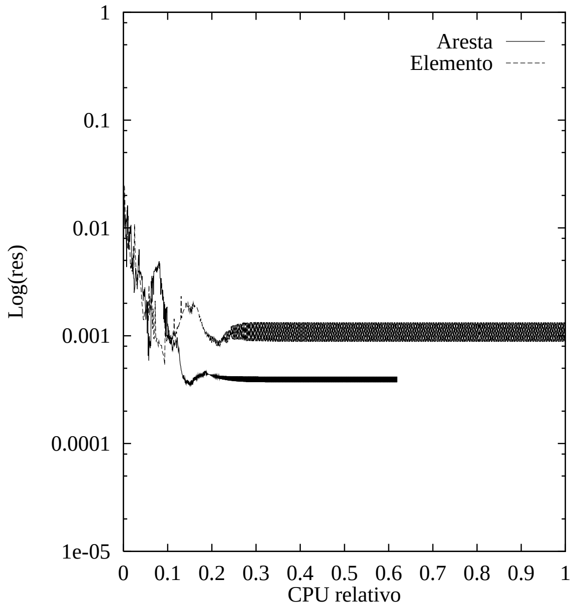


Figura 4.13: Choque Refletido (2D) - Malha adaptada com 3594 elementos, 1837 nós e 5430 arestas - 1000 passos do resíduo da densidade com operador CD para as formulações com aresta e elemento.

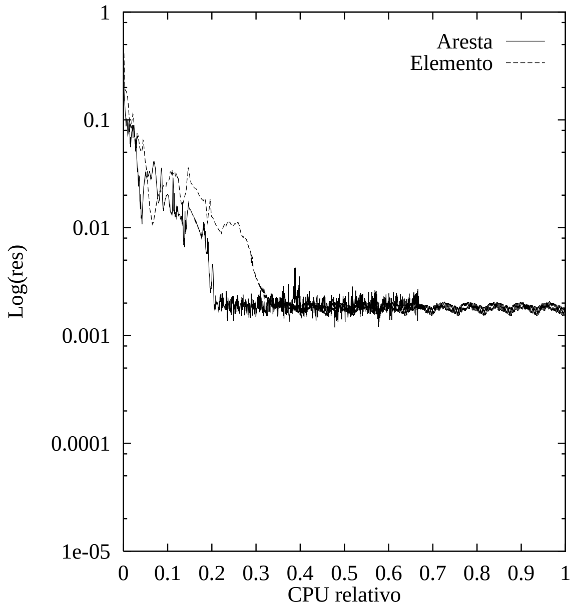


Figura 4.14: Choque Refletido (2D) - Malha adaptada com 3594 elementos, 1837 nós e 5430 arestas - 1000 passos do resíduo da densidade com operador CAU para as formulações com aresta e elemento.

Tabela 4.4 mostra o tempo relativo de CPU e a taxa de operações de ponto flutuante (*flop/s*) nas rotinas que dominam o tempo de execução, sendo `tri2d` a rotina que monta a matriz efetiva, `resid` a rotina que monta o vetor de resíduo, `matvec` a rotina que calcula o produto matriz-vetor e `sblkd` a rotina que executa as operações relativas à ação do precondicionador bloco diagonal nodal.

Tabela 4.4: Desempenho vetorial das principais rotinas da implementação por aresta - Choque Oblíquo - Malha 80×80.

Rotina	tempo (%)	*Mflop/s*
`tri2d`	50.7	97.0
`resid`	41.8	107.6
`matvec`	3.4	105.9
`sblkd`	1.1	129.0

Observa-se um tempo muito pequeno para as principais rotinas de solução do sistema linear (`matvec` e `sblkd`) quando comparado com o tempo das rotinas responsáveis pela montagem do sistema (`tri2d` e `resid`). Este fato já é esperado, pois, além da rotina `matvec` estar bastante otimizada, a tolerância do método GMRES é de apenas 0.1, necessitando de um número muito pequeno de iterações para a convergência. As taxas de *Mflop/s* alcançadas pelas principais rotinas situam-se em torno ou acima de 50% da performance de pico da máquina (200 *Mflop/s*). Comparando estas taxas com dados conhecidos, verifica-se que o desempenho alcançado pelo *benchmark* da NASA CG-Classe A [43] no CRAY J90 é de 95 *Mflop/s*. Este *benchmark* emprega o método dos gradientes conjugados, e foi projetado para testar cálculos típicos de problemas em malhas não-estruturadas. O desempenho do *benchmark* é dominado pelo produto matriz-vetor esparso. Desta forma, observa-se que o desempenho da rotina para o produto matriz-vetor do presente trabalho é um pouco superior ao do *benchmark*. A Figura 4.15 mostra o *speed-up* paralelo, considerando a execução com 8 CPU's. Neste exemplo o algoritmo de coloração de malha definiu 6 blocos de cores. Note o ótimo desempenho paralelo das principais rotinas do código. Além disso, observa-se que o programa como um todo (`job`) apresenta um desempenho paralelo muito bom, quando comparado ao alcançado pelo *benchmark* da NASA CG, que é de 4.57 em 8 CPU's.

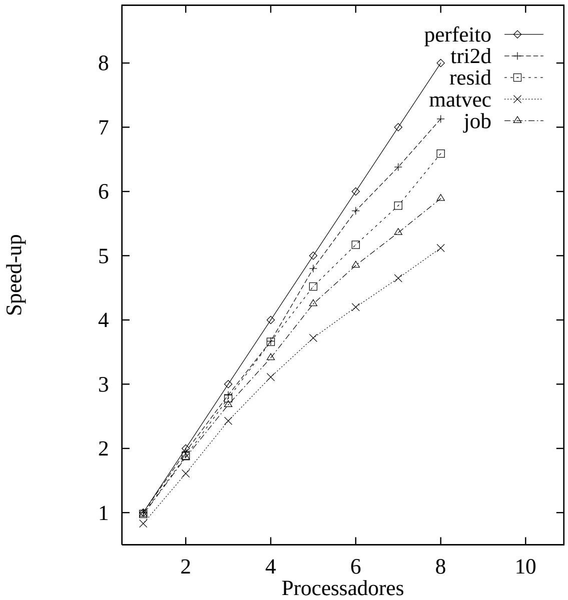


Figura 4.15: *Speed-up* Paralelo no CRAY J90 - Choque Oblíquo (2D) - Malha 80×80.

Capítulo 5

Técnicas de Aceleração de Convergência na Solução da Equação de Euler

Neste capítulo são apresentadas estratégias que visam acelerar o processo de solução da formulação descrita nos capítulos 3 e 4 para a equação de Euler bidimensional.

5.1 Estratégia de passo local no avanço do tempo

Estratégias de escolha do passo adequado nos algoritmos de avanço no tempo são um fator importante para a economia de tempo de processamento. Vários trabalhos enfatizando técnicas de escolha do passo tem sido desenvolvidos. Baseado na teoria de controle, COUTINHO e ALVES [44] implementaram uma escolha automática do passo para algoritmos transientes de escoamentos miscíveis, aumentando a robustez do código. Usando variáveis de entropia na solução das equações de Navier-Stokes, JOHAN, HUGHES e SHAKIB [45] apresentam uma heurística automática no algoritmo de controle do passo usando relações simples entre as variáveis físicas do problema, o último passo calculado e um passo máximo especificado. Os resultados demonstram que tais heurísticas reduzem os custos computacionais.

Outro tipo de estratégia, que também vem sendo usada com sucesso, é definir tamanhos diferentes de passos em regiões distintas do domínio. O tempo de CPU diminui bastante quando [20] define passos de tempo distintos para cada nó da malha a partir das condições de estabilidade do algoritmo de avanço no tempo, usando uma formulação espaço-tempo GLS (*Galerkin Least Squares*) com variáveis de entropia. Considerando uma formulação SUPG com variáveis conservativas LE BEAU e TEZDUYAR [46] usam um algoritmo de integração do tipo preditor multicorretor que emprega passos de tempo diferentes em regiões distintas do domínio. São considerados grupos de elementos que possuem passos de tempo apropriados com procedimentos de interação entre esses grupos. DE SAMPAIO e MOREIRA [47] apresentam uma formulação unificada para escoamentos compressíveis e fracamente incompressíveis usando uma estratégia de passo local originalmente desenvolvida para problemas incompressíveis em regime transiente com refinamento de malha. No caso das equações de Euler, a convergência para o regime permanente pode ser acelerada avançando-se no tempo cada nó da malha de acordo com um passo de tempo local. Para uma formulação explícita com volumes finitos, considerando malhas não estruturadas, MAVRIPLIS e JAMESON [24] definem um passo de tempo local que depende das propriedades difusivas e convectivas dos diversos volumes de controle ao redor dos pontos da malha. A partir dessas idéias este trabalho apresenta uma estratégia de passo local, definindo para cada nó da malha um passo de tempo diferente a partir de propriedades dos elementos (ou arestas) que concorrem neste nó. Em geral, nos algoritmos de avanço no tempo cada passo Δt é considerado fixo em toda a análise e em todo o domínio. Para se obter a solução estacionária em algoritmos transientes o passo Δt deve ser escolhido de forma a satisfazer certas condições de estabilidade e precisão. Em geral, pode-se dizer que o passo de tempo ideal pode ser expresso em função de,

$$\Delta t = g(CFL, h, \mathbf{u}) \tag{5.1}$$

onde CFL é o número de *Courant-Friedrichs-Lewy*, h é o parâmetro de malha que determina o tamanho do elemento e $\mathbf{u}$ é a velocidade do escoamento. Em algoritmos de integração explícitos a condição de estabilidade implica que $CFL < 1$. Já com algoritmos implícitos pode-se obter soluções suficientemente precisas com $CFL > 1$. Usualmente impõe-se uma condição de CFL para a malha, o que implica em se considerar um passo fixo, isto é,

$$\Delta t = min\{\Delta t_r\}, \quad r = 1, 2, \ldots, nel \;\; (\text{ou } nedges) \tag{5.2}$$

Contudo, observa-se que h e $\mathbf{u}$ são componentes locais, ou seja, para cada elemento da malha tem-se valores distintos. Assim, para um determinado valor fixo de Δt, em algumas partes do domínio essa relação poderia ser aproximadamente satisfeita, e no restante o valor de Δt poderia ser muito maior que o necessário. Este fato conduz a uma convergência lenta para o regime permanente e um consequente aumento dos custos computacionais. Alternativamente, pode-se determinar localmente o tamanho do passo mantendo condições ideais de estabilidade em todo o domínio. Considerando as condições de estabilidade no interior de cada elemento, a partir da expressão (3.10), tem-se:

$$\Delta t_e = CFL \frac{h_e}{(c_e + |\mathbf{u}_e \boldsymbol{\beta}_e|)} \tag{5.3}$$

sendo CFL constante para toda malha, h_e o parâmetro que mede o tamanho do elemento e, c_e a velocidade acústica, $\mathbf{u}_e$ é o vetor velocidade calculado no baricentro do elemento e e $\boldsymbol{\beta}_e$ é um vetor unitário arbitrário descrito em (3.11). Portanto, tem-se que o vetor passos de tempo local para o passo corrente no algoritmo de avanço no tempo pode ser definido por:

$$\Delta t(no_i) = min_{e \in \mathcal{A}}\{dt_e\} \quad \text{para } i = 1, 2, \ldots, nnos \tag{5.4}$$

onde $\mathcal{A}$ é o conjunto de todos os elementos que contém o nó no_i. Essa defінção é válida tanto para a estrutura de dados por elemento quanto para a estrutura de dados por aresta.

No algoritmo preditor multicorretor de avanço no tempo descrito na seção 3.5, para cada tempo t_n é calculado o vetor de resíduos $\mathbf{R}$ e a matriz efetiva $\mathbf{M}^*$, dados respectivamente por (3.67) e (3.90). Considerando o cálculo local de Δt é possível reescrever a matriz efetiva, considerando a estratégia por elemento esquematicamente por:

$$\begin{aligned} \mathbf{M}^* &= [\mathbf{m}^*_{ij}] \\ \mathbf{m}^*_{ij} &= \mathop{\mathsf{A}}_{e=1}^{nel} (\mathbf{m}^e_{ij} + 0.5\Delta t(noi)\mathbf{c}^e_{ij}) \end{aligned} \tag{5.5}$$

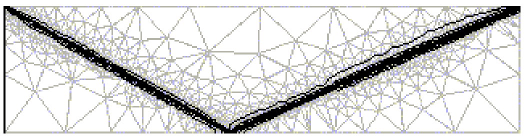
para $i, j = 1, 2, 3$. $\mathbf{m}^e_{ij}$ e $\mathbf{c}^e_{ij}$ são submatrizes de ordem 4 e foram definidas na seção 3.3. No algoritmo de avanço no tempo foi considerado a regra trapezoidal ($\alpha = 0.5$). Na estrutura de dados por aresta a matriz efetiva $\mathbf{M}^*$ é representada de forma similar, podendo ser escrita esquematicamente por:

$$\begin{aligned} \mathbf{M}^* &= [\mathbf{m}^*_{ij}] \\ \mathbf{m}^*_{ij} &= \mathop{\mathsf{A}}_{s=1}^{nedges} (\mathbf{m}^s_{ij} + 0.5\Delta t(noi)\mathbf{c}^s_{ij}) \end{aligned} \tag{5.6}$$

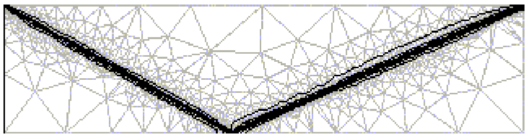
para $i, j = 1, 2$. $\mathbf{m}_{ij}^s$ e $\mathbf{c}_{ij}^s$ são submatrizes de ordem 4 e foram definidas na seção 4.3. Note que a implementação usando o passo local necessita somente de um arranjo a mais para armazenar os passos de tempos locais em cada nó da malha.

5.1.1 Experimentos Numéricos

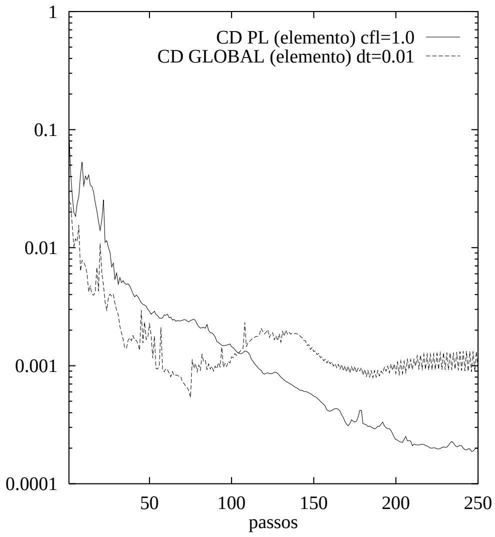
Considera-se o problema do choque refletido descrito na seção 3.6. As Figuras 5.1(a) e 5.1(b) mostram, respectivamente, as isocurvas de densidade para os operadores CD e CAU para a malha adaptada com 3594 elementos, usando a estrutura de dados por elementos e estratégia de passo local, considerando $CFL = 1.0$ ($0.002 < \Delta t < 0.2$). Observa-se que as isocurvas de densidade com o operador CD são praticamente iguais àquelas obtidas com o passo fixo, $\Delta t = 0.01$ e $CFL_{max} \approx 4.5$ (Figura 3.19). As Figuras 5.1(c) e 5.1(d) mostram comparações de resíduos de densidade para passo local e passo fixo para a estratégia por elemento, respectivamente, com os operadores CD e CAU. Observa-se que para os dois operadores o resíduo de densidade decresce mais na implementação com passo local durante os primeiros 250 passos.



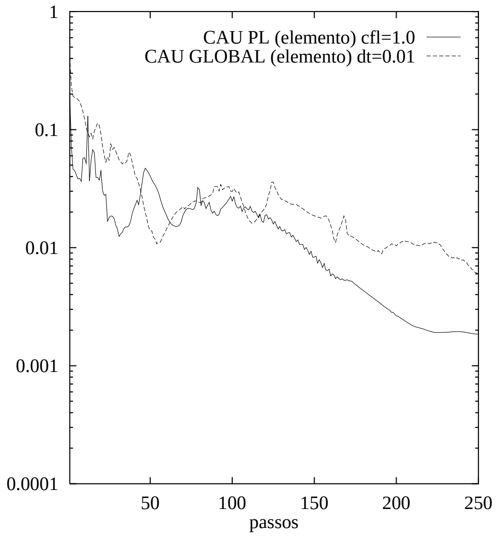
(a) Isocurva de Densidade com operador CD (Passo Local)



(b) Isocurva de Densidade com operador CAU (Passo Local)



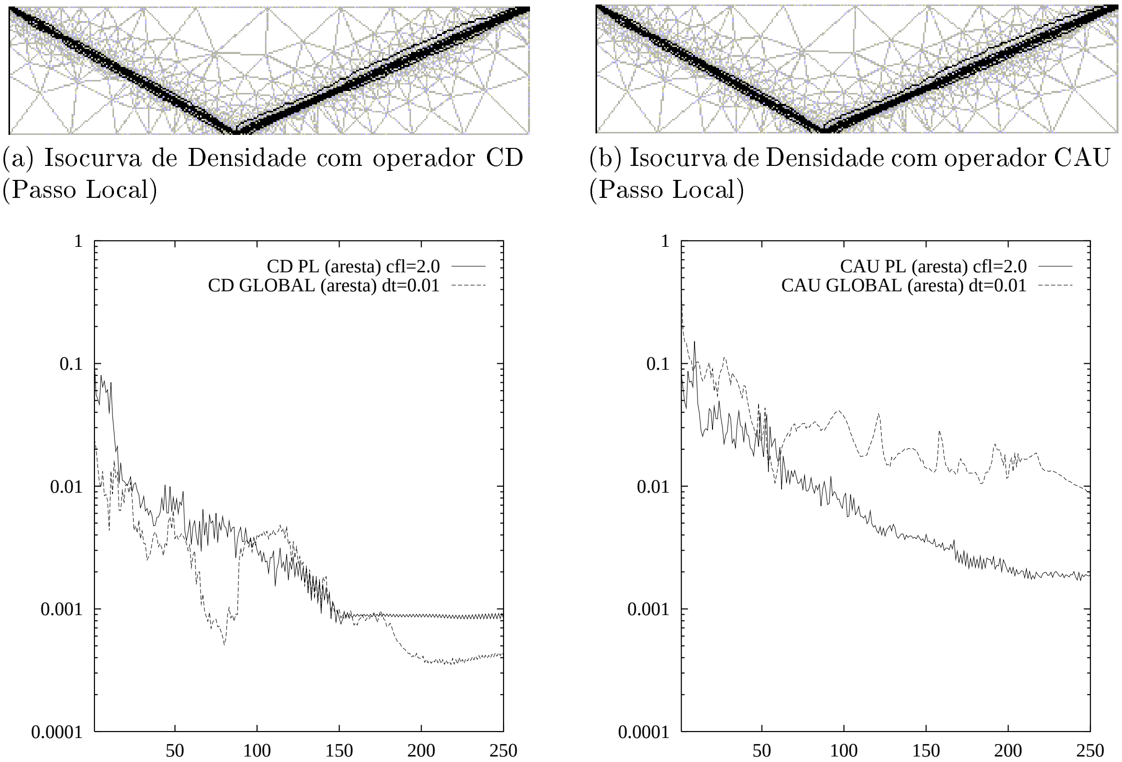
(c) Resíduo com operador CD



(d) Resíduo com operador CAU

Figura 5.1: Choque Refletido (2D) - Malha não estruturada adaptada com 3594 elementos e 1837 nós - Comparações de Densidade - Passo local com estratégia por elemento

As Figuras 5.2(a) e 5.2(b) mostram, respectivamente, as isocurvas de densidade para os operadores CD e CAU para a malha adaptada com 5430 arestas, usando a estrutura de dados por aresta e estratégia de passo local, considerando $CFL = 2.0$ ($0.004 < \Delta t < 0.4$). Observa-se que as isocurvas de densidade com o operador CD são praticamente iguais àquelas obtidas com o passo fixo, $\Delta t = 0.01$ e $CFL_{max} \approx 4.5$ (Figura 4.10). As Figuras 5.2(c) e 5.2(d) mostram comparações de resíduos de densidade para passo local e passo fixo para a estratégia por aresta, respectivamente, com os operadores CD e CAU. Observa-se que os resíduos de densidade são menores após 250 passos apenas com o operador CAU. Quando o operador CD é considerado o resíduo de densidade estagna em valores superiores àqueles obtidos usando passo fixo de avanço no tempo.



(a) Isocurva de Densidade com operador CD (Passo Local)

(b) Isocurva de Densidade com operador CAU (Passo Local)

(c) Resíduo com operador CD

(d) Resíduo com operador CAU

Figura 5.2: Choque Refletido (2D) - Malha não estruturada adaptada com 3594 elementos, 1837 nós e 5430 arestas - Comparações de Densidade - Passo local com estratégia por aresta

Na maioria dos casos estudados observa-se que após 200 passos os resíduos estagnam, tanto com passo local, quanto com passo fixo. O próximo item apresenta uma estratégia que visa acelerar a convergência fazendo com que os resíduos diminuam até a precisão da máquina.

5.2 Congelamento do operador de captura de descontinuidades

A solução das equações de Euler implica na resolução de um problema discreto altamente não-linear. JOHAN, HUGHES e SHAKIB [45] usam uma formulação de elementos finitos GLS (*Galerkin Least Square*) espaço-tempo em variáveis de entropia para as equações de N-S e observam que a linearização do sistema produz soluções que estagnam. Este fato ocorre uma vez que a linearização dos Jacobianos se dá de forma inconsistente. Para resolver esse problema os autores sugerem o uso de uma estratégia livre de matrizes, onde o efeito do Jacobiano é aproximado por duas avaliações sucessivas de resíduos. Este método é usado com sucesso por muitos autores, por exemplo, [34, 48, 49]. Por outro lado, VENKATAKRISHNAN [32], usando métodos explícitos em volumes finitos com esquemas *upwind* de alta ordem, também chega a soluções que estagnam. É investigado então o uso de técnicas de congelamento de limitadores das grandezas físicas do problema no contexto do método dos volumes finitos e proposto um novo limitador, não monotônico, que converge para a precisão da máquina. O mesmo problema de estagnação da convergência foi observado por LYRA et al [28] em formulações TVD explícitas de elementos finitos para as equações de Euler. Nesse caso foi utilizado a estratégia de congelar o operador de captura de descontinuidade quando o resíduo decresce em torno de duas ordens de grandeza.

Para a formulação implícita de elementos finitos em variáveis conservativas deste trabalho observa-se que, a partir de um determinado número de passos, o resíduo também não decresce. A estratégia a ser adotada visa congelar o operador de captura de descontinuidade a partir da estagnação do resíduo de densidade. Para justificar tal procedimento é considerado o exemplo do choque refletido, descrito na Figura 3.14, para a malha com 60×20 células, considerando 2 elementos em cada célula. A Figura 5.3 mostra os valores do parâmetro do operador de captura de descontinuidades δ_{CAU} em cada elemento da malha nos passos 1,2, 100, 200 e 300, com a estratégia por elemento. O operador CAU foi escolhido, pois seu valor é proporcional ao resíduo no interior do elemento. Observa-se que a partir de um determinado número de passos existem valores de δ_{CAU} significativamente mais elevados que os demais, conduzindo à estagnação do resíduo. Quando os operadores são congelados a aproximação no passo corrente é obtida sem a atualização dos efeitos não lineares do operador de captura de descontinuidades e, consequentemente, o resíduo pode até atingir a precisão da máquina.

Para congelar o operador de captura adota-se uma heurística que mede a variação média do resíduo dentro de um determinado número de passos. Para tal, considera-se dentro de um grupo de n_{passos}, o resíduo médio (res_{med}), o resíduo mínimo (res_{min}), o resíduo máximo (res_{max}), a razão mínima de variação (v_{min}) e a razão máxima de variação (v_{max}). No processo de avanço de passos é considerado o seguinte algoritmo para congelar o operador de captura:

`para` $i_{passo} = 1, 2, 3, \ldots$ `faça`

$\vdots$

calcular resíduo de i_{passo}

$\vdots$

`em cada grupo de` n_{passos} `faça`

`se` $res_{min} \geq v_{min} * res_{med}$ `e` $res_{max} \leq v_{max} * res_{med}$ `faça`

congelar o operador a partir de i_{passo}

`fim do se`

`fim do grupo de` n_{passos}

reinicializar componentes para o próximo grupo de n_{passos}

$\vdots$

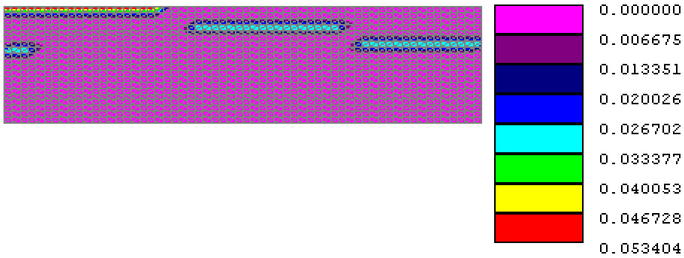
continuar procedimentos de i_{passo}

$\vdots$

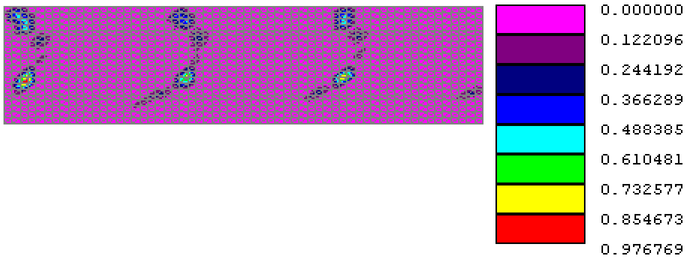
`fim do para`

5.2.1 Experimentos Numéricos

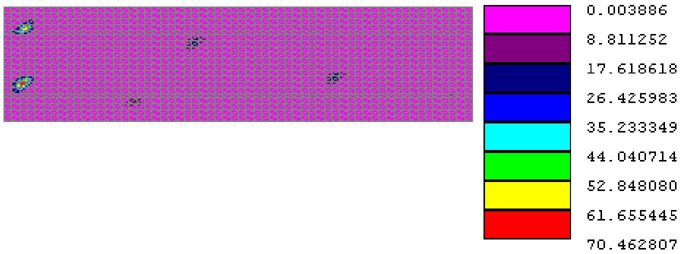
Mais uma vez é considerado o problema do choque refletido descrito na seção 3.6. A Figura 5.4 mostra os resultados obtidos para a malha adaptada com 3594 elementos, usando a estrutura de dados por elemento com congelamento para os operadores CD e CAU. Os parâmetros de congelamento adotados foram $n_{passos} = 30$, $v_{min} = 0.8$ e $v_{max} = 1.2$ para os dois operadores, sendo que o congelamento ocorreu no passo 240. As Figuras 5.4(a) e 5.4(b) mostram as isocurvas de densidade, respectivamente, para os operadores CD e CAU. Observa-se que os resultados são praticamente iguais àqueles obtidos sem o congelamento, mostrados, respectivamente, nas Figuras 5.1(a)



(a) Operador de captura CAU no passo 1

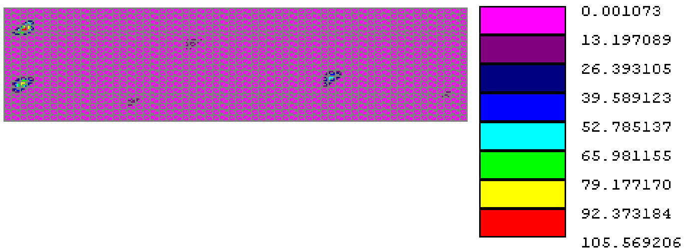


(b) Operador de captura CAU no passo 2



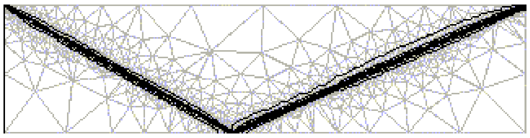
(c) Operador de captura CAU no passo 100

(d) Operador de captura CAU no passo 200

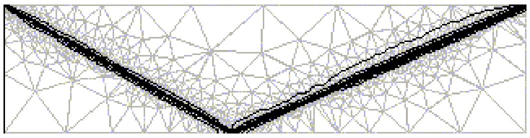


(e) Operador de captura CAU no passo 300

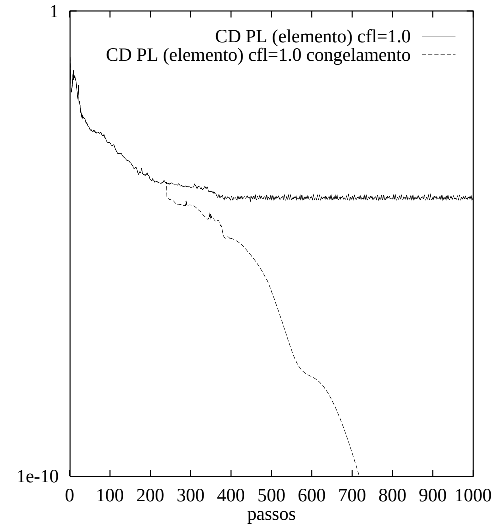
Figura 5.3: Choque Refletido (2D) - Malha 60×20 - Operador CAU - Estratégia por elemento e 5.1(b). As Figuras 5.4(c) e 5.4(d) comparam o resíduo de densidade, respectivamente, para os operadores CD e CAU. Observa-se que a solução converge para um tolerância de 10^{-10} em aproximadamente 700 passos para os dois operadores.



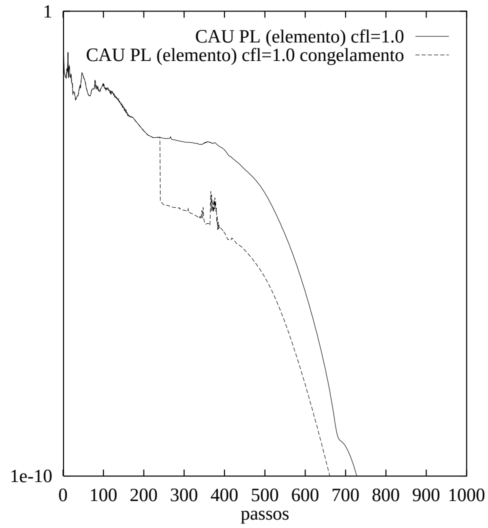
(a) Isocurva de Densidade com operador CD (congelamento)



(b) Isocurva de Densidade com operador CAU (congelamento)



(c) Resíduo com operador CD

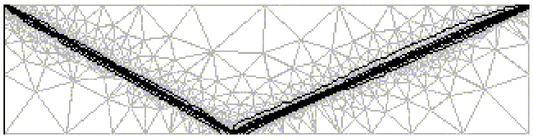


(d) Resíduo com operador CAU

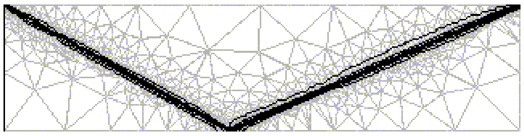
Figura 5.4: Choque Refletido (2D) - Malha não estruturada adaptada com 3594 elementos e 1837 nós - Passo local com Congelamento (estratégia por elemento)

A Figura 5.5 mostra os resultados obtidos para a malha adaptada com 5430 arestas, usando a estrutura de dados por aresta com congelamento para os operadores CD e CAU. Os parâmetros de congelamento adotado foram $n_{passos} = 30$, $v_{min} = 0.8$ e $v_{max} = 1.2$ para os dois operadores, sendo que o congelamento ocorreu no passo 180 para o operador CD e no passo 210 para o operador CAU. As Figuras 5.5(a) e 5.5(b) mostram as isocurvas de densidade, respectivamente, para os operadores CD e CAU. Observa-se que os resultados são praticamente iguais àqueles obtidos sem o congelamento mostrados, respectivamente, nas Figuras 5.2(a) e 5.2(b). As Figuras 5.5(c) e 5.5(d) comparam o resíduo de densidade, respectivamente, para os operadores CD e CAU. Observa-se que a solução converge para um tolerância de 10^{-10} em menos de 500 passos para os dois operadores.

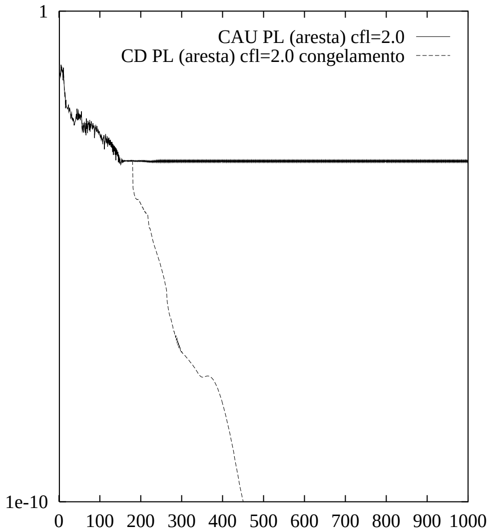
O resíduo de densidade decresce mais rápido na estratégia por aresta que na estratégia por elemento. Conclui-se, portanto, que a implementação usando passo local com congelamento permite obter a convergência até a precisão da máquina.



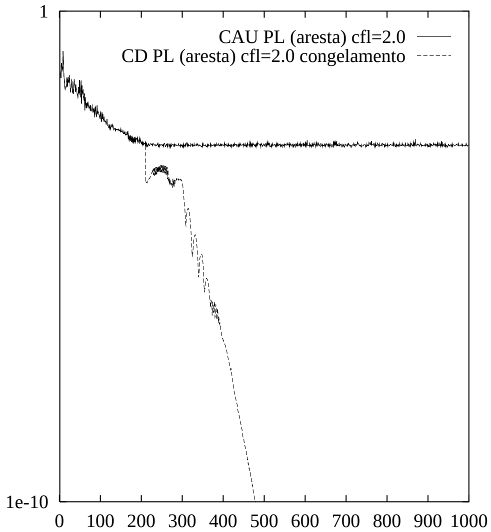
(a) Isocurva de Densidade com operador CD (congelamento)



(b) Isocurva de Densidade com operador CAU (congelamento)



(c) Resíduo com operador CD



(d) Resíduo com operador CAU

Figura 5.5: Choque Refletido (2D) - Malha não estruturada adaptada com 3594 elementos, 1837 nós e 5430 arestas - Passo local com Congelamento (estratégia por aresta)

Capítulo 6

Conclusões

Este trabalho apresentou uma formulação semi-discreta estabilizada usando variáveis conservativas para as equações de Euler, empregando uma estrutura de dados baseada nas arestas de malha de elementos finitos composta por triângulos. Foi implementada uma formulação implícita semi-discreta com parâmetros de estabilização e algoritmo de avanço no tempo do tipo preditor/multicorretor que simulam com sucesso escoamentos na presença de choques. É usada uma matriz de estabilização SUPG diagonalizada, simplificando bastante a formulação. São utilizados dois operadores de captura de descontinuidades, denominados CD e CAU. O operador CAU, originalmente definido em variáveis de entropia, neste trabalho é reescrito em variáveis conservativas através de transformações matriciais. Observa-se que as soluções encontradas para os dois operadores são praticamente iguais e bastante precisas, ou seja, representam bem o choque. Entretanto, a temperatura apresenta pequenas oscilações. Além disso, observa-se que em todos os exemplos testados os resíduos são maiores quando o operador CAU é usado.

A estrutura de dados por aresta, quando comparada com a estrutura de dados por elemento, reduz o tempo de processamento, necessitando de menos memória. A memória para armazenar os coeficientes da matriz efetiva necessária no algoritmo preditor/multicorretor na estratégia por aresta é praticamente a metade daquela necessária para armazenar os coeficientes na estratégia por elemento. Além disso, a estrutura por aresta reduziu o tempo de processamento em praticamente 30% em todas as aplicações executadas. Foram desenvolvidas técnicas para executar o produto matriz-vetor usando a estratégia por aresta que diminuem bastante o número de *flop* e endereçamentos indiretos, deixando os códigos bastante otimizados. Observa-se que para aplicações de escoamentos em regime permanente o método de solução do sistema linear resultante (GMRES) converge para tolerâncias muito baixas. Verificou-se também que em todos os exemplos testados as soluções por aresta estagnam com resíduos mais baixos para os dois operadores de captura de descontinuidades.

Uma importante consideração recai sobre o termo SUPG da formulação implícita, que além de oferecer maior estabilidade à solução também permite que a convergência seja mais rápida, como mostra a Figura 3.10(a). Este fato implica que a implementação por aresta desenvolvida nesse trabalho possui maior estabilidade que as implementações usando algoritmos tipo TVD [28, 30], uma vez que nesse tipo de implementação é desconsiderado o termo SUPG.

O tempo de execução é dominado pelas rotinas de montagem das matrizes efetivas locais, montagem do vetor de resíduo e a resolução do sistema linear de equações. O tempo necessário para a resolução do sistema na estratégia por aresta é praticamente a metade do tempo necessário na estratégia por elemento. Na estrutura de dados por elemento o tempo é dominado pela solução do sistema, enquanto que na estrututa de dados por aresta o tempo é dominado pela montagem da matriz efetiva local e do vetor de resíduos. Isto deve-se ao fato de que na estrutura de dados por aresta os parâmetros de estabilização não são calculados em cada aresta, mas avaliados nos dois elementos adjacentes, ocasionando um número maior de *flop* e endereçamentos indiretos.

As técnicas de aceleração de convergência no processo de avanço no tempo desenvolvidas contribuem para que a implementação apresentada neste trabalho convirja até a precisão da máquina na maioria dos casos estudados. Com a estratégia de passo local obtem-se soluções com $CFL \geq 1$ tão precisas quanto aquelas obtidas com CFL menores. O congelamento do operador de captura permite atingir a precisão da máquina em um número reduzido de passos.

Muito ainda há por realizar empregando estruturas de dados por arestas. Uma extensão natural seria a implementação de aplicações tridimensionais com termos viscosos (equações de Navier-Stokes). As equações de águas rasas também representam uma classe de problemas que possuem muitas características comuns com as equações de Euler, sendo portanto outra extensão natural. Além disso, novas estratégias de aceleração de convergência podem ser desenvolvidas, tais como novos pré-condicionadores e controle automático do tamanho do passo de tempo. Para diminuir o tempo de montagem das matrizes efetivas locais e o vetor de resíduos podem ser estudadas aproximações consistentes do cálculo dos parâmetros de estabilização diretamente na aresta, e não nos elementos adjacentes.

Apêndice A

Método GMRES

Seja o sistema

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{A.1}$$

onde $\mathbf{x}^t = \{x_1, \ldots, x_{neq}\}^t$ é o vetor de variáveis nodais, sendo neq o número de incógnitas; $\mathbf{A}$ é a matriz não simétrica $neq \times neq$; $\mathbf{b}$ é o vetor de termos independentes também de ordem neq.

A matriz $\mathbf{A}$ e o vetor $\mathbf{b}$ são construídos a partir da montagem das contribuições dos elementos ou das arestas da malha de triângulos lineares, isto é,

$$\mathbf{A} = \sum_{r=1}^{nn} \mathbf{A}_r \tag{A.2}$$

$$\mathbf{b} = \sum_{r=1}^{nn} \mathbf{b}_r \tag{A.3}$$

onde $\mathbf{A}_r$ e $\mathbf{b}_r$ podem ser descritos descritos a nível de cada elemento ou de cada aresta da malha.

O método iterativo GMRES [35], [20] tem por objetivo básico minimizar a norma residual do sistema (A.1). Considerando $\mathbf{x}_0$ a solução inicial, uma solução aproximada apresenta a forma $\mathbf{x}_0 + \mathbf{z}$ onde $\mathbf{z}$ é um vetor do espaço de Krylov $K = span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\}$, sendo $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ e k é a dimensão de K. O método GMRES determina $\mathbf{z}$ tal que a norma do resíduo seja mínima, isto é,

$$\mathbf{x}_0 + \mathbf{z} \text{ é solução de (A.1) se } \|\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{z})\| \text{ é mínima.}$$

onde $\|.\|$ denomina norma euclideana.

Uma base ortonormal de K pode ser obtida pelo Processo Modificado de Gram-Schmidt [20]:

$\mathbf{u}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$

`Para` $i = 1, \ldots, k$ `faça`

  $\tilde{\mathbf{u}}_{i+1} = \mathbf{A}\mathbf{u}_i$

  `Para` $j = 1, \ldots, i$ `faça`

    $\beta_{i+1,j} = (\tilde{\mathbf{u}}_{i+1}, \mathbf{u}_j)$

    $\tilde{\mathbf{u}}_{i+1} \leftarrow \tilde{\mathbf{u}}_{i+1} - \beta_{i+1,j}\mathbf{u}_j$

  `Fim do` j

  $\mathbf{u}_{i+1} = \frac{\tilde{\mathbf{u}}_{i+1}}{\|\tilde{\mathbf{u}}_{i+1}\|}$

`Fim do` i

Seja $\mathbf{U}_k = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$. Pode-se então, mostrar que $\mathbf{A}\mathbf{U}_k = \mathbf{U}_{k+1}\mathbf{H}_k$ onde $\mathbf{H}_k$ é a matriz superior de Hessenberg de ordem $(k+1) \times k$,

$$\mathbf{H}_k = \begin{bmatrix} \beta_{2,1} & \beta_{3,1} & \ldots & \beta_{k,1} & \beta_{k+1,1} \\ \|\tilde{\mathbf{u}}_2\| & \beta_{3,2} & \ldots & \beta_{k,2} & \beta_{k+1,2} \\ 0 & \|\tilde{\mathbf{u}}_3\| & \ddots & \vdots & \vdots \\ \vdots & 0 & \ddots & \beta_{k,k-1} & \vdots \\ \vdots & \vdots & \ddots & \|\tilde{\mathbf{u}}_k\| & \beta_{k+1,k} \\ 0 & 0 & \vdots & 0 & \|\tilde{\mathbf{u}}_{k+1}\| \end{bmatrix} \tag{A.4}$$

Seja $\mathbf{z} = \sum_{j=1}^{k} y_j u_j$ e $\mathbf{e} = \{\|\mathbf{r}_0\|, 0, \ldots, 0\}^T$ com $k+1$ termos. Observe que $\mathbf{r}_0 = \mathbf{U}_{k+1}\mathbf{e}$, portanto, tem-se:

$$\begin{aligned} \|\mathbf{b} - \mathbf{A}(x_0 + z)\| &= \|\mathbf{r}_0 - \mathbf{A}(\textstyle\sum_{j=1}^{k} y_j u_j)\| \\ &= \|\mathbf{r}_0 - A\mathbf{U}_k\mathbf{y}\| \\ &= \|U_{k+1}(\mathbf{e} - \mathbf{H}_k\mathbf{y})\| \\ &= \|\mathbf{e} - \mathbf{H}_k\mathbf{y}\| \end{aligned} \tag{A.5}$$

Assim o problema de minimização em questão pode ser reescrito por:

$$min_{\mathbf{z}\in K}\|\mathbf{b} - \mathbf{A}(x_0 + z)\| = min_{\mathbf{y}\in\Re^k}\|\mathbf{e} - \mathbf{H}_k\mathbf{y}\| \tag{A.6}$$

A matriz $\mathbf{H}_k$ tem um estrutura quase triangular superior, necessitando-se somente eliminar a diagonal contendo $\|\tilde{\mathbf{u}}_2\|, \|\tilde{\mathbf{u}}_3\|, \ldots, \|\tilde{\mathbf{u}}_{k+1}\|$. A transformação de $\mathbf{H}_k$ em uma matriz triangular superior pode ser feita pelo algoritmo QR, usando em cada passo o processo de rotações de Givens para obter um novo problema de minimização, maiores detalhes podem ser vistos em [20]. Assim, obtem-se:

$$min_{\mathbf{y}\in\Re^k} \|\bar{\mathbf{e}} - \bar{\mathbf{H}}_k \mathbf{y}\| \tag{A.7}$$

onde $\bar{\mathbf{H}}_k$ é uma matriz triangular superior. O sistema pode então, ser resolvido facilmente por retrosubstituição e um novo resíduo é calculado com a aproximação encontrada. Se a convergência não for atingida dentro de um ciclo (formado pelo número de vetores na base de Krylov), a solução mais recente será usada para iniciar a próxima iteração. Resumindo, uma iteração do Método GMRES consiste de:

- formação de uma base ortogonal para o espaço de Krylov pelo processo modificado de Gram-Schmidt.
- triangularização da Matriz de Hessenberg pelo algoritmo QR.
- Retrosubstituição para obter a solução aproximada.

Com estas considerações descreve-se um algoritmo padrão do método GMRES. Para tal define-se os parâmetros:

$$\begin{array}{lll} \varepsilon_{tol} & \rightarrow & \text{tolerância} \\ k & \rightarrow & \text{número de vetores de Krylov} \\ l_{max} & \rightarrow & \text{número máximo de iterações} \end{array}$$

`Dados` $\mathbf{A}$, $\mathbf{b}$, k, ε_{tol} `e` l_{max}

(Inicializações)

$\varepsilon \leftarrow \varepsilon_{tol} b$

$\mathbf{x} = 0$

(ciclo GMRES)

`Para` $l = 1, \ldots, l_{max}$ `faça`

  $\mathbf{u}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}$

$\bar{e} = \|\mathbf{u}_1\|$

$\mathbf{u}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$

(iterações do GMRES)

`Para` $i = 1, \ldots, k$ `faça`

$\mathbf{u}_{i+1} = \mathbf{A}\mathbf{u}_i$

(Ortogonalização modificada de Gram-Schmidt)

`Para` $j = 1, \ldots, i$ `faça`

$\beta_{i+1,j} = (\mathbf{u}_{i+1}, \mathbf{u}_j)$

$\mathbf{u}_{i+1} \leftarrow \mathbf{u}_{i+1} - \beta_{i+1,j}\mathbf{u}_j$

`Fim do` j

(Fim da ortogonalização modificada de Gram-Schmidt)

$\bar{\mathbf{h}}^{(i)} = \{\beta_{i+1,1}, \ldots, \beta_{i+1,i}, \|\mathbf{u}_{i+1}\|\}^T$

$\mathbf{u}_{i+1} = \frac{\mathbf{u}_{i+1}}{\|\mathbf{u}_{i+1}\|}$

(Algoritmo QR)

`Para` $j = 1, \ldots, i-1$ `faça`

$$\left\{ \begin{array}{c} \bar{h}_j^{(i)} \\ \bar{h}_{j+1}^{(i)} \end{array} \right\} \leftarrow \left[\begin{array}{cc} c_j & s_j \\ -s_j & c_j \end{array} \right] \left\{ \begin{array}{c} \bar{h}_j^{(i)} \\ \bar{h}_{j+1}^{(i)} \end{array} \right\}$$

`Fim do` j

$r = \sqrt{(\bar{h}_i^{(i)})^2 + (\bar{h}_{i+1}^{(i)})^2}$

$c_i = \frac{\bar{h}_i^{(i)}}{r}$

$s_i = \frac{\bar{h}_{i+1}^{(i)}}{r}$

$\bar{h}_i^{(i)} \leftarrow r$

$\bar{h}_{i+1}^{(i)} \leftarrow 0$

$\bar{e}_{i+1} = -s_i\bar{e}_i$

$\bar{e}_i = c_i\bar{e}_i$

(Fim do algoritmo QR)

`Teste de convergência:  Se` $|\bar{e}_{i+1}| \leq \varepsilon_{tol}$, `fim do` i

(Fim da iteração GMRES)

`Resolver` $\mathbf{y}$:

$$\left[\begin{array}{cccc} \bar{h}_1^{(1)} & \ldots & \bar{h}_1^{(i-1)} & \bar{h}_1^{(i)} \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \bar{h}_{i-1}^{(i-1)} & \bar{h}_{i-1}^{(i)} \\ 0 & \ldots & 0 & \bar{h}_i^{(i)} \end{array} \right] \left\{ \begin{array}{c} y_1 \\ \vdots \\ y_{i-1} \\ y_i \end{array} \right\} = \left\{ \begin{array}{c} \bar{e}_1 \\ \vdots \\ \bar{e}_{i-1} \\ \bar{e}_i \end{array} \right\}$$

`Cálculo da Solução:`

$$\mathbf{x} \leftarrow \mathbf{x} + \sum_{j=1}^{i} y_j u_j$$

`Teste de Convergência:` `Se` $|\bar{e}_{i+1}| \leq \varepsilon$, `fim do`$l$

(Fim do ciclo GMRES)

`x é a solução aproximada`

A fragilidade dos métodos iterativos com relação a rapidez de convergência é amplamente conhecida, portanto técnicas que permitam acelerar a convergência tem sido desenvolvidas. Tais técnicas são caracterizadas pelo uso de pré-condicionadores. Grosseiramente falando, um pré-condicionamento é uma maneira explícita ou implícita de modificar um sistema original de forma que se possa resolvê-lo "facilmente" por um dado método iterativo. Por exemplo, o escalonamento de todas as linhas de um sistema linear feita pelos coeficientes da diagonal principal é uma forma explícita de pré-condicionamento. Outros exemplos poderiam ser:

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} \tag{A.8}$$

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{N}^{-1}\mathbf{N}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} \tag{A.9}$$

$$\mathbf{A}\mathbf{N}^{-1}\mathbf{N}\mathbf{x} = \mathbf{b} \tag{A.10}$$

onde (A.8) é conhecido como pré-condicionador à esquerda, (A.9) é conhecido por pré-condicionador à esquerda e direita e finalmente (A.10) é conhecido como pré-condicionador à direita. $\mathbf{M}$ e $\mathbf{N}$ podem representar simples decomposições ou complicadas transformações de $\mathbf{A}$. Em geral, a eficiência do pré-condicionador é inversamente proporcional a complexidade da transformação, pois levando em consideração que o maior objetivo é diminuir o tempo de execução, os pré-condicionadores com decomposições simples acabam sendo mais vantagosos. Segundo Saad em [35], encontrar um bom pré-condicionador para resolver um sistema linear é frequentemente visto com uma combinação de arte e ciência. Resultados teóricos são raros e algumas técnicas oferecem resultados surpreendentes para algumas aplicações, outras nem tanto. A eficiência da operação matriz-vetor, mais uma vez é a maior preocupação, pois nela baseia-se a transformação do sistema.

Bibliografia

[1] ZIENKIEWICZ, O.C., TAYLOR, R.L., *The Finite Element Method - Solid and Fluid Mechanics Dynamics and Non-Linearity*, Vol. 2, 41ª ed., London, McGraw-Hill Book Company, 1991.

[2] HIRSH, C., *Numerical Computation of Internal and External Flows - Computational Methods for Inviscid and Viscous Flows*, Vol. 2, Chichester, John Wiley and Sons Ltd, 1992.

[3] BROOKS, A.N., HUGHES, T.J.R., "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations", *Comp. Meth. Appl. Mech. and Engng.*, v.32, pp. 199-259, 1982.

[4] MALVERN, L.E., *Introduction to the mechanics of a continuous medium*, Prentice Hall, Englewood Cliffs, NJ, 1969.

[5] HUGHES, T.J.R., *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis*, Englewood Cliffs, NJ, Prentice-Hall, 1987.

[6] HUGHES, T.J.R., TEZDUYAR, T.E.,"Finite Element Methods for First-Order Hyperbolic Systems with Particular Emphasis on the Compressible Euler Equations", *Comp. Meth. Appl. Mech. and Engng.*, v.45, pp. 217-284, 1984.

[7] ALMEIDA, R.C., GALEÃO, A.C., "An Adaptative Petrov-Galerkin Formulation for the Compressible Euler and Navier-Stokes Equation",*Comp Appl. Mech. and Engng.*, v.129, pp. 157-176, 1996.

[8] HUGHES, T.J.R., FRANCA, L.P., MALLET, M., "A New Finite Element Formulation for Computational Fluid Dynamics: I. Symmetric Forms of the Compressible Euler and Navier-Stokes Equations and The Second Law of Thermodynamics", *Comp. Meth. Appl. Mech. and Engng.*, v.54, pp. 223-234, 1986.

[9] HUGHES, T.J.R., MALLET, M., "A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalized Streamline Operator for Multidimensional Advective-Diffusive Systems",*Comp. Meth. Appl. Mech. and Engng.*, v.58, pp. 305-328, 1986.

[10] ALIABADI, S.K., TEZDUYAR, T.E., "Space-time finite element computation of compressible flows involving moving boundaries and interfaces", *Comp. Meth. Appl. Mech. and Engng.*, v.107, pp. 209-223, 1993.

[11] LE BEAU, G.J., RAY, S.E., ALIABADI, S.K. et al., "SUPG finite element computations of compressible flows with the entropy and conservation variables formulations", *Comp. Meth. Appl. Mech. and Engng.*, v.104, pp. 397-422, 1993.

[12] ALIABADI, S.K., RAY, S.E., TEZDUYAR, T.E., "SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations", *Computational Mechanics*, v.11, pp. 300-312, 1993.

[13] ALIBADI, S.K., TEZDUYAR, T.E., "Parallel fluid dynamics computations in aerospace applications", *Int. J. Num. Meth. Fluids*, v.21, pp. 783-805, 1995.

[14] TEZDUYAR, T.E., ALIABADI, S.K., BEHR, M., et al., "Flow simulation and high performance computing", *Computational Mechanics*, v.18, pp. 397-412, 1996.

[15] SOULAÏMANI, A., FORTIN, M., "Finite element solution of compressible viscous flows using conservative variables, *Comp. Meth. Appl. Mech. and Engng.*, v.118, pp. 319-350, 1994.

[16] HAUKE, G., HUGHES, T.J.R., "A unified approach to compressible and incompressible flows", *Comp. Meth. Appl. Mech. and Engng.*, v.113, pp. 389-395, 1994.

[17] HAUKE, G., HUGHES, T.J.R., "A comparative study of different sets of variables for solving compressible and incompressible flows", *Comp. Meth. Appl. Mech. and Engng.*, v.153, pp. 1-44, 1998.

[18] MITTAL, S., TEZDUYAR, T., "A unified finite element formulation for compressible and incompressible flows using augmented conservation variables", AHPCRC Preprint 97-056, 1997.

[19] SAMPAIO, P.A.B., MOREIRA, M.L., “A new finite element formulation for both compressible and nearly incompressible fluid dynamics”, *Int. J. Numerical Methods in Fluids*, v.32, pp. 51-78, 2000.

[20] SHAKIB, F., *Finite element analysis of the compressible Euler and Navier-Stokes equations*, Ph.D. dissertation, Stanford University, Palo Alto, California, USA, 1988.

[21] MARTINS, M.A.D., COUTINHO, A.L.G.A., ALVES, J.L.D, “Parallel iterative solution of finite element systems of equations employing edge-based data structures”, In: *SIAM Conf. on Parallel Processing for Scientific Computing*, CD-ROM, 1997.

[22] PERAIRE, J., MORGAN, K., PEIRO, J., “Unstructured grid methods for compressible flows”, In: Report R-787 *on Special Course on Unstructured Grid Methods for Advection Dominated Flows*, pp. 5.1–5.39, 1992.

[23] BAUM, J.D., LUO, H., LOHNER, R., “Edge–based finite element scheme for Euler equations”, *AIAA J.*, v.32, pp. 1183–1190, 1994.

[24] MAVRIPLIS, D.J., JAMESON, A., ”Multigrid solution of the Navier–Stokes equations on triangular meshes”, *AIAA J.*, v.28(8), pp. 1415–1425, 1990.

[25] MAVRIPLIS, D.J., “On Convergence Acceleration Techniques for Unstructured Meshes”.In: Report ICASE 98-44, Institute for Computer Applications in Science and Engineering (NASA Langley Research Center), Hampton, VA, 1998.

[26] CAREY, G.F., *Computational Grids: Generation, Adaptation and Solution Strategies*, Taylor and Francis, Washington, DC, USA, 1997.

[27] BARTH, T.J., *Numerical aspects of computings viscous high Reynolds number flows on unstructures meshes*, AIAA 91–0721, 1991.

[28] LYRA, P.R.M., PERAIRE, J., MORGAN, K., PEIRÓ, J., “TVD algorithms for the solution of the compressible Euler equations on unstructured meshes”, *Int. J. Num. Meth. Fluids*, v.19, pp. 849–863, 1994.

[29] LOHNER, R., “Edges, stars, superedges and chains”, *Comp. Meth. Appl. Mech and Engng.*, v.111, pp. 255-263, 1994.

[30] LYRA, P.R.M., WILLMERSDORF, R.B., MARTINS, M.A.D., COUTINHO, A.L.G.A., "Parallel implementation of edge-based finite element schemes for compressible flows on unstructured meshes", In: *Proceedings of the 3th International Meeting on Vector and Parallel Processing, VECPAR'98* , Porto, Portugal, 1998.

[31] VENKATAKRISHNAN, V., MAVRIPLIS, D.J., "Implicit solvers on ustructured meshes", *J. Comp. Physics*, v.105, pp. 83–91, 1993.

[32] VENKATAKRISHNAN, V., "Parallel computation of Ax and $A^t x$", *Int. Journal for High Speed Computing*, v.6, pp. 324–342, 1994.

[33] MARTINS, M.A.D., *Solução Iterativa em Paralelo de Sistemas de Equações do Método dos Elementos Finitos Empregando Estrutura de Dados por Arestas*, Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, 1996.

[34] LUO, H., BAUM, J.D., LOHNER, R., "A fast matrix-free implicit methods for compressible flows on unstructured grids", *J. Computational Physics*, v.146, pp. 664–690, 1998.

[35] SAAD, Y., *Iterative methods for sparse linear systems*,Boston, PWS Publishing Company, 1996.

[36] HSIEH, S.H., PAULINO, G.H., ABEL, J.F., "Recursive Spectral Algotithms for Automatic Domain Partitioning in Parallel Finite Element Analysis", *Comp. Meth. Appl. Mech. and Engng.*, v.121, pp. 137–162, 1995.

[37] HUGHES, T.J.R., FERENCZ, R., HALLQUIST, J.O., "Large-scale vectorized implicit calculations in solid mechanics on a CRAY X-MP/48 utilizing EBE preconditioned conjugate gradients", *Comp. Meth. Appl. Mech and Engng.*, v.61, pp. 21–248, 1987.

[38] PINAR, A., HEATH, M.T., "Improving Performance of Sparse Matrix-Vector Multiplication", In: *Proccedings of the SuperComputing 99*, CD-ROM, Portland. Oregon, USA, Nov., 1999.

[39] van GIJZEN, M.B., "Large Scale Finite Element Computations with GMRES-Like Methods on a Cray Y-MP", *Applied Numerical Mathematics*, v.19, pp. 51–62, 1995.

[40] LOHNER, R., "Some useful renumbering strategies for unstructured grids", *International Journal for Numerical Methods in Engineering*, v.36, pp. 3259–3270, 1993.

[41] LOHNER, R., "Renumbering strategies for unstructured-grid solvers operating in shared-memory, cache-based parallel machines", *Computer Methods in Applied Mechanics and Engineering*, v.163, pp. 95–109, 1998.

[42] LOHNER, R., YANG, C., ONATE, E., et al., "An unstructured grid- parallel free surface solver", *Applied Numerical Mathematics*, v.31, pp. 271–293, 1999.

[43] SAINI, S., BAILEY, D.H., *NAS Parallel Benchmarks (version 1.0) Results 11-96*, Report NAS-96-18, 1996, Numerical Aerodynamic Simulation Group, NASA Ames Research Center.

[44] COUTINHO, A. L. G. A., ALVES, J. L. D., "Parallel finite element simulation of msicible displacement in porous media", *SPE Journal*, v.4(1), pp. 487–500, 1996.

[45] JOHAN, Z., HUGHES, T., J., R., SHAKIB, F., "A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids",*Comp. Meth. Appl. Mech. and Engng.*, v.87, pp. 281–304, 1991. LE BEAU e TEZDUYAR

[46] LE BEAU, G.J., TEZDUYAR, T.E., "Finite Element Solution of Flow Problems with Mixed-Time Integration", *Journal of Engineering Mechanics*, v.117(6), pp. 1311–1330, 1991.

[47] DE SAMPAIO, P. A. B., MOREIRA, M. L., "A new finite element formulation for both compressible and nearly incompressible fluid dynamics", *Int. J. Num. Meth. Fluids*, v.32, pp. 51–78, 2000.

[48] TEZDUYAR, T. E., ALIABADI, S., BEHR, A., et al., "Parallel Finite Element Computation of 3D Flows", *IEEE Computer*, octorber, pp.27–36,1993.

[49] HAGER, J. O., LEE, K. D., "Effects of implicit preconditioners on solution acceleration schemes in cfd", *Int. J. Num. in Fluids*, v.22,pp. 1023–1035, 1996.