

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Civil

**Simulação Computacional de Escoamentos
Compressíveis Utilizando Adaptação de Malhas
Anisotrópica**

Renato Vaz Linn

Porto Alegre

2013

Renato Vaz Linn

**Simulação Computacional de Escoamentos
Compressíveis Utilizando Adaptação de Malhas
Anisotrópica**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Mestre em Engenharia. Orientação: Prof. Dr. Armando Miguel Awruch. Co-orientação: Prof. Dr. João Ricardo Masuero.

Porto Alegre

2013

L180s Linn, Renato Vaz
Simulação Computacional de Escoamentos Compressíveis Utilizando
Adaptação de Malhas Anisotrópica / Renato Vaz Linn. – 2013.

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul.
Escola de Engenharia. Programa de Pós-Graduação em Engenharia Civil.
Porto Alegre, BR-RS, 2013

Orientador: Prof. Dr. Armando Miguel Awruch.

Co-orientador: Prof. Dr. João Ricardo Masuero.

1. Dinâmica de Fluidos Computacional. 2. Adaptação Anisotrópica de
Malhas. 3. Elementos finitos (Engenharia). I. Awruch, Armando Miguel,
orient. II. Masuero, João Ricardo, coorient. III. Título.

CDU-624.04(043)

Renato Vaz Linn

Simulação Computacional de Escoamentos Compressíveis Utilizando Adaptação de Malhas Anisotrópica

Esta dissertação de mestrado foi julgada adequada para a obtenção do título de MESTRE EM ENGENHARIA, na Área de Concentração de ESTRUTURAS, e aprovada em sua forma final pelo professor orientador e pelo Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul.

Porto Alegre, 15 de Março de 2013

Prof. Armando Miguel Awruch
Dr. COPPE/UFRJ
Orientador

Prof. João Ricardo Masuero
Dr. PPGEC/UFRGS
Co-orientador

Ph.D. Luis Carlos Pinto da Silva Filho
Coordenador do PPGEC/UFRGS

Comissão Julgadora:

Alexandre Luis Braun
Dr. PPGEC/UFRGS

Horácio Antônio Vielmo
Dr. POSMEC/UFSC

Adriane Prisco Petry
Dr. PROMEC/UFRGS

Dedico este trabalho para minha mãe Maria Sirlei e meu irmão Rodrigo

Agradecimentos

Agadeço primeiramente a minha família, em especial minha mãe Maria Sirlei e meu irmão Rodrigo pelo apoio, paciência e carinho.

Ao professor Awruch pelo constante incentivo, dedicação e orientação do trabalho.

Ao professor Masuero pela valiosa ajuda.

Ao meu cão Toddy pelo companheirismo.

Ao Felipe Espath, um amigo.

A todos os professores que de alguma forma contribuíram com minha formação.

À Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior – CAPES e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, pela bolsa de estudos.

A todos os destilados e fermentados que me ajudaram a manter a mente lúcida e ávida ao longo deste período conturbado.

Ao grupo MP3 pelos diversos momentos prazerosos de distração jogando Magic.

Ao pessoal do Diablo 3 pelo momentos épicos compartilhados.

Ao CESUP, por possibilitar o uso do Cluster via acesso remoto, sem o qual grande parte deste trabalho não poderia ter sido realizado.

A todos os meus amigos e colegas, sem os quais o percurso ao longo da faculdade e da vida não seria tão agradável.

Epígrafe

Ни шагу назад!
(*Nenhum passo para trás!*)

[...] *O Supremo Quartel-General do Exército Vermelho ordena:*

[...] *Os conselhos militares dos exércitos e em primeiro lugar os comandantes de exército devem;*

- 1. Incondicionalmente remover do corpo de oficiais do seu exército comandantes e comissários que aceitaram retiradas de tropas de posições ocupadas sem a ordem do comando do exército e os enviar aos conselhos militares das frentes para corte marcial;*
- 2. Formar, dentro dos limites de cada exército, de 3 à 5 esquadrões defensivos bem armados (até 200 homens em cada), e pô-los diretamente atrás das divisões instáveis e requisitá-los no caso de pânico e de retiradas dispersas de elementos das divisões, para fuzilarem no local os disseminadores de boatos e os covardes e assim ajudar os honestos soldados das divisões a executarem a sua obrigação para com a Pátria;*
- 3. Formar, dentro dos limites de cada exército, até dez (dependendo da situação) companhias penais (de 150 a 200 homens em cada) para onde, os soldados rasos e os comandantes subalternos que forem culpados de falha de disciplina devido à covardia ou confusão, serão enviados e pô-los em difíceis setores do exército para lhes dar uma oportunidade para redimir com sangue os seus crimes contra a Pátria.*

[...]

Ordem No. 227 de Stalin. Moscou, 28 de julho de 1942.

Resumo

Linn, R.V. **Simulação Computacional de Escoamentos Compressíveis Utilizando Adaptação de Malhas Anisotrópica**. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 2013.

Esta dissertação tem por objetivo a implementação de um método adaptativo anisotrópico de malhas e seu acoplamento a um programa de simulação de escoamentos de fluidos compressíveis numa faixa ampla de número de Mach utilizando o Método dos Elementos Finitos (Bubnov-Galerkin). Mais especificamente, implementa-se um código para simulação de escoamentos compressíveis utilizando o método das linhas ou direções características (CBS), o qual possui boa performance para uma amplo espectro de números de Mach. A discretização espacial é feita utilizando-se elementos finitos triangulares lineares e tetraédricos lineares para problemas bidimensionais e tridimensionais, respectivamente. A discretização temporal é feita por uma expansão em séries de Taylor. Implementa-se uma adaptação anisotrópica com capacidade de refinamento e desrefinamento do domínio computacional para se obter soluções precisas a baixos custos computacionais, onde a estimativa de erro é conduzida por uma métrica Riemanniana, a qual permite o tratamento contínuo do erro através de um tensor de erro anisotrópico. O código é modificado para possibilitar o uso de arquiteturas de memória compartilhada a fim de aumentar a velocidade de processamento da simulação. Por fim, diversos estudos, tanto para problemas bidimensionais quanto tridimensionais, são realizados, validando a metodologia e realizando estudos adicionais em problemas mais complexos.

Palavras-chave: Dinâmica de Fluidos Computacional, Adaptação Anisotrópica de Malhas, Elementos Finitos.

Abstract

Linn, R.V. **Computational Simulation of Copressible Flow Using Anisotropic Mesh Adaptation.** Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 2013.

The objectives of this dissertation are the implementation of an anisotropic mesh adaptation procedure and its coupling with a compressible flow simulation code with large Mach range using the Finite Element Method (Bubnov-Galerkin). Specifically, a computer code using the Characteristic Based Split method called CBS, which has a good performance for a wide range of Mach numbers, is implemented. The spatial discretization is performed using linear triangular and linear tetrahedral finite elements for two and three dimensional cases, respectively. The temporal discretization is done with Taylor series. An anisotropic mesh adaptation technique, based on a Riemann metric to evaluate the errors (which allows continuous error analysis through an anisotropic error tensor evaluation), including mesh refinement and mesh coarsening procedures is also implemented looking for accurate and computationally cheaper solutions. The code is able to use shared memory architecture in order to reduce computer process time. Finally, two-dimensional and three-dimensional studies are performed in order to validate the proposed methodology and additional complex tests are also analysed.

Keywords: Computational Fluid Dynamics, Anisotropic Mesh Adaptation, Finite Element Method.

Sumário

Lista de Figuras	11
Lista de Tabelas	14
1 Introdução	15
1.1 Solução Numérica do Escoamento	15
1.2 Métodos de Adaptação de Malhas	18
1.3 Estimativa de Erro	20
1.4 Computação de Alto Desempenho	21
1.5 Objetivos	22
1.6 Metodologia	23
2 Equações de Conservação e Aproximação Numérica	24
2.1 Equações de Conservação	24
2.1.1 Condições Iniciais e de Contorno	27
2.1.2 Forma Adimensional das Equações de Conservação	28
2.2 Solução Numérica das Equações de Conservação	29
2.2.1 O Método das Linhas ou Direções Características (<i>Characteristic Based Split</i> – CBS)	30
2.2.2 Método CBS Aplicado à Mecânica de Fluidos	33
2.2.3 Discretização Espacial	35
2.2.4 Tratamento de Problemas de Escoamentos Compressíveis	37
2.2.5 Incremento de Tempo	40
2.2.6 Simulações Permanentes e Transientes	41
3 Estimativa de Erro	42
3.1 Métrica	43
3.1.1 Espaço Métrico Euclidiano	43
3.1.2 Espaço Métrico Riemanniano	44
3.2 Interpretação Geométrica	45
3.3 Relacionando Métrica Riemanniana e Erro	46
3.4 Determinação da Matriz Hessiana	48
4 Adaptação Anisotrópica	50
4.1 Geometria e Topologia	50
4.1.1 Simplex	50
4.1.2 Malha	51
4.2 Adaptação-h	52
4.2.1 Refinamento	52

4.2.2	Desrefinamento	55
4.2.3	Alternância de Arestas	57
4.2.4	Alternância de Faces	58
4.3	Condições Angulares	59
4.3.1	Condição Angular Mínima	59
4.3.2	Condição Angular Máxima	60
4.4	Algoritmo de Adaptação	62
4.5	Acoplamento com Código CBS	63
5	Computação de Alto Desempenho	65
5.1	Otimização Sequencial	65
5.1.1	Otimizações Simples	66
5.1.2	Compiladores	69
5.2	Otimização de Acesso à Memória	71
5.2.1	Acesso à Arranjos	71
5.2.2	Renumeração Nodal	72
5.3	Otimização Paralela em Computadores de Memória Compartilhada	75
5.3.1	Atribuição de Valores Prescritos	75
5.3.2	Escalonamento de <i>Loops</i>	76
5.3.3	Sincronizações e Atualizações	78
5.3.4	Escalabilidade de Pequenas Tarefas	78
5.4	Particularidades da Implementação	79
5.5	Definições	79
5.5.1	<i>Speedup</i>	80
5.5.2	Eficiência de Paralelização	80
6	Resultados	81
6.1	Adaptação de Malhas Bidimensionais	83
6.1.1	Exemplo 1: Solução Analítica Hiperbólica	83
6.1.2	Exemplo 2: Escoamento Supersônico Não-Viscoso em Canal com Rampa de Compressão e Expansão	96
6.1.3	Exemplo 3: Escoamento Supersônico Não-Viscoso Sobre Aerofólio NACA0012	100
6.1.4	Exemplo 4: Escoamento Transônico Viscoso Sobre Aerofólio NACA0012	107
6.1.5	Exemplo 5: Escoamento Subsônico não-Viscoso Sobre Aerofólio NACA0012	114
6.1.6	Exemplo 6: Escoamento Transiente Viscoso Sobre Aerofólio NACA0012	120
6.2	Adaptação de Malhas Tridimensionais	127
6.2.1	Exemplo 7: Solução Analítica de Função Trigonométrica	127
6.2.2	Exemplo 8: Análise de Desempenho Computacional	131
6.2.3	Exemplo 9: Escoamento Supersônico Não-Viscoso Sobre Esfera	134
6.2.4	Exemplo 10: Escoamento Transônico Não-Viscoso Sobre Asa Onera M6	141
6.2.5	Exemplo 11: Escoamento Transônico Não-Viscoso Sobre Avião	149
6.2.6	Exemplo 12: Escoamento Supersônico Não-Viscoso Sobre Míssil AGM-114 Hellfire	157

6.2.7	Exemplo 13: Escoamento Supersônico Não-Viscoso Sobre Avião Lockheed SR-71	169
7	Conclusões e Sugestões Para Trabalhos Futuros	173
7.1	Conclusões	173
7.2	Sugestões Para Trabalhos Futuros	175
	Referências Bibliográficas	177

Lista de Figuras

2.1	Domínio (genérico) e regiões do contorno.	28
2.2	Natureza do comportamento em forma de onda.	31
2.3	Problema de convecção linear. Linhas características.	31
3.1	Isovalores da função $f(\mathbf{x}) = \ell_{\mathcal{M}}(\mathbf{o}\mathbf{x})$, onde \mathbf{o} é a origem do plano (centro), para diferentes espaços métricos Riemannianos. Na esquerda, o espaço Euclidiano canônico $([-1,1] \times [-1,1], \mathbf{I}_2)$. No centro, o espaço métrico Euclidiano $([-1,1] \times [-1,1], \mathcal{M})$ com \mathcal{M} constante. À direita, um espaço métrico Riemanniano $(\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in [-1,1]^2}$ com um tensor métrico variável.	45
3.2	Interpretação geométrica de $\Psi_{\mathcal{M}}(1)$. v_i são os autovetores de \mathcal{M} e h_i^{-2} são os autovalores de \mathcal{M}	46
4.1	Configurações para subdivisão de triângulos.	53
4.2	Configurações para subdivisão de tetraedros.	53
4.3	Desrefinamento por colapso de aresta seguida de degeneração de elementos adjacentes.	56
4.4	Alternância de arestas.	58
4.5	Alternância de faces de elementos no contorno.	58
4.6	Alternância de faces de elementos no interior do domínio.	59
5.1	Renumeração do tipo <i>advancing-front</i>	73
5.2	Tipos de escalonamentos em OpenMP para um <i>loop</i> de 20 iterações e executado por 3 <i>threads</i>	77
6.1	Malha e solução inicial do exemplo 1.	83
6.2	Malha ao longo das iterações de adaptação do exemplo 1.	86
6.3	Solução ao longo das iterações de adaptação do exemplo 1.	87
6.4	Detalhe aumentado de uma porção da malha adaptada final do exemplo 1.	88
6.5	Detalhe aumentado de uma porção da malha adaptada final do exemplo 1 sem alternância de arestas.	90
6.6	Malha ao longo das iterações de adaptação do exemplo 1 alterando-se o limiar superior para $\eta_S = 1.6$	91
6.7	Malha ao longo das iterações de adaptação do exemplo 1 alterando-se o limiar inferior para $\eta_I = 0.3$	92
6.8	Malha e solução inicial do exemplo 1 modificado.	93
6.9	Malha ao longo das iterações de adaptação do exemplo 1 modificado.	94
6.10	Solução ao longo das iterações de adaptação do exemplo 1 modificado.	95
6.11	Geometria e condições de contorno do exemplo 2.	97
6.12	Convergência do exemplo 2.	98

6.13	Massa específica em função de x para $y = 0.5$ do exemplo 2.	98
6.14	Malha e solução inicial do exemplo 2.	99
6.15	Malha e solução final do exemplo 2.	99
6.16	Geometria e condições de contorno do exemplo 3.	100
6.17	Malha inicial do exemplo 3.	101
6.18	Solução inicial do exemplo 3.	103
6.19	Solução final com adaptação do exemplo 3.	104
6.20	Malha adaptada final do exemplo 3.	105
6.21	Coefficiente de pressão para solução adaptada do exemplo 3.	105
6.22	Convergência do exemplo 3.	106
6.23	Solução (campo de M) obtida por Dolejší e Felcman (2004).	106
6.24	Solução inicial do exemplo 4.	109
6.25	Solução final adaptada do exemplo 4.	110
6.26	Solução obtida por Bono (2008).	111
6.27	Linhas de corrente para a solução inicial e adaptada final do exemplo 4. . .	111
6.28	Malha adaptada final do exemplo 4.	111
6.29	Convergência do exemplo 4.	112
6.30	Coefficiente de sustentação C_L ao longo das iterações do exemplo 4.	112
6.31	Coefficiente de arrasto C_D ao longo das iterações do exemplo 4.	113
6.32	Coefficiente de pressão C_p para solução final adaptada do exemplo 4.	113
6.33	Solução inicial do exemplo 5.	116
6.34	Solução final adaptada do exemplo 5.	117
6.35	Malha adaptada final do exemplo 5.	118
6.36	Convergência do exemplo 5.	118
6.37	Comparação de ρ próxima à região de estagnação do exemplo 5.	119
6.38	Coefficiente de pressão C_p para solução final adaptada do exemplo 5.	119
6.39	Campo de ρ para diferentes instantes de tempo do exemplo 6.	122
6.40	Campo de e para diferentes instantes de tempo do exemplo 6.	123
6.41	Campo de M para diferentes instantes de tempo do exemplo 6.	124
6.42	Malha para diferentes instantes de tempo do exemplo 6.	125
6.43	Detalhe da malha para diferentes instantes de tempo do exemplo 6.	126
6.44	Malha e solução inicial do exemplo 7.	128
6.45	Malha e isocontornos da solução ao longo das iterações de adaptação do exemplo 7.	129
6.46	Isosuperfícies de $u = -0.597$ para a solução inicial e final do exemplo 7. . .	130
6.47	Detalhe aumentado de uma porção da malha adaptada final do exemplo 7. . .	130
6.48	Detalhe em corte (plano em $z = 1$) de uma porção da malha adaptada final do exemplo 7.	131
6.49	Malhas utilizadas no exemplo 8.	132
6.50	Geometria e condições de contorno do exemplo 9.	135
6.51	Malha inicial do exemplo 8.	136
6.52	Solução inicial no plano xz com $y = 0$ do exemplo 8.	137
6.53	Solução final adaptada no plano xz com $y = 0$ do exemplo 8.	138
6.54	Malha da solução adaptada final no plano yz do exemplo 8.	139
6.55	Isosuperfícies da solução adaptada final do exemplo 8.	139
6.56	Convergência do exemplo 9.	140
6.57	Coefficiente de arrasto C_D ao longo das iterações do exemplo 9.	140
6.58	Geometria detalhada da asa Onera M6 do exemplo 10.	142

6.59	Condições de contorno do exemplo 10.	142
6.60	Malha inicial do exemplo 10.	143
6.61	Solução inicial do exemplo 10.	144
6.62	Solução final adaptada do exemplo 10.	145
6.63	Malha da solução adaptada final no plano xy do exemplo 10.	146
6.64	Isosuperfícies da solução adaptada final do exemplo 10.	146
6.65	Coeficientes de pressão C_P para diferentes seções da asa do exemplo 10.	147
6.66	Convergência do exemplo 10.	148
6.67	Coeficiente de arrasto C_D ao longo das iterações do exemplo 10.	148
6.68	Coeficiente de sustentação C_L ao longo das iterações do exemplo 10.	149
6.69	Malha Inicial do exemplo 11.	151
6.70	Malha Inicial do exemplo 11.	151
6.71	Malha Inicial do exemplo 11.	152
6.72	Solução adaptada final no plano xy do exemplo 11.	153
6.73	Malha adaptada final do exemplo 11.	154
6.74	Campo de p (vista em perspectiva) do exemplo 11.	155
6.75	Campo de M (vista em perspectiva) do exemplo 11.	155
6.76	Convergência do exemplo 11.	156
6.77	Coeficiente de sustentação C_L ao longo das iterações do exemplo 11.	156
6.78	Míssil AGM-114 Hellfire do exemplo 12.	159
6.79	Condições de contorno do exemplo 12.	160
6.80	Malha Inicial do exemplo 12.	160
6.81	Solução inicial no plano yz com $x = 0$ do exemplo 12.	161
6.82	Solução adaptada final no plano yz com $x = 0$ do exemplo 12.	162
6.83	Campos da solução adaptada final sobre o míssil e em perspectiva do exemplo 12.	163
6.84	Cortes no plano xy para o campo de M do exemplo 12.	164
6.85	Malha final adaptada do exemplo 12.	165
6.86	Isosuperfícies para de M do exemplo 12.	166
6.87	Regiões de interferência do arrasto para um míssil com 4 canards dispostos simetricamente (Gregorek, 1970).	167
6.88	Convergência do exemplo 12.	167
6.89	Coeficiente de arrasto C_D ao longo das iterações do exemplo 12.	168
6.90	Avião SR-71 do exemplo 13.	170
6.91	Malha inicial do exemplo 13.	171
6.92	Malha adaptada final do exemplo 13.	171
6.93	Campo da solução adaptada final sobre o avião do exemplo 13.	172

Lista de Tabelas

6.1	Dados das iterações de adaptação do exemplo 1	85
6.2	Dados das iterações de adaptação do exemplo 1 sem alternância de arestas	89
6.3	Dados das iterações de adaptação do exemplo 2	100
6.4	Dados das iterações de adaptação do exemplo 3	102
6.5	Dados das iterações de adaptação do exemplo 4	108
6.6	Coefficientes aerodinâmicos do exemplo 4	114
6.7	Variáveis no ponto de estagnação do exemplo 5	120
6.8	Dados das iterações de adaptação do exemplo 5	121
6.9	Dados das iterações de adaptação do exemplo 7	130
6.10	Dados das malhas utilizadas no exemplo 8	133
6.11	Performance computacional do exemplo 8	133
6.12	Performance computacional sem escalonamento de tarefas do exemplo 8	133
6.13	Dados das iterações de adaptação do exemplo 9	136
6.14	Coefficientes aerodinâmicos do exemplo 9	136
6.15	Dados das iterações de adaptação do exemplo 10	143
6.16	Coefficientes aerodinâmicos do exemplo 10	143
6.17	Coefficientes aerodinâmicos do exemplo 11	150
6.18	Dados das iterações de adaptação do exemplo 11	150
6.19	Dados das iterações de adaptação do exemplo 12	159
6.20	Dados das iterações de adaptação do exemplo 13	170

Capítulo 1

Introdução

Uma breve revisão bibliográfica sobre cada assunto relacionado e analisado nesta dissertação é apresentada neste capítulo, bem como o delineamento, justificativa e motivação da abordagem adotada.

1.1 Solução Numérica do Escoamento

Estudos na área da Dinâmica de Fluidos Computacional (*Computational Fluid Dynamics* – CFD) têm sido desenvolvidos cada vez mais devido, em grande parte, aos avanços conjuntos em termos de recursos computacionais e algoritmos numéricos. É uma área importante e já incorporada a diversos campos da engenharia, como, por exemplo, o da engenharia aeronáutica e aeroespacial, tanto em fase de projeto como de pesquisa e investigação. Entre os objetivos de tais simulações, e em especial no contexto desta dissertação, visa-se obter com precisão os coeficientes de pressão, arrasto e sustentação de escoamentos sobre corpos imersos, bem como a distribuição de velocidades no entorno de configurações geométricas complexas como aviões (considerando a geometria total), mísseis e asas. Dentre as vantagens obtidas com o emprego de simulações numéricas para estes fins, pode-se citar a possibilidade de obter uma ampla quantidade de dados, os quais podem servir de base para diversos estudos paramétricos do problema, assim como a facilidade com que as simulações podem ser alteradas, com a simples mudança das condições de contorno do problema e a capacidade de simular fenômenos, efeitos e situações de difícil ou custosa realização via análise experimental. Entre algumas

dificuldades encontradas na simulação numérica de escoamentos pode-se citar o elevado custo computacional bem como a dependência dos modelos utilizados para obtenção de resultados válidos. O custo computacional elevado se deve em parte pela necessidade de empregar-se uma discretização espacial suficientemente refinada sem os quais bons resultados não são obtidos. Outra parcela se deve à escala dos fenômenos envolvidos, os quais quanto mais precisos e detalhados forem analisados, tão mais refinada deverá ser a discretização, o que por sua vez geralmente resulta em grandes reduções de incrementos de tempo utilizados na simulação devido às limitações impostas pela física do problema sobre o algoritmo numérico. A dependência do modelo utilizado para obtenção da solução numérica varia de acordo com o problema analisado, onde, por exemplo, escoamentos de caráter altamente turbulento serão representados com a incorporação de um modelo de turbulência, tais como os modelos *Large-Eddy Simulation* (LES) e *Reynolds-Averaged Navier-Stokes Equations* (RANS). Pode-se também realizar a simulação direta (*Direct Numerical Simulation* – DNS) do escoamento, entretanto, o custo computacional requerido é muito mais elevado neste caso (Löhner, 2008). Outro exemplo é a capacidade do método em lidar com amplas variações de compressibilidade e velocidade, o que pode ser um limitante de emprego de determinado algoritmo para uma dada simulação em alguns casos (Zienkiewicz et al., 2011).

Para a simulação numérica de escoamentos não-permanentes necessita-se de dois níveis de discretização: uma espacial e outra temporal. Entre os métodos numéricos mais populares empregados para discretização espacial pode-se citar o Método das Diferenças Finitas, o Método dos Volumes Finitos e o Método dos Elementos Finitos (Galerkin), em que a discretização espacial é chamada de malha. O Método das Diferenças Finitas baseia-se na aproximação de derivadas nodais (forma diferencial), sendo um método mais simples, fácil de ser obtido e implementado, assim como eficiente computacionalmente. Entretanto, é um método em que a discretização da geometria torna-se limitada a malhas estruturadas (em blocos). O Método dos Volumes Finitos, por sua vez, se baseia na aproximação de integrais, tendo como característica o tratamento conservativo (na forma integral) do problema e adaptável a praticamente qualquer tipo de geometria. O Método dos Elementos Finitos deriva de uma aproximação variacional de resíduos ponderados sobre as equações governantes (forma fraca), sendo muito flexível e generalizado, adequado para geometrias complexas, tanto quanto o Método dos Volumes Finitos. Como o presente

trabalho visa tratar de escoamentos no entorno de geometrias complexas, adotou-se o emprego do Método dos Elementos Finitos para a discretização espacial do domínio, utilizando malhas não estruturadas formadas por triângulos e tetraedros lineares em duas e três dimensões, respectivamente. O Método dos Volumes Finitos e o Método dos Elementos Finitos podem ser empregados muitas vezes de forma equivalente no contexto de simulações de escoamentos (Löhner, 2008). Para a discretização espacial há uma grande variedade de metodologias, podendo-se citar o método de Taylor-Galerkin (de 1 e de 2 passos e também suas variantes), o método Galerkin-Descontínuo e o Método das Linhas ou Direções Características (*Characteristic Based Split – CBS*). Os métodos de Taylor-Galerkin se baseiam em uma expansão em série de Taylor no tempo truncada. No contexto dos escoamentos compressíveis, seu emprego se mostra adequado para escoamentos de alta velocidade, onde as equações governantes assumem um comportamento predominantemente hiperbólico (o método foi, de fato, concebido originalmente para solução de equações diferenciais parciais puramente hiperbólicas). Velocidades baixas nestas aproximações resultam em instabilidade numérica devido a oscilações de pressão (Zienkiewicz et al., 2011). O método de Galerkin-Descontínuo tem como característica a descontinuidade de suas funções de forma e possui performance mais adequada para uma variedade ampla de problemas, não apenas os hiperbólicos, mas também os elípticos, parabólicos e mistos. O Método das Linhas ou Direções Características (CBS) baseia-se em uma transformação de coordenadas da malha que é variável ao longo do tempo, de forma que altere o comportamento das equações governantes para otimizar a discretização espacial. A otimização se dá pelo fato do operador das equações governantes se tornar auto-adjunto nesta metodologia. Este método foi desenvolvido no intuito de poder ser utilizado num amplo espectro de velocidades, desde problemas de escoamentos subsônicos até problemas de escoamentos supersônicos sem apresentar grandes oscilações. A desvantagem do Método das Linhas Características, que é a de necessitar mais cálculos a cada incremento de tempo comparado com outros métodos, é superada pela redução do número de passos requeridos para a convergência (Zienkiewicz et al., 2011). Neste sentido, adotou-se o Método das Linhas ou Direções Características (CBS) no presente trabalho. O método é empregado na forma explícita no presente trabalho. As vantagens de uma formulação explícita são o menor requerimento de memória computacional e a facilidade de paralelização, entretanto,

apresenta a desvantagem de requerer mais iterações, principalmente para escoamentos permanentes. Além disto, tendo em vista a extensão a trabalhos futuros incluindo a interação fluido-estrutura, o método explícito torna-se mais indicado neste sentido.

Para simulação de escoamentos compressíveis, tratamentos adicionais devem ser empregados. Um destes tratamentos refere-se às regiões em que há formação de ondas de choque, que são implementadas conjuntamente com alguma metodologia que permita gerar alguma difusão forçada confinada a apenas estas regiões. A viscosidade e condutividade térmica também não podem ser consideradas constantes nestes problemas e sim função da temperatura (Zienkiewicz et al., 2011). Adota-se neste trabalho dois modelos distintos para tratar ondas de choque e a Lei de Sutherland para relacionar viscosidade e temperatura. A metodologia adotada para simulação de escoamentos compressíveis utilizando o método CBS é detalhada no capítulo 2.

1.2 Métodos de Adaptação de Malhas

A adaptação de malhas visa remodelar automaticamente a discretização espacial do problema através de alguma metodologia, de forma a reduzir o esforço computacional e maximizar a precisão da solução obtida. Atualmente, dispõe-se de uma grande variedade de programas para geração automática de malhas para discretização do domínio analisado. É possível, algumas vezes com certo esforço, gerar malhas adequadas para simulação de certos casos de escoamentos. Entretanto, tais malhas dificilmente são apropriadas. Isto decorre do fato destas malhas, mesmo quando geradas por pessoas experientes com este tema, possuírem seu processo construtivo fundamentado em métodos heurísticos. Além disto, a malha mais apropriada para um dado escoamento no entorno de uma geometria fixada irá apresentar grande variação, dependendo das condições empregadas no escoamento como número de Reynolds e número de Mach, entre outros. Portanto a malha apropriada para a simulação de um dado escoamento depende de todas as características deste escoamento (Habashi et al., 2000). Tais observações não são restritas a problemas de simulação de escoamentos. Na verdade, praticamente toda simulação via elementos finitos enquadra-se nesta problemática.

As estratégias de adaptação de malhas são geralmente divididas em quatro possíveis

abordagens:

- **Adaptação h :** onde infere-se diretamente na topologia da malha através de inserção e remoção de nós. Neste procedimento, partem-se arestas, ou elementos, em outros menores para criar regiões mais densas na malha onde ocorram grandes gradientes para obter-se transições suaves na solução. Também aplica-se o processo inverso nas regiões de baixo gradiente, diminuindo a densidade da malha para reduzir a quantidade de graus de liberdade em regiões já suficientemente discretizadas. Entre as vantagens principais nesta metodologia pode-se citar a capacidade de reduzir significativamente o erro a níveis muito baixos, o fato do procedimento ser conservativo quando aumenta-se o número de graus de liberdade, e que o procedimento é adequado a problemas transientes onde zonas de transição variam continuamente, requerendo grande flexibilidade para adequação da malha.
- **Adaptação r :** nesta abordagem mantém-se a topologia da malha sem aumentar ou diminuir o número de graus de liberdade. Move-se os nós da malha para adequá-la melhor ao problema. A principal vantagem do método é a de manter fixa a topologia da malha conjuntamente com uma maior simplicidade de implementação. Entretanto a limitação do número de graus de liberdade pode comprometer a capacidade de reduzir ou captar adequadamente todas regiões de altos gradientes e também ser relativamente mais limitada para captar certos problemas físicos complexos (Löhner, 2008).
- **Adaptação p :** neste método, eleva-se o grau do polinômio interpolador para melhorar a precisão do resultado. Não é um método muito popular para problemas de dinâmica de fluidos devido às grandes complicações computacionais que o método proporciona. Combinações de métodos são possíveis, assim, pode-se utilizar um método $h-p$, $h-r$ ou $r-p$, por exemplo. A combinação de metodologias tende a criar métodos mais completos, nos quais um método complementa as dificuldades encontradas pelo outro método. Todos estes três procedimentos anteriormente citados trabalham localmente sobre a malha.
- **Remalhamento:** A vantagem principal deste método é a possibilidade de sempre adaptar o domínio por completo, o qual também é sua desvantagem devido ao

maior custo computacional envolvido e dificuldades extras encontradas quanto ao problema de como especificar quais regiões devem ser mais ou menos refinadas de forma automática. A utilização de um gerador de malhas eficiente e capaz de lidar com especificações complexas de geração de malhas se torna imprescindível nesta metodologia.

O presente trabalho contempla apenas a implementação do método h , contendo etapas tanto de refinamento quanto de desrefinamento.

Métodos clássicos de adaptação refinam ou desrefinam regiões de forma isotrópica em todas as direções. Tais malhas só podem ser ótimas para escoamentos que possuam gradientes espaciais que variam igualmente em todas direções. Disto resulta que fluxos direcionais tais como ondas de choque, camada limite, ondas e vórtices não são necessariamente adaptados com a maior eficiência e menor custo possível, sendo que o número de elementos cresce rapidamente a cada etapa de refinamento (Tam et al., 2000). No presente trabalho, utiliza-se um procedimento de adaptação anisotrópico, no qual adapta-se a malha de forma direcional, consistindo uma abordagem mais eficiente com menor custo computacional envolvido, a qual é detalhada no capítulo 4.

1.3 Estimativa de Erro

Para qualquer metodologia de adaptação de malhas, torna-se necessário que se indique ou estime onde e em que intensidade há erros. Há duas metodologias principais de estimativa de erro: *a priori* e *a posteriori*. A metodologia *a priori* é principalmente uma abordagem qualitativa enquanto a *a posteriori* produz informação tanto qualitativa quanto quantitativa. A estimativa *a priori* é usualmente global por natureza, já a estimativa *a posteriori*, usualmente fornece tanto informação local quanto global. A diferença principal entre estas duas abordagens recai quanto à informação para a estimativa do erro. No método *a priori* o erro é mensurado a partir de informações do fluxo ou geométricas. Já na abordagem *a posteriori*, utiliza-se a solução aproximada para estimar o erro (Loiselle, 2008). A metodologia *a posteriori*, por sua vez, também possui diversas abordagens distintas. Entre elas, pode-se citar o indicador de erro baseado em pulos e descontinuidades, os obtidos por simples interpolação, por comparação de

derivadas, os obtidos através de resíduos em pontos adjacentes da malha, os baseados em normas energéticas e os conduzidos por métrica (Löhner, 2008). Para a implementação de uma adaptação anisotrópica, torna-se indispensável que o indicador de erro possa informar estas direções de anisotropia bem como suas respectivas magnitudes (Bottasso, 2004). Neste sentido, no presente trabalho utiliza-se a metodologia de indicação de erro conduzido por métrica para a abordagem de estimativa *a posteriori*. A métrica adotada é a Riemanniana. A vantagem desta abordagem, além de possibilitar o tratamento anisotrópico, é a de estimar o erro de forma proporcional à grandeza do erro estimado de forma eficiente e de forma contínua (na forma integral, em contraponto a uma abordagem discreta). O capítulo 3 aborda a metodologia e conceitos teóricos envolvidos com a implementação da estimativa de erro conduzida pela métrica Riemanniana.

1.4 Computação de Alto Desempenho

A Computação de Alto Desempenho - *High Performance Computing* (HPC) se refere, em geral, ao uso de supercomputadores, *clusters* e *workstations* e até mesmo computadores simples para tarefas que requerem grandes recursos de computação, tais como simulações numéricas. Os exemplos mais claros são o uso de diversos computadores ligados por uma rede trabalhando juntos para resolver um problema e o uso de um único computador com diversos processadores. Mas estes são apenas exemplos fisicamente distintos. A HPC engloba também questões menos óbvias que passam despercebidas, como o uso eficiente do *hardware* dos computadores, o meio de acesso à memória, a forma de comunicação, a forma de divisão de tarefas, dentre tantas outras. Outros pontos relacionados à HPC são de natureza muito mais conceitual, como por exemplo a escolha de um determinado tipo de algoritmo para solução de um problema. Do ponto de vista matemático, muitas vezes é indiferente a solução proveniente de diferentes algoritmos, mas computacionalmente muitas vezes há diferenças marcantes. Em problemas computacionalmente pequenos, estas questões, em geral, não requerem atenção significativa. Mas em problemas complexos, a escolha de um algoritmo ineficiente pode significar a diferença de um tempo de execução de apenas um dia para até uma semana ou mesmo um mês.

Distinguem-se dois tipos básicos de programação: a de memória distribuída e a de

memória compartilhada. No primeiro caso, tem-se usualmente diversos computadores ligados através de uma rede através da qual se comunicam. Já na segunda, tem-se apenas um computador com diversos processadores. Do ponto de vista da implementação de códigos computacionais, a metodologia de memória distribuída é mais robusta, uma vez que pode lidar com problemas maiores, mas também possui implementação mais complexa. Já a de memória compartilhada é mais simples, principalmente devido à não necessidade de comunicação através de uma rede, mas não possui tanta capacidade de expansão como a distribuída, pois o *hardware* para memória compartilhada tem sua complexidade aumentada exponencialmente com o aumento do número de processadores. O presente trabalho utiliza apenas a programação em memória compartilhada e utilizou-se diretivas OpenMP para isto.

O capítulo 5 deste trabalho apresenta as diversas abordagens de programação de alto desempenho adotadas na implementação do código desenvolvido. Embora muitas das informações contidas no capítulo possam parecer triviais, ou não usuais em uma dissertação, ainda parece, sob a ótica do autor do trabalho, a melhor forma de expô-las idéias. Idéias estas que dificilmente encontram-se claramente difundidas, focalizadas especialmente na aplicação do trabalho (adaptação de malhas e simulação de escoamentos com elementos finitos). É bem verdade também que há um grande despreparo de alunos (e até mesmo professores) das áreas científicas e engenharias quanto à programação e conhecimentos em ciências da computação, devido (prioritariamente) à falta de ensino nas instituições. Neste sentido, o capítulo 5 também visa de alguma forma fornecer informações úteis àqueles que futuramente venham a ler o capítulo em busca de informações que ajudem a implementar um código eficiente.

1.5 Objetivos

Os objetivos do presente trabalho são listados a seguir:

- Implementar a adaptação anisotrópica de malhas não estruturadas de elementos triangulares e tetraédricos lineares para os casos bidimensionais e tridimensionais, respectivamente. O procedimento inclui etapas de refinamento, desrefinamento e de

alternância de arestas e faces;

- Implementar o método CBS tridimensional para solução de escoamentos compressíveis;
- Acoplar adaptação anisotrópica de malhas com o código CBS, tanto para o caso tridimensional quanto bidimensional;
- Testar exemplos comparativos que validem a metodologia;
- Simular escoamentos complexos e comparar com a literatura existente.

1.6 Metodologia

Utiliza-se a linguagem Fortran 90/95 para implementar os objetivos citados. Fez-se uso do código de simulação de escoamentos compressíveis bidimensionais fornecido pelo Professor P. Nithiarasu da Universidade de Swansea como ponto de partida do trabalho. Este código original era bastante simplificado e não possuía nenhum tipo de implementação de adaptação de malhas, programação paralelizada, bem como carecia de melhorias, extensões e otimizações. Reformulou-se e reestruturou este código na linguagem de programação utilizada assim como o pré e o pós-processamento dos resultados foram modificados. A partir deste código foi desenvolvido um novo código para simulação de escoamentos compressíveis tridimensionais. Adicionalmente, implementou-se no código tridimensional desenvolvido a paralelização para computadores de memória compartilhada para possibilitar o uso de malhas mais robustas em um intervalo de tempo aceitável. As simulações tridimensionais foram executadas no cluster do CESUP/UFRGS (Centro Nacional de Super Computação).

Capítulo 2

Equações de Conservação e Aproximação Numérica

As equações de conservação de quantidade de movimento, de massa e de energia constituem um sistema de equações diferenciais parciais que descrevem o movimento e o cumprimento da primeira Lei da Termodinâmica em um meio contínuo (Malvern, 1969). Para solução numérica deste sistema de equações através do Método dos Elementos Finitos (MEF) emprega-se neste trabalho o método dos resíduos ponderados do tipo Bubnov-Galerkin para a discretização espacial do problema. Utiliza-se malhas não estruturadas com elementos triangulares e tetraédricos lineares para problemas bidimensionais e tridimensionais, respectivamente. Para a discretização temporal utiliza-se uma série de Taylor e o esquema das Direções ou Linhas Características CBS é utilizado, a qual modifica a discretização espacial em função do tempo.

2.1 Equações de Conservação

As equações de conservação constituem um sistema de equações diferenciais em derivadas parciais que pode classificar-se em um tipo mais amplo de problema denominado de problema de convecção-difusão. Estudos mais aprofundados e detalhados sobre as equações de conservação na Mecânica dos Fluidos podem ser encontrados em Schlichting (1979) e White (1991). Na forma conservativa, em uma descrição cinemática Euleriana, tem-se que a conservação de massa é expressa pela equação da continuidade

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) \quad (2.1)$$

onde ρ é a densidade, u_i é a componente da velocidade do fluido segundo a direção de x_i , sendo x_i ($i = 1,2,3$) as direções dos eixos coordenados no espaço cartesiano e t indica a variável tempo. A equação da continuidade expressa a relação entre a massa acumulada no interior de um volume de controle com o fluxo através de seu contorno. A partir da Segunda Lei de Newton, deriva-se a equação de conservação de quantidade de movimento, que vem dada por

$$\frac{\partial(\rho u_j)}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i u_j) - \frac{\partial \tau_{ij}}{\partial x_i} + \frac{\partial p}{\partial x_j} - \rho g_j = 0 \quad (2.2)$$

onde p é a pressão termodinâmica, g_j é a força por unidade de massa atuando na direção de x_j e τ_{ij} são as tensões viscosas, as quais são proporcionais ao gradiente de velocidades. Adotando-se a hipótese de Stokes, a qual relaciona efeitos viscosos com a variação de volume, obtém-se para a relação constitutiva das tensões viscosas a seguinte expressão

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \quad (2.3)$$

em que μ é a viscosidade dinâmica (de cisalhamento) do fluido e δ_{ij} é o delta de Kronecker. A conservação de energia é derivada da Primeira Lei da Termodinâmica, em conjunto com a adoção da Lei de Condutibilidade Térmica de Fourier (a qual adota a hipótese de que o fluxo de calor deve ser proporcional ao gradiente de temperatura local), resultando

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i E) - \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right) + \frac{\partial}{\partial x_i}(p u_i) - \frac{\partial}{\partial x_i}(\tau_{ij} u_j) - \rho g_i u_i + q = 0 \quad (2.4)$$

onde T é a temperatura absoluta, k é a condutibilidade térmica (isotrópica), q é uma fonte ou sumidouro de calor e E é a energia total por unidade de massa, a qual inclui a energia interna específica por unidade de massa e e a energia cinética por unidade de massa, ou seja que (desprezando-se a energia potencial gravitacional)

$$E = e + \frac{1}{2} u_i u_i \quad (2.5)$$

As equações de conservação podem ser escritas em uma forma mais generalizada da seguinte maneira

$$\frac{\partial \Phi}{\partial t} + \frac{\partial \mathbf{F}}{\partial x_i} + \frac{\partial \mathbf{G}}{\partial x_i} + \mathbf{Q} = \mathbf{0} \quad (2.6)$$

onde Φ , \mathbf{F} , \mathbf{G} e \mathbf{Q} são dadas por

$$\Phi = \begin{Bmatrix} \rho \\ \rho u_i \\ \rho E \end{Bmatrix} \quad (2.7)$$

$$\mathbf{F} = \begin{Bmatrix} \rho u_i \\ \rho u_j u_i + p \delta_{ji} \\ (\rho + p) E u_i \end{Bmatrix} \quad (2.8)$$

$$\mathbf{G} = \begin{Bmatrix} 0 \\ -\tau_{ji} \\ -\tau_{ij} u_j - k \frac{\partial T}{\partial x_i} \end{Bmatrix} \quad (2.9)$$

$$\mathbf{Q} = \begin{Bmatrix} 0 \\ -\rho g_i \\ -\rho g_i u_i - q \end{Bmatrix} \quad (2.10)$$

onde $i, j = 1, 2, 3$. O vetor Φ contém as variáveis de conservação, \mathbf{F} contém os termos advectivos, \mathbf{G} possui os termos difusivos e \mathbf{Q} os termos de campo e fonte. Embora formulado para casos tridimensionais, as equações anteriores podem ser facilmente bidimensionalizadas através da redução da variação dos índices i, j e k de 3 para 2. Quando desprezam-se os termos difusivos, ou seja, $\mathbf{G} = \mathbf{0}$, o sistema resultante constitui as Equações de Euler (despreza-se a viscosidade do fluido e o balanço de energia fica modificado de forma a desconsiderar a difusão de calor e o trabalho mecânico das tensões viscosas). O sistema de equações (2.6) possui mais variáveis do que número de equações e portanto não pode ser resolvido ainda. É necessário a inclusão das chamadas equações de estado para o seu completo fechamento. Neste sentido, adota-se a hipótese de gás perfeito para o fluido, a qual relaciona a massa específica ρ com a pressão p e a temperatura absoluta T através da constante universal dos gases R

$$\rho = \frac{p}{RT} \quad (2.11)$$

entretanto, a equação anterior é mais comumente expressa em função da relação entre os coeficiente de calor específico a pressão e a volume constante, C_p e C_v , respectivamente, ficando

$$p = (\gamma - 1) \rho e \quad (2.12)$$

onde $\gamma = \frac{C_p}{C_v}$. Adota-se adicionalmente a hipótese de que o gás seja termicamente perfeito, ou seja, que ele não esteja sujeito a reações químicas e que sua energia interna e e entalpia h sejam proporcionais exclusivamente à temperatura T , isto é, $e = e(T)$ e $h = h(T)$, de forma que, na forma diferencial:

$$\frac{de}{dT} = C_v \quad \frac{dh}{dT} = C_p \quad (2.13)$$

Por fim, adota-se a hipótese de gás caloricamente perfeito, na qual, em adição, o calor específico é constante

$$e = C_v T \quad h = C_p T \quad (2.14)$$

O sistema está agora fechado, contendo 5 equações diferenciais parciais para 5 variáveis independentes ρ , u_i ($i = 1,2,3$) e e juntamente com fórmulas algébricas para o cálculo de p , τ e T . Resta apenas especificar condições iniciais e de contorno adequadas.

2.1.1 Condições Iniciais e de Contorno

As condições iniciais que devem ser estabelecidas para se resolver as equações do sistema (2.6) são as seguintes

$$\rho|_{t=t_0} = \rho_0(\mathbf{x}), \quad u_i|_{t=t_0} = u_{i0}(\mathbf{x}), \quad e|_{t=t_0} = e_0(\mathbf{x}) \quad \text{em } \Omega \quad (2.15)$$

em que \mathbf{x} é a posição e Ω representa o domínio. As condições de contorno possíveis são aplicadas em 3 tipos distintos de contorno (Fig. 2.1). O primeiro deles é o contorno onde há fluxo de entrada $\Gamma_{in} = \{x \in \Omega : u_i n_i < 0\}$ (sendo n_i a normal externa segundo a direção de x_i), onde prescreve-se densidade, energia e velocidade, ou seja

$$\rho = \rho_{in}, \quad u_i = u_{iin}, \quad e = e_{in} \quad \text{em } \Gamma_{in} \quad (2.16)$$

O segundo tipo de contorno é o de fluxo de saída $\Gamma_{out} = \{x \in \Omega : u_i n_i > 0\}$, onde prescreve-se a velocidade ou tensões de cisalhamento nulas, portanto

$$u_i n_i = u_{out} \quad \text{ou} \quad -p + \tau_{ij} n_i n_j = 0 \quad \text{em } \Gamma_{out} \quad (2.17)$$

e por último, há o contorno do tipo parede sólida $\Gamma_w = \{x \in \Omega : u_i n_i = 0\}$, onde prescreve-se a condição de não deslizamento conjuntamente com a imposição de uma dada temperatura ou um fluxo prescrito, ou seja

$$u_i = 0, \quad T = T_w \quad \text{ou} \quad \frac{\partial T}{\partial n_i} = q_w \quad \text{em } \Gamma_w \quad (2.18)$$

Para escoamentos com viscosidade desprezível (Euler), o contorno de parede sólida é tratado de forma distinta, anulando-se apenas a componente da velocidade normal ao contorno. A união de cada uma das partes do contorno fornece o contorno total $\Gamma = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_w$. Com estas condições, o problema fica bem definido. Condições

iniciais ou de contorno insuficientes ou incorretas podem levar a soluções errôneas, ou a nenhuma solução. As condições iniciais e os valores prescritos nos contornos utilizados no presente trabalho são sempre as de corrente livre.

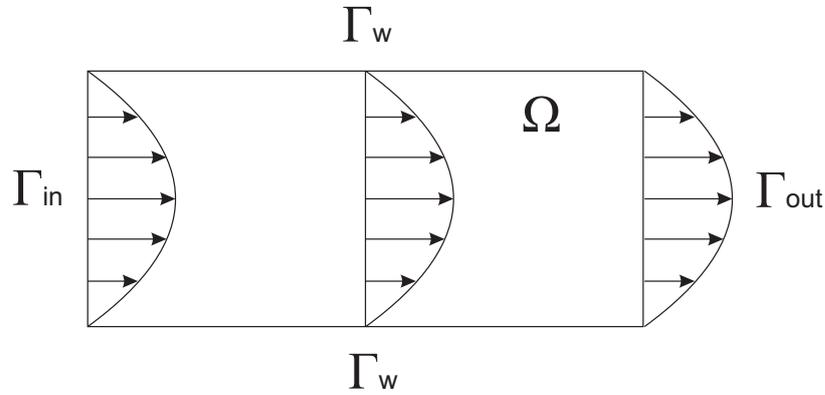


Figura 2.1: Domínio (genérico) e regiões do contorno.

2.1.2 Forma Adimensional das Equações de Conservação

Embora as equações de conservação apresentadas anteriormente possam ser utilizadas para solução numérica, resulta mais conveniente utilizá-las de forma adimensionalizada. Entre as vantagens obtidas nesse procedimento pode-se citar a facilidade do escalonamento das equações para obtenção de resultados em situações reais de fluxos, o erro de arredondamento devido à manipulação de números em escalas muito distintas é minimizado e, finalmente, o acesso direto a termos de fundamental importância para o modelo. Os termos adimensionalizados utilizados são

$$\begin{aligned} \bar{t} &= \frac{tu_\infty}{L} & \bar{x}_i &= \frac{x_i}{L} & \bar{\rho} &= \frac{\rho}{\rho_\infty} & \bar{p} &= \frac{p}{\rho_\infty u_\infty^2} \\ \bar{u}_i &= \frac{u_i}{u_\infty} & \bar{E} &= \frac{E}{u_\infty} & \bar{T} &= \frac{TC_p}{u_\infty^2} & \bar{c}^2 &= \frac{c^2}{u_\infty^2} \end{aligned} \quad (2.19)$$

onde uma linha sobre a variável indica que ela está adimensionalizada, L é o comprimento de referência, o sub-índice ∞ representa uma quantidade de corrente livre e c é a velocidade do som no meio. Convém também definir os seguintes números adimensionais

$$Re = \frac{\rho_\infty u_\infty L}{\mu_\infty} \quad Pr = \frac{C_p \mu_\infty}{k_\infty} \quad M = \frac{|u_i|}{c_\infty} \quad (2.20)$$

em que Re é o número de Reynolds (relação entre forças de inércia e forças de viscosidade), Pr é o número de Prandtl (relação entre taxas de difusão viscosa e taxa de difusão

térmica) e M é o número de Mach (relação entre velocidade do fluido e velocidade do som). Utilizando-se as variáveis adimensionalizadas nas equações governantes obtém-se a seguinte forma para a equação da conservação de massa

$$\frac{\partial \bar{\rho}}{\partial \bar{t}} = \frac{1}{\bar{c}^2} \frac{\partial \bar{p}}{\partial \bar{t}} = - \frac{\partial \bar{U}_i}{\partial \bar{x}_i} \quad (2.21)$$

onde considerou-se o escoamento como isentrópico e $U_i = \rho u_i$ é o fluxo de massa. A equação de conservação de quantidade de movimento fica

$$\frac{\partial \bar{U}_i}{\partial \bar{t}} = \frac{\partial}{\partial \bar{x}_j} (\bar{u}_j \bar{U}_i) + \frac{1}{Re} \frac{\partial (\bar{\nu} \bar{\tau}_{ij})}{\partial \bar{x}_j} - \frac{\partial \bar{p}}{\partial \bar{x}_i} + \bar{\rho} \bar{g}_i \quad (2.22)$$

em que $\nu = \frac{\mu}{\rho}$ é a viscosidade cinemática e

$$\bar{\nu} = \frac{\nu}{\nu_\infty} \quad \bar{g}_i = \frac{g_i L}{u_\infty^2} \quad (2.23)$$

A equação de conservação de energia adimensionalizada é dada por

$$\frac{\partial (\bar{\rho} \bar{E})}{\partial \bar{t}} = - \frac{\partial}{\partial \bar{x}_i} (\bar{\rho} \bar{u}_i \bar{E}) + \frac{1}{RePr} \frac{\partial}{\partial \bar{x}_i} \left(\bar{k} \frac{\partial \bar{T}}{\partial \bar{x}_i} \right) - \frac{\partial}{\partial \bar{x}_i} (\bar{p} \bar{u}_i) + \frac{1}{Re} \frac{\partial}{\partial \bar{x}_i} (\bar{\nu} \bar{\tau}_{ij} \bar{u}_j) \quad (2.24)$$

onde

$$\bar{k} = \frac{k}{k_\infty} \quad (2.25)$$

A equação de estado adimensionalizada resulta

$$\bar{p} = \bar{\rho} \frac{(\gamma - 1)}{\gamma} \bar{T} \quad (2.26)$$

Por motivo de clareza, as equações na forma original serão mantidas no restante do trabalho.

2.2 Solução Numérica das Equações de Conservação

A aplicação do Esquema das Linhas ou Direções Características (CBS) sobre as equações de conservação de quantidades de movimento e energia é capaz de proporcionar a separação dos termos advectivos e difusivos. Nesta separação, o operador diferencial de cada uma das equações a serem discretizadas torna-se auto-adjunto, o que otimiza a aplicação de uma técnica variacional ou de resíduos ponderados.

2.2.1 O Método das Linhas ou Direções Características (*Characteristic Based Split – CBS*)

Os métodos CBS têm sido muito utilizados em diversas áreas da mecânica dos fluidos bem como da mecânica dos sólidos (Zienkiewicz et al., 2011). Para tornar mais claro o entendimento do método, considera-se inicialmente a equação de advecção-difusão para o caso unidimensional na forma não-conservativa

$$\frac{\partial \phi}{\partial t} + U \frac{\partial \phi}{\partial x} - \frac{\partial}{\partial x} \left(k \frac{\partial \phi}{\partial x} \right) + Q = 0 \quad (2.27)$$

onde ϕ é uma variável de conservação. O problema assim definido é não-linear a menos que U seja independente de ϕ . Entretanto, a forma não-conservativa da equação acima admite variação espacial de U . Aplica-se então uma mudança da variável independente x da seguinte forma

$$dx' = dx - U dt \quad (2.28)$$

Observando-se que $\phi = \phi(x', t)$ tem-se a seguinte derivada

$$\left. \frac{\partial \phi}{\partial t} \right|_{x=const} = \frac{\partial \phi}{\partial x'} \frac{\partial x'}{\partial t} + \left. \frac{\partial \phi}{\partial t} \right|_{x'=const} = -U \frac{\partial \phi}{\partial x'} + \left. \frac{\partial \phi}{\partial t} \right|_{x'=const} \quad (2.29)$$

a qual pode ser substituída em (2.27), resultando

$$\frac{\partial \phi}{\partial t} - \frac{\partial}{\partial x'} \left(k \frac{\partial \phi}{\partial x'} \right) + Q(x') = 0 \quad (2.30)$$

e uma equação deste tipo possui um operador auto-adjunto e pode ser adequadamente discretizada espacialmente via Galerkin. O sistema de coordenadas descrito por (2.28) descreve as linhas ou direções características. Como simples corolário desta mudança de coordenadas tem-se que, na ausência de difusão ($k = 0$) e termos de fonte ($Q = 0$), a equação (2.27) torna-se simplesmente

$$\frac{\partial \phi}{\partial t} = 0 \quad (2.31)$$

para o caso unidimensional, ou, de outra forma

$$\phi(x') = \phi(x - Ut) = \text{constante} \quad (2.32)$$

ao longo de uma linha ou direção característica (considerando-se U constante). Esta é uma típica equação de onda propagando-se com velocidade U na direção x , como mostra

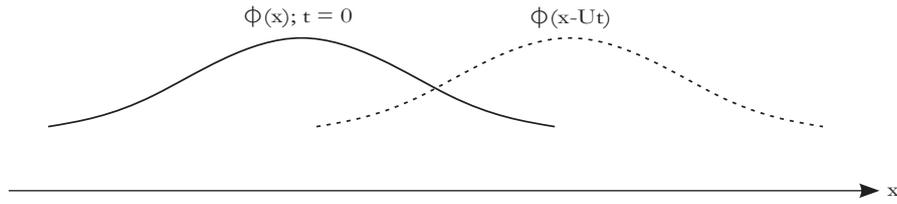


Figura 2.2: Natureza do comportamento em forma de onda.

a Fig. 2.2. A natureza do comportamento assemelhada à propagação de uma onda é evidente neste problema, mesmo para o caso em que a difusão não seja nula, onde deve-se obter uma onda que se atenua à medida que se propaga.

Nas condições anteriormente estabelecidas, pode-se escrever a seguinte igualdade

$$\phi(x')^{n+1} = \phi(x)^n \quad (2.33)$$

em que o super-índice indica tempo, ou seja, a variável propagada ao longo da linha ou direção característica é constante independentemente do tempo no qual a partícula é analisada. Na Fig. 2.3 é mostrada a propagação de linhas características no plano $x - t$ em que a velocidade de convecção é constante.

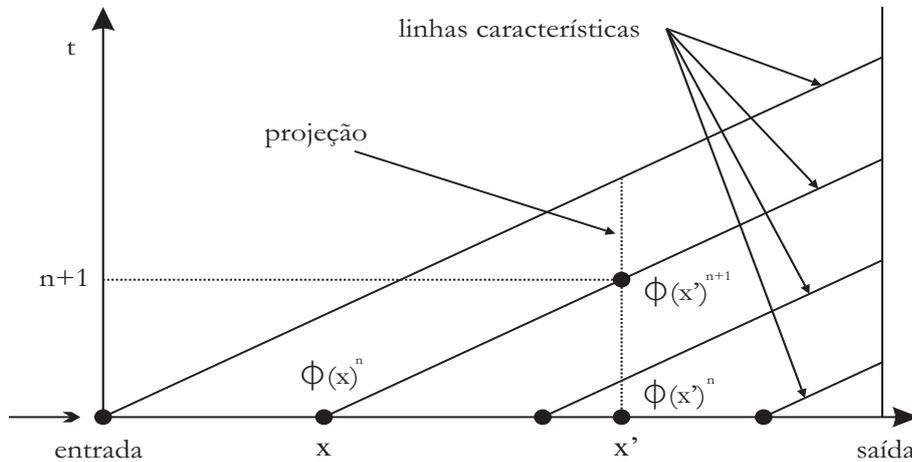


Figura 2.3: Problema de convecção linear. Linhas características.

A partir da equação de advecção escalar unidimensional ao longo da linha característica, dada em (2.31), é possível uma discretização no tempo da forma

$$\frac{\phi(x')^{n+1} - \phi(x)^n}{\Delta t} = 0 \quad (2.34)$$

em que Δt é o passo de tempo. Se $\phi(x')^{n+1}$ puder ser analiticamente representado em x , o problema de projeção é evitado. Alternativamente, $\phi(x)^n$ pode ser expressado em termos

de x' . É possível integrar (2.31) através de um método indireto; entretanto, um método explícito utilizando uma expansão de série de Taylor para determinar uma relação entre $\phi(x)^n$ em termos de quantidades em x' torna o problema significativamente simplificado (Nithiarasu et al., 2006). A seguinte expansão de Taylor é utilizada em um entorno $(x' - x)$ da distância entre $\phi(x')^n$ e $\phi(x)^n$ em um tempo n

$$\phi(x)^n = \phi(x')^n - (x' - x) \frac{\partial \phi(x')^n}{\partial x} + \frac{(x' - x)^2}{2} \frac{\partial^2 \phi(x')^n}{\partial x^2} - \frac{(x' - x)^3}{6} \frac{\partial^3 \phi(x')^n}{\partial x^3} + \dots \quad (2.35)$$

Na expressão acima, a distância $(x' - x)$ pode ser expressada em termos do passo de tempo e da velocidade média \bar{u} ao longo da linha ou direção característica na forma

$$x' - x = \bar{u} \Delta t \quad (2.36)$$

A eq. (2.35) pode ser substituída em (2.34), que, juntamente com (2.36), resulta na seguinte expressão

$$\frac{\phi(x')^{n+1} - \phi(x')^n}{\Delta t} = -\bar{u} \frac{\partial \phi(x')^n}{\partial x} + \frac{\Delta t \bar{u}^2}{2} \frac{\partial^2 \phi(x')^n}{\partial x^2} - \frac{\Delta t^2 \bar{u}^3}{6} \frac{\partial^3 \phi(x')^n}{\partial x^3} + \mathcal{O}(\Delta t^3) \quad (2.37)$$

Embora deduzida para o caso de advecção linear, a eq. (2.37) pode ser estendida ao caso não-linear ao se considerar a seguinte aproximação para $u\phi(x)$

$$u\phi(x)^n = u\phi(x')^n - (x' - x) \frac{\partial}{\partial x} (u\phi(x')^n) + \frac{(x' - x)^2}{2} \frac{\partial^2}{\partial x^2} (u\phi(x')^n) - \dots \quad (2.38)$$

Considerando-se a aproximação $\Delta t = \frac{(x' - x)}{u}$ e substituindo-se na eq. (2.34), obtém-se:

$$\frac{\phi(x')^{n+1} - \phi(x')^n}{\Delta t} = -\frac{\partial}{\partial x} (u\phi(x')^n) + \frac{\Delta t}{2} u \frac{\partial}{\partial x} \left[\frac{\partial}{\partial x} (u\phi(x')^n) \right]^n + \mathcal{O}(\Delta t^2) \quad (2.39)$$

A equação acima é a forma conservativa do método CBS explícito para um problema de advecção escalar. Para um problema de advecção-difusão, o método CBS, sem termos de alta ordem, tem a seguinte forma

$$\begin{aligned} \frac{\phi(x')^{n+1} - \phi(x')^n}{\Delta t} = & -\frac{\partial}{\partial x} (u\phi(x')^n) + \frac{\partial}{\partial x} \left(k \frac{\partial \phi}{\partial x} (x') \right)^n \\ & + \frac{\Delta t}{2} u \frac{\partial}{\partial x} \left[\frac{\partial}{\partial x} (u\phi(x')^n) - \frac{\partial}{\partial x} \left(k \frac{\partial \phi}{\partial x} (x') \right) \right]^n + \mathcal{O}(\Delta t^2) \end{aligned} \quad (2.40)$$

Os termos de ordem maior ou igual a 3 na equação acima podem ser negligenciados para elementos com função de interpolação lineares, entretanto, para elementos de ordem superior devem ser incluídos. Isto se deve ao fato de que, embora ao se considerar mais termos a equação torna-se de ordem maior no tempo, as derivadas espaciais

não acompanham esta discretização. Para problemas não escalares, há uma diferença fundamental devido ao fato de que a variável que sofre convecção não necessariamente se torna uma variável característica e a velocidade na qual ocorre essa convecção não necessariamente é a velocidade característica. A próxima seção trata do método CBS aplicado às equações de conservação de quantidade de movimento e energia.

2.2.2 Método CBS Aplicado à Mecânica de Fluidos

Para a utilização do método CBS nas equações de conservação aplica-se uma separação dos termos advectivos e difusivos. Considera-se inicialmente a equação da continuidade juntamente com a equação da conservação de quantidade de movimento a ser resolvida para as variáveis u_i e p . Com exceção do termo de pressão, esta equação é similar à equação de advecção-difusão. Este termo pode, contudo, ser tratado como uma quantidade conhecida (do tipo fonte), fornecendo uma forma independente de se calcular a pressão (Zienkiewicz et al., 2011). O objetivo aqui é que esta separação proposta forneça uma forma de se calcular uma solução (U_i^{n+1}, p^{n+1}) no tempo t^{n+1} a partir dos valores (U_i^n, p^n) no tempo t^n . Assume-se que o incremento de tempo é $\Delta t = t^{n+1} - t^n$. O fluxo de massa é então dividido em duas parcelas (Zienkiewicz et al., 2011)

$$U_i^{n+1} = U_i^n + \Delta U_i^* + \Delta U_i^{**} \quad (2.41)$$

Discretiza-se no tempo também a aproximação para o intervalo $t^n \leq t \leq t^{n+1}$

$$\frac{\partial U_i^n}{\partial t} = \frac{U_i^{n+1} - U_i^n}{\Delta t} = \frac{\Delta U_i^*}{\Delta t} + \frac{\Delta U_i^{**}}{\Delta t} \quad (2.42)$$

Utilizando-se a eq. (2.40), estendendo-a ao caso multidimensional e substituindo-se u por u_j e ϕ por U_i obtém-se

$$U_i^{n+1} - U_i^n = \Delta t \left[-\frac{\partial}{\partial x_j} (u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} + (\rho g_i) \right]^n - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k^n \frac{\partial}{\partial x_k} \left[\frac{\partial}{\partial x_j} (u_j U_i) - \frac{\partial \tau_{ij}}{\partial x_j} - \rho g_i \right]^n + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial p^{n+\theta_2}}{\partial x_i} \right) \quad (2.43)$$

Na equação acima adotou-se

$$\frac{\partial p^{n+\theta_2}}{\partial x_i} = \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \quad (2.44)$$

onde $0 \leq \theta_2 \leq 1$, sendo $\theta_2 = 0$ para um esquema explícito e $\theta_2 > 0$ para esquemas semi-implícitos e com

$$\Delta p = p^{n+1} - p^n \quad (2.45)$$

Observa-se que a velocidade e os termos da tensão viscosas são calculados em t^n ao passo que a pressão é calculada em $t^{n+\theta_2}$. Utilizando as variáveis auxiliares ΔU_i^* e ΔU_i^{**} divide-se a eq. (2.43) em duas partes. Neste ponto há pelo menos duas alternativas. A primeira é eliminar a pressão do cálculo inicial para ser posteriormente introduzida. A segunda alternativa é omitir apenas o gradiente de pressão no começo do passo de tempo no cálculo inicial, o qual é re-introduzido novamente em outro passo (Zienkiewicz et al., 2011). Utiliza-se a segunda alternativa mencionada no presente trabalho, com a omissão do gradiente de pressão no começo do primeiro passo de tempo. Para a solução das equações de conservação, o método CBS pode ser resumido nos 4 passos seguintes (Zienkiewicz et al., 2011):

1. Resolver as equações de conservação de momento omitindo o gradiente de pressão no início do passo para calcular ΔU_i^* ,
2. Calcular o incremento de massa $\Delta \rho$ ou de pressão Δp ,
3. Calcular a correção das velocidades ΔU_i^{**} para finalmente calcular o valor de U_i em t^{n+1} ,
4. Calcular variáveis escalares adicionais como temperatura e massa específica através da equação de conservação da energia.

Para o passo 1 tem-se a seguinte equação a ser resolvida

$$\Delta U_i^* = \Delta t \left[-\frac{\partial}{\partial x_j} (u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j} (u_j U_i) + \frac{\partial p}{\partial x_i} - \rho g_i \right) \right]^n \quad (2.46)$$

No passo 2 calcula-se o incremento de massa ou de pressão

$$\Delta \rho = \left(\frac{1}{c^2} \right)^n \Delta p = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \left(\frac{\partial^2 p^n}{\partial x_i^2} + \theta_2 \frac{\partial^2 \Delta p}{\partial x_i^2} \right) \right] \quad (2.47)$$

onde $1/2 \leq \theta_1 \leq 1$. Conhecido o incremento de massa ou de pressão, calcula-se a correção das velocidades no passo 3

$$\Delta U_i^{**} = -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \quad (2.48)$$

Finalmente, calcula-se U_i em t^{n+1} . Resta apenas o passo 4 para atualização das variáveis escalares, ou seja

$$\begin{aligned} \frac{(\rho E)^{n+1} - (\rho E)^n}{\Delta t} = & - \frac{\partial}{\partial x_i} (u_i \rho E)^n + \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right)^n - \frac{\partial}{\partial x_i} (u_i p)^n + \frac{\partial}{\partial x_i} (\tau_{ij} u_j)^n - \rho g_i u_i^n \\ & + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left[\frac{\partial}{\partial x_i} (u_i \rho E) - \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right) \right. \\ & \left. + \frac{\partial}{\partial x_i} (u_i p) - \frac{\partial}{\partial x_i} (\tau_{ij} u_j) + \rho g_i u \right]^n \end{aligned} \quad (2.49)$$

Desta forma, tem-se que o método proporciona um operador que é auto-adjunto nas variáveis Δp (ou $\Delta \rho$) para uma discretização espacial do tipo Galerkin e, portanto, tal discretização é empregada de forma otimizada. É importante observar que a sequência de passos utilizada no algoritmo CBS promove uma forma eficiente de se revolver as equações de conservação com adequada estabilização por convecção. Como resultante, o método é adequado para resolver problemas de escoamentos em ampla faixa de velocidade, desde problemas subsônicos, transônicos, supersônicos e até hipersônicos. A forma conservativa empregada na formulação garante que as ondas de choque serão posicionadas corretamente e que a solução encontrada (se houver) será única (Zienkiewicz et al., 2011). Observa-se que mesmo em problemas de escoamentos supersônicos, existem regiões de estagnação onde, de fato, não há grande compressibilidade. Este tipo de região causa grandes oscilações e instabilidades em outros métodos, como por exemplo o de Taylor-Galerkin, gerando também perda de velocidade de convergência em problemas de escoamentos compressíveis de baixas velocidades. Cabe ressaltar também que para malhas convencionais é necessário algum tipo de amortecimento numérico na região próxima aos choques por razões de estabilidade numérica. Este tópico é discutido em outra seção deste trabalho.

2.2.3 Discretização Espacial

Uma discretização espacial padrão do tipo Bubnov-Galerkin é empregada sobre as equações correspondentes aos 4 passos do método CBS. Utiliza-se neste trabalho apenas elementos triangulares e tetraédricos lineares para problema bidimensionais e tridimensionais, respectivamente. A discretização espacial das variáveis utiliza as

seguintes aproximações

$$\begin{aligned} U_i &= \mathbf{N}_u \tilde{U}_i, & \Delta U_i &= \mathbf{N}_u \Delta \tilde{U}_i, & \Delta U_i^* &= \mathbf{N}_u \Delta \tilde{U}_i^*, & \Delta U_i^{**} &= \mathbf{N}_u \Delta \tilde{U}_i^{**} \\ u_i &= \mathbf{N}_u \tilde{u}_i, & p &= \mathbf{N}_p \tilde{p}, & \rho &= \mathbf{N}_\rho \tilde{\rho} \end{aligned} \quad (2.50)$$

onde o símbolo \sim acima da variável indica uma quantidade nodal, sendo \mathbf{N}_i as funções de interpolação tal que

$$\begin{aligned} \tilde{U}_i &= [U_i^1, U_i^2, \dots, U_i^a, \dots, U_i^m]^T, & i &= 1, 2, 3 \\ \mathbf{N}_j &= [N_j^1, N_j^2, \dots, N_j^a, \dots, N_j^m] & j &= u, p, \rho \end{aligned} \quad (2.51)$$

em que a é o nó (ou variável), o qual varia entre 1 e m . As equações obtidas para o método CBS aplicado às equações de conservação são agora ponderadas por \mathbf{N}^T e integradas sobre o domínio. Empregando-se as expressões dadas em (2.50) e integrando-se por partes, obtém-se a discretização final utilizada, sendo ela a seguinte:

- Passo 1

$$\begin{aligned} \mathbf{M} \Delta \tilde{U}_i^* &= \Delta t [(\mathbf{C} - \mathbf{S}) (\tilde{\mathbf{u}}_j \tilde{U}_i) + (\mathbf{K}_\tau - \mathbf{T}_\tau) \tau_{ij}]^n \\ &\quad + \frac{\Delta t^2}{2} \tilde{\mathbf{u}}_k [(\mathbf{K}_u - \mathbf{T}_u) (\tilde{\mathbf{u}}_j \tilde{U}_i) + (\mathbf{K}_p - \mathbf{T}_p) \tilde{p}]^n \end{aligned} \quad (2.52)$$

- Passo 2

$$\mathbf{M} \Delta \tilde{\rho} = \Delta t [(\mathbf{D} - \mathbf{R}_u) (\tilde{U}_i + \theta_1 \Delta \tilde{U}_i^*) - \Delta t \theta_1 \mathbf{K} \tilde{p}]^n \quad (2.53)$$

- Passo 3

$$\mathbf{M} \tilde{U}_i = \mathbf{M} \Delta \tilde{U}_i^* + \Delta t [(\mathbf{D} - \mathbf{R}_u) \tilde{p}]^n \quad (2.54)$$

- Passo 4

$$\begin{aligned} \mathbf{M} \Delta \tilde{\rho} \tilde{\mathbf{E}} &= \Delta t \left[(\mathbf{C} - \mathbf{S}) \tilde{\mathbf{u}}_j (\tilde{\rho} \tilde{\mathbf{E}} + \tilde{p}) + (\mathbf{T}_\tau - \mathbf{K}_\tau) \left(k \frac{\partial T}{\partial x_i} + \tau_{ij} \tilde{\mathbf{u}}_j \right) \right]^n \\ &\quad - \frac{\Delta t^2}{2} \left[\tilde{\mathbf{u}}_k (\mathbf{K}_u - \mathbf{T}_u) \tilde{\mathbf{u}}_j (\tilde{\rho} \tilde{\mathbf{E}} + \tilde{p}) \right]^n \end{aligned} \quad (2.55)$$

onde as matrizes e vetores que foram introduzidos são dados por

$$\mathbf{M} = \int_{\Omega} \mathbf{N}^T \mathbf{N} d\Omega \quad (2.56a)$$

$$\mathbf{C} = \int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_j} \mathbf{N} d\Omega \quad (2.56b)$$

$$\mathbf{D} = \int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_i} \mathbf{N} d\Omega \quad (2.56c)$$

$$\mathbf{K} = \int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_i} \frac{\partial \mathbf{N}}{\partial x_i} d\Omega \quad (2.56d)$$

$$\mathbf{K}_{\tau} = \int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_j} d\Omega \quad (2.56e)$$

$$\mathbf{K}_u = \int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_k} \frac{\partial \mathbf{N}}{\partial x_j} d\Omega \quad (2.56f)$$

$$\mathbf{K}_p = \int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_k} \frac{\partial \mathbf{N}}{\partial x_i} d\Omega \quad (2.56g)$$

$$\mathbf{R}_u = \int_{\Omega} \mathbf{N}^T \mathbf{N} n_i d\Omega \quad (2.56h)$$

$$\mathbf{S} = \int_{\Gamma} \mathbf{N}^T \mathbf{N} n_j d\Gamma \quad (2.56i)$$

$$\mathbf{T}_{\tau} = \int_{\Gamma} \mathbf{N}^T n_j d\Gamma \quad (2.56j)$$

$$\mathbf{T}_u = \int_{\Gamma} \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial x_j} n_k d\Gamma \quad (2.56k)$$

$$\mathbf{T}_p = \int_{\Gamma} \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial x_i} n_k d\Gamma \quad (2.56l)$$

Convém comentar as seguintes considerações, as quais são necessárias para se chegar à forma final das equações apresentadas acima. Os termos contendo variáveis de força de campo g_i foram desconsiderados. Os passos 2 e 3 são calculados explicitamente em termos da massa ou da pressão, ou seja, adota-se $\theta_2 = 0$. Termos referentes à conservação de energia e ao gradiente da temperatura no passo 4 são calculados aproximadamente a nível de elemento, utilizando-se as funções de forma para tal.

2.2.4 Tratamento de Problemas de Escoamentos Compressíveis

Para melhor desempenho do método CBS (bem como para outros métodos) em encontrar soluções suaves para problemas de escoamentos compressíveis numa ampla faixa de velocidade, tratamentos adicionais são requeridos. Entre eles, os aqui utilizados são a Lei de Sutherland para relacionar a viscosidade com a temperatura, a difusão artificial para captura de choques em problemas com número de Mach alto e suavização de variáveis

para problemas com número de Mach baixo.

Lei de Sutherland

Em escoamentos com altos gradientes de temperatura, como é o caso de escoamentos compressíveis, a viscosidade e a condutividade térmica não podem ser consideradas constantes, mas sim função da temperatura. Adota-se neste trabalho a Lei de Sutherland (Hirsch, 1988) a qual estima a viscosidade em função da temperatura através da seguinte relação, válida para o ar

$$\mu = \frac{1.45T^{3/2}}{T + 110} \times 10^{-6} \quad (2.57)$$

onde as unidades são as do Sistema Internacional (SI).

Difusão Artificial

Em problemas transônicos, supersônicos e de velocidade mais altas, o escoamento apresenta um efeito distinto ao dos escoamentos subsônicos e escoamentos de fluidos incompressíveis com a formação de ondas de choques em certas regiões do domínio. Claramente, utilizando-se aproximações de elementos finitos nas quais todas as variáveis são interpoladas utilizando-se continuidade C_0 não é possível a exata reprodução de choques. Em tal aproximação, os choques são representados simplesmente como regiões de alto gradiente. A situação ideal é obtida se esta alta variação for confinada a uma região onde realmente ocorre a formação do choque. Infelizmente, esta aproximação da descontinuidade introduz oscilações locais as quais podem persistir por uma área considerável do domínio. Por esta razão, introduz-se na análise de elementos finitos viscosidades adicionais as quais ajudam a amortecer as oscilações causadas pelos choques no intuito de tentar reproduzir a descontinuidade o melhor possível. Adota-se neste trabalho a seguinte metodologia para corrigir as variáveis de conservação Φ para os casos de escoamentos compressíveis bidimensionais (Zienkiewicz et al., 2011)

$$\tilde{\Phi}_s^{n+1} = \tilde{\Phi}^{n+1} + \Delta t \mathbf{M}_L^{-1} \frac{C_e S_e}{\Delta t_e} (\mathbf{M} - \mathbf{M}_L) \tilde{\Phi}^n \quad (2.58)$$

onde o subíndice s indica a variável suavizada, C_e é uma constante especificada pelo usuário, \mathbf{M}_L é a matriz de massa discreta obtida da forma

$$M_{L,ab} = \delta_{ab} \int_{\Omega} N_a d\Omega \quad (2.59)$$

sendo que o termo Δt_e é o passo de tempo local do elemento e S_e é o sensor de pressão do elemento. S_e é avaliado segundo a média dos sensores de pressão nodais do elemento S_a os quais são dados por

$$S_a = \frac{|\sum_e (p_a - p_k)|}{\sum_e |p_a - p_k|} \quad (2.60)$$

onde o índice a indica um nó e k indica os nós conectados ao nó a . Para escoamentos compressíveis tridimensionais utiliza-se a seguinte formulação para captura dos choques, a qual baseia-se no Hessiano da pressão (Nithiarasu et al., 1998)

$$\tilde{\Phi}_s^{n+1} = \tilde{\Phi}^{n+1} - \Delta t \mathbf{M}_L^{-1} C_e h^3 \frac{|\mathbf{u}| + c}{\hat{p}} \left| \frac{\partial^2 p}{\partial x_i^2} \right|_e^n \left(\int_{\Omega} \frac{\partial \mathbf{N}^T}{\partial x_i} \frac{\partial \mathbf{N}}{\partial x_i} d\Omega \right) \tilde{\Phi}^n \quad (2.61)$$

em que h é o tamanho característico do elemento, \hat{p} é a pressão média no elemento e o subíndice e indica um elemento. Observa-se que as duas metodologias apresentam características importantes que devem estar presentes em qualquer tipo de difusão artificial: ao passo que o tamanho do elemento ou o gradiente de pressão tende a zero, a difusão artificial também deve tender a zero. Além disto, o sensor de pressão ajuda a limitar a região a ser suavizada a apenas a região no entorno do choque. Um dos tipos de difusão artificial acima mencionado é adicionado ao final do passo 1, 2 e 4 do método CBS em problemas transônicos e supersônicos.

Suavização de Variáveis à Baixas Velocidades

Em problemas subsônicos ($M < 0.8$) torna-se conveniente incluir um procedimento para suavização das variáveis se o acoplamento entre energia e outras variáveis existir. Outra possibilidade seria a de utilizar aproximações isotérmicas para tal objetivo. Entretanto, para resolver-se um problema compressível em regimes de baixo número de Mach sem a remoção do acoplamento térmico utiliza-se neste trabalho a seguinte suavização proposta por Thomas e Nithiarasu (2005)

$$\Phi_S = \left[\frac{1}{1 + 0.5\alpha_S} \Phi + \frac{\alpha_S}{1 + 0.5\alpha_S} (\mathbf{M} - \mathbf{M}_D) \mathbf{M}_L^{-1} \Phi \right] \quad (2.62)$$

onde α_S é um parâmetro de suavização o qual varia entre 0 e 0.05 e \mathbf{M}_D é a matriz \mathbf{M} sem os termos não-diagonais.

2.2.5 Incremento de Tempo

O método explícito é condicionalmente estável e portanto é importante adotar-se um passo de tempo que seja inferior ou no máximo igual ao passo de tempo crítico. Para problemas permanentes, a solução final deve ser independente do tempo. Portanto, a variável tempo é puramente computacional. Neste sentido, adota-se neste trabalho diferentes incrementos de tempo para a malha (passos de tempo locais), de forma que cada nó possua seu próprio incremento. Este procedimento conduz a uma grande aceleração na solução de problemas estacionários. O uso de um passo de tempo mínimo para a malha em problemas permanentes conduz a soluções demasiadamente mais custosas, sem qualquer ganho físico sobre a solução final encontrada (Löhner, 2008). No entanto, para problemas transientes, torna-se necessário adotar para o cálculo do incremento de tempo Δt o incremento mínimo de tempo global da malha (passo de tempo global). O incremento de tempo é calculado utilizando-se o valor mínimo de incremento de tempo entre os incrementos convectivos Δt_{conv} e difusivos Δt_{diff}

$$\Delta t = \min(\Delta t_{conv}, \Delta t_{diff}) CFL \quad (2.63)$$

onde

$$\Delta t_{conv} = \frac{h}{c + |\mathbf{u}|} \quad (2.64)$$

e

$$\Delta t_{diff} = \frac{h^2}{2\nu} \quad (2.65)$$

em que o termo h é uma dimensão característica do elemento. O termo CFL é o número de Courant–Friedrichs–Lewy, o qual tem um papel de coeficiente de segurança neste caso (adotado entre 0 e 1, dependendo do problema e da malha adotada). A condição Courant–Friedrichs–Lewy implica em um limite de passo de tempo máximo, no qual a física do problema não é violada. Por ser demasiadamente custosa essa determinação, aplica-se uma redução sobre o passo de tempo incorporado sobre a forma do valor de CFL pré multiplicado pelo incremento de tempo (tipicamente um valor entre 0.1 e 0.4). É também preciso determinar a dimensão característica h para cada um dos nós da malha. Embora essa determinação não seja única, adota-se a seguinte metodologia neste trabalho para o caso bidimensional

$$h_i = \min(2A/\hat{L})_{ie} \quad (2.66)$$

onde o valor mínimo é selecionado entre o número de elementos ie conectados ao nó i , A é a área do elemento e \hat{L} indica o comprimento da aresta oposta ao nó i no elemento. Para o caso tridimensional adota-se

$$h_i = \min \left(3V/\hat{A} \right)_{ie} \quad (2.67)$$

onde V é o volume do elemento e \hat{A} indica a área da face oposta do nó i no elemento.

2.2.6 Simulações Permanentes e Transientes

Para o caso de simulação de problemas permanentes, os 4 passos do algoritmo CBS são realizados utilizando-se a matriz de massa \mathbf{M} na sua forma discreta (ou diagonalizada) \mathbf{M}_L e o programa é executado até um limite de incrementos de tempo ou até a convergência a qual deve atender à tolerância do resíduo objetivado R_T , dada por

$$R_T = \sqrt{\frac{\sum_{a=1}^{nnos} (\rho_a^{n+1} - \rho_a^n)^2}{\sum_{a=1}^{nnos} (\rho_a^{n+1})^2}} \quad (2.68)$$

onde $nnos$ indica o número de nós da malha e ρ_a é a massa específica do nó a . A tolerância adimensional, em geral, é reduzida até um valor mínimo próximo de 10^{-5} para se obter um estado permanente do fluxo. Para problemas transientes, a matriz de massa \mathbf{M} é utilizada na forma completa e calcula-se iterativamente as variáveis da seguinte forma (Löhner, 2008)

$$\Phi_{i+1} = \mathbf{M}_L^{-1} \mathbf{r}_n + \mathbf{M}_L^{-1} (\mathbf{M} - \mathbf{M}_L) \Phi_i \quad (2.69)$$

onde obtém-se o termo \mathbf{r}_n que representa o lado direito da igualdade das equações dos passos do método CBS iterada n vezes. Tipicamente, $n = 3$ é suficiente e aplica-se o método nos passos 2 e 4 quando utilizado (nos outros passos utiliza-se a matriz de massa discreta \mathbf{M}_L).

Capítulo 3

Estimativa de Erro

Para um procedimento adaptativo anisotrópico, é evidente que torna-se necessário uma estimativa de erro capaz de indicar os locais e direções em que esta anisotropia ocorrerá. Para isto, estima-se o erro em cada aresta da malha. O erro na aresta é estimado a partir do comprimento que a mesma apresenta em uma métrica Riemanniana. Isso torna-se possível quando a curvatura do espaço na métrica Riemanniana é proporcional à grandeza do erro a ser medido. A vantagem desta abordagem, além de possibilitar o tratamento anisotrópico, é a de estimar o erro de forma proporcional à grandeza do erro estimado de forma eficiente e de forma contínua (na forma integral, em contraponto a uma abordagem discreta). A utilização da métrica Riemanniana torna implicitamente necessário que se avalie o erro através de um tensor métrico, o qual é feito através da matriz Hessiana. A determinação do valor absoluto do erro é uma tarefa extremamente custosa ou mesmo impossível, não possuindo uma aplicação prática em termos de adaptação de malhas. Além disto, o erro exato torna-se desnecessário neste sentido, pois o processo adaptativo utiliza o erro apenas como um indicador. Neste capítulo, apresenta-se uma pequena descrição dos temas relacionados com medições de comprimentos na métrica Euclidiana e Riemanniana, a ligação entre curvatura e estimativa de erro, uma breve descrição da interpretação geométrica deste procedimento e, por fim, o procedimento de avaliação das matrizes Hessianas.

3.1 Métrica

Para tornar mais clara a abordagem adotada, apresenta-se nessa seção um resumo sobre noções de geometria diferencial, as quais são utilizadas para o processo adaptativo. O enfoque principal refere-se ao cálculo de distâncias para diferentes tipos de espaços: o espaço Euclidiano, espaços métricos Euclidianos e espaços métricos Riemannianos. Um estudo completo sobre esse assunto é encontrado em Carmo (1976). Adota-se a notação de \mathbf{a} , \mathbf{b} , \mathbf{u} , \mathbf{v} e \mathbf{x} para vetores ou pontos em \mathbb{R}^3 . As coordenadas vetoriais são denotadas por $\mathbf{x} = (x_1, x_2, x_3)$ e o produto escalar entre dois vetores \mathbf{u} e \mathbf{v} de \mathbb{R}^3 por $\langle \mathbf{u}, \mathbf{v} \rangle$.

3.1.1 Espaço Métrico Euclidiano

Um espaço métrico Euclidiano $(\mathbb{R}^3, \mathcal{M})$ é um espaço vetorial finito onde o produto escalar é definido através de uma matriz simétrica positiva-definida \mathcal{M}

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}} = \mathbf{u}^T \mathcal{M} \mathbf{v} \quad \text{para } (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^3 \times \mathbb{R}^3 \quad (3.1)$$

\mathcal{M} é escrita como uma matriz 3×3 a qual verifica as seguintes propriedades

1. (simétrica) $\forall (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^3 \times \mathbb{R}^3, \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}} = \langle \mathbf{v}, \mathbf{u} \rangle_{\mathcal{M}}$
2. (positiva) $\forall \mathbf{u} \in \mathbb{R}^3, \quad \langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{M}} \geq 0$
3. (definida) $\langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{M}} = 0 \Rightarrow \mathbf{u} = \mathbf{0}$

Estas propriedades garantem que \mathcal{M} defina um produto escalar. A matriz \mathcal{M} é também chamada de tensor métrico ou simplesmente métrica. O caso mais comumente utilizado de espaço métrico Euclidiano é o definido pela métrica igual à matriz identidade $\mathcal{M} = \mathbf{I}_3$ o qual expande o espaço Euclidiano canônico em \mathbb{R}^3 .

O produto escalar definido através de \mathcal{M} torna \mathbb{R}^3 um espaço vetorial normado $(\mathbb{R}^3, \|\cdot\|_{\mathcal{M}})$ e um espaço métrico vetorial $(\mathbb{R}^3, d_{\mathcal{M}}(\cdot, \cdot))$ auxiliado pela seguinte definição de norma e distância, respectivamente:

$$\bullet \quad \forall \mathbf{u} \in \mathbb{R}^3, \quad \|\mathbf{u}\|_{\mathcal{M}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{M}}}$$

$$\bullet \forall (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^3 \times \mathbb{R}^3, \quad d_{\mathcal{M}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_{\mathcal{M}}$$

Nestes espaços, o comprimento de um segmento de reta $\mathbf{ab} = [\mathbf{a}, \mathbf{b}]$ é dado pela distância entre suas extremidades: $\ell_{\mathcal{M}}(\mathbf{ab}) = d_{\mathcal{M}}(\mathbf{a}, \mathbf{b})$. É importante observar que esta propriedade é válida apenas no espaço Euclidiano.

3.1.2 Espaço Métrico Riemanniano

Em um espaço métrico Riemanniano, a métrica varia continuamente em todo o domínio Ω . Denota-se este espaço por $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Omega}$. Utiliza-se esta métrica para se calcular o comprimento das arestas no procedimento de estimativa de erro do presente trabalho. É importante observar que, em um espaço métrico Riemanniano, calcular o comprimento de um segmento de reta difere de calcular a distância entre as extremidades de seu segmento. De fato, a linha reta neste espaço deixa de ser o menor caminho entre dois pontos o qual é dado pela geodésica (curvatura). Para levar em consideração a variação da métrica ao longo da aresta, o comprimento da aresta é calculado através de integração. Assim, em um espaço métrico Riemanniano $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Omega}$, o comprimento da aresta \mathbf{ab} é calculado utilizando-se uma parametrização do tipo $\gamma(t) = \mathbf{a} + t\mathbf{ab}$ onde $t \in [0, 1]$

$$\ell_{\mathcal{M}}(\mathbf{ab}) = \int_0^1 \|\gamma'(t)\|_{\mathcal{M}} dt = \int_0^1 \sqrt{\mathbf{ab}^T \mathcal{M}(\mathbf{a} + t\mathbf{ab}) \mathbf{ab}} dt \quad (3.2)$$

Para a integração acima, utiliza-se a abordagem proposta por Castro-Díaz et al. (1997) na qual aproxima-se a variação como sendo linear, ou seja

$$\begin{aligned} \ell_{\mathcal{M}}(\mathbf{ab}) &= \int_0^1 \sqrt{\xi_0^2 + t(\xi_1^2 - \xi_0^2)} dt \\ &= \frac{2}{3} \frac{\xi_0^2 + \xi_0\xi_1 + \xi_1^2}{\xi_0 + \xi_1} \end{aligned} \quad (3.3)$$

em que $\xi_i = \sqrt{\mathbf{ab}^T \mathcal{M}(\mathbf{a} + t\mathbf{ab}) \mathbf{ab}} \Big|_{t=i}$, $i = 0, 1$.

A Fig. 3.1 mostra as curvas de isovalores de segmentos de mesmo comprimento até a origem (centro) para diferentes espaços métricos Riemannianos. A função plotada é $f(\mathbf{x}) = \ell_{\mathcal{M}}(\mathbf{ox})$ onde \mathbf{o} é a origem do plano. Os isovalores são isotrópicos para o caso de um espaço Euclidiano definido por \mathbf{I}_2 . Eles são anisotrópicos no caso de um espaço métrico Euclidiano definido por \mathcal{M} , apresentando duas direções principais bem definidas. No caso de um espaço métrico Riemanniano \mathbf{M} , todas as simetrias são perdidas.

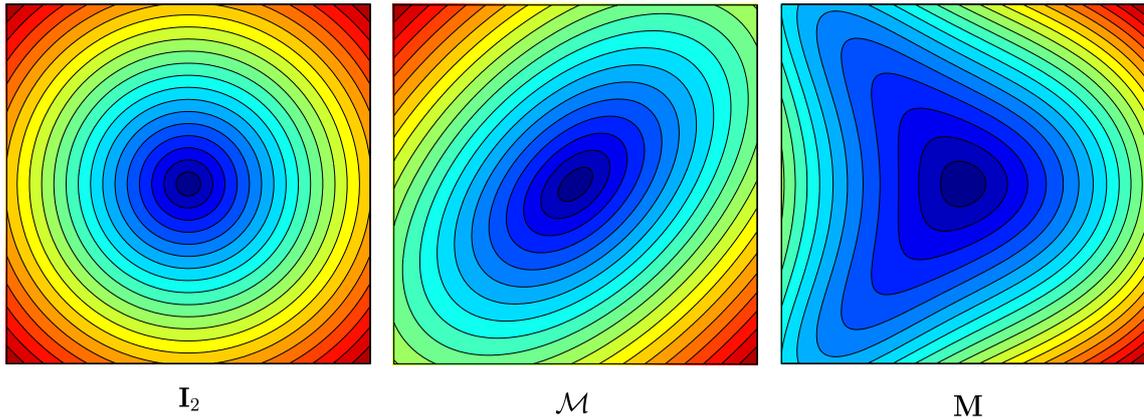


Figura 3.1: Isovalores da função $f(\mathbf{x}) = \ell_{\mathcal{M}}(\mathbf{o}\mathbf{x})$, onde \mathbf{o} é a origem do plano (centro), para diferentes espaços métricos Riemannianos. Na esquerda, o espaço Euclidiano canônico $([-1,1] \times [-1,1], \mathbf{I}_2)$. No centro, o espaço métrico Euclidiano $([-1,1] \times [-1,1], \mathcal{M})$ com \mathcal{M} constante. À direita, um espaço métrico Riemanniano $(\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in [-1,1]^2}$ com um tensor métrico variável.

3.2 Interpretação Geométrica

Existem diferentes formas de se interpretar uma métrica. A partir das propriedades que a matriz \mathcal{M} apresenta (simétrica, positiva-definida), pode-se reescrever a métrica \mathcal{M} através de uma decomposição espectral

$$\mathcal{M} = \mathcal{R}\mathbf{\Lambda}\mathcal{R}^T \quad (3.4)$$

onde \mathcal{R} é uma matriz ortonormal a qual satisfaz $\mathcal{R}^T\mathcal{R} = \mathcal{R}\mathcal{R}^T = \mathbf{I}_3$. \mathcal{R} é composta pelo autovetores associados à \mathcal{M} . $\mathbf{\Lambda}$ é uma matriz diagonal composta pelos autovalores associados à \mathcal{M} . Por ser positiva-definida, os autovalores $(\lambda_i)_{i=1,3}$ são estritamente positivos e diferentes de zero. A interpretação geométrica do tensor métrico é feita através de uma região na vizinhança $\mathcal{V}(\mathbf{a})$ de um ponto \mathbf{a} . Este conjunto de pontos, os quais estão a uma distância ε de \mathbf{a} , são dados por

$$\Psi_{\mathcal{M}}(\varepsilon) = \left\{ \mathbf{x} \in \mathcal{V}(\mathbf{a}) : (\mathbf{x} - \mathbf{a})^T \mathcal{M} (\mathbf{x} - \mathbf{a}) \leq \varepsilon^2 \right\} \quad (3.5)$$

É suficiente descrever apenas a região $\Psi_{\mathcal{M}}(1)$ uma vez que a região $\Psi_{\mathcal{M}}(\varepsilon)$ pode ser deduzida para todo ε por homogeneidade. Assumindo-se que a forma quadrática inicial $(\mathbf{x} - \mathbf{a})^T \mathcal{M} (\mathbf{x} - \mathbf{a})$ torna-se $(\tilde{\mathbf{x}} - \tilde{\mathbf{a}})^T \mathbf{\Lambda} (\tilde{\mathbf{x}} - \tilde{\mathbf{a}})$ através da decomposição espectral, então $\Psi_{\mathcal{M}}(1)$ é reescrito nesta base da seguinte forma (Loseille e Alauzet, 2011)

$$\Psi_{\mathcal{M}}(1) = \left\{ \tilde{\mathbf{x}} \in \mathcal{V}(\tilde{\mathbf{a}}) : \sum_{i=1}^3 \left(\frac{\tilde{x}_i - \tilde{a}_i}{h_i} \right)^2 \leq 1 \right\} \quad (3.6)$$

A última relação define um elipsóide centrado em \mathbf{a} com seus eixos alinhados segundo as direções principais de \mathcal{M} . Os comprimentos segundo essas direções são dados por $h_i = \lambda_i^{-\frac{1}{2}}$. Denota-se por $\mathcal{E}_{\mathcal{M}}$ este elipsóide. A Fig. 3.2 representa $\mathcal{E}_{\mathcal{M}}$. É interessante observar que quando a métrica Riemanniana é constante e igual à matriz identidade, isto é, $\mathcal{M} = \mathbf{I}_3$, então o elipsóide torna-se uma esfera. Este resultado é uma forma alternativa de demonstrar que um espaço Euclidiano canônico apresenta curvatura constante para todo o domínio.

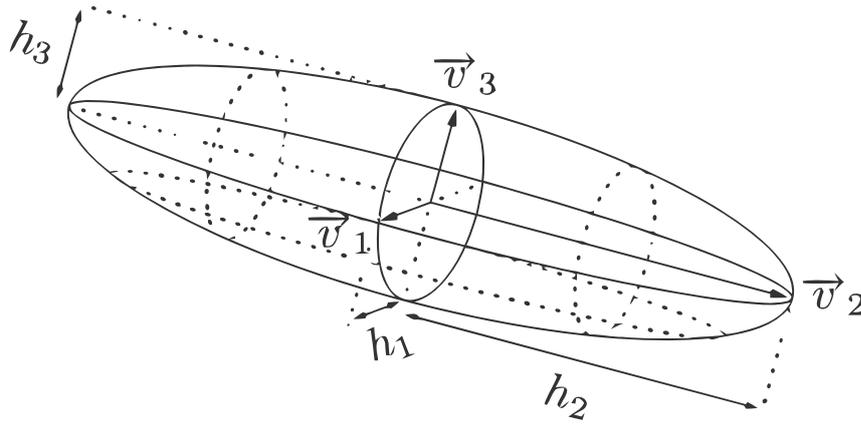


Figura 3.2: Interpretação geométrica de $\Psi_{\mathcal{M}}(1)$. v_i são os autovetores de \mathcal{M} e h_i^{-2} são os autovalores de \mathcal{M} .

3.3 Relacionando Métrica Riemanniana e Erro

A determinação do comprimento de uma aresta em uma métrica Riemanniana só é possível a partir da descrição de como varia a métrica $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Omega}$ ao longo do espaço. A ideia é tornar esta variação proporcional ao erro estimado. Desta forma, o comprimento da aresta na métrica Riemanniana assume o papel de curvatura, a qual é tão mais acentuada quanto maior o erro associado e orientada segundo as direções de anisotropia do erro.

Para um elemento tetraédrico (ou triangular) linear K , a solução para uma variável u é interpolada linearmente ao longo de seus vértices: $K = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$. Assume-se que o erro E para uma variável escolhida avaliada nos nós do elemento é dado pela diferença entre a solução exata u e a solução interpolada $\Pi_h u$ de u em K . Como a solução exata não é disponível, substitui-se a solução exata por uma função quadrática do tipo

$$\forall \mathbf{x} \in \Omega, \quad u(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} \quad (3.7)$$

em que \mathbf{H} é a matriz Hessiana. Desta forma, o erro máximo avaliado (diferença máxima entre a interpolação quadrática e linear) na norma L_2 avaliado na aresta \mathbf{ab} é dado por (Habashi et al., 2000)

$$E(\mathbf{ab}) = \|u - \Pi_h u\|_{L_2} \leq c_0 \|\mathbf{ab}\|^2 \mathbf{v}^T |\mathbf{H}| \mathbf{v} \quad (3.8)$$

onde \mathbf{v} são vetores unitários (segundo direção da aresta), $\|\mathbf{ab}\|^2$ é o módulo do comprimento da aresta \mathbf{ab} na métrica Euclidiana e c_0 uma constante indeterminada. A matriz Hessiana \mathbf{H} é simétrica e admite uma decomposição espectral. Nesta decomposição, toma-se o valor em módulo dos termos da matriz de autovalores associados, obtendo-se assim a matriz $|\mathbf{H}|$, a qual é positiva-definida.

Para relacionar a curvatura do espaço métrico Riemanniano com o erro, admite-se que a métrica \mathbf{M} possui variação espacial proporcional ao Hessiano positivo-definido $|\mathbf{H}|$

$$\mathbf{M}(\mathbf{x}) = |\mathbf{H}(\mathbf{x})| \quad (3.9)$$

e o comprimento desta aresta nesta métrica, avaliado de forma parameterizada, tal como em (3.2), indicará o erro estimado na aresta \mathbf{ab} e vem dada por (levando em conta a expressão (3.9)):

$$E(\mathbf{ab}) \equiv \ell_{\mathbf{M}}(\mathbf{ab}) = \int_0^1 \sqrt{(\|\mathbf{ab}\| \cdot \mathbf{v})^T |\mathbf{H}(\mathbf{a} + t \mathbf{ab})| (\|\mathbf{ab}\| \cdot \mathbf{v})} dt \quad (3.10)$$

a qual pode ser calculada através do procedimento de integração descrito na eq. (3.3). Quando a variação espacial é tomada como sendo igual à variável cujo erro é mensurado, ou seja, $\mathbf{x} = u$, fica completa a determinação da relação entre estimativa de erro e a curvatura (comprimento) avaliada em um espaço Riemanniano. Quanto menor a variância do erro ao longo do domínio calculado com a eq. (3.10), mais bem distribuído o erro se encontrará, sem grandes variações no domínio. O objetivo é tornar esta curvatura tão mais uniforme quanto possível em toda a malha (domínio). Valores muito elevados com relação à média do erro representam regiões em que o erro é grande, requerendo um processo de refinamento. Já valores muito inferiores à média indicam regiões as quais o erro estimado é muito baixo, podendo ser submetidas a processos de desrefinamento. É importante salientar que este procedimento estima o erro de apenas uma variável u escolhida. Os detalhes de como utilizar a estimativa de erro no processo de adaptação são explicados com mais detalhes no próximo capítulo.

3.4 Determinação da Matriz Hessiana

Para obter-se a estimativa do erro através da eq. 3.10, torna-se necessário a informação da matriz Hessiana \mathbf{H} . Entretanto, utiliza-se elementos lineares no presente trabalho, e tal informação não está diretamente disponível. Torna-se necessário um procedimento o qual aproxime o Hessiano. Uma possível alternativa para isto, a qual é adotada neste trabalho, consiste em determinar a primeira variação de uma dada variável u ao longo das direções x_k com k variando de 1 até 3 (ou 2, para o caso bidimensional), onde assume-se que esta derivada seja proporcional à função de forma N_i (Löhner, 2008)

$$\frac{\partial u}{\partial x_k} \approx \tilde{N}_i \frac{\partial \tilde{u}_i}{\partial x_k} \quad (3.11)$$

onde o subíndice i indica o nó do elemento. Do pressuposto sobre a variação de u tem-se também que

$$u \approx N_i \tilde{u}_i \quad (3.12)$$

Derivando-se a última expressão com relação às direções x_i , obtém-se

$$\frac{\partial u}{\partial x_k} \approx \frac{\partial N_i}{\partial x_k} \tilde{u}_i \quad (3.13)$$

Comparando-se as equações (3.13) e (3.11) pode-se escrever que

$$\tilde{N}_i \frac{\partial \tilde{u}_i}{\partial x_k} = \frac{\partial N_j}{\partial x_k} \tilde{u}_j \quad (3.14)$$

Sobre esta eq. (3.14) aplica-se uma função ponderadora W_i e integra-se no domínio Ω

$$\int_{\Omega} W_i \tilde{N}_j \frac{\partial \tilde{u}_j}{\partial x_k} d\Omega = \int_{\Omega} W_i \frac{\partial N_j}{\partial x_k} \tilde{u}_j d\Omega \quad (3.15)$$

Para o caso particular em que $W_i = \tilde{N}_i = N_i$, o qual corresponde ao método dos resíduos ponderados de Galerkin, as derivadas de primeira ordem são reduzidas à

$$\int_{\Omega} N_i N_j \frac{\partial u_j}{\partial x_k} d\Omega = \int_{\Omega} N_i \frac{\partial N_j}{\partial x_k} u_j d\Omega \quad (3.16)$$

Observa-se que no lado esquerdo da equação acima há a matriz de massa consistente. Para simplificar o cálculo, adota-se a matriz de massa diagonalizada neste procedimento, tornado a inversão direta. A eq. (3.16) fornece o valor da derivada de primeira ordem para cada nó da malha. Para isto, é preciso somar a contribuição de cada elemento que compartilha um determinado nó I para o cálculo das derivadas da variável u no nó I

segundo as direções x_k . Com isto, pode-se repetir todo o procedimento novamente, mas substituindo-se a variável u pela sua respectiva derivada em cada uma das direções x_k . O resultado final são as derivadas de segunda ordem, as quais são requeridas para o cálculo da matriz Hessiana \mathbf{H} . As derivadas de segunda ordem, quando calculadas desta forma, apresentam um erro mais acentuado nas regiões do contorno, onde termos de contorno deveriam ser empregados. Entretanto, para fins práticos, esse pequeno erro ocasionado nos contornos torna-se pouco relevante no contexto da adaptação de malhas (Zienkiewicz et al., 2011). Alguns autores como Pain et al. (2001) utilizam técnicas para melhorar o Hessiano nos contornos, entretanto o resultado final obtido para adaptação de malhas não parece apresentar diferenças significativas e portanto não incluem-se neste trabalho tratamentos adicionais neste sentido.

Capítulo 4

Adaptação Anisotrópica

Após realizada a estimativa do erro, infere-se sobre a discretização espacial localmente, refinando-se onde esta discretização é insuficiente e desrefinando-se onde o nível de refinamento é excessivo. Este capítulo apresenta a metodologia utilizada no presente trabalho, definindo inicialmente os conceitos topológicos e geométricos utilizados, os procedimentos de refinamento, desrefinamento, alternância de arestas e faces, os embasamentos teóricos relativos à convergência do método de Galerkin para elementos de alta distorção e, por fim, a metodologia do acoplamento com o código de solução de escoamentos compressíveis através do método CBS.

4.1 Geometria e Topologia

Os conceitos geométricos e topológicos fundamentais utilizados neste trabalho são apresentados a seguir, de forma genérica, no intuito de que a exposição fique clara para qualquer dimensão adotada.

4.1.1 Simplex

Denomina-se k -simplex ($0 \leq k \leq d$) o envólucro convexo de $k + 1$ pontos independentes em \mathbb{R}^d denominados vértices. Geometricamente, um 0 -simplex é um ponto, um 1 -simplex é um segmento de reta, um 2 -simplex é um triângulo, um 3 -simplex é um tetraedro e assim por diante. O presente trabalho aborda apenas triângulos e tetraedros

embora a extensão a outras dimensões seja direta. A partir deste ponto, toda vez que for referenciado um $k - simplex$, deve-se interpretar que k é válido para 2 ou 3.

Define-se V_K como o conjunto dos $k + 1$ vértices v de um dado $k - simplex$ K

$$V_K = \{v_0, \dots, v_k\} \quad (4.1)$$

Define-se A_K como o conjunto dos $3k - 3$ vetores aresta a de um dado $k - simplex$ K

$$A_K = \{a_0, \dots, a_{3k-4}\} \quad (4.2)$$

A medida $|T_K|$ de um dado $k - simplex$ K (área ou volume orientado) é dada por

$$|T_K| = \frac{\det \begin{pmatrix} v_1 - v_0 & v_2 - v_0 & \dots & v_{k-1} - v_0 & v_k - v_0 \end{pmatrix}}{k!} \quad (4.3)$$

Um $k - simplex$ K é dito não-degenerado se e somente se

$$|T_K| > 0 \quad (4.4)$$

Define-se $k - face$ de um simplex K como um $k - simplex$ subconjunto de K . Em particular, $2 - face$ é chamado de face, $1 - face$ é chamado de aresta e $0 - face$ é chamado de vértice. Adicionalmente, define-se $(-1) - face$ como um conjunto vazio.

4.1.2 Malha

Dado um conjunto finito \mathcal{N} de pontos (chamados nós) pertencentes a um domínio Ω e um conjunto \mathcal{T} de $k - simplexes$ não-degenerados (chamados elementos) cujos vértices pertencem à \mathcal{N} e denotando-se por \mathcal{F} o conjunto das $m - faces$ dos $k - simplexes$, então, $(\mathcal{N}, \mathcal{T})$ é uma malha de Ω se e somente se (Gruau, 2004):

1. cada $(k - 1) - face$ de \mathcal{F} partilha no máximo dois elementos de \mathcal{T}
2. a intersecção de dois elementos distintos de \mathcal{T} é uma $k - face$ de seus dois elementos tal que

$$k \in \{-1, 0, \dots, k - 1\} \quad (4.5)$$

- 3.

$$\bigcup_{T \in \mathcal{T}} T = \Omega \quad (4.6)$$

Também define-se que:

- o conjunto \mathcal{T} é também chamado de topologia da malha;
- a vizinhança de um nó S é o conjunto dos outros nós dos elementos da malha que compartilham S ;
- a fronteira (no sentido topológico) de uma malha é denotada por $\partial\mathcal{T}$, correspondendo ao conjunto de $(k - 1) - faces$ que compartilham apenas um elemento.

A seguir, serão descritos os procedimentos de alteração da topologia da malha adotados neste trabalho.

4.2 Adaptação-h

Nesta metodologia altera-se o número de nós (graus de liberdade) de uma dada malha através do procedimento de refinamento (inserção de nós) e desrefinamento (remoção de nós). Adicionalmente, aplica-se o procedimento de alternância de arestas e faces para melhorar a qualidade da malha. A adaptação do tipo h é empregada neste trabalho, cujos procedimentos são detalhados a seguir. Para captura da anisotropia do erro, a adaptação é conduzida a partir das arestas da malha.

4.2.1 Refinamento

Quando uma aresta é submetida ao refinamento, um novo nó é inserido no ponto médio da mesma. Deve-se também garantir que este procedimento gere elementos válidos, os quais não violem a topologia da malha \mathcal{T} (condições 1, 2 e 3 acima). Para isto, enquadra-se o particionamento de um elemento de acordo com a quantidade e posição das arestas a serem refinadas no mesmo. Para o caso de triângulos, apenas 4 casos são possíveis, totalizando um total de 8 casos de anisotropia (Fig. 4.1):

- *Nenhuma aresta a ser refinada*: neste caso, nada é preciso ser feito;

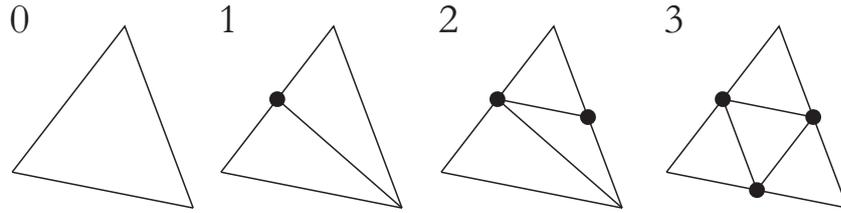


Figura 4.1: Configurações para subdivisão de triângulos.

- *Uma aresta a ser refinada*: o triângulo é dividido ao longo da linha que une o novo nó até o nó oposto do triângulo, gerando dois novos elementos;
- *Duas arestas a serem refinadas*: os dois novos nós são conectados. O quadrilátero remanescente é dividido partindo de algum dos dois novos nós até o nó oposto, gerando três novos elementos;
- *Três arestas a serem refinadas*: todos os novos nós são conectados, gerando quatro novos elementos.

A subdivisão dos elementos a partir das arestas é única e sempre será possível (Ruprecht e Müller, 1994) (condição de existência e unicidade). A metodologia, portanto, garante que a topologia da malha não seja violada. Para o caso de tetraedros, existem 11 possibilidades de subdivisão, totalizando um total de 64 casos de anisotropia, entretanto apenas 8 tipos de subdivisão são considerados neste trabalho, totalizando 43 casos possíveis de anisotropia. São eles (Fig. 4.2):

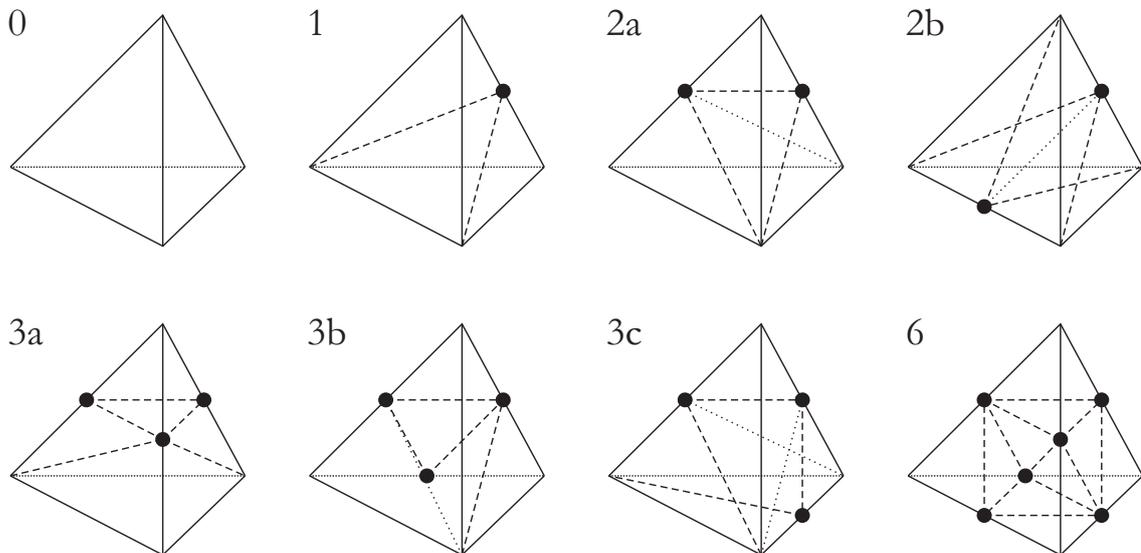


Figura 4.2: Configurações para subdivisão de tetraedros.

- *Nenhuma aresta a ser refinada:* neste caso, nada é preciso ser feito;
- *Uma aresta a ser refinada:* os dois triângulos adjacentes à aresta refinada são divididos pela metade. As duas novas arestas, juntamente com a aresta do tetraedro diagonalmente oposta à aresta refinada, formam um triângulo interno no tetraedro dividindo-o pela metade, gerando dois novos elementos;
- *Duas arestas a serem refinadas na mesma face:* o triângulo contendo as arestas refinadas é dividido em três triângulos. Cada um destes forma um novo tetraedro com o nó oposto do tetraedro original, gerando três novos elementos;
- *Duas arestas a serem refinadas em faces opostas:* todos os triângulos na superfície são divididos pela metade. Adiciona-se uma aresta interna nova conectando os dois novos nós, gerando-se quatro novos elementos;
- *Três arestas a serem refinadas que compartilham um nó:* as arestas que unem as arestas divididas separam uma porção do tetraedro original, dando origem a um tetraedro e um prisma. Este prisma pode ser subdividido em três tetraedros se as novas arestas as quais dividem o quadrilátero formam uma linha poligonal simples. Este sempre será o caso uma vez que o novo nó inserido na maior das três arestas a ser subdividida é comum a duas das diagonais adicionadas. Independentemente da orientação da terceira diagonal, ela será sempre conectada a uma das outras duas, gerando-se quatro novos elementos;
- *Três arestas a serem refinadas na mesma face:* um dos triângulos é dividido em quatro pelas arestas que conectam os três novos pontos. Cada um destes novos triângulos forma um novo tetraedro com o nó oposto, gerando-se ao final quatro novos elementos;
- *Três arestas a serem refinadas e que não compartilham um único nó ou face:* adiciona-se uma nova aresta interna conectando os novos nós diagonalmente opostos. Esta aresta, juntamente com as arestas criadas nos dois triângulos que contêm apenas um único novo nó, separa do tetraedro original um novo tetraedro e duas pirâmides deformadas as quais compartilham uma superfície triangular. Cada uma destas duas pirâmides podem ser partidas em dois novos tetraedros independentes, gerando-se ao total cinco novos elementos;

- *Seis arestas a serem refinadas:* partindo-se em um novo tetraedro cada vértice do elemento, resta um octaedro no interior, o qual pode ser dividido em quatro com a adição de uma nova aresta ligando dois nós diagonalmente opostos, gerando-se ao total oito novos elementos.

Assim como no caso dos triângulos, a subdivisão dos tetraedros baseados nas arestas é única e sempre possível. Os casos de quatro e cinco arestas a serem refinadas em um tetraedro são automaticamente forçadas ao caso de seis arestas, atualizando-se elementos vizinhos, entretanto é perfeitamente possível a implementação destes casos de refinamento.

4.2.2 Desrefinamento

O procedimento de desrefinamento adotado neste trabalho é o de colapso de arestas seguida por degeneração total de elementos. Entre as vantagens principais do método tem-se o baixo custo computacional e a independência de malha para desrefinamento, isto é, não se trata de um procedimento que anula um processo de refinamento realizado, retornando apenas à malha original, mas sim um procedimento capaz de desrefinar qualquer porção e nível da malha, seja ela a inicial ou uma porção já previamente refinada (Trotts et al., 1998). Como desvantagem pode-se citar que os elementos gerados pelo procedimento não possuem qualquer tipo de controle de qualidade, diferentemente de metodologias tais como triangulações Delaunay, que visam gerar elementos com condições angulares mais bem distribuídas (Cutler et al., 2004). Entretanto, demonstrar-se-á em um dos subcapítulos seguintes que estas condições angulares não representam problemas reais para a estabilidade do método dos elementos finitos. Além disto, a triangulação Delaunay clássica e outros procedimentos análogos sofrem fortemente com a precisão numérica bem como instabilidades e dualidades de solução que podem ocorrer devido à necessidade de tratamento de diversos procedimentos da geometria analítica via operações com pontos flutuantes (Löhner, 2008), procedimentos estes muito menos preponderantes na metodologia implementada no presente trabalho.

Por motivos de clareza, o procedimento é demonstrado a seguir para o caso de triângulos, sendo sua extensão para o caso de tetraedros direta, mas muito mais complexa de se visualizar. O procedimento parte da aresta a a ser desrefinada. Esta aresta possui

nó inicial v_1 e nó final v_2 . Define-se Q como o conjunto de todos os elementos que compartilham uma aresta com a

$$Q = \{K : a \in A_K\} \quad (4.7)$$

e define-se o conjunto B dos elementos que compartilham o nó v_1 mas não pertençam ao conjunto Q (Fig. 4.3)

$$B = \{K : v_1 \in V_K \text{ e } a \notin A_K\} \quad (4.8)$$

O colapso da aresta é realizado inicialmente de v_1 para v_2 , reduzindo a aresta a a um

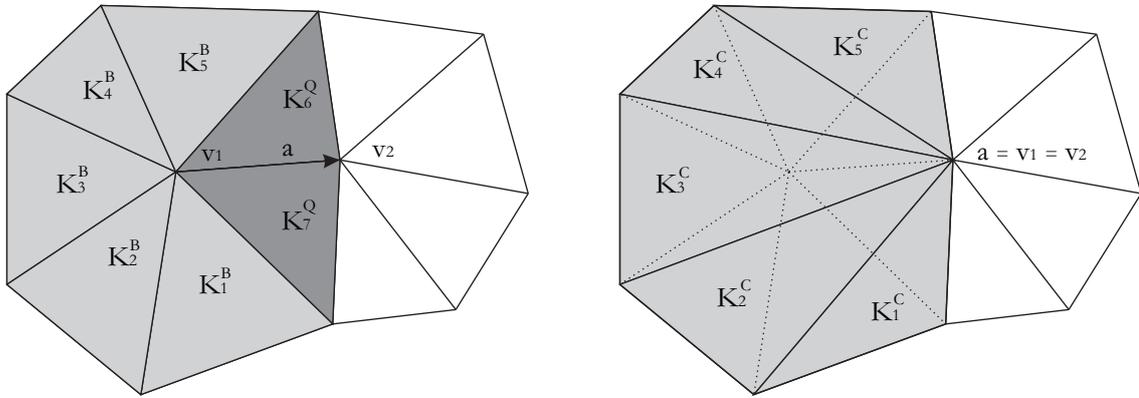


Figura 4.3: Desrefinamento por colapso de aresta seguida de degeneração de elementos adjacentes.

ponto $a = v_2 = v_1$. Degenera-se os elementos pertencentes ao conjunto Q . Desta forma, os elementos originais pertencentes ao conjunto B transformam-se em um novo conjunto C através da seguinte transformação linear

$$C = \{K \in B : (v_1 \in V_K) \mapsto v_2\} \quad (4.9)$$

Mas para realizar o colapso, a condição de não-degeneração deve ser necessariamente verificada

$$|T_K| \quad \forall K \in C > 0 \quad (4.10)$$

bem como a condição de conservação da massa após a aplicação da transformação linear

$$\sum_{K \in Q \cup B} |T_K| = \sum_{K \in B^C} |T_K| \quad (4.11)$$

Em caso da não verificação das duas condições acima, verifica-se a possibilidade de colapsar a aresta a no sentido contrário, de v_2 para v_1 , verificando-se novamente as condições necessárias. No caso em que mesmo assim não se verifiquem as condições,

a aresta não poderá ser colapsada. Como corolário da segunda condição a ser verificada, tem-se que uma aresta que possui apenas um único nó pertencente ao contorno do domínio só pode ser colapsada para o nó do contorno.

4.2.3 Alternância de Arestas

O procedimento de alternância de arestas (em inglês *Edge Swapping*), ou também chamado de reconecção de malha é um procedimento que modifica a conexão existente entre elementos vizinhos, mantendo fixo o número de nós da malha. Pode-se implementar o procedimento para o caso de malhas compostas por triângulos (bidimensionais) bem como para tetraedros (tridimensionais). No caso de triângulos, existe apenas um tipo de alternância em contraponto ao caso dos tetraedros, onde infinitas possibilidades surgem em decorrência da ilimitada possibilidade de elementos pertencerem a uma determinada aresta. Implementou-se a alternância de arestas apenas em casos bidimensionais neste trabalho. O procedimento visa melhorar a qualidade dos elementos da malha. Entretanto, o critério de qualidade de elemento é adotada por cada autor na literatura de formas diferentes. A definição de qualidade de um elemento mais comumente adotada é a de que quanto mais bem distribuídos estiverem os ângulos do mesmo, melhor sua qualidade (algumas vezes ponderada pelo volume ou área, uma forma indireta de medição de ângulos sólidos). A qualidade do elemento medido desta forma é portanto otimizado por triangularizações do tipo Delaunay. Porém, do ponto de vista do método dos elementos finitos, elementos distorcidos e, portanto, em contraste ao critério de qualidade de elementos com ângulos bem distribuídos, conduzem à solução correta e convergência do método se condições angulares máximas forem respeitadas. Mais do que isso, elementos distorcidos poupam recursos computacionais pois descrevem a porção do domínio de forma mais eficiente, quando distorcidos segundo as direções de anisotropia dos campos estudados (a estabilidade do método nestas condições é discutida mais adiante).

O procedimento de alternância de arestas (no caso de malhas compostas por elementos triangulares) pode ser realizado em arestas que não pertençam ao contorno $\partial\mathcal{T}$ da malha. Estas arestas a compartilham dois elementos K_1 e K_2 . A alternância consiste em trocar a posição da aresta interna do subdomínio composto por $K_1 \cup K_2$, como indica a Fig. 4.4. Portanto, tanto o número de elementos quanto o número de nós da malha permanece

constante neste caso. O critério utilizado para que se realize o procedimento é o de que

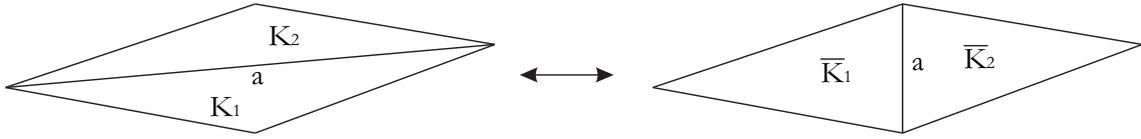


Figura 4.4: Alternância de arestas.

os elementos novos \bar{K} gerados devam possuir ângulos internos α_i ($i = 1,2,3$) máximos menores que os elementos originais K

$$\max \{ \alpha_i \in (\bar{K}_1 \cup \bar{K}_2) \} < \max \{ \alpha_i \in (K_1 \cup K_2) \} \quad (4.12)$$

A alternância só pode ser realizada caso o critério de não-degeneração seja verificado; em caso contrário, a topologia é violada e o procedimento é inconsistente.

4.2.4 Alternância de Faces

O procedimento de alternância de faces (em inglês *Face Swapping*) é análogo ao de alternância de arestas. Porém, só é possível em malhas tridimensionais. Outra diferença fundamental reside na possibilidade de alternar faces pertencentes ao contorno. É o caso apresentado na Fig. 4.5 onde dois elementos vizinhos contendo uma face de contorno são substituídos por outros dois após alternância da face interna. Para dois elementos internos, os quais compartilhem uma face, o procedimento de alternância de face pode ser aplicado ao inserir-se uma nova aresta interna, resultando três novos elementos (Freitag e Ollivier-Gooch, 1997) (o número de elementos da malha varia mas o de nós permanece constante). O procedimento inverso também é possível, ou seja, passar de três para dois elementos através da remoção de uma aresta, como mostra a Fig. 4.6. Tais procedimentos foram

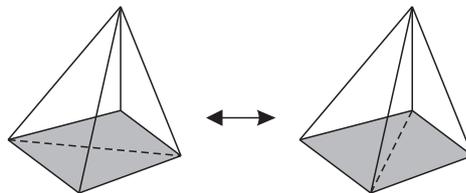


Figura 4.5: Alternância de faces de elementos no contorno.

implementados no presente trabalho segundo o critério de diminuição de ângulo (neste caso usou-se o ângulo diedral) de forma análoga ao apresentado na eq. 4.12 estendida ao caso trimensional.

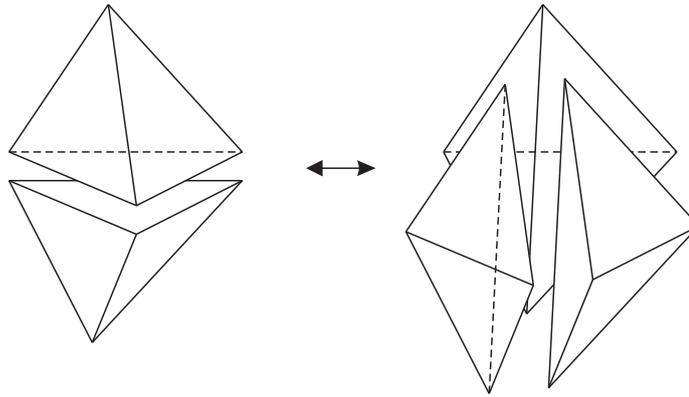


Figura 4.6: Alternância de faces de elementos no interior do domínio.

4.3 Condições Angulares

Até o presente momento, apenas a questão topológica do processo adaptativo foi abordado. A adaptação anisotrópica pode modificar a topologia da malha de elementos finitos de forma muito acentuada, gerando elementos com ângulo internos próximos de 0 e π . Por isto, a presente seção visa explicar até que ponto o método dos elementos finitos é capaz de convergir para a solução da equação (ou sistema de equações) diferencial da qual se busca a solução em função dos ângulos extremos presentes na malha. A questão é discutida a seguir.

4.3.1 Condição Angular Mínima

Também chamada de condição de Zlámal, foi originalmente concebida para problemas de valor de contorno elípticos com coeficientes suaves aproximada por elementos finitos triangulares quadráticos, mas cuja solução é análoga para elementos lineares (Babuska e Aziz, 1976). Assume-se que o domínio Ω possui uma solução suficientemente suave u a qual é aproximada por uma solução linear contínua por partes de elementos finitos u_h em uma triangularização \mathcal{T}_h sobre Ω . Zlámal (1968) indica que a convergência de u_h para u quando $h \rightarrow 0$ depende da chamada condição angular mínima

$$\exists \alpha_0 > 0 \quad \forall \mathcal{T}_h \in \mathcal{F} \quad \forall K \in \mathcal{T}_h \quad : \quad \alpha_K \geq \alpha_0 \quad (4.13)$$

onde α_K é o menor ângulo de um triângulo K e $\mathcal{F} = \{\mathcal{T}_h\}_{h \rightarrow 0}$ é uma família de triangularizações. A mesma constante $\alpha_0 > 0$ é utilizada para definir de forma análoga a condição para elementos tetraédricos lineares (Brandts et al., 2008), os quais devem

possuir ângulos diedrais α_D (ângulos entre faces) e ângulos sólidos φ (entre arestas) tais que

$$\alpha_D \geq \alpha_0 \quad \text{e} \quad \varphi \geq \alpha_0 \quad (4.14)$$

4.3.2 Condição Angular Máxima

A condição angular máxima para elementos triangulares lineares foi formulada por Babuska e Aziz (1976) e é definida por

$$\exists \gamma_0 < \pi \quad \forall \mathcal{T}_h \in \mathcal{F} \quad \forall K \in \mathcal{T}_h \quad : \quad \gamma_K \leq \gamma_0 \quad (4.15)$$

onde γ_K é o maior ângulo de K . Como corolário, a condição angular mínima implica na condição angular máxima, embora a implicação inversa não seja verdadeira, por isso diz-se que a condição máxima é mais generalizada e demonstra que a condição angular mínima não é essencial para a convergência do método dos elementos finitos. Mais do que isso, se a condição 4.15 é verdadeira (portanto uma condição suficiente) então a seguinte relação também é válida

$$\|u - u_h\|_1 \leq Ch|u|_2 \quad (4.16)$$

em que $|\cdot|_k$ denota a semi-norma no espaço Sobolev $H^k(\Omega)$, $\|\cdot\|_k$ denota a norma $H^k(\Omega)$ Sobolev, $u \in H^2(\Omega)$ é a solução fraca, u_h é a aproximação linear por partes de elementos finitos (triângulos lineares) e C é uma constante que depende de u mas é independente do parâmetro de discretização espacial h (o diâmetro máximo de todos os triângulos em \mathcal{T}_h). Nestas condições, u_h converge para u quando $h \rightarrow 0$. O limite superior do erro foi mais tarde generalizado para espaços que não são de Hilbert por Krížek (1991). Em particular, se 4.15 é verificada, então, em termos de uma desigualdade Jensen-Minkowski-Cauchy-Schwarz e triangular, tem-se

$$\|u - u_h\|_{1,p} \leq C_p h |u|_{2,p} \quad p > 1 \quad (4.17)$$

em que $\|\cdot\|_{k,p}$ e $|\cdot|_{k,p}$ denotam a norma padrão e a semi-norma no espaço Sobolev $H^{k,p}(\Omega)$, respectivamente. Do ponto de vista da topologia total, ou seja, toda a malha e não um limite superior para um simples elemento, a seguinte propriedade é verificada como consequência direta de 4.15 (Babuska e Aziz, 1976)

$$\|v - \pi_h v\|_{1,\infty} \leq Ch|v|_{2,\infty} \quad (4.18)$$

onde $\pi_h v$ é a interpolação contínua por partes sobre toda a malha de uma função suficientemente suave v definida em um espaço \mathbb{R}^2 . As demonstrações e provas de cada limite são relativamente extensas e podem ser encontradas nas referências. O importante para este trabalho é em que estes resultados implicam.

Devido às eqs. (4.17) e (4.18) (Brandts et al., 2011), tem-se uma justificativa matemática de que não deve haver receios de se utilizar malhas contendo elementos com ângulos pequenos, mesmo que quase nulos, pois ângulos pequenos não podem ser responsáveis por falha de convergência ou soluções deterioradas no método dos elementos finitos. Assim, elementos que são praticamente degenerados não devem ser encarados como problema; pelo contrário, eles são muito úteis para preencher regiões com gradiente elevado direcional (regiões anisotrópicas) e até mesmo regiões geométricas muito estreitas (tais como ranhuras, fendas, etc.). Embora um ângulo possa ser arbitrariamente pequeno, em geral, dois ângulos não o podem, pois violariam a condição (4.15).

A extensão a tetraedros implica em uma condição análoga, tal que uma família $\mathcal{F} = \{\mathcal{T}_h\}_{h \rightarrow 0}$ de partições face-a-face de um poliedro que cobre Ω satisfaz a condição angular máxima se (Krížek, 1992)

$$\exists \gamma_0 < \pi \quad \forall \mathcal{T}_h \in \mathcal{F} \quad \forall K \in \mathcal{T}_h \quad : \quad \alpha_{DK} \leq \gamma_0 \quad \text{e} \quad \varphi_K \leq \gamma_0 \quad (4.19)$$

onde α_{DK} é o ângulo diedral máximo entre as faces do tetraedro K e φ_K é o ângulo sólido máximo entre todas as arestas de K . Estes ângulos constituem condições diferentes as quais devem ser ambas verificadas. A utilização de elementos tetraédricos lineares quase degenerados ou muito distorcidos, da mesma forma que os triangulares lineares, não constituem um problema para a convergência do método dos elementos finitos.

Embora a condição angular máxima seja um condição suficiente para a convergência, ela não constitui uma condição necessária. É com base nestas condições angulares que o presente trabalho não utiliza procedimentos que visem uniformizar a malha ao longo do processo de adaptação, mas sim apenas tentando reduzir ângulos acentuados presentes (por exemplo as condições apresentadas para alternância de arestas e faces). Embora muitos trabalhos tanto de geração quanto de adaptação de malhas possuam grande enfoque sobre qualidade dos elementos, a qual é definida geralmente por distribuição o mais uniforme possível de ângulos internos, não parece haver justificativa matemática

para tal no contexto da aplicação do método dos elementos finitos utilizando elementos lineares.

Mesmo que uma solução seja encontrada pelo método dos elementos finitos, não significa que a mesma tenha significado físico correto. Considerando o avanço da convergência do problema, pode-se encontrar uma solução aceitável matematicamente mas fisicamente incoerente contendo, por exemplo, quantidades tais como concentração, temperatura (em Kelvin) e densidade negativas. O estudo pioneiro nesta questão em equações parciais diferenciais resolvidas por métodos numéricos foi realizado por Courant et al. (1928). Não apenas esquemas explícitos mas também os implícitos possuem restrições de avanço no tempo que limitam o passo de tempo máximo possível. A determinação exata deste passo de tempo aceitável é muito custosa (Courant et al., 1928), recaindo sobre um problema de autovalores e autovetores. Devido a isto, opta-se por utilização da condição CFL como um número fornecido pelo usuário, atuando como coeficiente de segurança.

4.4 Algoritmo de Adaptação

Diversas abordagens, combinações e ordens podem ser utilizadas na implementação do algoritmo de adaptação. A utilizada no presente trabalho segue a seguinte metodologia, baseada em Habashi et al. (2000):

1. Refinar todas as arestas cujo erro esteja acima de um dado limiar superior η_S ;
2. Alternar faces/arestas;
3. Desrefinar todas as arestas cujo erro esteja abaixo de um dado limiar inferior η_I ;
4. Alternar faces/arestas.

Os 4 passos acima definem um ciclo de adaptação e são repetidos N_A vezes, de acordo com o informado pelo usuário. O critério que deve ser verificado para que uma dada aresta a seja refinada é

$$E(a) \geq \eta_S \sum_{i=1,n} \frac{E(i)}{n} \quad (4.20)$$

e para desrefinamento

$$E(a) \leq \eta_I \sum_{i=1,n} \frac{E(i)}{n} \quad (4.21)$$

sendo n o número total de arestas da malha. Ou seja, refina-se ou desrefina-se a aresta quando seu erro desloca-se do erro médio da malha ponderados por η_S e η_I , respectivamente. Quanto menor o valor de η_S , menor a tolerância de erro aceitável para refinamento e quanto maior η_I menor a tolerância de erro aceitável para o desrefinamento.

Quando o erro é totalmente equidistribuído sobre a malha, diz-se que a malha está completamente adaptada para o nível de resolução solicitado (malha ótima). Entretanto, tal condição torna-se bastante difícil de se verificar utilizando-se apenas adaptação-h (seria necessário a utilização de movimento nodal em conjunto com a adaptação-h para tal). Portanto, relaxa-se a condição para casos onde o erro encontra-se simplesmente melhor equidistribuído quanto possível. Avalia-se esta condição através da variância do erro sobre as arestas da malha no presente trabalho.

4.5 Acoplamento com Código CBS

O algoritmo de adaptação é, na verdade, independente da aplicação (e assim foi implementado). Após a manipulação da malha, deve-se atualizar e reinicializar todas variáveis do sistema (por exemplo velocidade do som nodal, pressão, tamanho característico do elemento..., etc.). O procedimento de adaptação é incorporado no código CBS de forma a ser executado a uma dada frequência fixada de passos de tempo $F_{\Delta t}$. Ou seja, a cada $F_{\Delta t}$ passos de tempo, o algoritmo de adaptação é acionado no código CBS. Entretanto, para casos estacionários, o procedimento só é acionado após a convergência da malha inicial do problema e é executado um número fixo de vezes N_F . Após a convergência da malha inicial e da realização de N_F ciclos de adaptação esparsos $F_{\Delta t}$ passos de tempo um do outro a contar do passo onde ocorra a convergência, o algoritmo encaminha a última malha adaptada para a convergência (final), encerrando a solução, finalmente. Ao contrário de convergir todos passos intermediários, é mais vantajoso utilizar mais passos intermediários com resoluções piores, permitindo maior flexibilidade da malha para se adaptar à evolução da solução (Habashi et al., 2000). Em problemas transientes, utiliza-se um limitante h_{min} o qual indica o tamanho mínimo de aresta que pode ser

refinada. Optou-se neste trabalho por utilizar as seguintes variáveis de conservação para estimar o erro: p para escoamentos subsônicos; ρ para escoamentos transônicos e supersônicos não-viscosos e $Mach$ para escoamentos transônicos e supersônicos viscosos. Esta escolha está relacionada à representatividade destas variáveis como caráter primário nos escoamentos nos quais são empregados. Desta forma, escoamentos subsônicos possuem baixa compressibilidade e a variável p torna-se mais significativa. Escoamentos não-viscosos transônicos e supersônicos apresentam grandes compressibilidades diretamente relacionadas com as grandes pressões e velocidades desenvolvidas, portanto utiliza-se ρ como variável característica nestes casos. Em escoamentos viscosos, o gradiente de velocidades é bastante diferenciado e importante, portanto adota-se M como variável característica a qual define a estimativa de erro. O número de ciclos de adaptação N_F para cada problema é função da malha inicial e do tipo de problema analisado, variando desde 2 até 7 ciclos no presente trabalho. A frequência de adaptação $F_{\Delta t}$ é função da malha inicial e do caráter do escoamento, variando de 500 (transiente) até 2000 (permanente) nos exemplos estudados. O limiar superior η_S é tomado como um valor maior que 1.0 e menor do que 2.0 e o limiar inferior η_I como um valor entre 0.1 e 0.7. Em geral, quanto mais uniforme as intensidades dos gradientes para o problema e quanto mais adaptações forem realizadas, maior pode ser o valor do limiar superior e menor o do limiar inferior.

Capítulo 5

Computação de Alto Desempenho

Devido à grande demanda de recursos computacionais envolvidos, principalmente no caso de simulações tridimensionais, torna-se indispensável o uso da Computação de Alto Desempenho - *High Performance Computing* (HPC). Neste sentido, implementou-se o código visando otimizar o uso do *hardware* disponível e de forma a se adequar automaticamente para computadores de memória compartilhada utilizando diretivas OpenMP. Este capítulo aborda todas as técnicas implementadas no código, levando-se em conta a linguagem de programação adotada: Fortran 90/95. Uma exposição ampla sobre a linguagem de programação Fortran 90/95 é encontrado em Chapman (2007) e uma descrição concisa de programação de alto desempenho e programação OpenMP pode ser encontrada em Chapman et al. (2007) e Hager e Wellein (2011).

5.1 Otimização Seqüencial

Um código a ser paralelizado deve ser eficiente quando executado seqüencialmente, caso contrário, todas as ineficiências serão multiplicadas quando executado de forma paralela, comprometendo e limitando severamente a performance computacional. As técnicas a seguir detalhadas são implementadas para otimizar a execução seqüencial.

5.1.1 Otimizações Simples

Mudanças muito simples podem muitas vezes resultar em melhoras significativas de performance. Embora possam parecer triviais, são comumente ignoradas (ou mesmo esquecidas). O resumo de algumas delas é descrito a seguir.

Redução de Trabalho Dispensável

Re-arranjar o código de forma a reduzir gastos computacionais desnecessários é uma abordagem simples com bons resultados. Um caso típico desta situação ocorre quando percorre-se um *loop* sobre determinado número de objetos no intuito de verificar se algum deles apresenta determinada propriedade, sendo realmente necessário que apenas um deles apresente a propriedade. Exemplifica-se o caso com um código que verifica se uma determinada aresta `edge1` pertence à alguma das seis arestas de um determinado tetraedro `tetraedge`:

```
logical :: flag
flag = .false.
do i = 1,6
  edge2 = tetra_edge(i)
  if ( edge1 .eq. edge2 ) then
    flag = .true.
  end if
end do
```

O código acima pode ser melhorado com a inclusão do comando `exit` para finalização do laço caso algum item seja verdadeiro:

```
logical :: flag
flag = .false.
do i = 1,6
  edge2 = tetra_edge(i)
  if ( edge1 .eq. edge2 ) then
    flag = .true.
    exit ! finaliza o laço do
  end if
end do
```

Cabe ressaltar que o código acima pode também ser reescrito utilizando o intrínseco `any`.

Evitar Operações Custosas

Muitas vezes os códigos são implementados traduzindo fórmulas para um código de forma direta. Embora isso torne o entendimento do código mais claro, em geral, é pouco otimizado no sentido computacional. Exemplos mais proeminentes são os que envolvem operações mais custosas tais como funções trigonométricas, exponenciais e logaritmos. Um exemplo típico ocorre quando se calcula $x**2.0$. O compilador não interpretará corretamente como $x*x$, acarretando o cálculo de uma função exponencial seguida de uma logaritmica, cujo custo computacional envolvido é muito superior.

Eliminação de Subcálculos Comuns

A eliminação de cálculos comuns (aqueles que são repetidos diversas vezes de forma desnecessária) é uma otimização padrão a qual os compiladores automaticamente tentam remediar. Porém, em estruturas mais complexas ou casos particulares, o compilador não irá detectar esta situação e cabe ao programador a otimização manual. Por exemplo:

```
do i = 1,n
  a(i) = a(i) + b * cos(c)  ! ineficiente
end do
```

onde claramente repete-se desnecessariamente n vezes o cálculo, o qual pode ser reescrito de forma eficiente como:

```
aux = b * cos(c)
do i = 1,n
  a(i) = a(i) + aux        ! eficiente
end do
```

Quanto maior a complexidade dos cálculos comuns eliminados e quantidade de acessos a serem realizados, tão maior será o ganho computacional envolvido.

Evitar Ramificações Condicionais

Laços sobre verificações condicionais escassas são sempre candidatos à uma linearização (também chamada de *pipelining*). Um código que multiplica uma matriz quadrada por um vetor é representado a seguir:

```
do j = 1,n
  do i = 1,n
    if ( i .ge. j ) then
      sign = 1.d0
    else if ( i .lt. j ) then
      sign = -1.d0
    else
      sign = 0.d0
    end if
    c(j) = c(j) + sign * a(i,j) * b(i)
  end do
end do
```

Nesta multiplicação, as partes triangulares superiores e inferiores adquirem sinais diferentes e a diagonal é ignorada. A cláusula condicional `if` serve para decidir este sinal. Entretanto, toda vez que a cláusula condicional é encontrada, o processador automaticamente tenta prever qual o resultado mais provável com uma lógica probabilística, mesmo sem saber o resultado correto (*branch prediction*). Com esta previsão, o processador avança os cálculos da ramificação. O problema real ocorre quando esta previsão falha (*branch miss*), sendo necessário retroceder na ramificação da execução do programa, resultando em perda de muitos ciclos de execução. Portanto, sempre que possível, o melhor procedimento a ser adotado é o de minimizar ou remover estas ramificações condicionais. O código a seguir é uma solução proposta:

```
do j = 1,n
  do i = j+1,n
    c(j) = c(j) + a(i,j) * b(i)
  end do
end do
do j=1,n
  do i = 1,j-1
    c(j) = c(j) - a(i,j) * b(i)
  end do
end do
```

Foi possível remover o condicional neste exemplo, resultando em melhor desempenho. Cabe ressaltar que é possível instruir o compilador a automaticamente tentar detectar este tipo de situação e remover os condicionais. Mas, como já mencionado, essa detecção automática nem sempre é bem sucedida, sendo necessária intervenção manual.

Uso de Intrínsecos

As funções intrínsecas da linguagem Fortran são fortemente otimizadas computacionalmente. Desta forma, em geral, resulta ser vantajoso o uso de tais funções. Pode-se citar como exemplos o de multiplicação de matrizes, produto interno, extração de raiz quadrada, dentre tantos outros.

Simplificação de Operações

O custo computacional envolvido em operações de adição, subtração, multiplicação, divisão, exponencial entre outras não é o mesmo. O custo de realizar adição ou subtração é o mesmo, e são as operações menos custosas. A multiplicação é sempre mais custosa que a adição e menos custosa que a divisão. Exponenciais e outros são mais custosos que a divisão. Deve-se dar preferência às operações mais simples ao escrever o código, porém, esta otimização é facilmente detectada pelo compilador e quase sempre automaticamente ajustada.

Uso de Subrotinas e Funções

Embora o uso de funções e subrotinas torne em geral o código mais claro, quanto maior o número de argumentos envolvidos na transição de dados do código em execução para a função ou subrotina e vice-versa, maior o custo computacional envolvido. Entretanto, muitas vezes é possível remediar essa situação através de diretivas de compilação (*inlining*) as quais são discutidas mais detalhadamente a seguir.

5.1.2 Compiladores

O uso correto do compilador e de técnicas de compilação possuem forte impacto na performance do código. Cada versão mais moderna de um compilador (em geral) possui maior capacidade de otimizar códigos a serem compilados. Se por um lado o compilador pode conseguir otimizar trechos de código extremamente complexos, por outro, pode não conseguir otimizar trechos muito simples. Por esta falta de garantias, o programador

deve sempre tentar otimizar o código manualmente na medida do possível. O compilador utilizado no presente trabalho foi o GFortran 4.7.

Opções Gerais de Compilação

A maioria dos compiladores oferece uma coleção de otimização padrão em diversos níveis (-O0, -O1, -O2... em ordem crescente de otimização). Quanto maior o nível de otimização requerido, maior o tempo de compilação, pois o compilador executa buscas cada vez mais complexas visando otimizar e verificar a consistência do código. O presente trabalho utiliza o nível -O3 de otimização de compilação.

Inlining

O termo *inlining* se refere à inserção de funções ou subrotinas diretamente no ponto do código em que é chamada. Sempre que uma dada função é chamada ao longo do código, um custo computacional é requerido pois argumentos devem ser passados para a função e de lá para o código em execução. Assim, registros e memória adicional são requeridos, dependendo do número de parâmetros envolvidos nessa função. A inserção da função diretamente no código não remove a necessidade do uso de variáveis auxiliares locais, as quais devem ser estabelecidas de qualquer forma, mas remove a necessidade de colocar variáveis na pilha (*stack*) e permite ao compilador usar registros quando necessário (e não segundo uma ordem convencional por chamar uma dada função ou subrotina), removendo, portanto, a pressão de registro (*pressure register*). Pressão de registro ocorre quando a CPU não possui registros suficientes para lidar com todas as operações dentro de uma dada computação complexa ou *loop*. Além disto, a diretiva *inlining* permite que o compilador visualize uma porção maior do código, realizando otimizações que não seriam disponíveis em um código não-linear fazendo uso de subrotinas e funções dispersas. O lado negativo do *inlining* é que o tamanho (físico) do código é aumentado consideravelmente, o que pode desencadear problemas com a capacidade de memória de instrução *cache* L1. Assim, se as instruções de um *loop* não couberem na memória L1, elas irão competir com a transferência de dados de e para memórias *caches* mais lentas ou até mesmo a memória principal. No compilador GFortran, existem diversas opções de *inlining*, e a utilizada no presente trabalho foi a padrão: `-finline-functions`. Cabe dizer também que ao

utilizar o nível `-O3` de otimização, automaticamente será utilizada a técnica de *inlining* padrão.

5.2 Otimização de Acesso à Memória

De todos os limitantes de performance para a HPC, o mais importante é o acesso à memória (Hager e Wellein, 2011). Microprocessadores costumam ser referenciados por sua grande capacidade de performance de pico, atingindo diversos *gigaflops*. Entretanto, os códigos gerados dificilmente atingem mais de 10% dessa capacidade (Löhner, 2008). Isso se deve ao fato de que a maioria dos códigos científicos consistem em grande parte de estruturas do tipo *loop* transferindo grandes quantidades de dados para CPU e desta para o código. Desta forma, a performance real do código fica limitada pelo acesso relativamente lento à memória ou até mesmo ao disco. A memória *cache* L1 é a mais rápida. Quando a memória L1 não é suficiente para alguma operação passa-se a utilizar a L2 e depois a L3 (diz-se que ocorreu um *cache-miss*), as quais possuem performance mais lenta. Caso ainda falte memória, passa-se a utilizar outras memórias como a principal e do disco, degradando-se cada vez mais a performance. Portanto, o código é tão mais eficiente quanto menos níveis de profundidade a memória for acessada ao longo de sua execução. Para o caso de simulação de escoamentos, em particular, ainda mais em malhas não estruturadas, diversas informações devem ser acessadas simultaneamente (nós, faces, arestas...) necessitando-se executar diversos endereçamentos indiretos à memória. Torna-se importante manter as informações a serem utilizadas conjuntamente o mais próximo possível na memória para otimizar seu acesso.

5.2.1 Acesso à Arranjos

O armazenamento de arranjos utilizados no código (elementos, coordenadas, variáveis, etc.) de forma compatível com a forma com que são acessados dentro de *loops* reduz apreciavelmente o *cache-miss*. No caso da linguagem Fortran, os arranjos são armazenados por coluna (Masuero, 2009), e portanto assim devem ser acessados para evitar o *cache-miss*, como no exemplo a seguir:

```

do i = 1,n          ! i varia lentamente
  do j = 1,m        ! j varia rapidamente
    a(j,i) = i * j  ! acessa a matriz por coluna
  end do
end do

```

A ordem de acesso da matriz ocorre por coluna: $a(1,1)$, $a(2,1)$, ... $a(m,1)$... $a(m,n)$ que é exatamente a forma como o arranjo está armazenado na memória. Caso os índices estivessem invertidos, a cada novo acesso à matriz seria necessário percorrer o arranjo ao longo da memória até ser encontrada a próxima posição.

5.2.2 Renumeração Nodal

Parte do código CBS (passos 1 a 4) pode ser resumida genericamente no seguinte código (caso tridimensional):

```

do element = 1,nelem          ! laco sobre os elementos
  node1 = intma(1,element)
  node2 = intma(2,element)
  node3 = intma(3,element)
  node4 = intma(4,element)
  rhs1(node1) = get_rhs0(node1,element,unknown) ! obtem incremento nodal
  rhs2(node2) = get_rhs0(node2,element,unknown)
  rhs3(node3) = get_rhs0(node3,element,unknown)
  rhs4(node4) = get_rhs0(node4,element,unknown)
  rhs0(node1) = rhs0(node1) + rhs1(node1)        ! adiciona o incremento
  rhs0(node2) = rhs0(node2) + rhs1(node2)        ! na variavel nodal
  rhs0(node3) = rhs0(node3) + rhs1(node3)        ! respectiva
  rhs0(node4) = rhs0(node4) + rhs1(node4)
end do

```

Percebe-se que o acesso sobre os elementos é feito de forma seqüencial mas que o acesso aos nós pertencentes ao elemento não necessariamente está ordenado. De fato, quanto mais esparsos esses nós estiverem, mais tempo será perdido acessando a memória. Dois níveis de renumeração nodal são implementados neste trabalho, os quais são descritos a seguir.

Renumeração Nodal Global

As conectividades de uma dada malha podem ser interpretadas como um grafo, ou mais simplesmente, uma matriz. Desta forma, diz-se que a largura de banda da malha é

minimizada quando esta matriz representando conectividades da malha torna-se menos dispersa. Existem diversas técnicas para se realizar esse procedimento, cada uma delas mais adequada para determinados casos. O método utilizado no presente trabalho é do tipo frente de avanço (*advancing-front* ou também *wave-renumbering*). Este tipo de método caracteriza-se por renumerar a malha começando por um dado nó da malha e seguir adicionando novas camadas de acordo com a menor distância de conectividade. A frente de avanço de nós renumerados segue até que toda a malha seja renumerada (Fig. 5.1). O método da forma de avanço adotado no presente trabalho é a técnica do

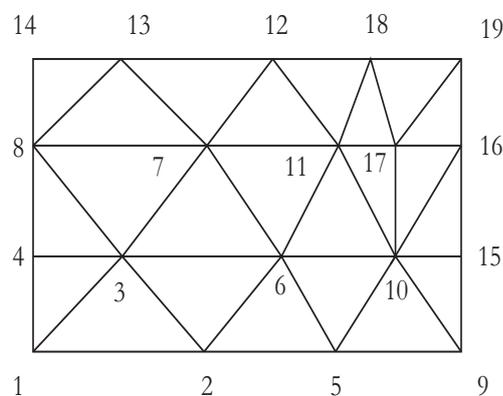


Figura 5.1: Renumeração do tipo *advancing-front*.

ordenamento reverso de Cuthill-McKee (RCM). Este método possui vantagem em casos onde o número de conectividades de cada nó é variável, que é o caso das malhas não-estruturadas.

Renumeração Nodal Local

O *cache-miss* à nível de elemento pode ser reduzido através da renumeração dos nós pertencentes a cada elemento. A ideia é acessar os nós de cada elemento da forma mais uniforme possível, utilizando-se para isto uma renumeração local crescente, como exemplificado a seguir para um tetraedro *ielem* cujas conectividades são dadas por `intma(1:4,ielem)`:

```
!sem renumeracao
intma(1,ielem) = 100
intma(2,ielem) = 5000
intma(3,ielem) = 400
intma(4,ielem) = 4600
```

Para o caso acima percebe-se que o acesso às conectividades ocorre de forma alternada, percorrendo-se a memória no sentido avante do primeiro para o segundo nó, depois retrocedendo-se no terceiro e avante novamente no quarto e último nó. Este avanço e retrocesso implica em custo computacional adicional percorrendo-se a memória. A forma a seguir é a mais adequada:

```
!com renumeracao local
intma(1,ielem) = 100
intma(2,ielem) = 400
intma(3,ielem) = 4600
intma(4,ielem) = 5000
```

É importante salientar que essa renumeração local deve ser tal que não afete as informações dos elementos (volume, funções de forma, etc.).

Uso de Listas

Em malhas não-estruturadas, faz-se uso de diversas informações dispersas ao longo da malha as quais devem ser acessadas de forma ordenada. Para otimizar este procedimento, faz-se uso intensivo de listas armazenando informações. Alguns exemplos, dentre muitos outros, são:

- elemento que pertençam a um nó;
- nós vizinhos a um dado nó;
- elementos vizinhos a outros elementos;
- conectividades de cada aresta;
- arestas de um elemento

As listas são armazenadas em forma de vetor e são acessadas de forma contínua a partir de um ponteiro auxiliar (para cada lista) o qual indica o ponto inicial e final a ser percorrido ao longo da mesma. Há dois tipos distintos de listas utilizados neste trabalho: listas estáticas e listas dinâmicas. As listas estáticas são as que não sofrem alterações para um determinado fim específico. As dinâmicas são aquelas que o número de itens varia ou não é conhecido *a priori*. Listas dinâmicas são comumente encontradas em procedimentos

adaptativos, onde, por exemplo, os elementos vizinhos de cada nó sofrem variação ao longo da execução do procedimento. Como em listas estáticas o número de itens é conhecido, armazena-se o vetor no tamanho exato ao requerido. Nas listas dinâmicas, por outro lado, optou-se por armazenar um dado espaço dedicado em cada vetor, o qual pode ou não ser utilizado totalmente, e que é percorrido pelos ponteiros de forma contínua ao longo da região de fato utilizada. Uma descrição completa de como gerar e gerenciar estas listas de forma eficiente pode ser encontrada em Löhner (2008).

5.3 Otimização Paralela em Computadores de Memória Compartilhada

Um computador de memória compartilhada é um computador no qual um número de CPUs trabalha em conjunto e compartilha um mesmo espaço físico. O modelo padrão de programação dominante em computadores de memória compartilhada é a OpenMP, a qual é utilizada no presente trabalho. A grande vantagem deste tipo de metodologia é a facilidade de implementação, bastando-se simples modificações ao longo do código seqüencial. A entidade central de programas com diretivas OpenMP é o *thread*. Um *thread* é uma linha ou encadeamento de tarefas e execuções, uma forma de um processo dividir a si mesmo em duas ou mais tarefas a serem executadas concorrentemente. A seguir são descritas as principais modificações utilizadas ao longo do código seqüencial para paralelização e também são comentados otimizações particulares.

5.3.1 Atribuição de Valores Prescritos

Ao longo de um código é comum atribuir um valor prescrito a uma variável (inicialização, cópia, simples prescrição). Se a variável for uma vetor, matriz ou arranjo de ordem maior, torna-se mais vantajoso atribuir-se estes valores de forma paralela quando essa prescrição ocorrer com freqüência significativa. O código a seguir é uma subrotina que atribui valor prescrito `val` à uma matriz `ma` de dimensões $a \times b$ de forma paralela eficientemente:

```
!$omp parallel do default(shared) private(i,j) collapse(2)
do j = 1,a
  do i = 1,b
    ma(i,j) = val
```

```
    end do
end do
!$omp end parallel do
```

O comando `$omp parallel do` indica início de laço a ser executado em paralelo, onde todas variáveis são compartilhadas (`default(shared)`) com exceção de `i` e `j` (`private(i, j)`), os quais são privados a cada *thread*. Porém, de forma genérica, não se sabe qual a extensão das dimensões `a` e `b` da matriz `ma`. A paralelização simples de qualquer uma dessas dimensões torna-se dependente da matriz específica em questão e do número de *threads* disponível. Uma alternativa mais eficiente para evitar este desbalanceamento é o uso da diretiva `collapse(2)`. Com ele, dois níveis consecutivos são expandidos em apenas um vetor, no caso acima, um único vetor de tamanho `ab`. E será este vetor final que será distribuído ao longo dos *threads*, resultando um balanço mais otimizado.

5.3.2 Escalonamento de *Loops*

A simples execução de um *loop* em paralelo é interpretada de forma padrão como uma distribuição estática uniforme, isto é, reparte-se a extensão do *loop* de forma uniforme ao longo dos *threads*. Porém, nem sempre a tarefa em si fica bem distribuída, pois a extensão não necessariamente está relacionada com o volume de trabalho a ser realizado. Como em geral o programa deve esperar que cada *thread* finalize suas tarefas para posterior sincronização, tem-se que toda a região paralela fica dependente do *thread* mais sobrecarregado, o qual atrasará todos os outros. Uma forma de melhorar essa situação é com a utilização da diretiva `schedule()`, a qual pode ser `static`, `dynamic` ou `guided`. No modo `static`, divide-se o *loop* de forma contínua ao longo dos *threads*. No modo `dynamic`, ao longo da execução, quando um dado *thread* finalizar uma parte de seu trabalho atribuída, ele se encarregará da próxima tarefa disponível. No modo `guided`, ocorre o mesmo que no modo `dynamic`, mas a distribuição é exponencial. No modo `dynamic`, quando a quantidade de trabalho a ser realizado é muito pequena com relação à distribuição de tarefas à cada *thread*, ocorre perda significativa de eficiência, pois o custo do gerenciamento dinâmico das tarefas passa a desempenhar papel significativo nestes casos. Nestes casos, o modo `guided` é mais vantajoso pois as divisões acabam sendo proporcionais ao número restante de tarefas disponíveis dividido pelo número de

threads. É possível incluir adicionalmente um número fixo de divisão para cada uma das três opções acima. A Fig. 5.2 exemplifica estas atribuições de escalonamento de forma genérica. Pode-se utilizar este escalonamento, por exemplo, para termos de contorno,

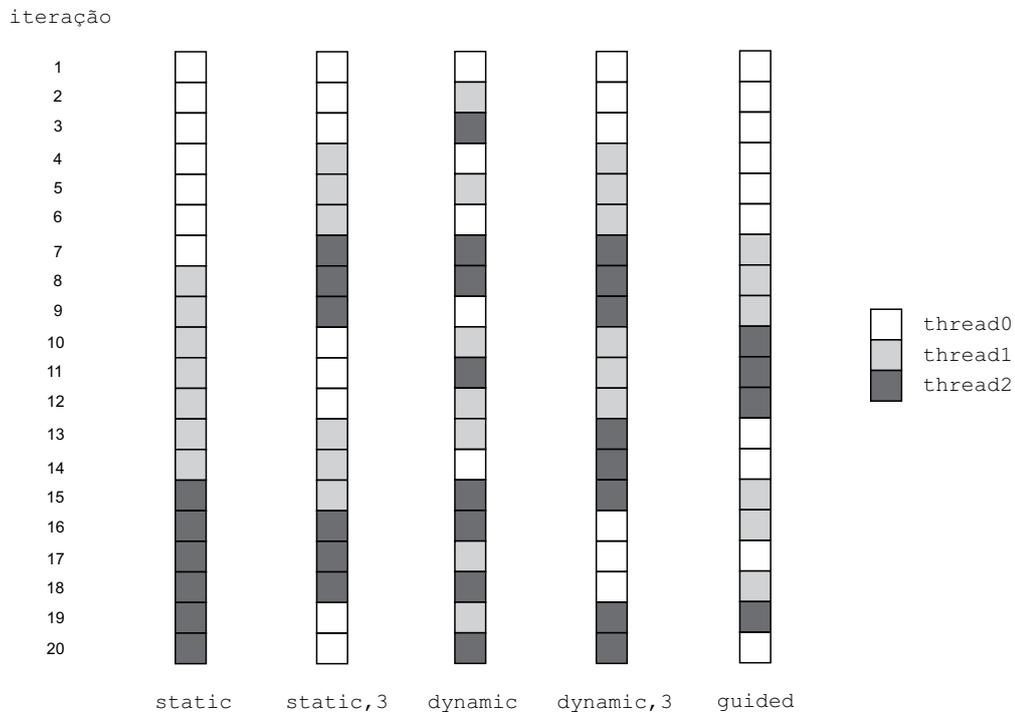


Figura 5.2: Tipos de escalonamentos em OpenMP para um *loop* de 20 iterações e executado por 3 *threads*.

como indica o trecho de código genérico a seguir:

```
!$omp parallel do default(shared) private(i,j,ip) schedule(guided)
do i = 1,nboun
  ! percorre faces do contorno
  if ( isido(5,i) .eq. 1 ) then ! isido(5,i) indica o tipo de contorno
    do j = 1,3
      ! percorre os tres nos da face
      ip = isido(j,i)
      unkno(1,ip) = val
      ! caso seja um dado tipo de contorno, aplica
      ! um valor val prescrito (exemplo: massa)
    end do
  end if
end do
!$omp end parallel do
```

O código acima percorre as faces de contorno, verificando se é de um tipo específico. Em caso afirmativo, percorre os três nós da mesma, aplicando uma condição sobre as variáveis de conservação de seus nós. Como nem todos termos do contorno são do tipo esperado e a carga de trabalho é pequena, a diretiva *guided* é a mais indicada neste caso.

5.3.3 Sincronizações e Atualizações

Sempre que possível, deve-se evitar sincronizações desnecessárias dos *threads*, pois isto acaba retirando parte ou toda a eficiência da execução paralela. Existem trechos do código os quais requerem mais atenção, como por exemplo o de atualização de uma dada variável global. É um caso típico onde cada *thread* realiza uma série de cálculos para ao final atualizar dadas variáveis (por exemplo, variáveis de conservação). Deve-se ter o cuidado que nesta atualização cada *thread* acesse e atualize a variável sem que outro realize o mesmo simultaneamente, caso contrário, o resultado torna-se incorreto. Outra alternativa é o de pré-condicionar cada *thread* a ser responsável por áreas independentes que jamais se interceptam, sem necessidade do cuidado anteriormente citado. A primeira abordagem é a mais simples e implementada no presente trabalho. Exemplifica-se o procedimento no trecho de código a seguir:

```
!$omp parallel do default(shared) private(element,node,i,rhs1)
do element = 1,nelem                                     ! laco sobre os elementos
  node(1:4) = intma(1:4,element)
  do i = 1,4
    rhs1(node(i)) = get_rhs0(node(i),element,unknown) ! obtem incremento nodal
  end do
  do i = 1,4
    !omp atomic
    rhs0(node(i)) = rhs0(node(i)) + rhs1(node(i))      ! adiciona o incremento
  end do                                               ! na variavel nodal
end do                                                 ! respectiva
!$omp end parallel do
```

A diretiva `atomic` indica que a linha a seguir é executada por apenas um *thread* de cada vez. Outras diretivas, que distinguem-se levemente desta podem ser utilizadas também, como a `critical` e a `reduction`. Obviamente estas diretivas implicam em uma perda intrínseca de eficiência de paralelização. Quanto maior a carga de trabalho e balancemendo de cada *thread* dentro do *loop* em execução, tão menor essa perda será.

5.3.4 Escalabilidade de Pequenas Tarefas

Nem todo o código pode ser paralelizável, o que implica que sempre haverá perdas em uma execução paralela. Entretanto, muitas vezes a carga de trabalho é paralelizável até certo número de *threads*. Após este número, percebe-se um decréscimo de eficiência de

paralelização. Muitas vezes este efeito de escalabilidade reside no fato da carga de trabalho a ser realizada ser muito pequena para um número grande de *threads*. Sempre que dispara-se uma execução paralela, milhares de *flops* são requeridos na distribuição da tarefa (Löhner, 2008). Assim, por exemplo, um *loop* que varia de 1 até 100000 provavelmente é mais eficiente quando executado com 12 *threads* do que com 24. Deve-se ponderar as cargas de trabalho com o número de *threads* disponíveis. No caso de malhas não estruturadas de elementos tetraédricos lineares, tem-se que o número de elementos em geral supera em quatro a cinco vezes o número de nós da malha. Já as faces de contorno são uma fração bem menor do número de elementos e se distinguirmos em cada tipo de contorno, menor ainda. Assim, percebe-se que as dimensões das estruturas básicas trabalhadas na malha de elementos finitos não são as mesmas, e uma tentativa unificada de distribuição de cargas de trabalho falha por falta de escalabilidade.

5.4 Particularidades da Implementação

Algumas observações foram realizadas na otimização do código implementado. Devido à metodologia de adaptação adotada, tem-se que o desrefinamento e as alternâncias de faces e arestas possuem dependência forte da execução, sendo a paralelização do procedimento complicada ou inviável. Com base nisso, implementou-se todas as abordagens descritas na parte de otimização seqüencial e de acesso à memória, em especial, e com mais ênfase, o uso de listas. O uso massivo de listas resultou em uma redução severa de custos computacionais da etapa de adaptação. Cabe também comentar a respeito de alocações. Alocar e desalocar arranjos com frequência é uma tarefa custosa e evitou-se, sempre que possível, este procedimento ao longo da etapa de adaptação, através do uso de alocações estáticas. Estas alocações são escalonadas estaticamente uma única vez ao início da etapa de adaptação. Já o código CBS faz pouco uso de listas e prioriza a paralelização OpenMP.

5.5 Definições

Para posterior análise, as seguintes definições são adotadas no presente trabalho.

5.5.1 *Speedup*

Define-se o *speedup* S_p como a relação entre o tempo de execução do código seqüencial T_1 e o tempo de execução do código paralelo utilizando p processadores, T_p

$$S_p = \frac{T_1}{T_p} \quad (5.1)$$

O *speedup* ideal é aquele que se aproxima de p , mas perdas sempre ocorrerão e este valor será menor. Quanto mais próximo de p maior a escalabilidade do código.

5.5.2 Eficiência de Paralelização

Define-se eficiência de paralelização E_p como o *speedup* dividido pelo número de processadores p

$$E_p = \frac{S_p}{p} \quad (5.2)$$

O valor, que varia entre 0 e 1, estima o quão bem os processadores estão sendo utilizados para resolver o problema, em comparação com quanto esforço é gasto em comunicação e sincronização.

Capítulo 6

Resultados

Este capítulo apresenta os resultados obtidos com a utilização da metodologia de adaptação de malhas desenvolvida. São estudadas aplicações do algoritmo bidimensional e tridimensional tanto para problemas analíticos quanto acoplados ao algoritmo CBS de simulação de escoamentos compressíveis desenvolvido. Os problemas analíticos estudados visam esclarecer e analisar com mais detalhes as propriedades do algoritmo de adaptação e as adaptações acopladas ao código CBS são aplicações diretas. Para todos exemplos, tanto bidimensionais quanto tridimensionais, excetuando-se os casos analíticos, os seguintes parâmetros comuns relativos ao código CBS são empregados:

- $\theta_1 = 0.5$;
- $\gamma = 1.4$;
- $Pr = 0.72$;
- $\rho_\infty = 1.0$;
- $L = 1$

Os dados do fluido empregado são relativos ao ar (propriedades médias próximas a $T = 300K$). São fornecidos os valores de corrente livre de cada problema, bem como a indicação das linhas ou superfícies onde as condições de contorno são aplicadas em cada exemplo (seguindo o estipulado em 2.1.1, considerando valores prescritos sempre como os de corrente livre). Os seguintes dados são fornecidos para cada simulação:

- M_∞ indica o número de Mach do escoamento;
- $u_{i\infty}$ indica a velocidade de corrente livre do escoamento nas direções x , y e z ;
- CFL indica o número de CFL empregado na simulação;
- C_E indica o coeficiente de difusão artificial empregado na simulação;
- R_T indica o resíduo para convergência da simulação (escoamento permanente);
- T_∞ indica a temperatura (em Kelvins) de corrente livre do escoamento;
- Re indica o número de Reynolds do escoamento (escoamentos viscosos);
- α_S indica o coeficiente de suavização de variáveis (problemas subsônicos)
- ϕ indica a variável utilizada para medir o erro em cada problema analisado;
- η_S indica o valor do coeficiente de limiar superior para refinamento;
- η_I indica o valor do coeficiente de limiar inferior para o desrefinamento;
- N_F indica o número de ciclos de adaptação empregado;
- $F_{\Delta t}$ indica a frequência de adaptação (intervalo de iterações);
- h_{min} indica o comprimento mínimo de aresta o qual pode ser submetido à refinamento (indicado apenas para problemas transientes)

É importante ressaltar que os dados do escoamento não possuem unidade pois são todos obtidos baseados em valores adimensionais tais como o número de Mach e de Prandtl, excetuando-se a temperatura, que possui unidade sempre. Desta forma, as velocidades de corrente livre informadas indicam a direção do escoamento e a massa específica de corrente livre é tomada como valor unitário. A pressão e outras variáveis de corrente livre são calculadas levando-se em conta as equações que relacionam os parâmetros adimensionais do escoamento e a massa específica de corrente livre, estado sempre todos corretamente relacionados. Os campos de solução apresentados são relativos à corrente livre. Assim, por exemplo, caso se queira calcular a massa específica considerando alguma unidade, basta utilizar a temperatura real fornecida e os números adimensionais do escoamento em conjunto com as propriedades do gás para relacionar diretamente com valores de corrente livre.

6.1 Adaptação de Malhas Bidimensionais

Nesta seção são analisadas aplicações da adaptação bidimensional para um caso analítico, um canal com rampa de compressão e expansão e diversas configurações de escoamento no entorno de um aerofólio NACA0012.

6.1.1 Exemplo 1: Solução Analítica Hiperbólica

Para demonstrar a capacidade, bem como as características do algoritmo de adaptação de malhas bidimensionais compostas por elementos triangulares lineares, apresenta-se um caso de estudo onde uma solução analítica é imposta a uma malha inicial. Tem-se, portanto, a solução correta nos nós da malha, mas devido à discretização espacial insuficiente, a solução interpolada linearmente entre os nós da malha possui baixa resolução. O problema analisado constitui-se de um domínio retangular $\Omega : \mathbf{x} \in [-5, -5] \times [5,5]$ onde impõe-se um campo hiperbólico analítico $u = \tanh(xy)$. A malha inicial é composta por 221 nós e 400 elementos. A malha inicial bem como a solução inicial são mostradas na Fig. 6.1. A variável utilizada para análise de erro ϕ foi a própria

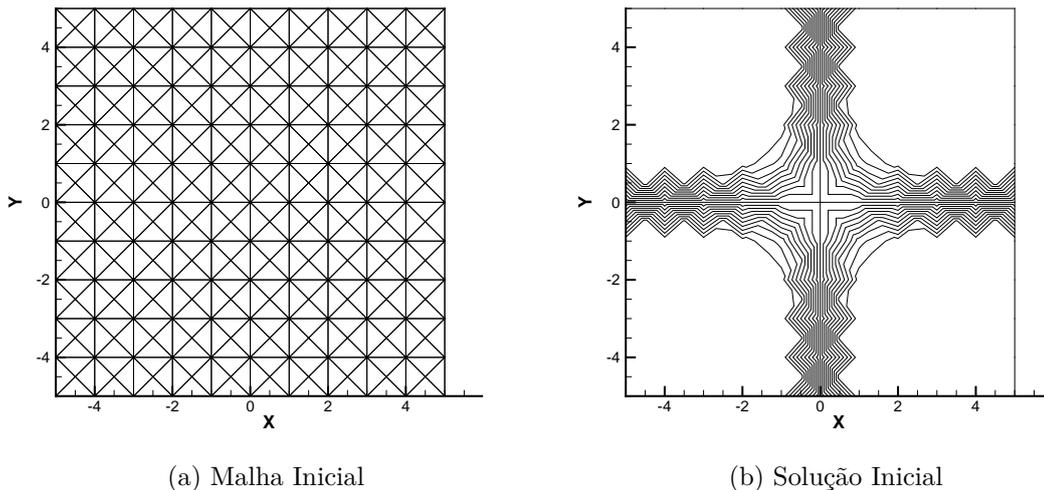


Figura 6.1: Malha e solução inicial do exemplo 1.

função analítica imposta $\phi = u$. A partir da solução inicial, realizou-se $N_F = 25$ ciclos de adaptação utilizando-se um limiar superior $\eta_S = 1.5$ e um limiar inferior $\eta_I = 0.2$. A Fig. 6.2 mostra as malhas e a Fig. 6.3 mostra os campos de soluções ao longo do avanço

da adaptação.

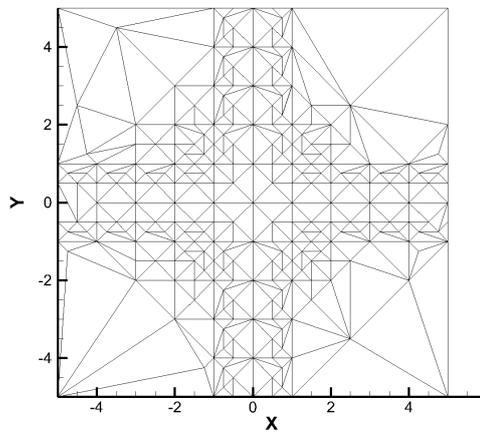
A Tab. 6.1 indica o número da adaptação com a correspondente quantidade de nós, elementos, o ângulo máximo interno entre todos triângulos (α_{max}), a porcentagem de ângulos que ultrapassam o limite de 160° (α_{160}) e a variância do erro das arestas da malha, $\sigma^2(E)$. Estes dados permitem uma análise mais quantitativa do resultado do procedimento. Primeiramente, observa-se uma diminuição de $\sigma^2(E)$ em todas as iterações (com exceção da adaptação 3). A variância $\sigma^2(E)$ indica o quão disperso da média os erros sobre todas as arestas se encontram. Portanto, quanto menor seu valor, mais equidistribuído estará o erro sobre a malha e portanto mais adaptada encontra-se a malha. O caso único onde não se verifica esta diminuição pode ser explicado por um procedimento de desrefinamento gerando elementos com arestas muito grandes, alterando de forma significativa $\sigma^2(E)$. O percentual α_{160} é um indicador de ângulos gerados por alta distorção dos elementos. Indica também casos onde o ângulo se aproxima da condição angular máxima. Juntamente com o ângulo máximo da malha α_{max} pode-se perceber que, embora muitas vezes o ângulo máximo torne-se elevado, este ângulo é predominantemente local, pois apenas uma parcela muito baixa da malha apresenta ângulos extremamente elevados, mas mesmo assim, sempre com valores inferiores à 180° .

Pode-se visualizar na Fig. 6.2 a capacidade de desrefinar a malha inicial bem como qualquer porção da malha ao longo do processo de adaptação. A Fig. 6.4 mostra uma região da malha adaptada (iteração 25) ampliada. É possível observar a malha distorcida segundo uma direção bem definida (vertical) na região superior, a qual torna-se cada vez mais isotrópica à medida que desloca-se para a região inferior e lateral da figura. Isto decorre do fato do gradiente ser acentuado na região superior e progressivamente menos acentuado à medida que se compara com as regiões mais abaixo e laterais. A região no entorno do eixo $x = 0$, por outro lado, possui gradiente muito baixo, formando uma região anisotrópica com nível de refinamento muito mais baixo que sua vizinhança, embora ainda em uma direção predominante. Isto demonstra a capacidade do algoritmo de adaptação de gerar zonas tanto anisotrópicas quanto isotrópicas, bem como regiões de alta ou baixa densidade de nós.

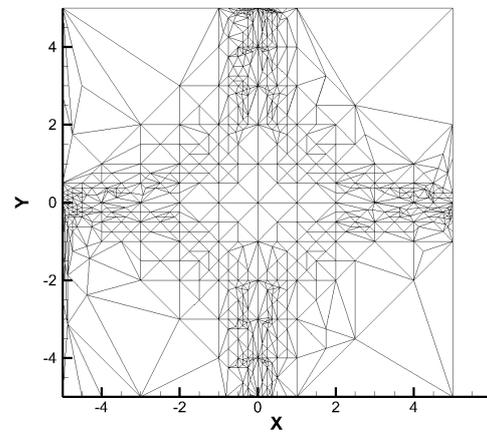
Analisa-se a seguir, de forma independente, o efeito da alternância de arestas e do limiar superior e inferior sobre a adaptação.

Tabela 6.1: Dados das iterações de adaptação do exemplo 1

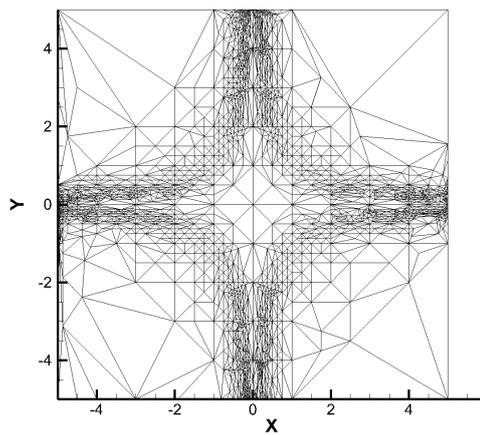
Adaptação	Nós	Elementos	α_{max}	α_{160}	$\sigma^2(E)$
0	221	400	90.00	0.000E-0	4.506E-2
1	261	496	150.25	0.000E-0	2.950E-2
2	336	640	161.56	1.041E-3	2.721E-2
3	452	864	164.87	2.314E-3	3.388E-2
4	527	1008	164.87	2.976E-3	2.582E-2
5	647	1239	168.69	3.497E-3	1.726E-2
6	819	1575	172.37	2.314E-3	1.173E-2
7	1039	2008	172.37	4.150E-3	8.269E-3
8	1293	2507	175.13	4.786E-3	5.950E-3
9	1617	3142	175.72	3.500E-3	4.895E-3
10	2057	4013	178.95	3.654E-3	3.449E-3
11	2588	5058	178.95	4.349E-3	2.665E-3
12	3186	6246	177.13	4.909E-3	2.175E-3
13	3861	7591	175.13	4.435E-3	1.738E-3
14	4729	9310	175.13	4.797E-3	1.406E-3
15	5815	11468	175.13	5.667E-3	1.133E-3
16	7135	14087	173.89	5.986E-3	8.951E-4
17	8758	17327	172.87	6.233E-3	7.152E-4
18	10750	21285	173.89	7.266E-3	5.845E-4
19	13024	25813	175.13	7.283E-3	4.825E-4
20	15781	31312	175.15	7.568E-3	3.955E-4
21	19201	38119	178.68	7.940E-3	3.194E-4
22	23423	46525	179.04	8.024E-3	2.589E-4
23	28878	57396	179.04	7.892E-3	2.098E-4
24	35325	70255	179.43	8.094E-3	1.714E-4
25	42988	85550	178.68	8.790E-3	1.409E-4



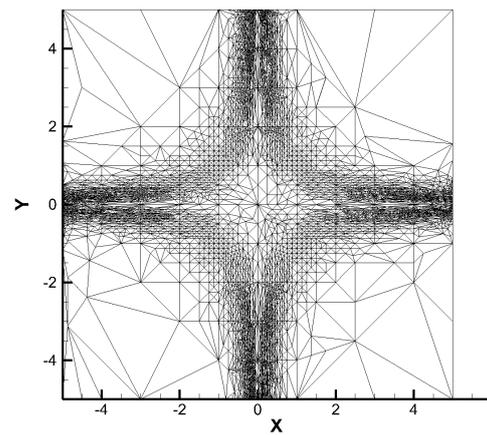
(a) Adaptação 1



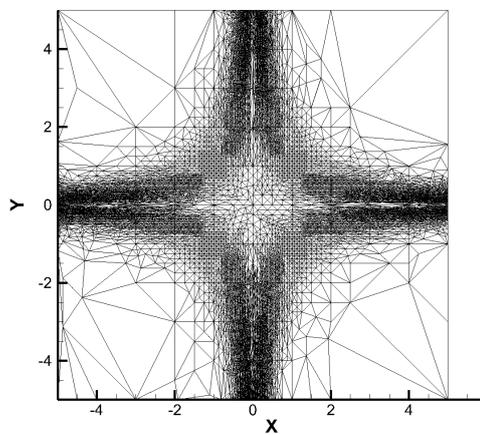
(b) Adaptação 5



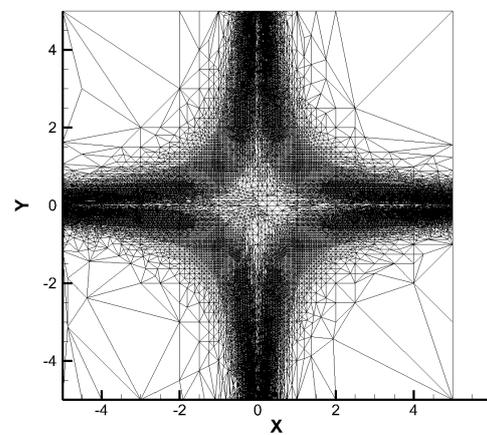
(c) Adaptação 10



(d) Adaptação 15

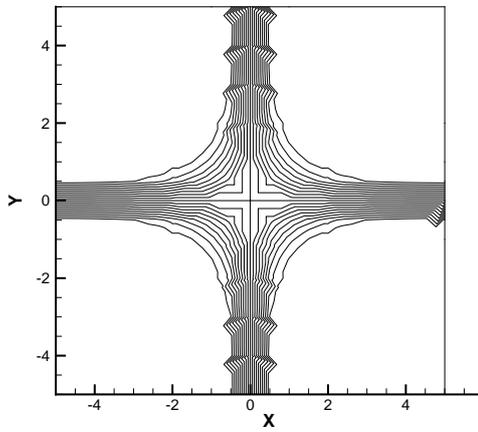


(e) Adaptação 20

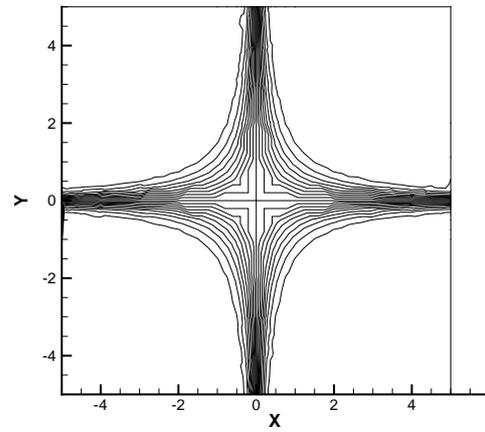


(f) Adaptação 25

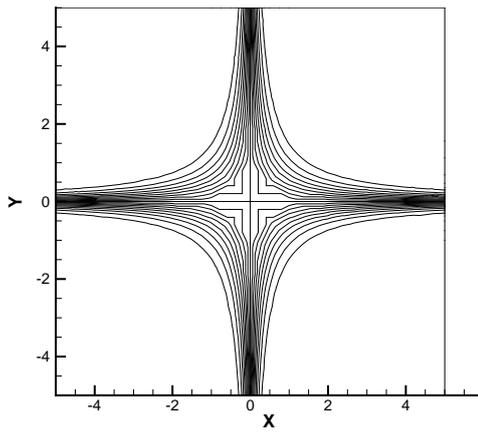
Figura 6.2: Malha ao longo das iterações de adaptação do exemplo 1.



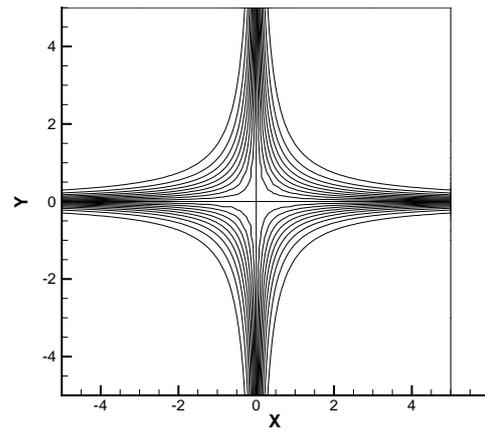
(a) Adaptação 1



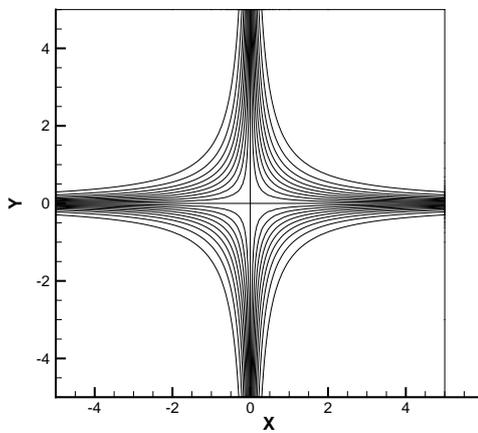
(b) Adaptação 5



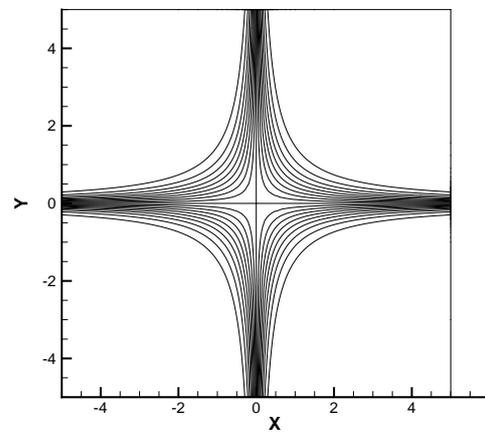
(c) Adaptação 10



(d) Adaptação 15



(e) Adaptação 20



(f) Adaptação 25

Figura 6.3: Solução ao longo das iterações de adaptação do exemplo 1.

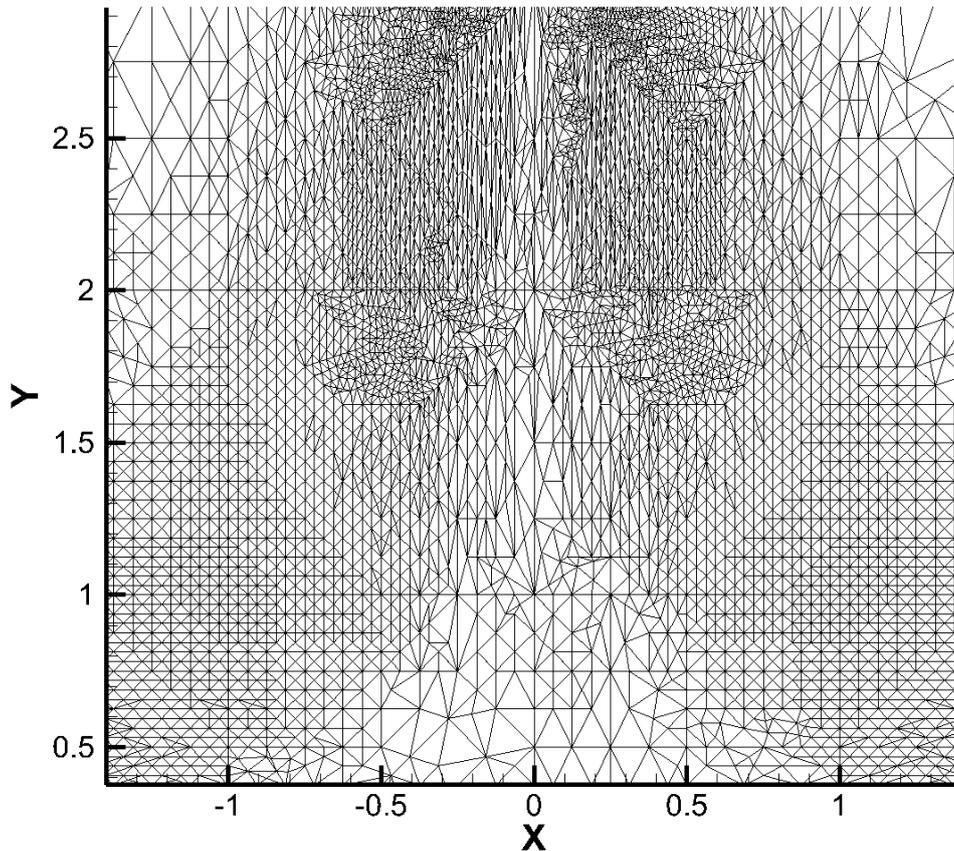


Figura 6.4: Detalhe aumentado de uma porção da malha adaptada final do exemplo 1.

Influência da Alternância de Arestas

O exemplo analítico apresentado anteriormente é novamente apresentado aqui, porém, sem utilizar a alternância de arestas após cada refinamento e desrefinamento realizado ao longo das iterações de adaptação. O resultado é apresentado na Tab. 6.2. Observa-se que, embora a alternância de arestas não modifique o número de nós ou elementos da malha, ela pode alterar as próximas iterações de adaptação. No presente caso, a falta de alternância de arestas resultou em um aumento de número de nós e de elementos ao longo das iterações de adaptação. Mas, acima de tudo, verifica-se que α_{160} é muito superior neste caso, indicando que a malha é muito mais suscetível a instabilidades decorrentes da violação da condição angular superior. O ângulo máximo α_{max} da malha possui comportamento variado comparando-se com o caso onde emprega-se a alternância de arestas, sendo em algumas iterações maior e em outras menor. Entretanto, verifica-se a baixa capacidade

Tabela 6.2: Dados das iterações de adaptação do exemplo 1 sem alternância de arestas

Adaptação	Nós	Elementos	α_{max}	α_{160}	$\sigma^2(E)$
0	221	400	90.00	0.000E-0	4.506E-2
1	262	497	150.25	0.000E-0	2.709E-2
2	334	638	161.56	1.044E-3	3.358E-2
3	445	849	168.69	5.104E-3	2.167E-2
4	561	1074	171.86	8.690E-3	2.004E-2
5	712	1374	173.65	1.043E-2	1.409E-2
6	926	1793	177.06	1.022E-2	9.450E-3
7	1178	2285	177.06	8.752E-3	7.895E-3
8	1449	2820	172.87	7.328E-3	6.053E-3
9	1838	3586	173.65	8.551E-3	4.170E-3
10	2281	4459	174.80	1.225E-2	2.924E-3
11	2867	5616	174.80	1.258E-2	1.974E-3
12	3550	6973	178.98	1.242E-2	1.680E-3
13	4448	8754	178.98	1.431E-2	1.163E-3
14	5608	11052	178.98	1.637E-2	8.269E-4
15	7049	13911	179.18	1.777E-2	6.063E-4
16	8666	17118	179.22	2.038E-2	4.696E-4
17	10653	21074	179.94	2.225E-2	3.738E-4
18	13093	25930	179.94	2.469E-2	2.939E-4
19	16113	31924	179.94	2.727E-2	2.273E-4
20	20013	39705	179.97	2.882E-2	1.777E-4
21	24391	48409	179.96	3.109E-2	1.478E-4
22	29290	58175	179.97	3.625E-2	1.241E-4
23	34711	68982	179.96	4.056E-2	1.380E-4
24	40965	81464	179.97	4.442E-2	1.567E-4
25	47999	95497	179.94	4.900E-2	2.109E-4

do algoritmo reduzir α_{max} ao longo das iterações quando não faz-se uso da alternância de arestas. A variância do erro $\sigma^2(E)$ apresenta comportamento inverso nesse caso. Ela tende a ser menor, indicando malhas com erro mais equidistribuídos. Isto explica-se pelo maior número de nós.

A Fig. 6.5 mostra a mesma região da malha adaptada (iteração 25) ampliada em detalhes que foi mostrada na Fig. 6.4. Neste caso, porém, sem o uso da alternância de arestas. É possível visualizar regiões com concentração de elementos mal orientados e com densidades anômalas. Com isso demonstra-se que a alternância de arestas utilizada segundo os critérios propostos conduz à malhas de qualidade superior e mais estáveis.

Influência do Limiar Superior

Altera-se neste exemplo o valor do limiar superior para $\eta_S = 1.6$ e mantêm-se o do limiar inferior de $\eta_I = 0.2$. A Fig. 6.6 mostra as malhas obtidas ao longo do processo

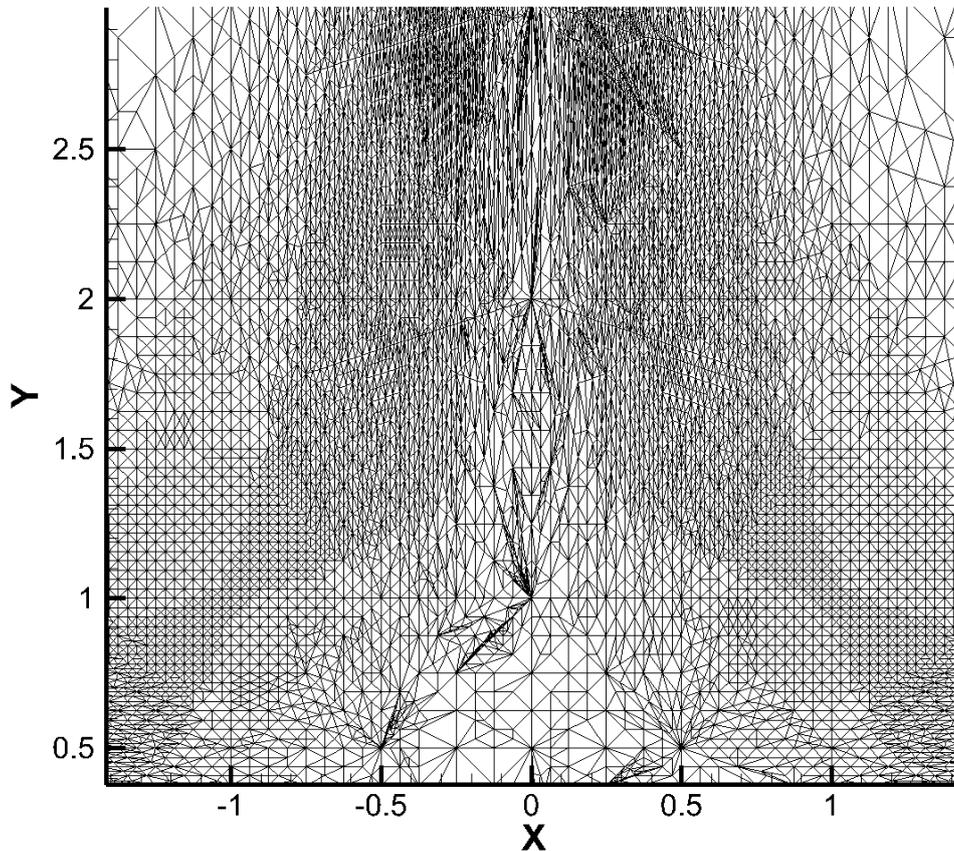
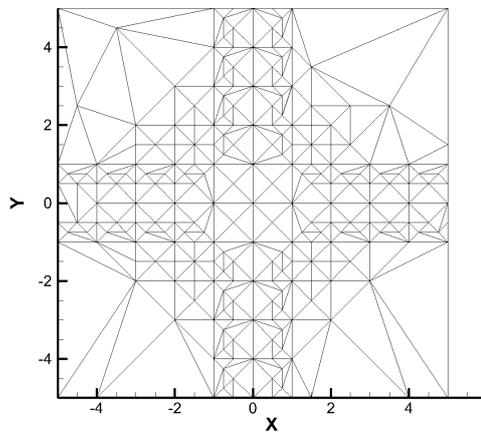


Figura 6.5: Detalhe aumentado de uma porção da malha adaptada final do exemplo 1 sem alternância de arestas.

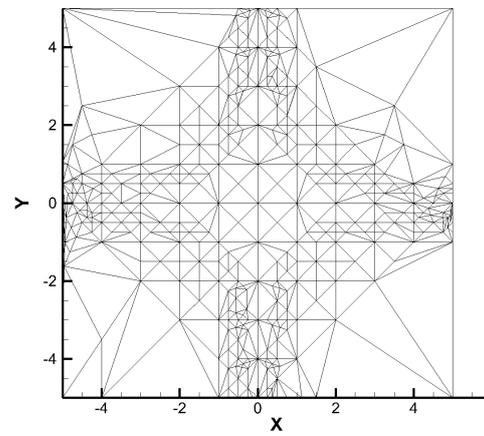
de adaptação com estes parâmetros. Torna-se claro que um aumento do limiar superior implica em maior tolerância ao erro no procedimento de refinamento. Como resultado, malhas menos densas são obtidas. A recíproca é verdadeira, já que ao se diminuir o valor do limiar superior a malha tende a ser mais densa.

Influência do Limiar Inferior

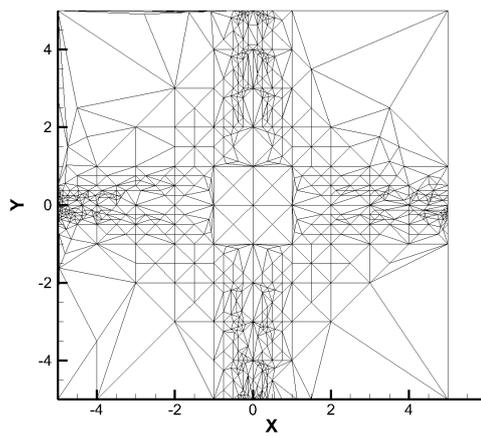
Altera-se neste exemplo o valor do limiar inferior para $\eta_I = 0.3$ e mantêm-se o do limiar superior original de $\eta_S = 1.5$. A Fig. 6.7 mostra as malhas obtidas ao longo do proceso de adaptação com estes parâmetros. Da mesma forma que para o limiar superior, o limiar inferior altera a tolerância ao erro mas com relação ao processo de desrefinamento. Quanto maior o limiar inferior, menos densa será a malha e vice-versa.



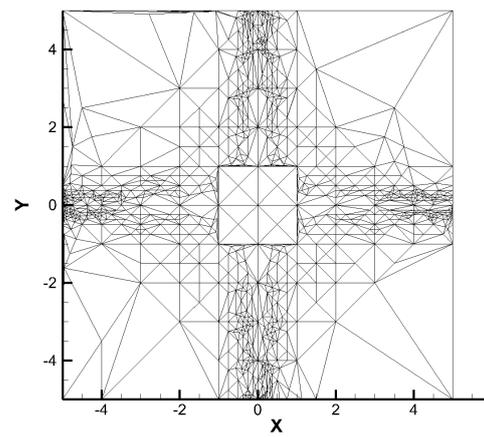
(a) Adaptação 1



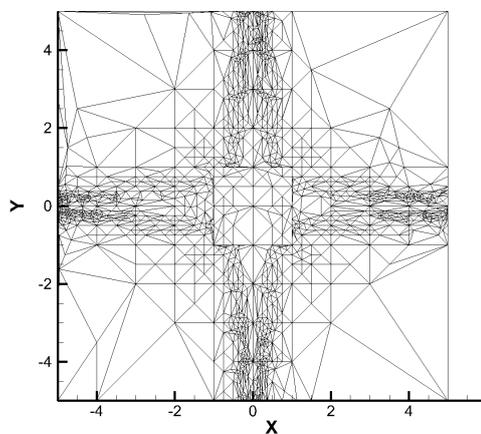
(b) Adaptação 5



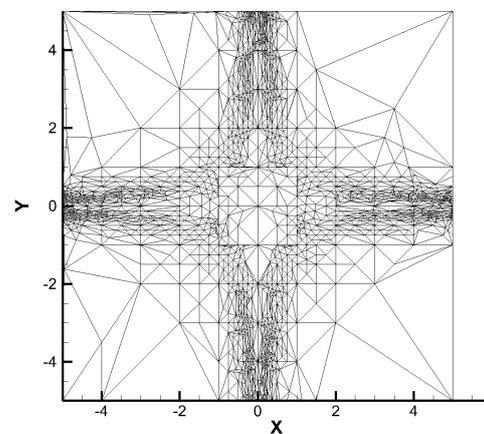
(c) Adaptação 10



(d) Adaptação 15

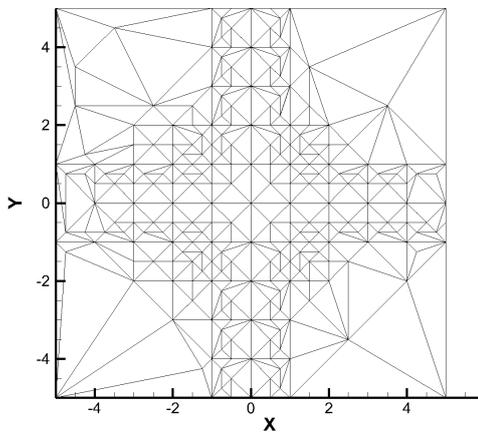


(e) Adaptação 20

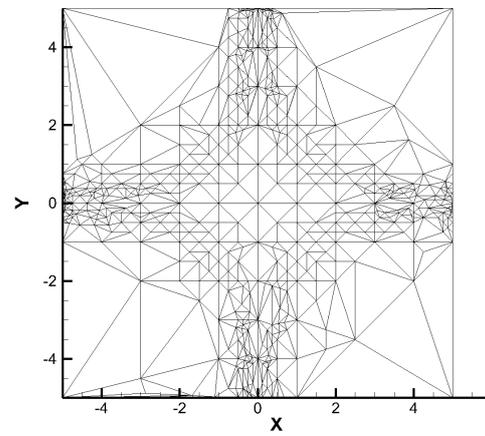


(f) Adaptação 25

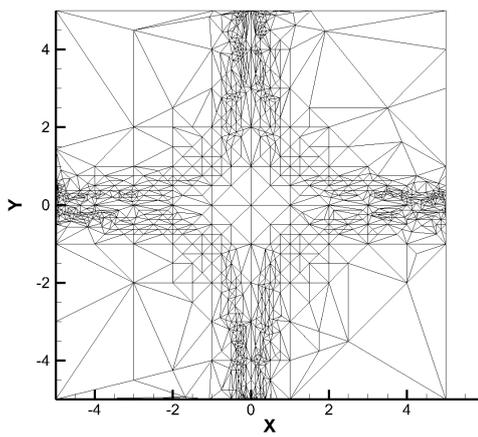
Figura 6.6: Malha ao longo das iterações de adaptação do exemplo 1 alterando-se o limiar superior para $\eta_S = 1.6$.



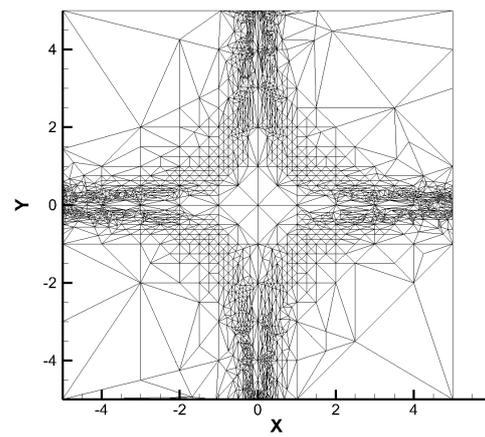
(a) Adaptação 1



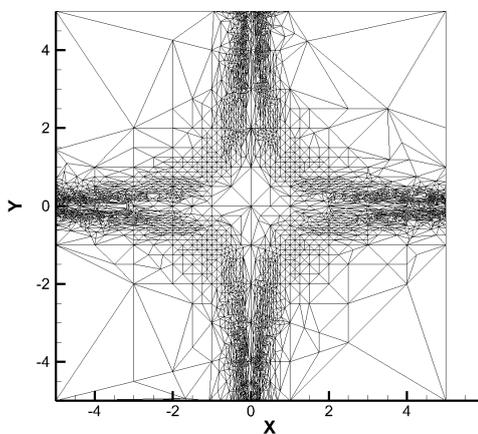
(b) Adaptação 5



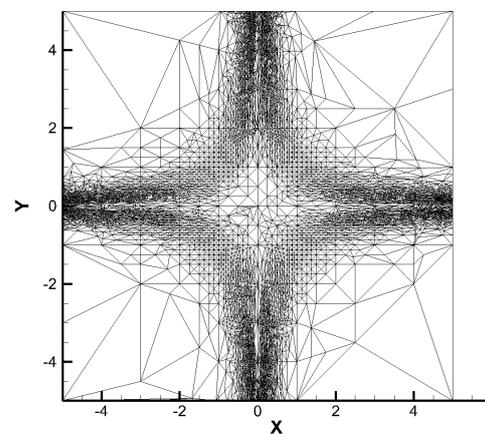
(c) Adaptação 10



(d) Adaptação 15



(e) Adaptação 20



(f) Adaptação 25

Figura 6.7: Malha ao longo das iterações de adaptação do exemplo 1 alterando-se o limiar inferior para $\eta_I = 0.3$.

Entretanto, desrefinar em excesso pode levar à perda de resolução da solução. Isto é, ao utilizar-se um limiar inferior demasiadamente grande, corre-se o risco do algoritmo desrefinar regiões com gradientes médios, onde a resolução após o desrefinamento torna-se suscetível à perda de qualidade nesta região.

Robustez da Metodologia

Para demonstrar a robustez do algoritmo de adaptação implementado, realiza-se novamente o mesmo exemplo, entretanto utilizando-se uma malha grosseira e uma solução inicial completamente deteriorada. A malha possui apenas 4 nós e 2 elementos, como indica a Fig. 6.8. A partir da solução inicial, realizou-se $N_F = 25$ ciclos de adaptação utilizando-se um limiar superior $\eta_S = 1.3$ e um limiar inferior $\eta_I = 0.2$. As malhas obtidas ao longo das iterações de adaptação são mostradas na Fig. 6.9 e as respectivas soluções na Fig. 6.10. Fica demonstrado a capacidade do algoritmo de captar o erro e de adaptar a malha para condições limites. É importante comentar, entretanto, que a adaptação acoplada ao algoritmo CBS (ou qualquer outro algoritmo de simulação) deve pressupor que a física esteja captada na solução inicial. Isto é, por mais robusto que o algoritmo de adaptação possa ser, ele jamais poderá adaptar a malha em casos onde a física esteja incorreta.

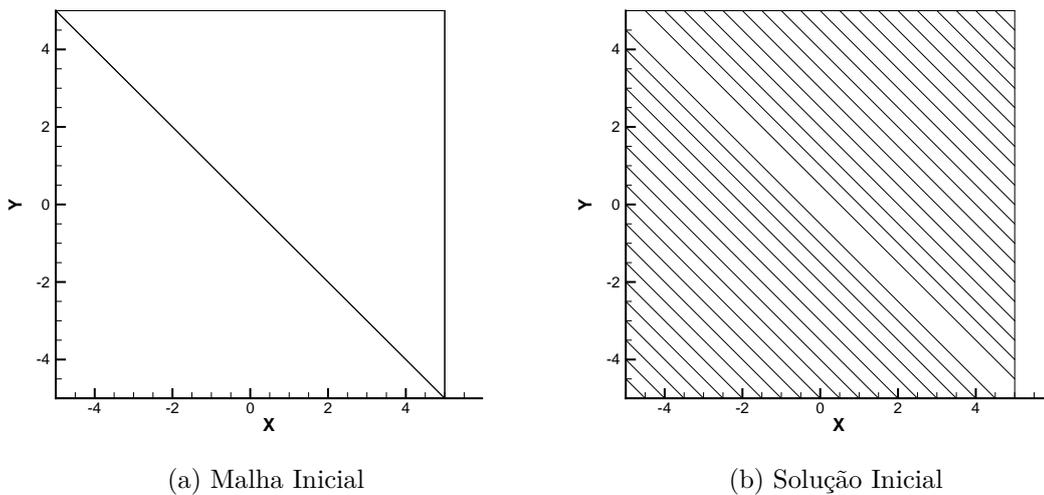
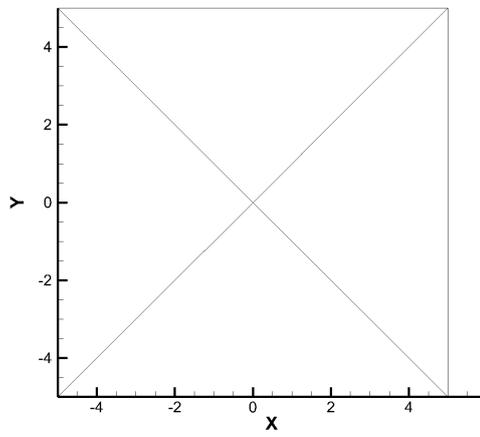
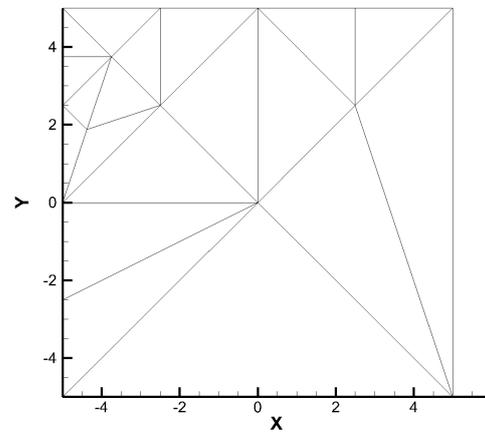


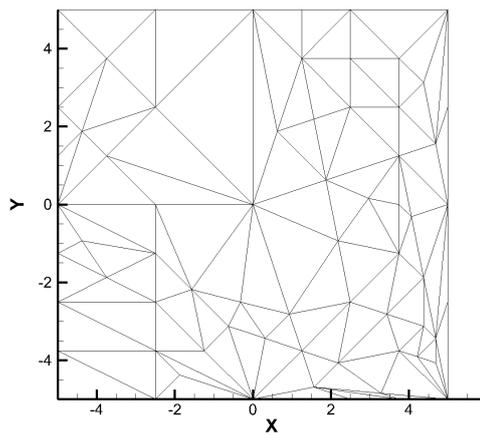
Figura 6.8: Malha e solução inicial do exemplo 1 modificado.



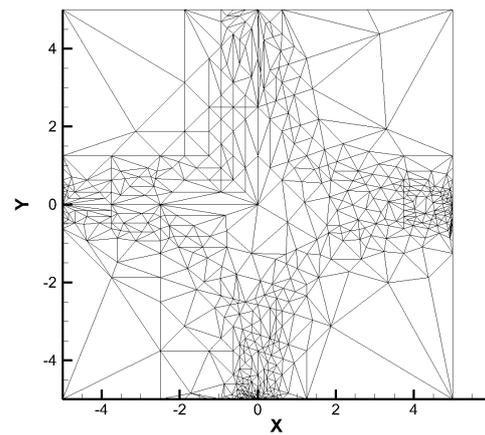
(a) Adaptação 1



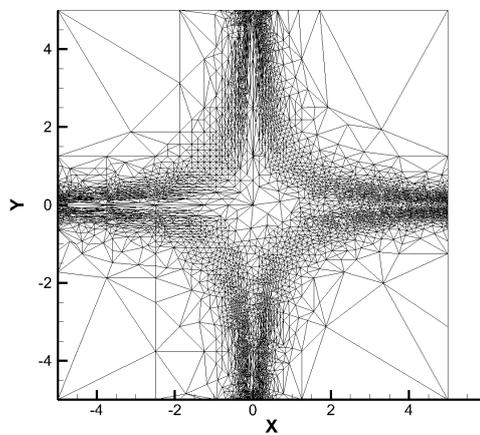
(b) Adaptação 5



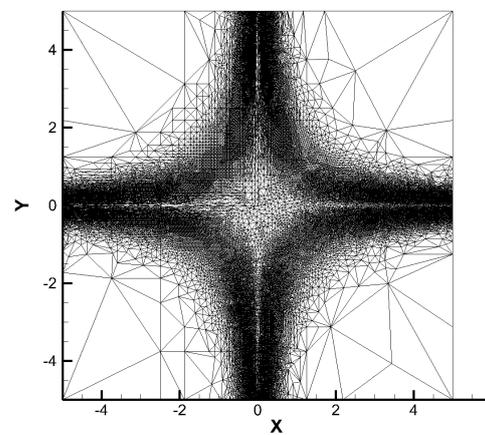
(c) Adaptação 10



(d) Adaptação 15

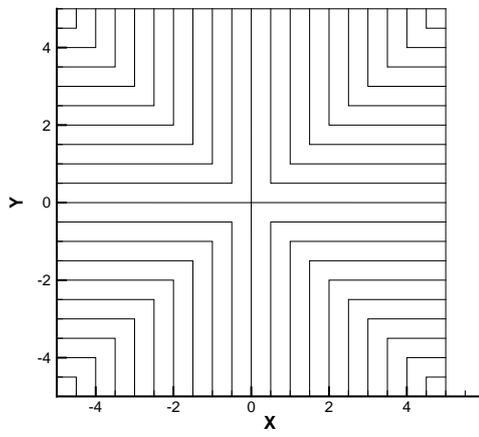


(e) Adaptação 20

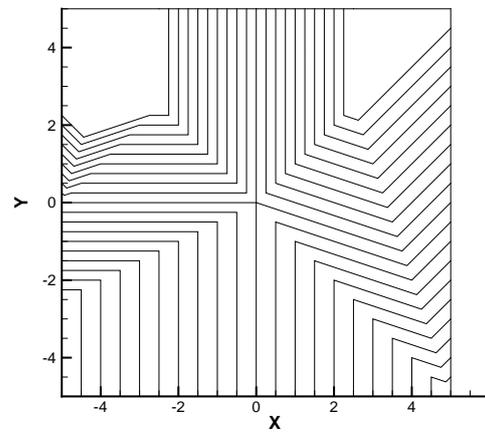


(f) Adaptação 25

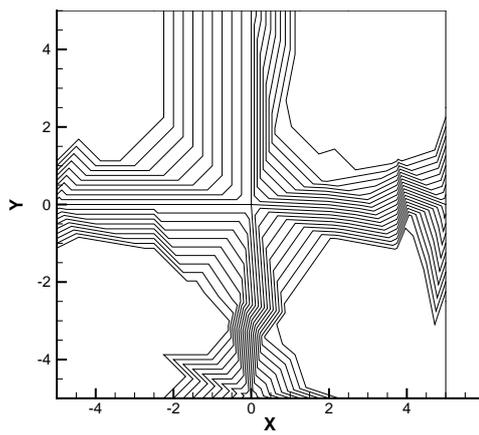
Figura 6.9: Malha ao longo das iterações de adaptação do exemplo 1 modificado.



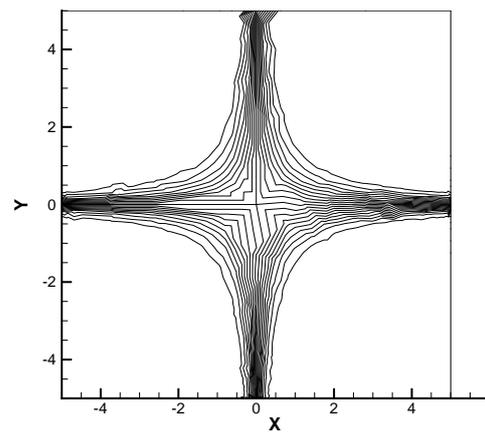
(a) Adaptação 1



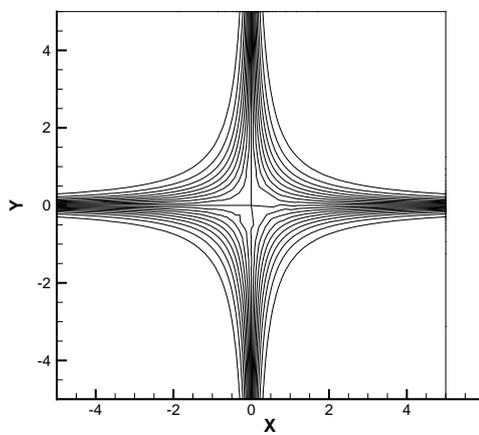
(b) Adaptação 5



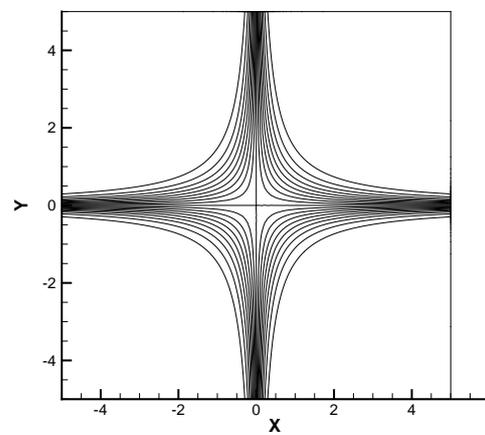
(c) Adaptação 10



(d) Adaptação 15



(e) Adaptação 20



(f) Adaptação 25

Figura 6.10: Solução ao longo das iterações de adaptação do exemplo 1 modificado.

6.1.2 Exemplo 2: Escoamento Supersônico Não-Viscoso em Canal com Rampa de Compressão e Expansão

Neste problema, o fluido é forçado a passar por um canal contendo uma rampa. Na região próxima ao início da rampa forma-se uma onda de choque por compressão, a qual sofre reflexão três vezes, entre as paredes do canal, até encontrar a saída. Ao final da rampa formam-se ondas de Mach de expansão, as quais expalham-se, atingindo a parede superior do canal e refletindo-se até o fim do canal. Estas ondas de Mach interferem sobre as ondas de choque decorrentes da compressão e vice-versa. A geometria e as condições de contorno do problema são mostradas na Fig. 6.11. Os parâmetros utilizados para o escoamento são:

- $M_\infty = 2.0$
- $u_{1\infty} = 1.0$
- $u_{2\infty} = 0.0$
- $CFL = 0.1$
- $C_E = 0.25$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 300K$

Os parâmetros de adaptação utilizados são:

- $\phi = \rho$
- $\eta_S = 1.5$
- $\eta_I = 0.2$
- $N_F = 7$
- $F_{\Delta t} = 500$

A Fig. 6.14 mostra a malha inicial e a respectiva solução em termos de massa específica. A malha inicial possui 3607 nós e 6896 elementos distribuídos de forma uniforme. As linhas de contorno para a solução indicada na Fig. 6.14 revelam que as ondas de choque não encontram-se confinadas à uma pequena região, mas sim esparsas e a onda de Mach de expansão possui largura inicial demasiadamente grande. Esta resolução ruim da solução é decorrente de uma discretização espacial insuficiente para o problema. A solução adaptada para esta solução inicial é mostrada na Fig. 6.15. Observa-se uma

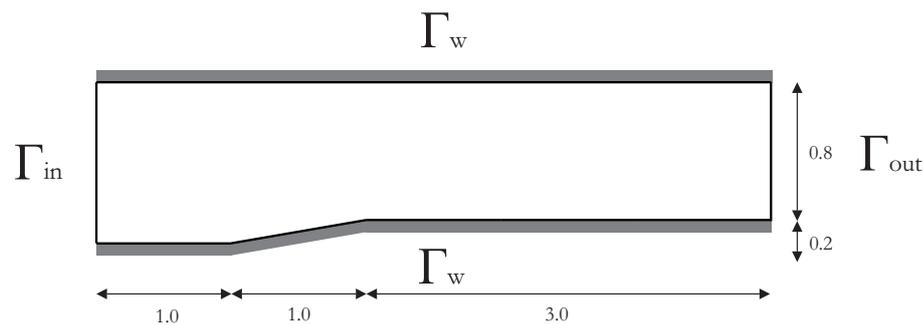


Figura 6.11: Geometria e condições de contorno do exemplo 2.

melhora significativa na qualidade do resultado final com a malha adaptada. As ondas de choque estão agora confinadas a uma região muito menor e a região de expansão ao final da rampa tem uma melhor resolução. O resíduo ao longo dos incrementos de tempo é mostrado na Fig. 6.12. Visualiza-se claramente a região de convergência inicial seguida por 7 ciclos de adaptação. Os ciclos intermediários avançam exatamente $F_{\Delta t} = 500$ passos cada, sendo que na última adaptação converge-se o resultado novamente. Fica clara a perda momentânea da convergência após cada ciclo de adaptação. Assim, o número de passos entre uma adaptação e outra funciona permitindo que se re-estabeleça novamente parte desta convergência entre os ciclos para que a nova adaptação já contenha mudanças no campo de solução. Na Fig. 6.13 apresenta-se a massa específica ρ em função da distância x ao longo da linha $y = 0.5$ para a solução inicial e a adaptada final. Visualiza-se que a descontinuidade da solução introduzida pelas ondas de choque é captada perfeitamente pela malha adaptada bem como regiões de gradiente intermediário provenientes da expansão de ondas de Mach também possuem a resolução melhorada. É possível observar também a diferença de intensidade de cada onda de choque, sendo a primeira praticamente de mesma intensidade que a segunda, que é sua primeira reflexão. Já a segunda e terceira onda refletida apresentam uma pequena diminuição de variação de massa entre o ponto inicial e final do choque.

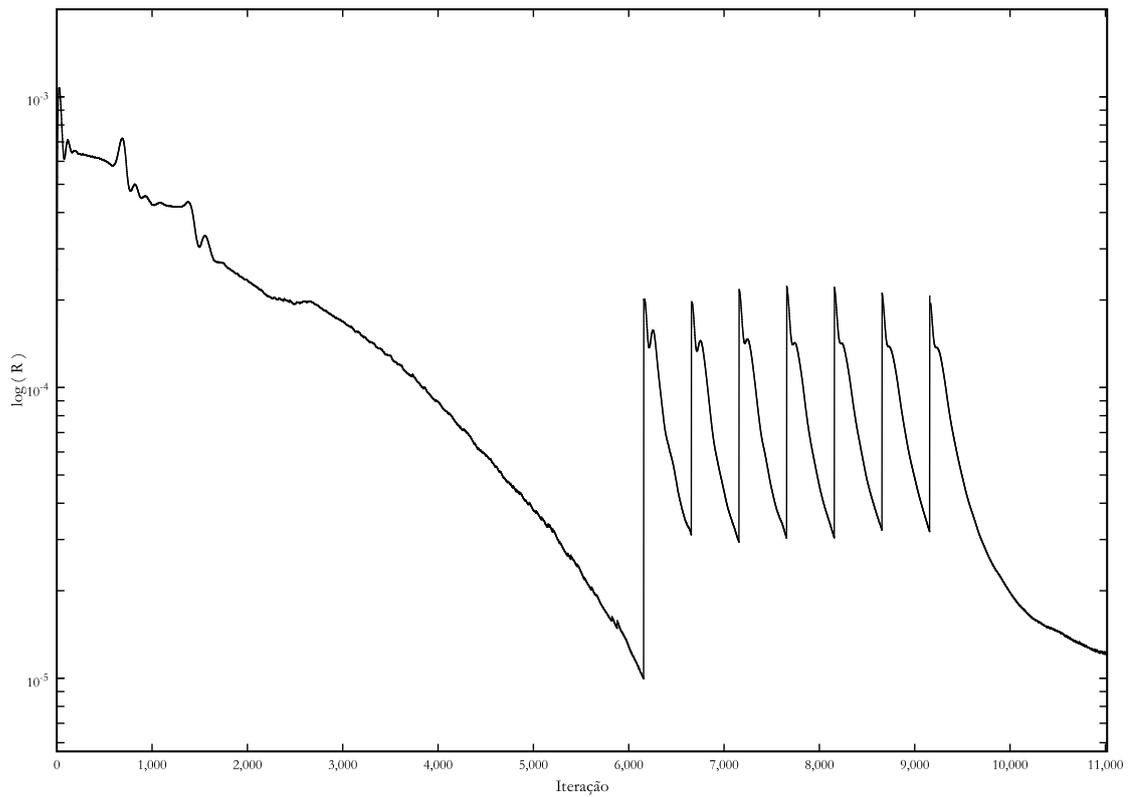
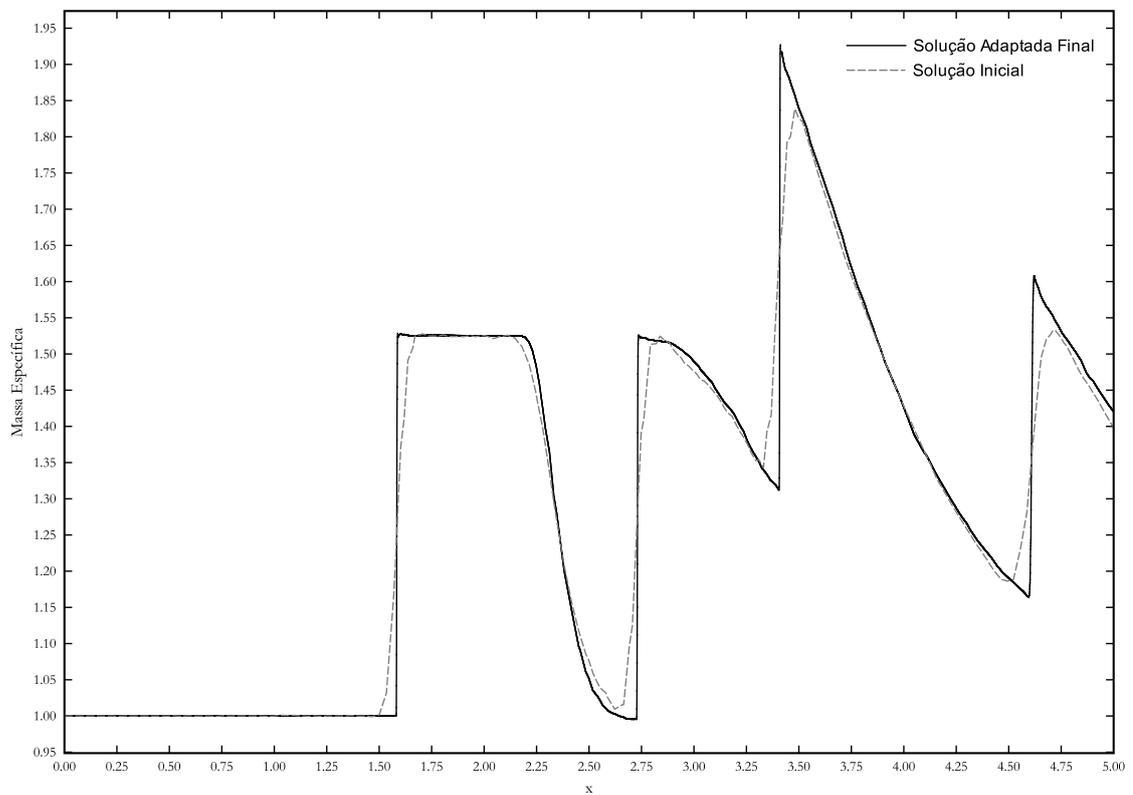
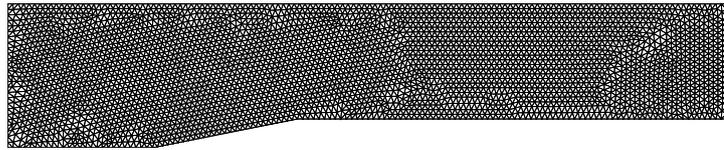


Figura 6.12: Convergência do exemplo 2.

Figura 6.13: Massa específica em função de x para $y = 0.5$ do exemplo 2.



(a) Malha Inicial

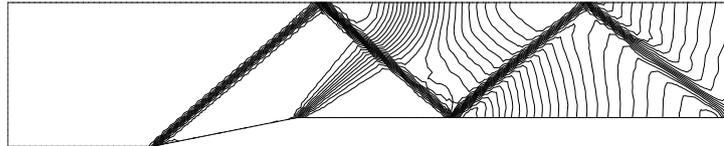
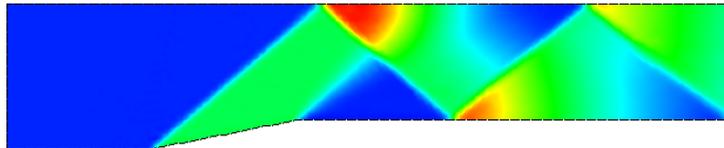
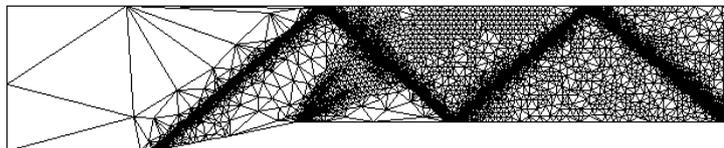
(b) Solução Inicial (isocontornos de ρ)(c) Solução Inicial (campo de ρ)

Figura 6.14: Malha e solução inicial do exemplo 2.



(a) Malha Final

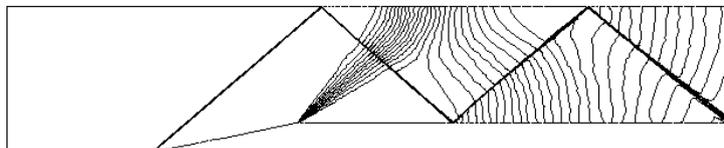
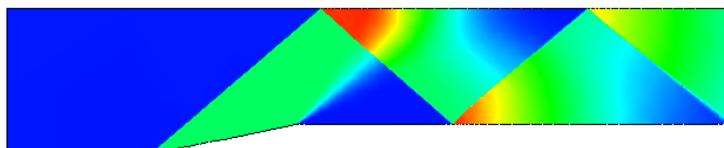
(b) Solução Final (isocontornos de ρ)(c) Solução Final (campo de ρ)

Figura 6.15: Malha e solução final do exemplo 2.

Os dados da malha ao longo das iterações de adaptação são mostrados na Tab. 6.3. Uma diferença fundamental com relação ao problema analítico é que $\sigma^2(E)$ ao final de uma adaptação não corresponde a $\sigma^2(E)$ do início da adaptação seguinte, uma vez que o campo de solução varia entre uma adaptação e outra. Os valores indicados correspondem ao valor ao final de cada adaptação.

Tabela 6.3: Dados das iterações de adaptação do exemplo 2

Adaptação	Nós	Elementos	α_{max}	α_{160}	$\sigma^2(E)$
0	3604	6896	105.30	0.000E-0	6.624E-3
1	4983	9705	169.05	1.030E-4	2.525E-3
2	8568	16845	173.00	3.957E-5	2.497E-3
3	14866	29423	173.00	2.265E-5	2.499E-3
4	25795	51255	175.63	1.951E-5	2.474E-3
5	43526	86704	174.94	5.306E-5	2.512E-3
6	72392	144429	178.30	1.846E-5	2.635E-3
7	119547	238783	178.86	3.343E-5	2.746E-3

6.1.3 Exemplo 3: Escoamento Supersônico Não-Viscoso Sobre Aerofólio NACA0012

Este exemplo analisa um escoamento supersônico sobre um aerofólio padrão NACA 0012 com ângulo de ataque nulo. Os dados utilizados para o escoamento são:

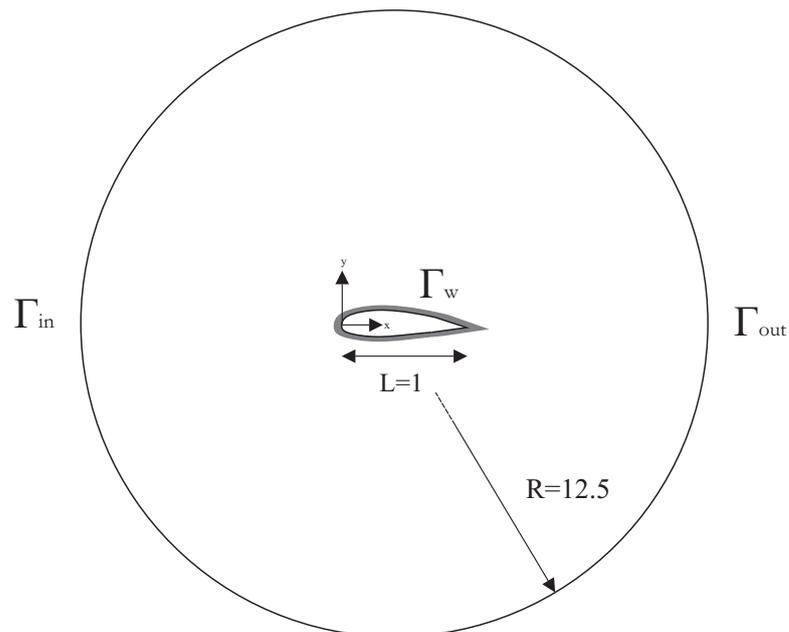
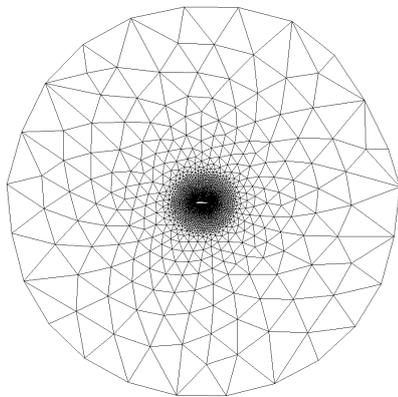


Figura 6.16: Geometria e condições de contorno do exemplo 3.

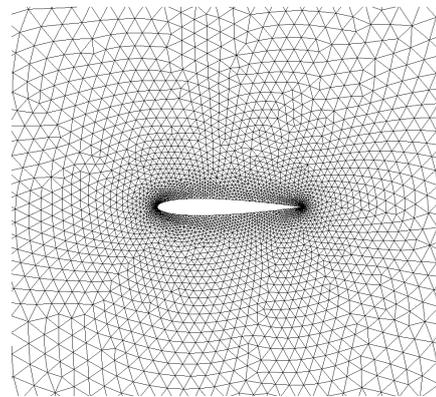
- $M_\infty = 2.0$
- $u_{1\infty} = 1.0$
- $u_{2\infty} = 0.0$
- $CFL = 0.1$
- $C_E = 0.25$
- $R_T = 1 \times 10^{-4}$
- $T_\infty = 300K$

Os parâmetros de adaptação utilizados são:

- $\phi = \rho$
- $\eta_S = 1.7$
- $\eta_I = 0.1$
- $N_F = 7$
- $F_{\Delta t} = 1500$



(a) Malha inicial (visão geral)



(b) Malha inicial (detalhe)

Figura 6.17: Malha inicial do exemplo 3.

A geometria do domínio e as condições de contorno empregadas são indicadas na Fig. 6.16. O domínio é circular, com diâmetro no valor de 25, sendo que o comprimento 1 equivale ao valor da corda média do aerofólio. A malha inicial é apresentada na Fig. 6.17 e a solução inicial é mostrada na Fig. 6.18. Percebe-se a falta de extensão da onda de choque, a qual dispersa-se rapidamente devido à falta de discretização espacial associada à difusão artificial. A Fig. 6.19 mostra a solução final para o problema adaptado. É possível observar que a onda de choque torna-se confinada a uma região muito mais estreita, melhorando a resolução tanto no bordo de ataque quando no de fuga. A discretização espacial melhorada também permite que a onda de choque se propague por uma região maior. Na malha inicial há muita difusão nesta região, não apenas limitando a propagação da onda de choque mas também curvando-a à medida que se dispersa. A Fig. 6.20 mostra a malha final adaptada para o problema. Devido à característica hiperbólica do problema (escoamento não-difusivo supersônico) e de sua geometria, não há nenhum gradiente na corrente não perturbada do escoamento na região anterior ao aerofólio. Esta região é praticamente totalmente desrefinada. A Fig. 6.21 apresenta o coeficiente de pressão obtido com a solução adaptada e a solução adaptada obtida por Dolejsí e Felcman (2004). Ambas soluções são muito próximas, indicando boa concordância. A Fig. 6.22 mostra a convergência do problema em função das iterações. Os dados da malha ao longo das iterações de adaptação são mostrado na Tab. 6.4. A Fig. 6.23 mostra o resultado em termos de Mach obtida por Dolejsí e Felcman (2004).

Tabela 6.4: Dados das iterações de adaptação do exemplo 3

Adaptação	Nós	Elementos	α_{max}	α_{160}	$\sigma^2(E)$
0	3753	7351	108.36	0.000E-0	9.501E-3
1	5690	11200	150.80	0.000E-0	5.520E-3
2	8450	16711	149.69	0.000E-0	6.116E-3
3	12305	24421	149.69	0.000E-0	6.984E-3
4	18066	35941	151.15	0.000E-0	7.759E-3
5	26940	53689	153.35	0.000E-0	8.886E-3
6	41100	82088	156.53	0.000E-0	9.927E-3
7	63111	126030	164.95	3.343E-5	1.071E-2

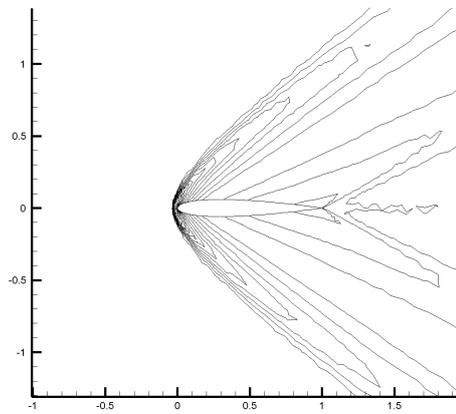
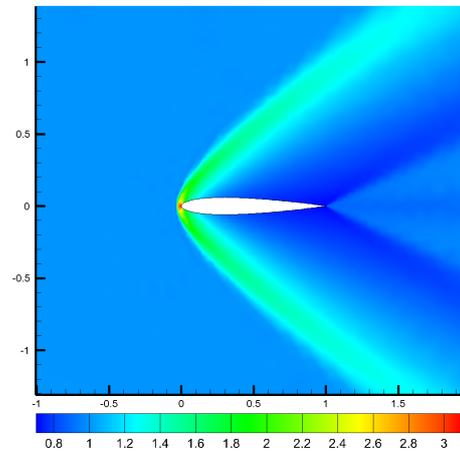
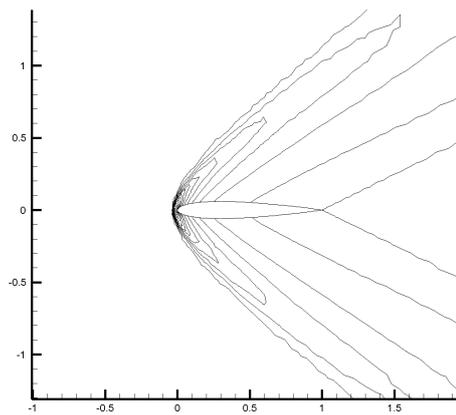
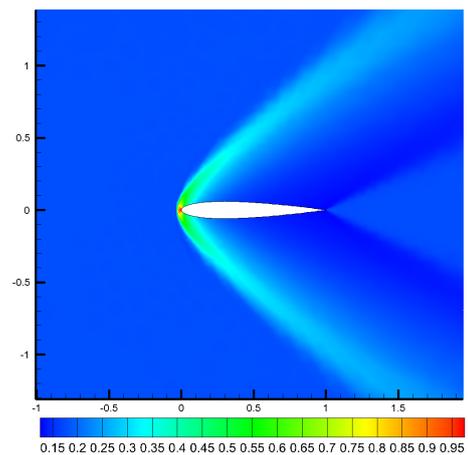
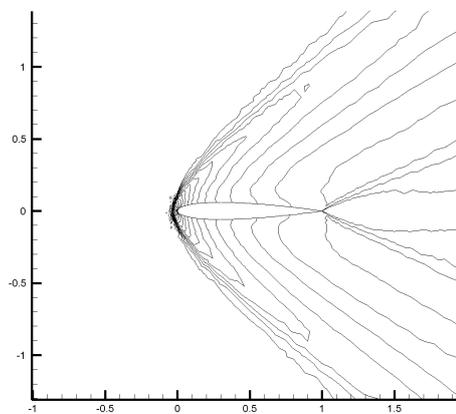
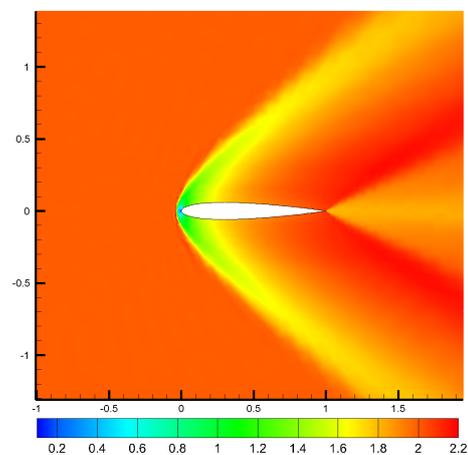
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.18: Solução inicial do exemplo 3.

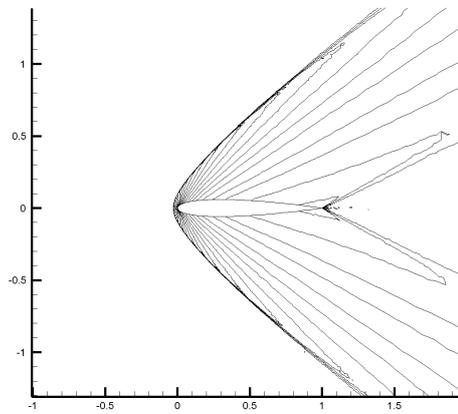
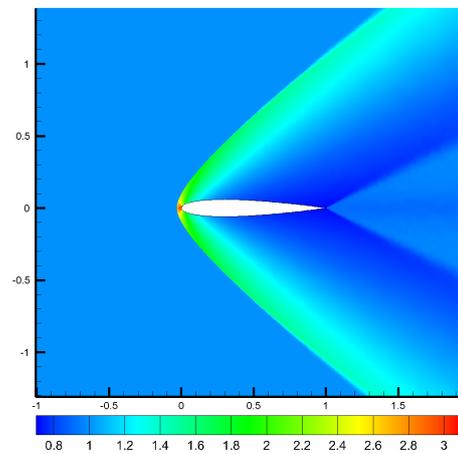
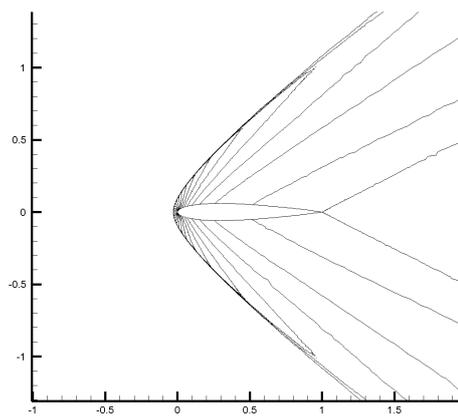
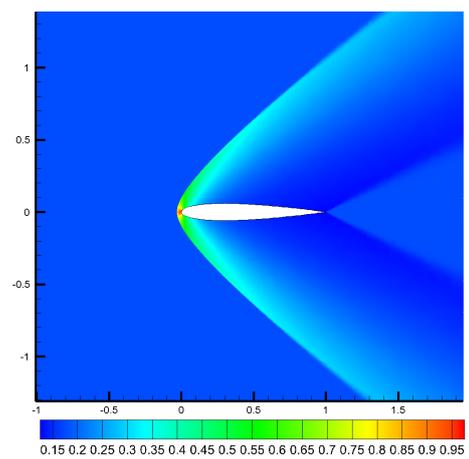
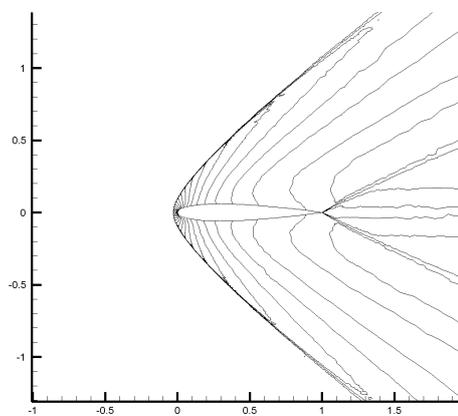
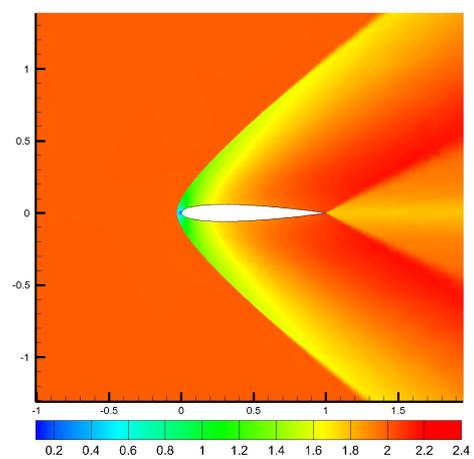
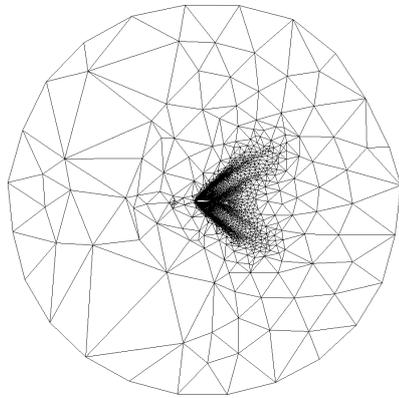
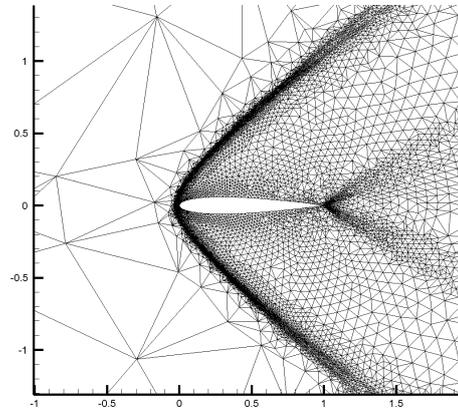
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.19: Solução final com adaptação do exemplo 3.



(a) Malha final (visão geral)



(b) Malha final (detalhe)

Figura 6.20: Malha adaptada final do exemplo 3.

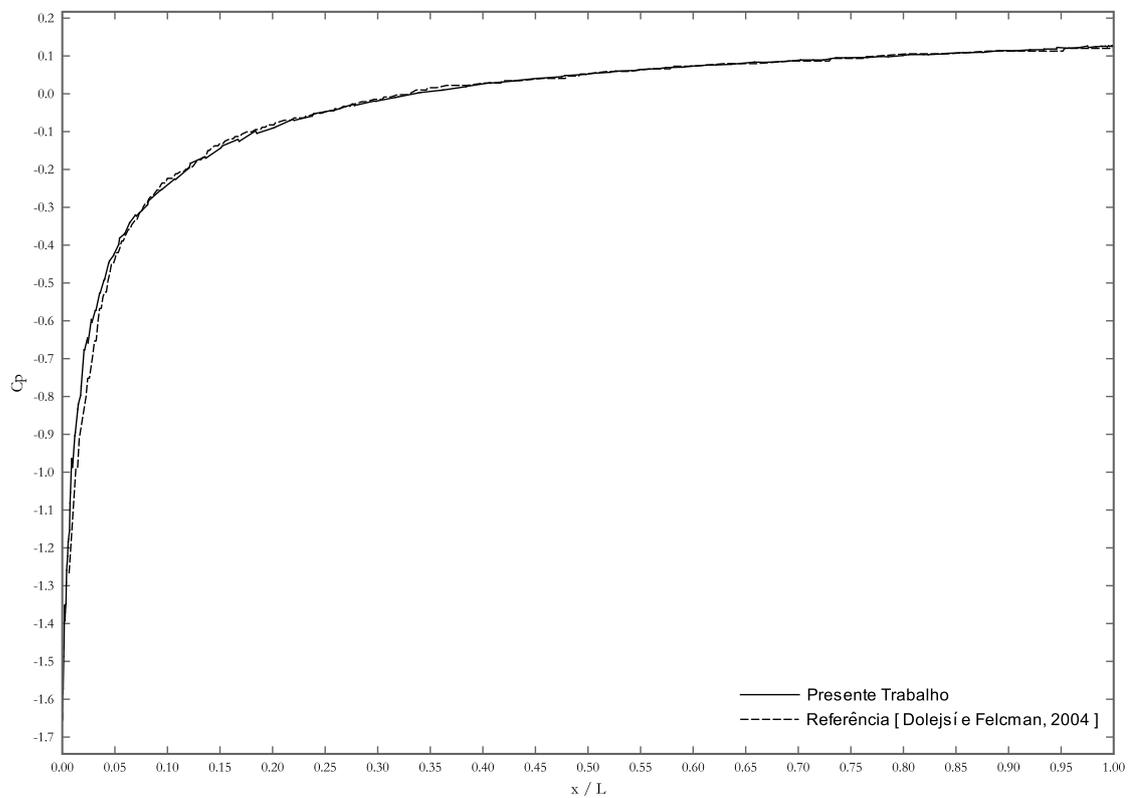


Figura 6.21: Coeficiente de pressão para solução adaptada do exemplo 3.

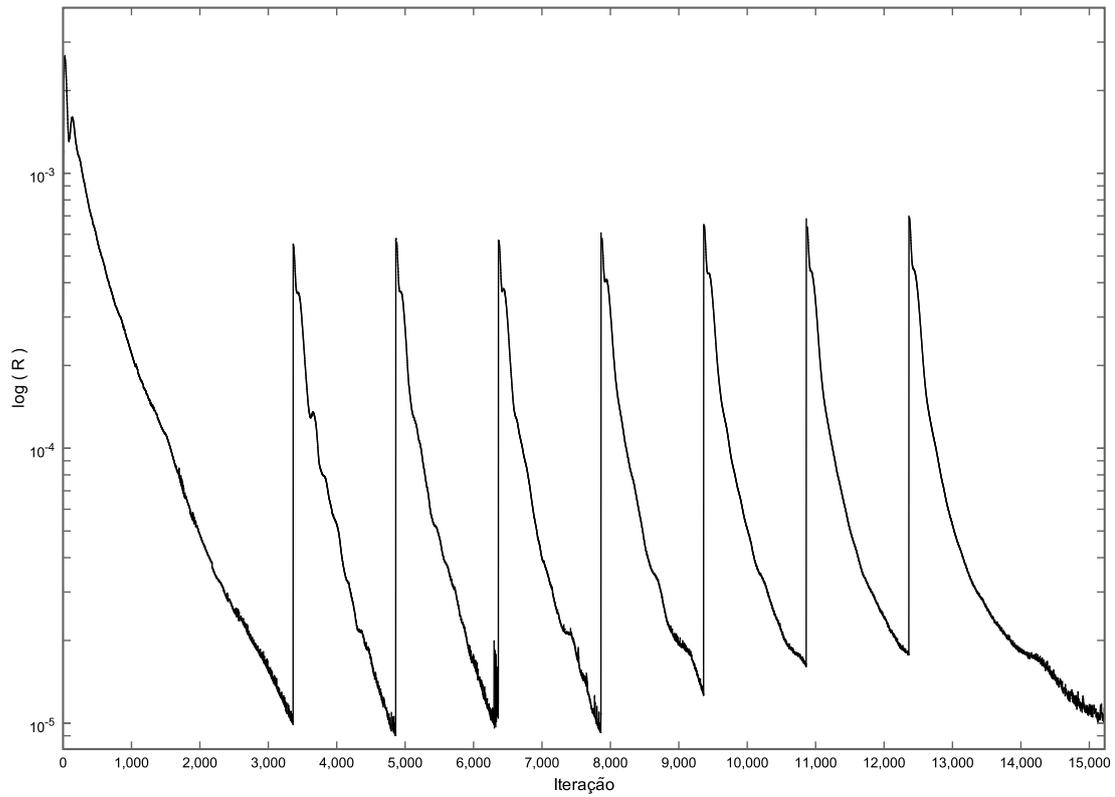


Figura 6.22: Convergência do exemplo 3.

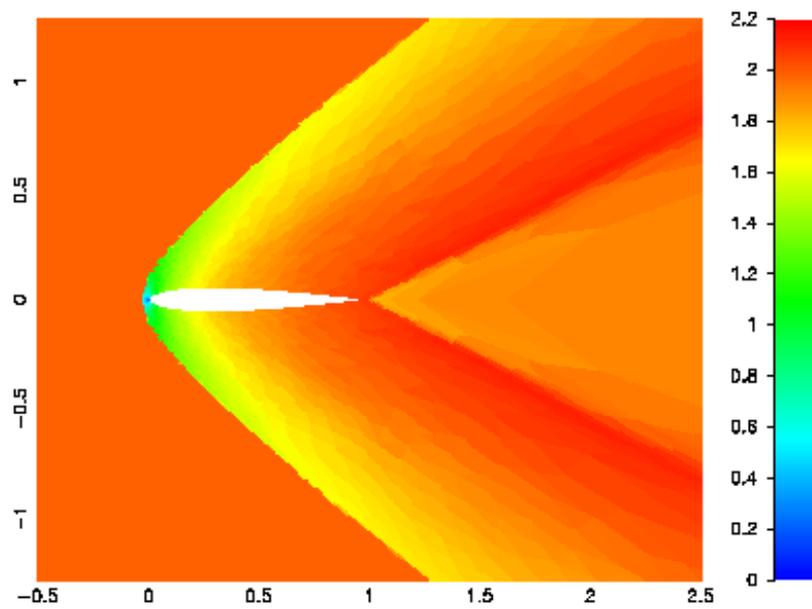


Figura 6.23: Solução (campo de M) obtida por Dolejší e Felcman (2004).

6.1.4 Exemplo 4: Escoamento Transônico Viscoso Sobre Aerofólio NACA0012

O presente exemplo utiliza a mesma malha e condições de contorno utilizadas no exemplo anterior (Fig. 6.16). Entretanto, o problema é transônico, difusivo e com um ângulo de ataque de 10° . É importante ressaltar que fisicamente o ar nesta velocidade desenvolveria um escoamento com número de Reynolds muito mais elevado. Entretanto, simula-se o problema com um número de Reynolds muito baixo para comparação com valores apresentados na literatura, mesmo que tal escoamento seja fisicamente incorreto, ou incoerente. Os dados empregados no escoamento são:

- $M_\infty = 0.8$
- $u_{1\infty} = 0.9848$
- $u_{2\infty} = 0.1736$
- $CFL = 0.1$
- $C_E = 0.25$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 300K$
- $Re = 500$

Os parâmetros de adaptação utilizados são:

- $\phi = M$
- $\eta_S = 1.7$
- $\eta_I = 0.1$
- $N_F = 7$
- $F_{\Delta t} = 1000$

Tabela 6.5: Dados das iterações de adaptação do exemplo 4

Adaptação	Nós	Elementos	α_{max}	α_{160}	$\sigma^2(E)$
0	3753	7351	108.36	0.000E-0	4.427E-3
1	6194	12207	149.47	0.000E-0	2.074E-3
2	9913	19624	152.77	0.000E-0	1.153E-3
3	14750	29271	152.03	0.000E-0	7.503E-4
4	21278	42294	166.43	1.576E-5	5.860E-4
5	30436	60579	166.43	2.200E-5	4.494E-4
6	44198	88045	168.60	2.271E-5	3.002E-4
7	64678	128922	169.94	2.585E-5	1.934E-4

Os resultados obtidos com a malha inicial são apresentados na Fig. 6.24 e os obtidos com a malha adaptada final encontram-se na Fig. 6.25. Os resultados obtidos por Bono (2008) são apresentados na Fig. 6.26 para comparação. Observa-se boa concordância com os campos de solução obtidos. As linhas de corrente obtidas no presente trabalho com a malha inicial e final são mostradas na Fig. 6.27. As linhas de correntes obtidas com a malha inicial assemelham-se muito às obtidas por Bono (2008) apresentadas na Fig. 6.26. Entretanto, a malha adaptada revela pequenas diferenças como o aumento da recirculação inferior e o prolongamento da extensão destas recirculações na direção do escoamento. A malha adaptada final é apresentada na Fig. 6.28. O histórico da convergência é mostrado na Fig. 6.29. Para este problema, a convergência logo após o procedimento de adaptação é rapidamente restabelecida, promovendo também uma diminuição do resíduo. As Fig. 6.30 e Fig. 6.31 mostram o coeficiente de sustentação C_L e de arrasto C_D , respectivamente, ao longo das iterações. Percebe-se uma grande variação destes coeficientes até o resultado final. A Fig. 6.32 mostra o coeficiente de pressão C_p comparado com o resultado obtido por Katz (2009), o qual utiliza um método de alta precisão. Os resultados apresentam ótima concordância. A Tab. 6.5 mostra os dados da malha ao longo das iterações e a Tab. 6.6 um resumo comparativo dos coeficientes aerodinâmicos obtidos para este exemplo. Os resultados possuem boa concordância com a referência além de mostrar uma variação significativa de C_L após a adaptação e uma variação muito baixa de C_D . Uma explicação possível é o fato da malha inicial ser muito inapropriada para simular escoamentos viscosos, pois não há uma camada limite, a qual é construída naturalmente ao longo da adaptação.

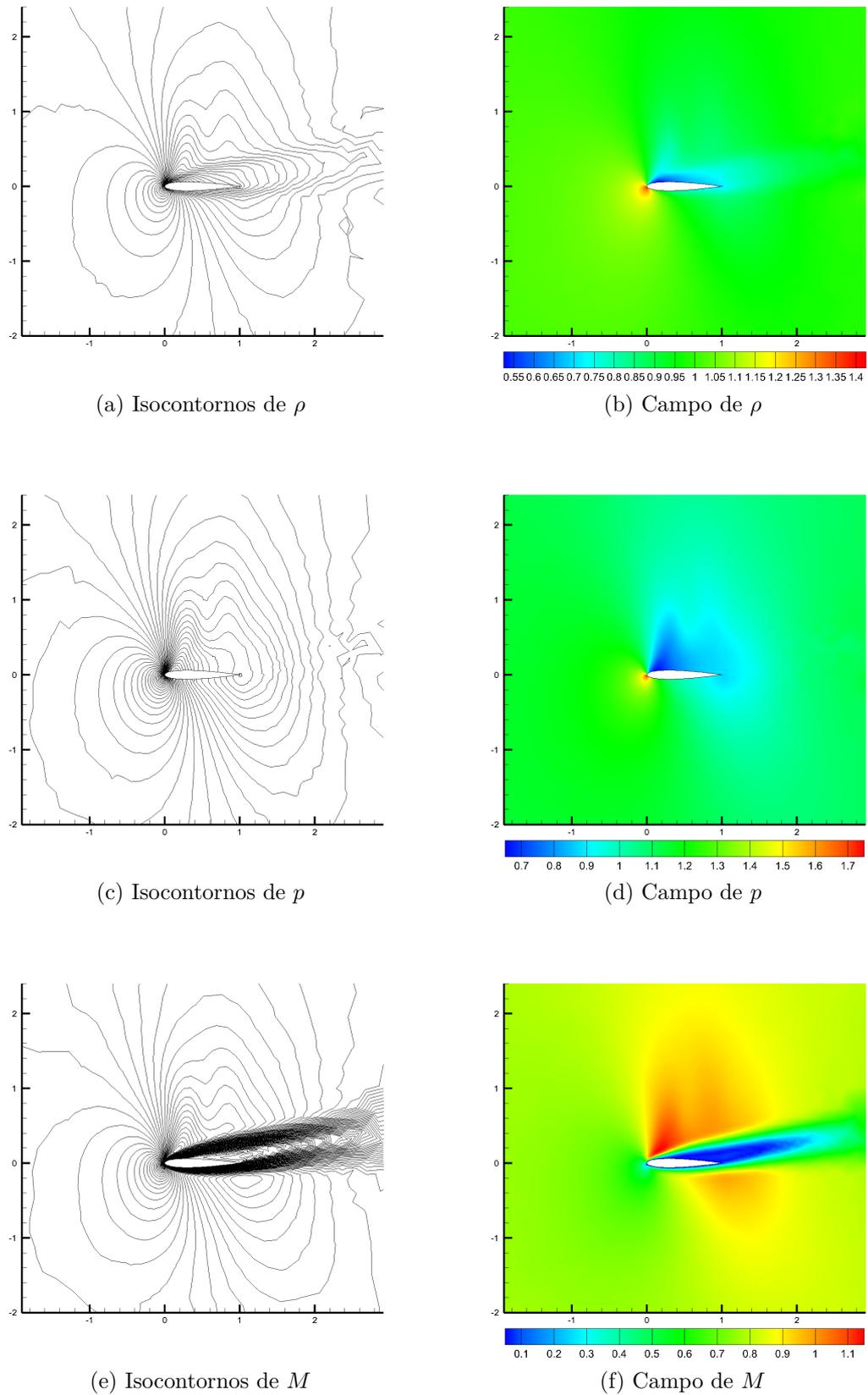


Figura 6.24: Solução inicial do exemplo 4.

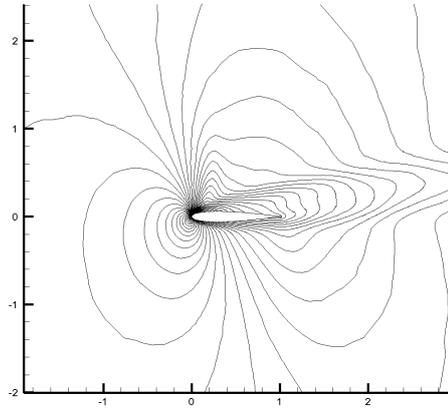
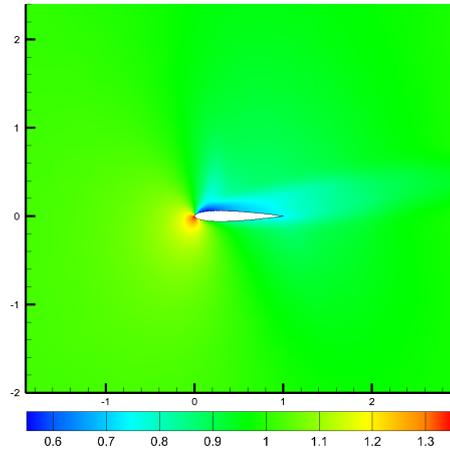
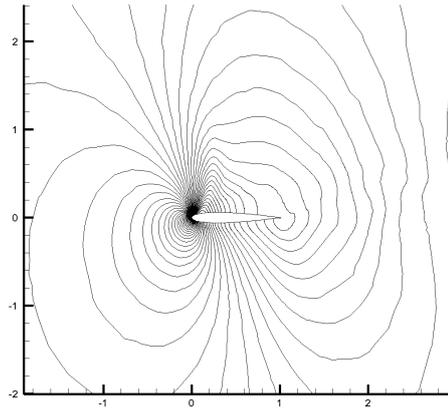
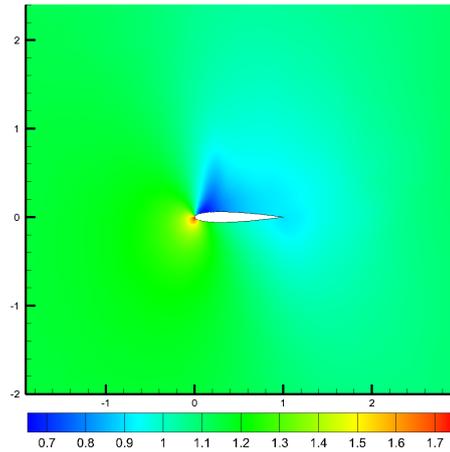
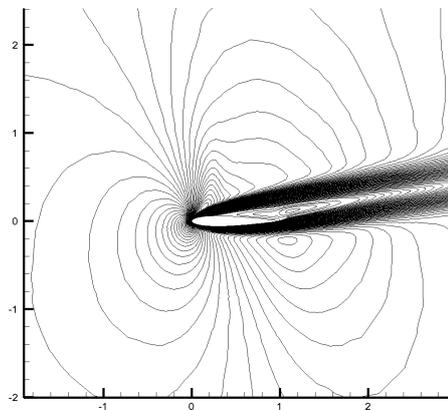
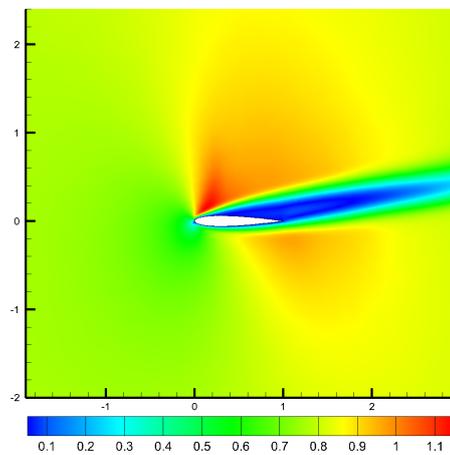
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.25: Solução final adaptada do exemplo 4.

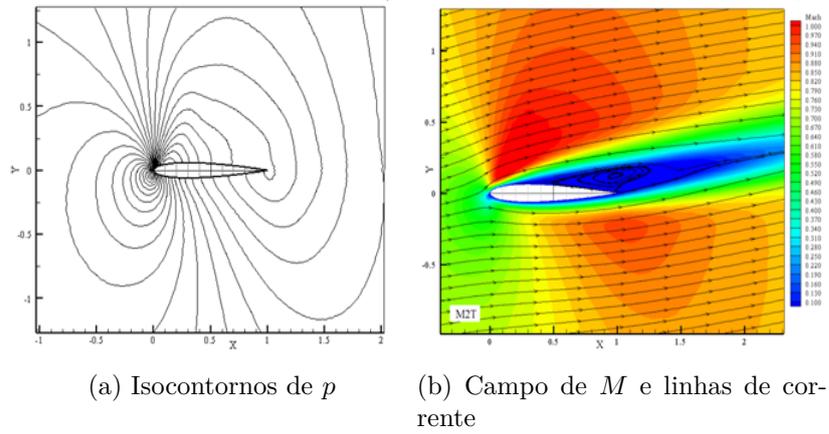


Figura 6.26: Solução obtida por Bono (2008).

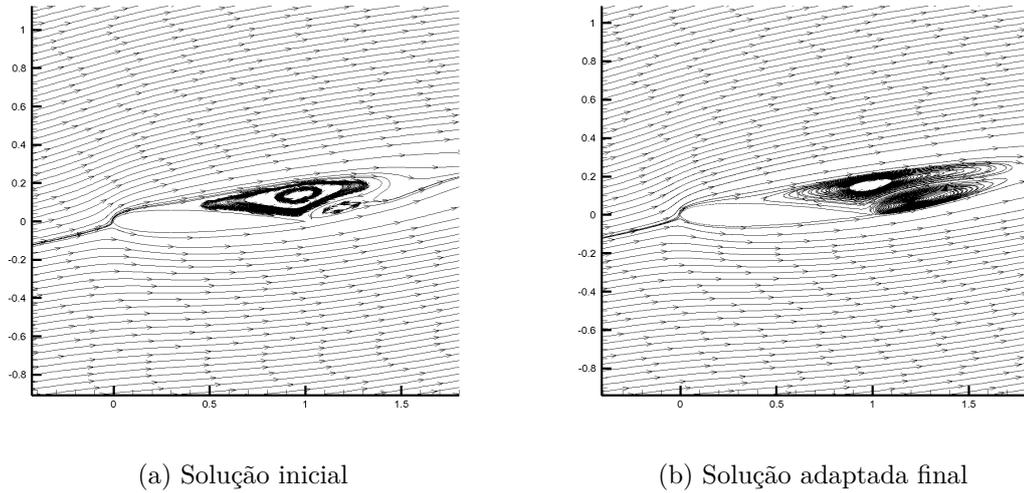


Figura 6.27: Linhas de corrente para a solução inicial e adaptada final do exemplo 4.

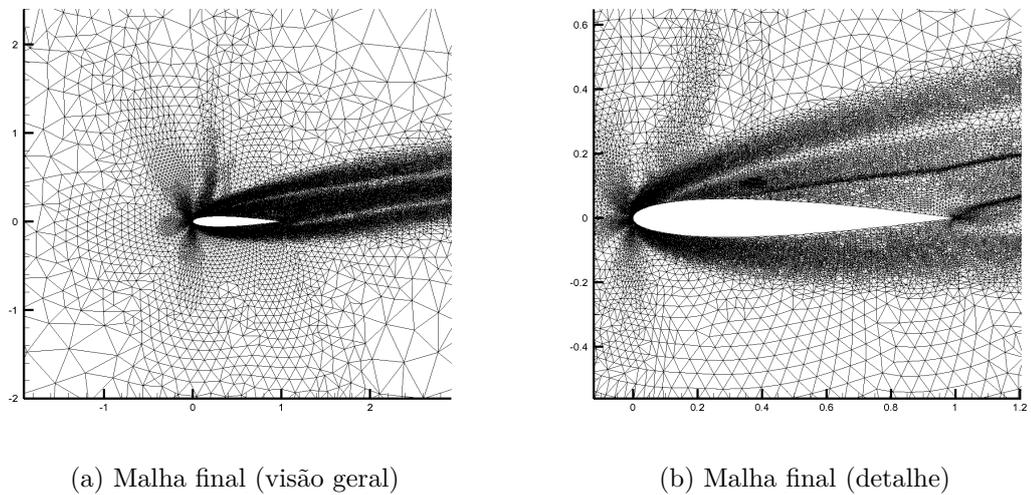


Figura 6.28: Malha adaptada final do exemplo 4.

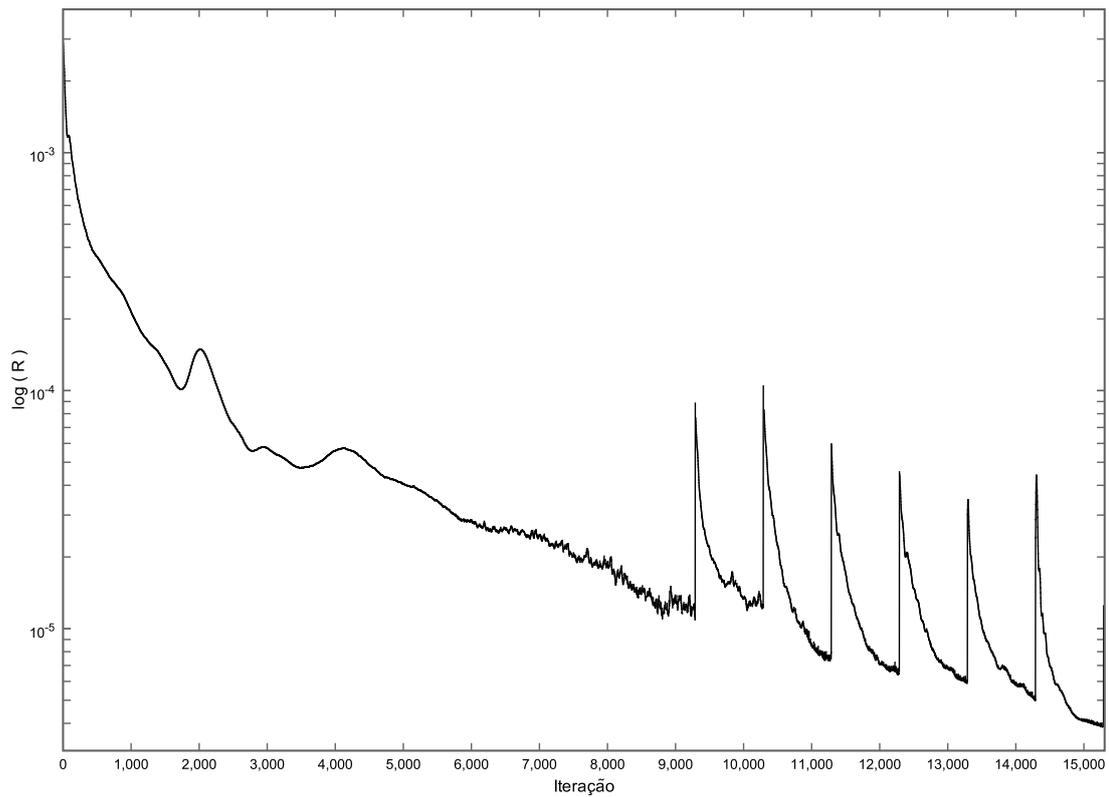
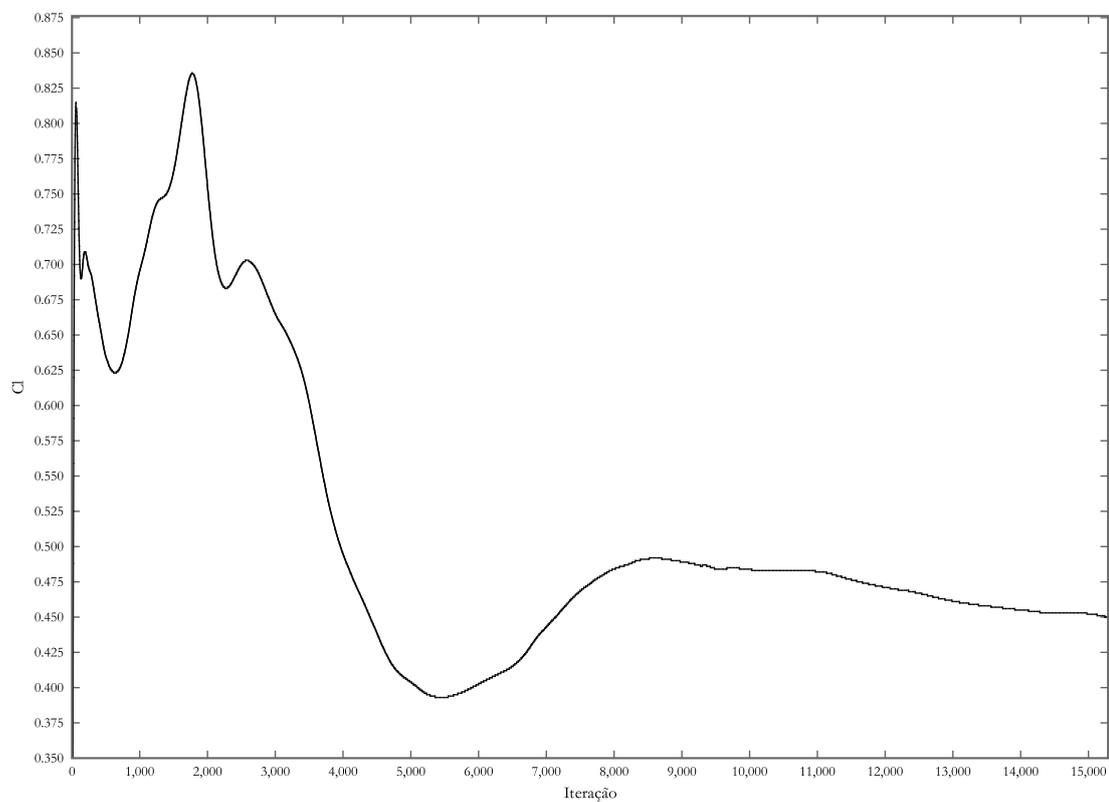


Figura 6.29: Convergência do exemplo 4.

Figura 6.30: Coeficiente de sustentação C_L ao longo das iterações do exemplo 4.

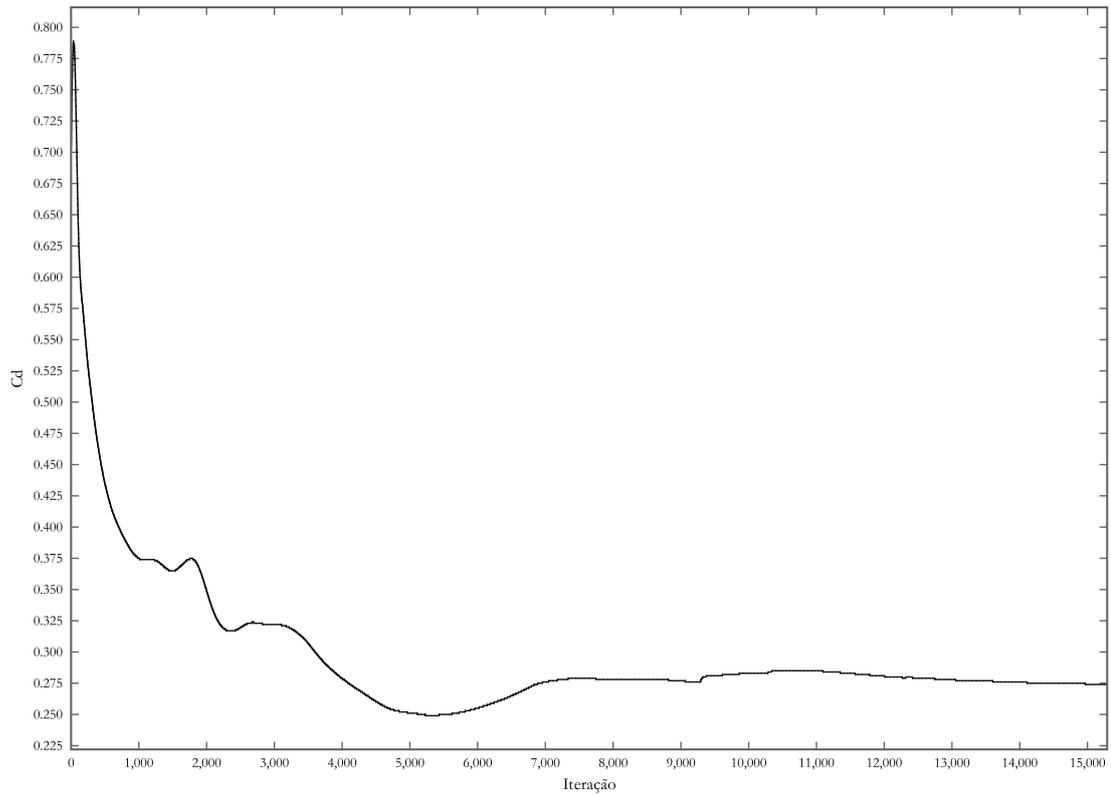


Figura 6.31: Coeficiente de arrasto C_D ao longo das iterações do exemplo 4.

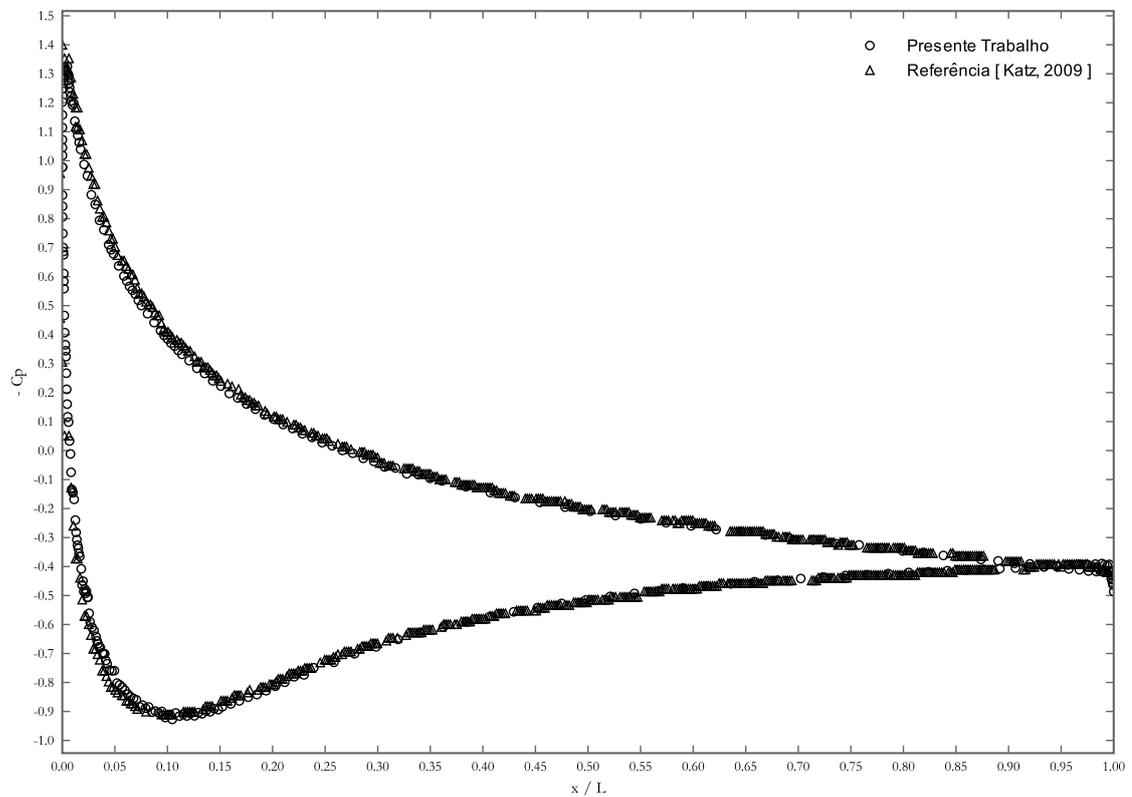


Figura 6.32: Coeficiente de pressão C_p para solução final adaptada do exemplo 4.

Tabela 6.6: Coeficientes aerodinâmicos do exemplo 4

Fonte	C_L	C_D
Presente Trabalho (solução inicial)	0.486	0.275
Presente Trabalho (solução adaptada final)	0.450	0.274
(Bono, 2008)	0.469	0.277

6.1.5 Exemplo 5: Escoamento Subsônico não-Viscoso Sobre Aerofólio NACA0012

O presente exemplo utiliza a mesma malha e condições de contorno utilizadas nos exemplos 3 e 4 anteriores (Fig. 6.16). Entretanto, o problema é agora subsônico e não-difusivo. Os dados empregados no escoamento são:

- $M_\infty = 0.5$
- $u_{1\infty} = 1.0$
- $u_{2\infty} = 0.0$
- $CFL = 0.1$
- $C_E = 0.0$
- $\alpha_S = 0.03$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 300K$

Os parâmetros de adaptação utilizados são:

- $\phi = p$
- $\eta_S = 1.4$
- $\eta_I = 0.4$
- $N_F = 2$
- $F_{\Delta t} = 1000$

Os resultados obtidos com a malha inicial são apresentados na Fig. 6.33 e os obtidos com a malha adaptada final na Fig. 6.34. Optou-se por adaptar apenas 2 vezes a malha pois, como observa-se, a malha inicial já é capaz de produzir uma resolução adequada da solução. Além disto, o escoamento é praticamente potencial, sem apresentar grandes efeitos anisotrópicos. Como indicado nos parâmetros do escoamento adotados, utilizou-se o procedimento de suavização de variáveis a baixas velocidades neste problema. Isto se deve ao fato do problema apresentar baixa compressibilidade e o algoritmo perder precisão e velocidade de convergência quanto mais baixa for a velocidade do escoamento nesta formulação (compressível com acoplamento térmico). Existem diversos procedimentos de pré-condicionamento das variáveis de conservação capazes de possibilitar formulações compressíveis para resolver problemas de baixa compressibilidade (Bono, 2008; Choi e Merkle, 1985). Entretanto, este pré-condicionamento funciona removendo energia térmica das equações de conservação de energia de forma artificial através, geralmente, da adição de termos extras que são derivadas temporais de ρ e ρu_i . O procedimento utilizado aqui, pelo contrário, mantém o acoplamento correto. A suavização redistribui a oscilação das variáveis de conservação no entorno dos nós da malha, removendo com isto a forte oscilação da convergência e possibilitando soluções mais suaves e corretas. Para demonstrar isso, a Fig. 6.36 apresenta a convergência do presente exemplo em conjunto com o mesmo problema simulado sem suavização (o problema foi encerrado antes de ser adaptado ou convergido, estando presente no gráfico apenas como comparação de convergência em função das iterações). A oscilação é alta e a taxa de convergência muito mais baixa no caso sem suavização, como pode ser observado. Convém mencionar que a presente solução não utiliza difusão artificial para estabilizar o algoritmo, demonstrando a estabilidade do método CBS em contraponto à métodos como o de Taylor-Galerkin. O método de Taylor-Galerkin, para este caso, gera oscilações espúrias no entorno da região de estagnação, devido à baixa compressibilidade, mesmo utilizando difusão artificial. A Fig. 6.37 mostra a variação da massa específica no entorno do ponto de estagnação para a malha inicial e final adaptada do presente trabalho conjuntamente com a solução obtida por Teixeira (2001), o qual utiliza a formulação de Taylor-Galerkin e cuja solução apresenta oscilações espúrias próximo ao ponto de estagnação. O coeficiente de pressão C_p é apresentado na Fig. 6.38 em comparação com o resultado obtido por Teixeira (2001), apresentando boa concordância. A Tab. 6.7 mostra um quadro comparativo das variáveis ρ e p no ponto de

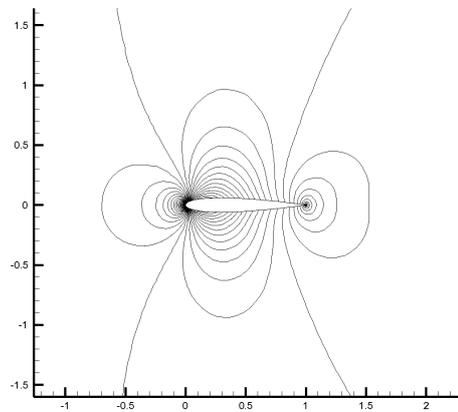
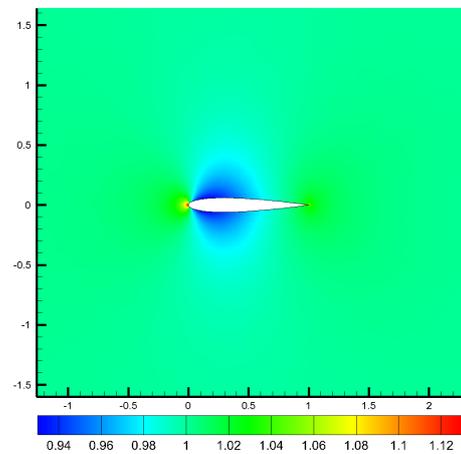
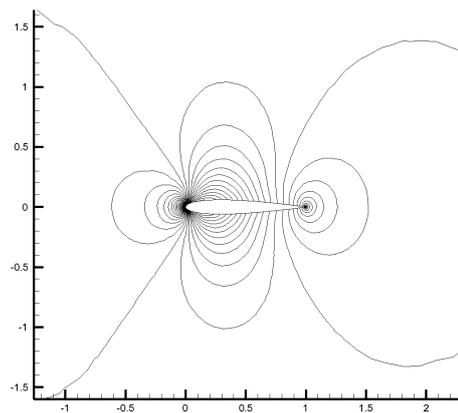
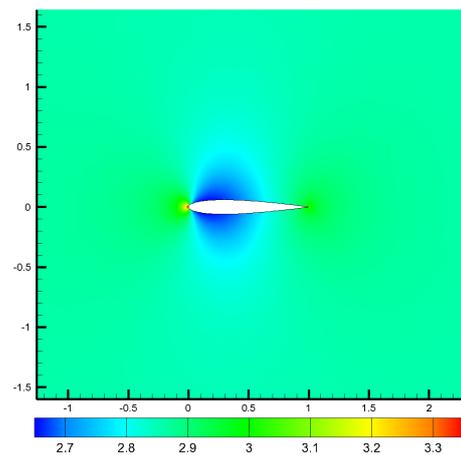
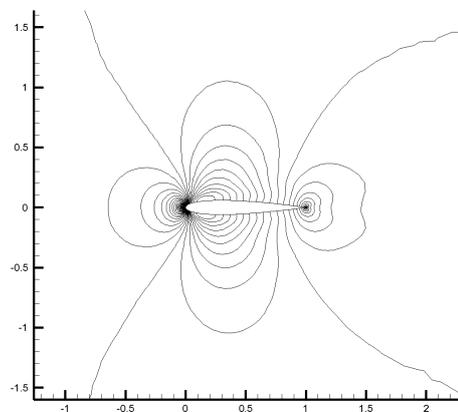
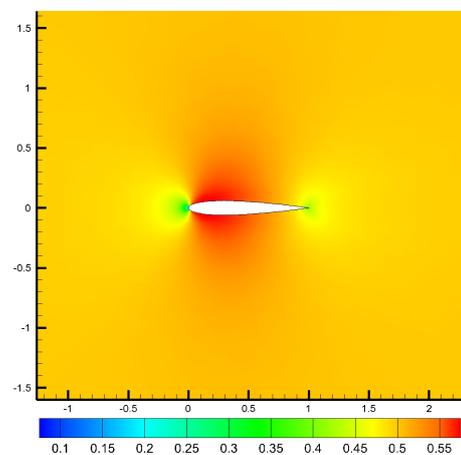
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.33: Solução inicial do exemplo 5.

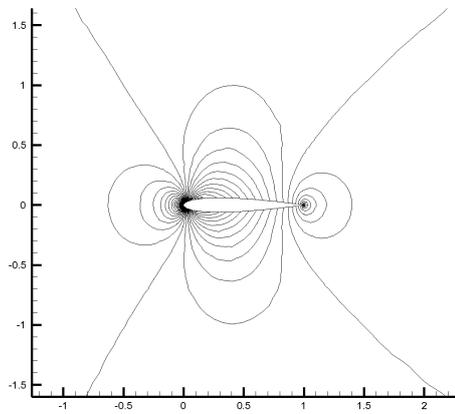
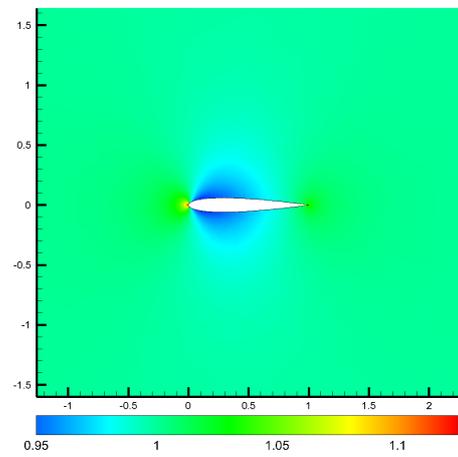
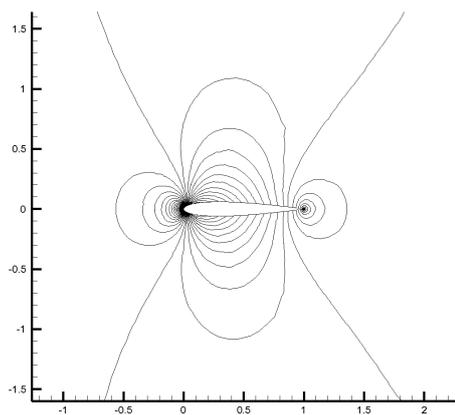
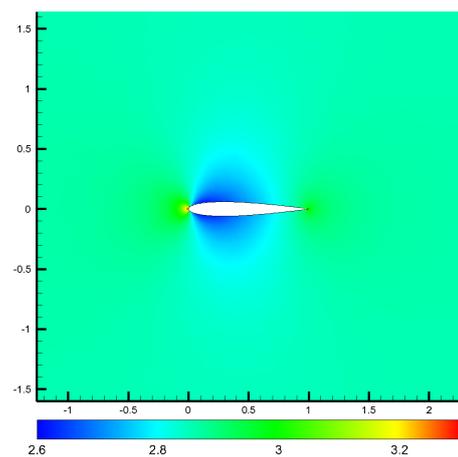
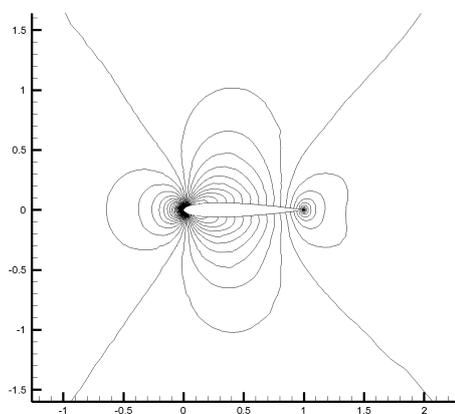
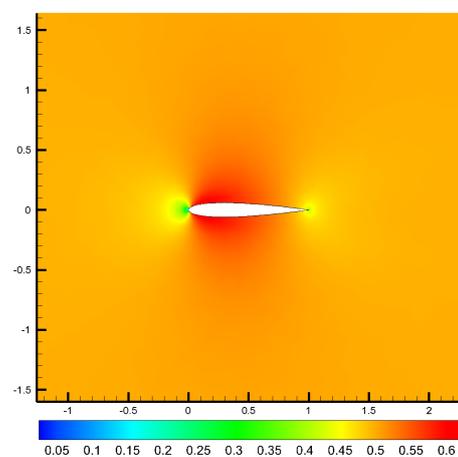
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.34: Solução final adaptada do exemplo 5.

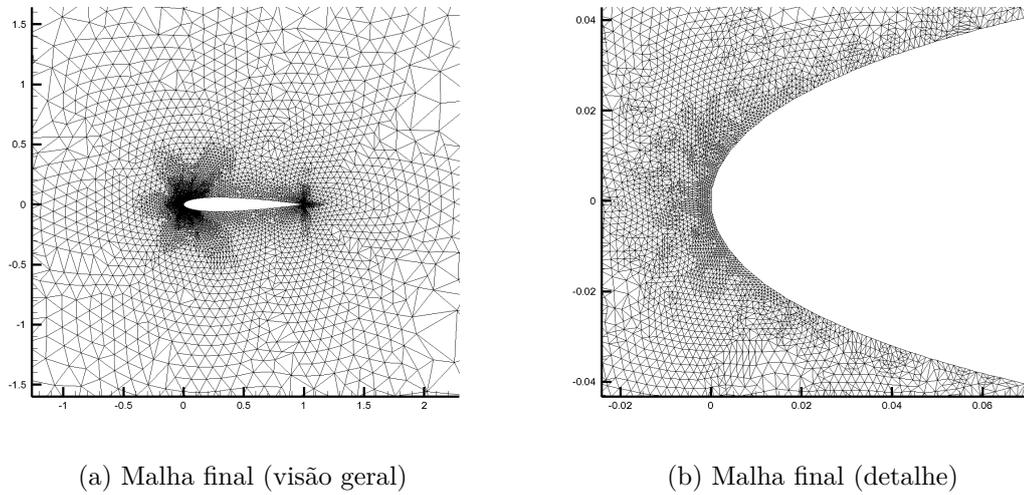


Figura 6.35: Malha adaptada final do exemplo 5.

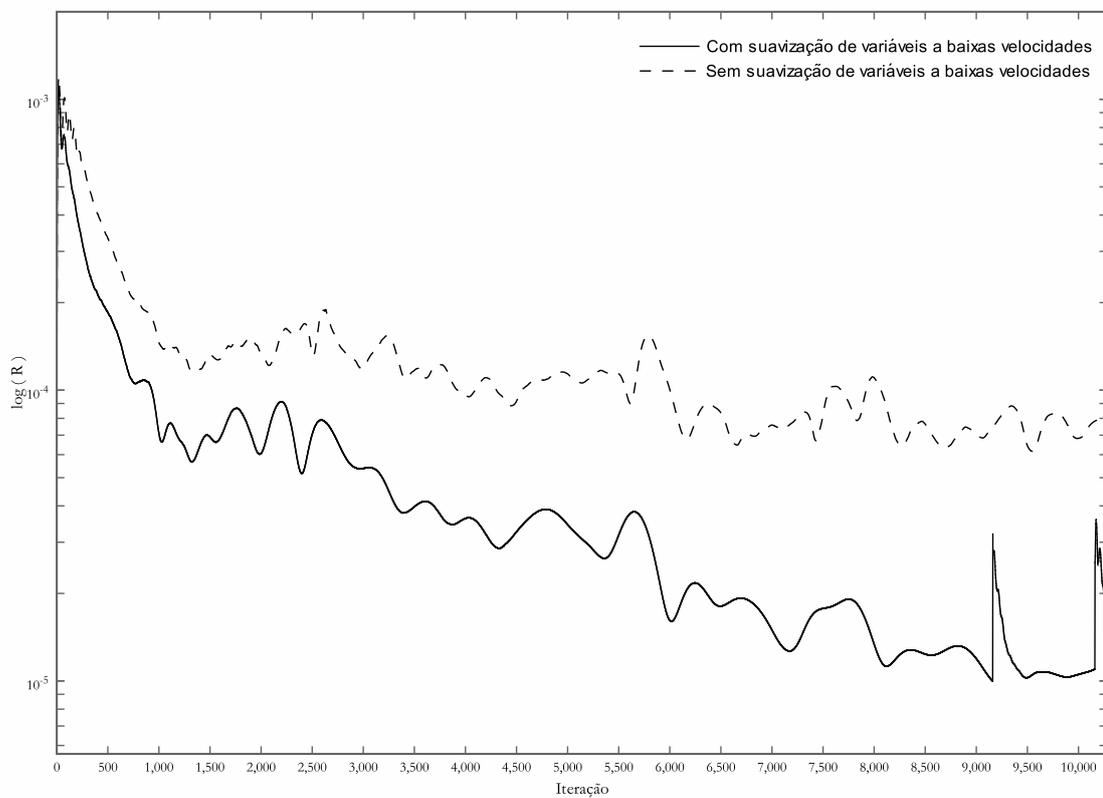


Figura 6.36: Convergência do exemplo 5.

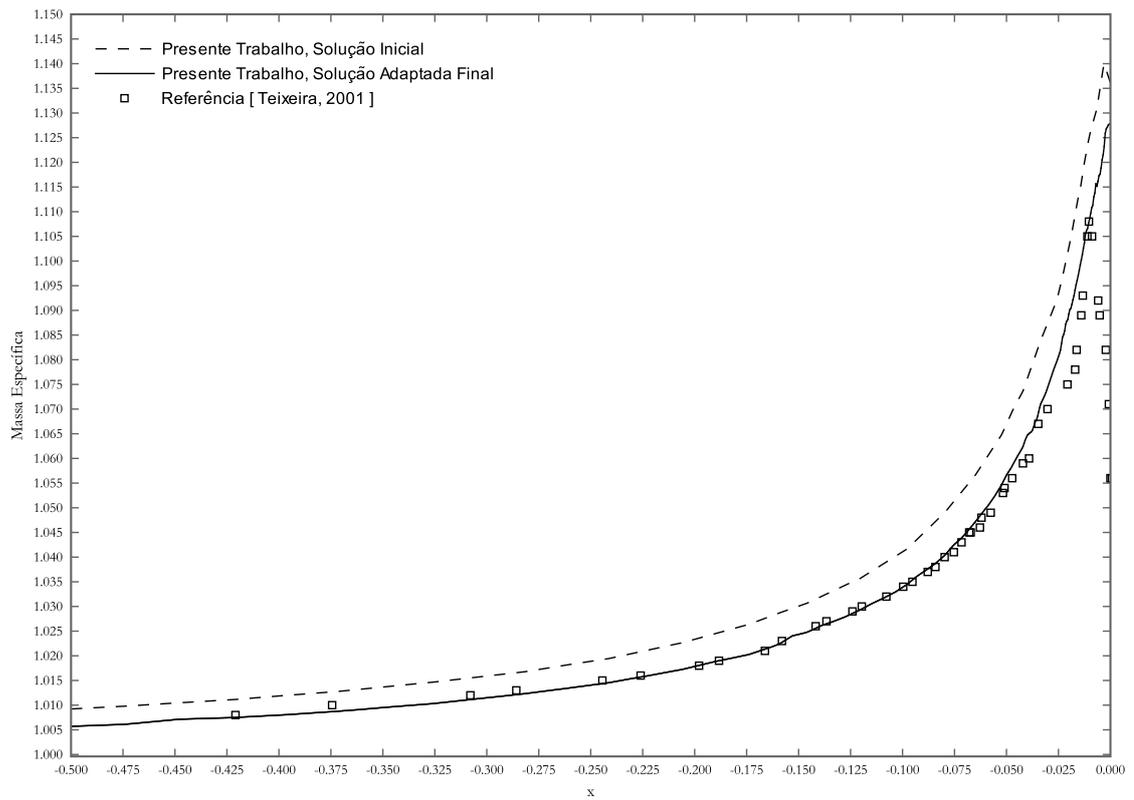


Figura 6.37: Comparação de ρ próxima à região de estagnação do exemplo 5.

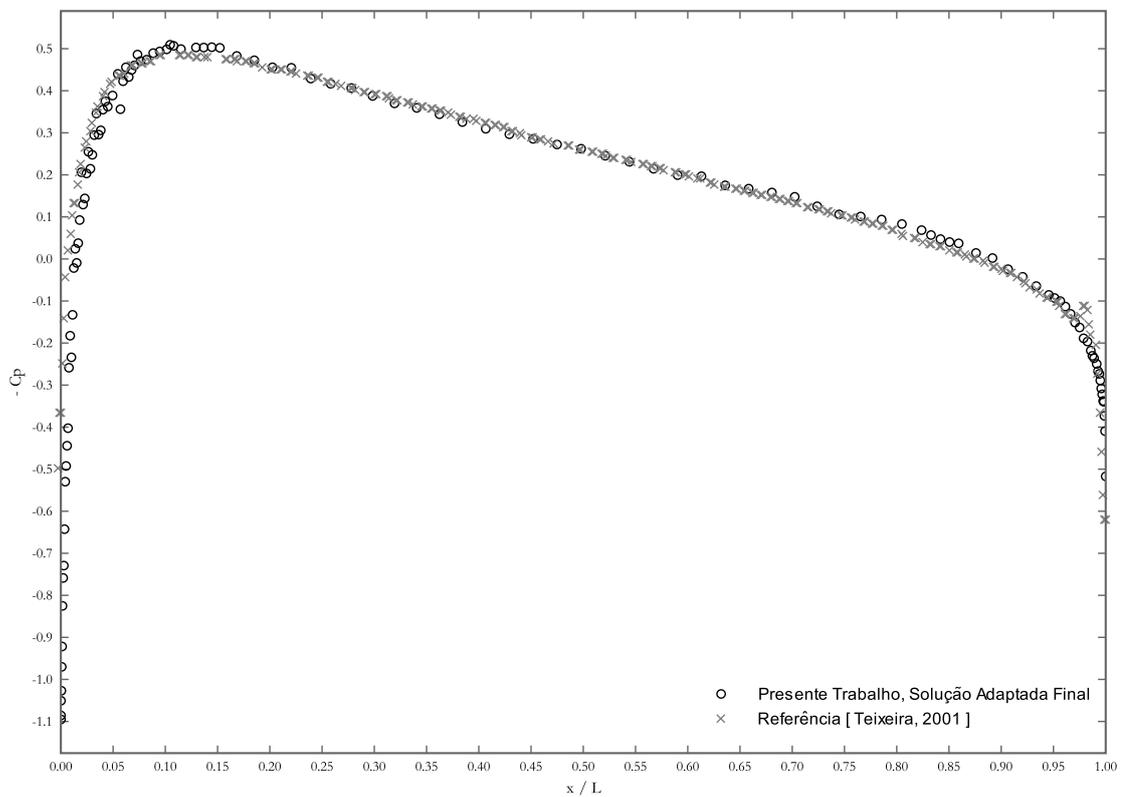


Figura 6.38: Coeficiente de pressão C_p para solução final adaptada do exemplo 5.

estagnação obtidos por diferentes autores e pela solução analítica teórica (obtida por teoria potencial). A Tab. 6.8 mostra os dados da malha ao longo das iterações de adaptação.

6.1.6 Exemplo 6: Escoamento Transiente Viscoso Sobre Aerofólio NACA0012

O presente exemplo utiliza a mesma malha inicial e condições de contorno utilizadas nos exemplos 3 a 5 anteriores (Fig. 6.16). O escoamento situa-se em um regime com número de Reynolds moderado, sobre ângulo de ataque nulo. Estuda-se o efeito transiente do escoamento sobre o aerofólio em regime transônico. Para uma ampla faixa de número de Reynolds (0.5 até 1×10^4), o escoamento é estacionário para um regime incompressível, entretanto, devido à efeitos de compressibilidade, o escoamento se torna instável até a transição à turbulência. Os mesmos comentários descritos no exemplo 4 são válidos neste caso, ou seja, há uma inconsistência física entre a velocidade e o número de Reynolds do escoamento, considerando o fluido como o ar. O escoamento real desenvolvido, portanto, seria diferente do simulado. Os dados empregados no escoamento são:

- $M_\infty = 0.85$
- $u_{1\infty} = 1.0$
- $u_{2\infty} = 0.0$
- $CFL = 0.4$
- $C_E = 0.25$
- $T_\infty = 300K$
- $Re = 5000$

Tabela 6.7: Variáveis no ponto de estagnação do exemplo 5

Fonte	ρ	p
Solução Analítica (solução potencial)	1.1297	3.3891
Presente Trabalho (solução inicial)	1.1250	3.3978
Presente Trabalho (solução adaptada final)	1.1297	3.3873
(Zienkiewicz et al., 1995) (CBS)	1.1268	3.2195
(Teixeira, 2001) (Taylor-Galerkin)	1.0541	3.1480

Tabela 6.8: Dados das iterações de adaptação do exemplo 5

Adaptação	Nós	Elementos	α_{max}	α_{160}	$\sigma^2(E)$
0	3753	7351	108.36	0.000E-0	6.683E-4
1	5468	10727	146.83	0.000E-0	2.541E-4
2	7776	15282	157.48	0.000E-0	1.599E-4

Os parâmetros de adaptação utilizados são:

- $\phi = M$
- $\eta_S = 1.8$
- $\eta_I = 0.2$
- $F_{\Delta t} = 500$
- $h_{min} = 8 \times 10^{-3}$

Sobre ângulo de ataque nulo e $M > 0.35$, o escoamento sobre o aerofólio NACA0012 apresenta uma instabilidade de von Kármán na região à jusante do bordo de fuga (Bourguet et al., 2007). A frequência, amplitude e tipo de esteira de von Kármán formada depende do número de Mach nestas condições. Para o caso estudado, onde $M = 0.85$, a instabilidade é induzida pelo alargamento da camada limite a jusante das regiões supersônicas, como indicam as Fig. 6.39 à Fig. 6.41-(b). A instabilidade propaga-se no sentido contrário ao escoamento, dirigindo-se ao bordo de fuga à medida que a esteira aumenta, como mostram as Fig. 6.39 à Fig. 6.41-(c e d). Quando a perturbação atinge as proximidades do bordo de fuga, como indicam as Fig. 6.39 à Fig. 6.41-(e), os vórtices desprendidos aumentam de amplitude até atingir um estado oscilatório periódico, como mostram as Fig. 6.39 à Fig. 6.41-(f). É interessante observar que no comportamento oscilatório final os vórtices realizam curvas delimitando uma região supersônica de alta energia e baixa massa específica, seguido de um comportamento inverso até realizar a próxima curva, onde novamente tem-se a região supersônica no contorno e assim por diante, desenvolvendo a oscilação com este comportamento. Na Fig. 6.42 e Fig. 6.43 pode-se visualizar que a densidade da malha acompanha diretamente a estrutura dos vórtices formados a cada instante de tempo, demonstrando a aplicabilidade da metodologia de adaptação anisotrópica de malhas implementada para simular problemas transientes.

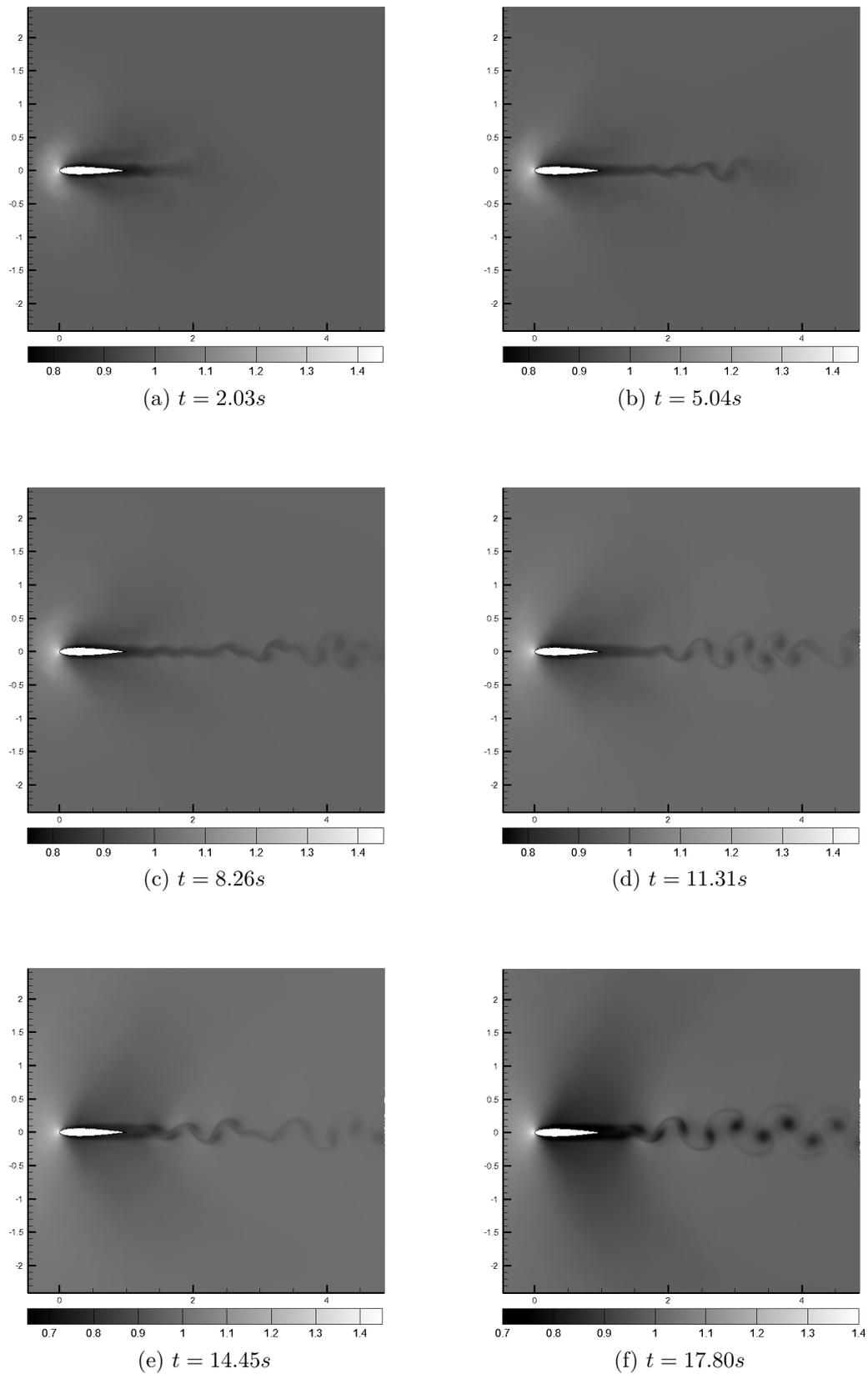


Figura 6.39: Campo de ρ para diferentes instantes de tempo do exemplo 6.

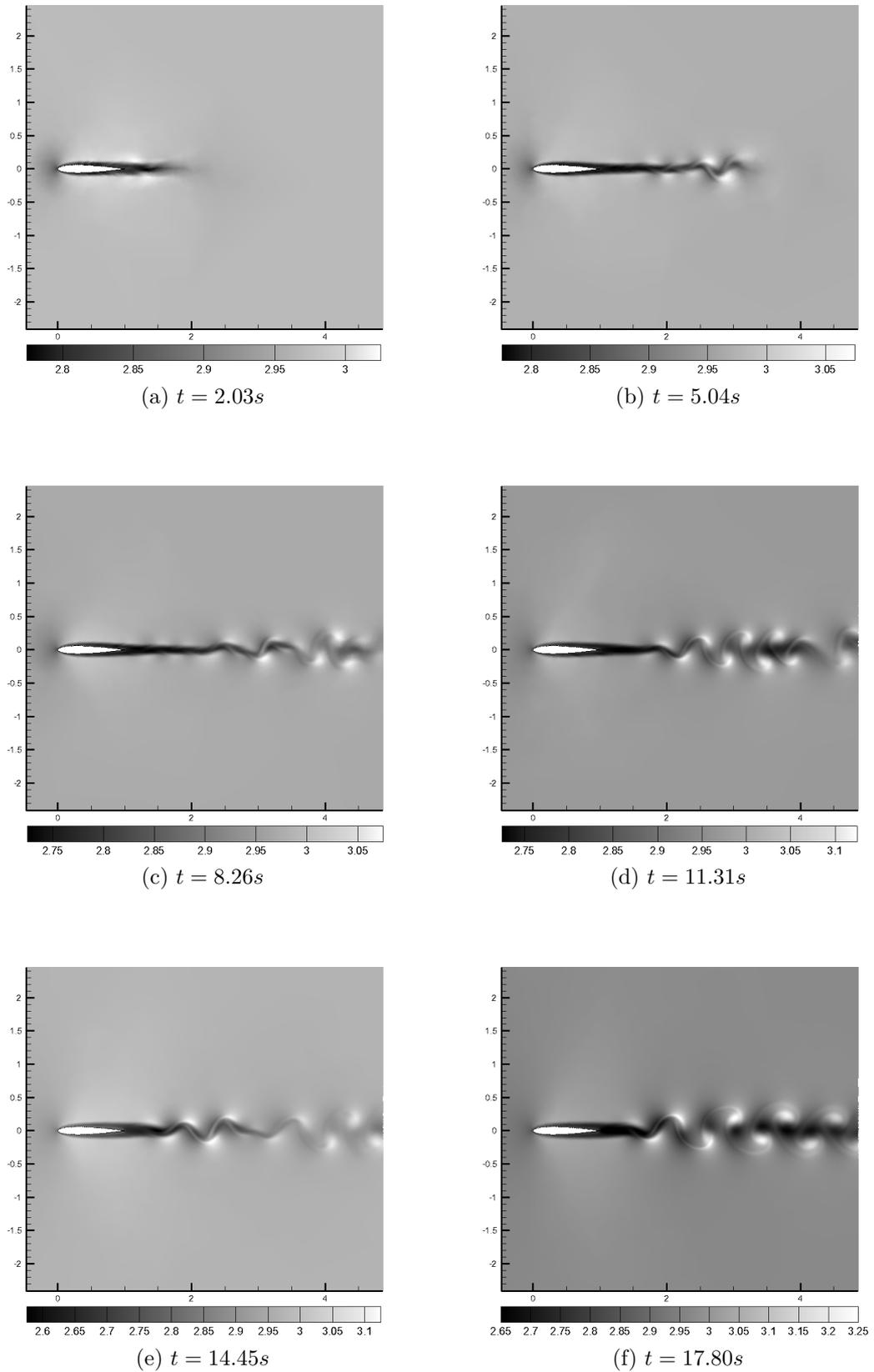


Figura 6.40: Campo de e para diferentes instantes de tempo do exemplo 6.

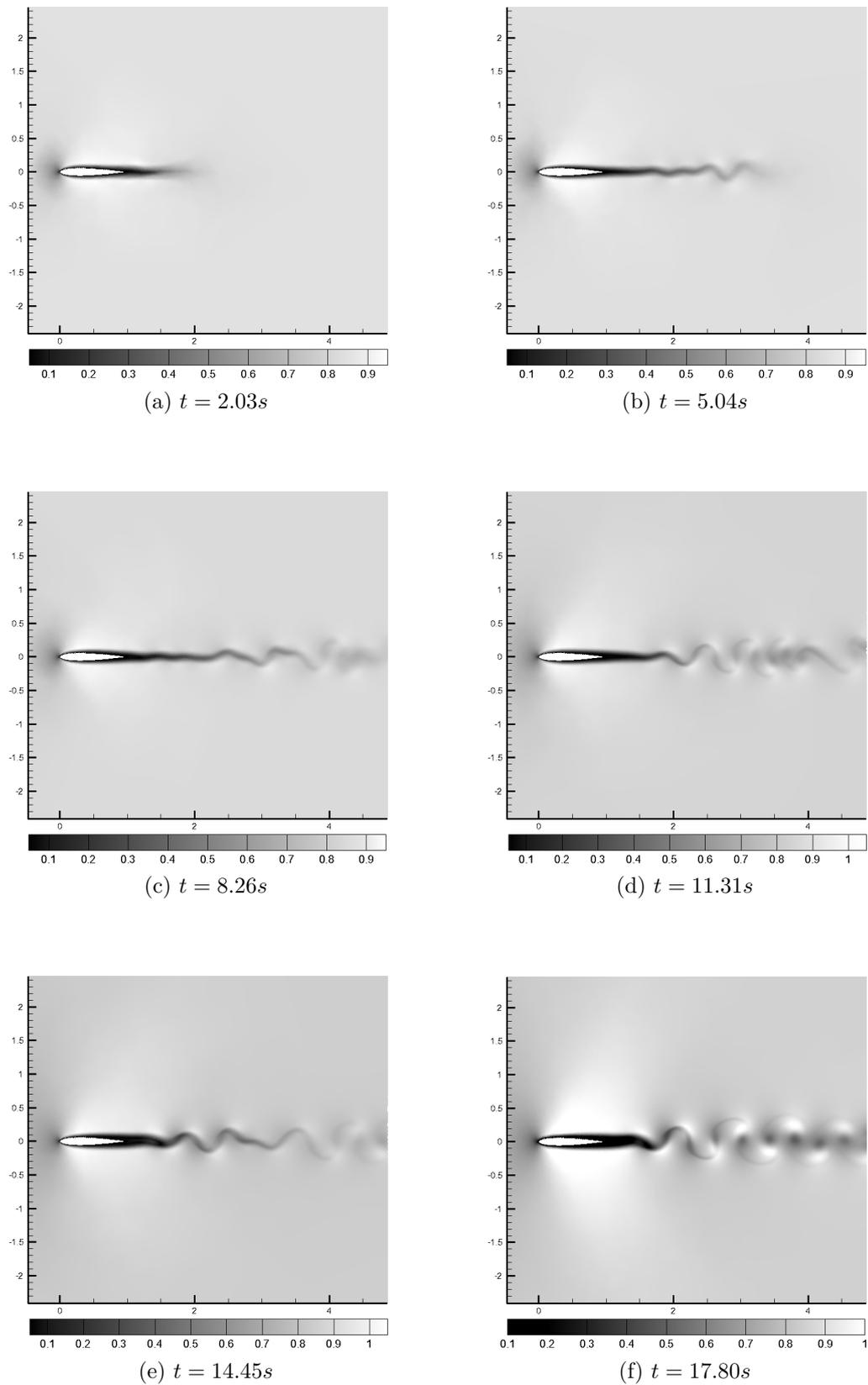


Figura 6.41: Campo de M para diferentes instantes de tempo do exemplo 6.

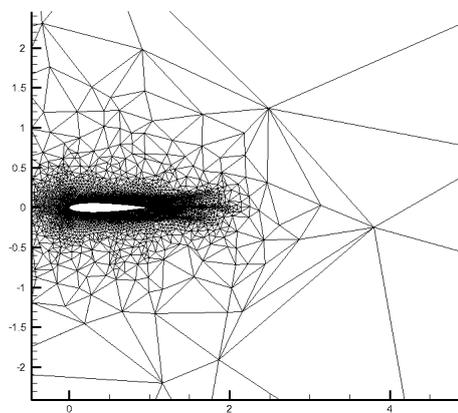
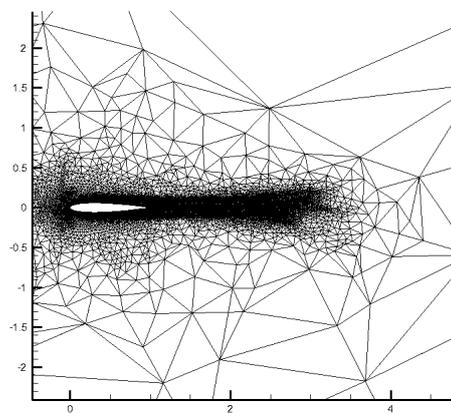
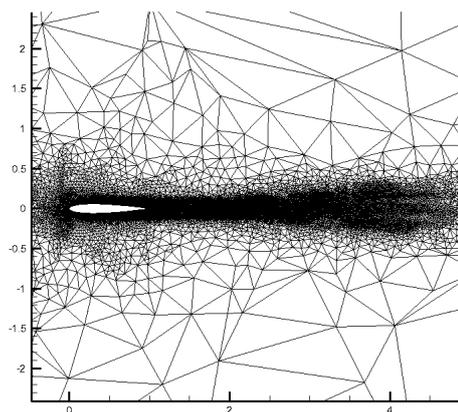
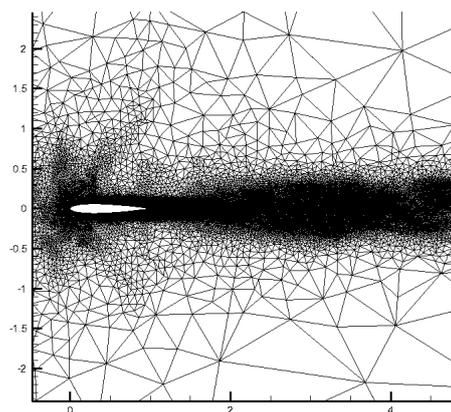
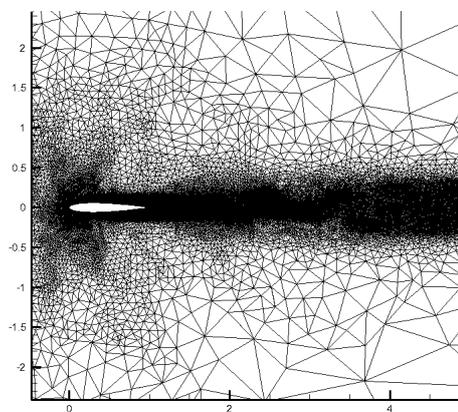
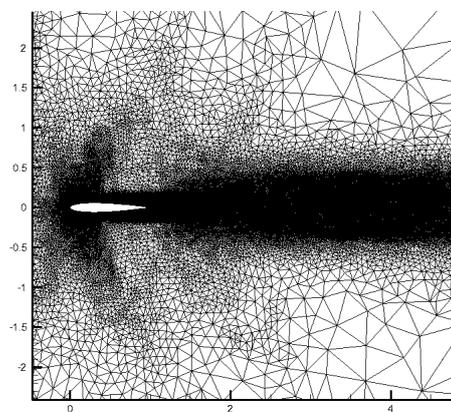
(a) $t = 2.03s$ (b) $t = 5.04s$ (c) $t = 8.26s$ (d) $t = 11.31s$ (e) $t = 14.45s$ (f) $t = 17.80s$

Figura 6.42: Malha para diferentes instantes de tempo do exemplo 6.

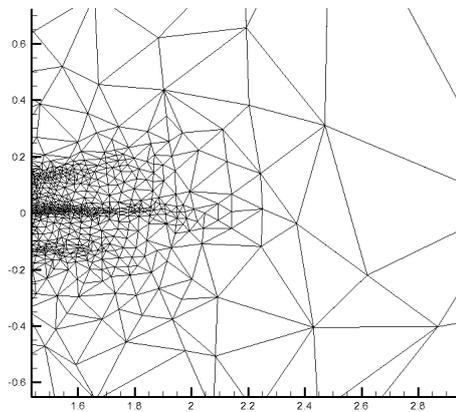
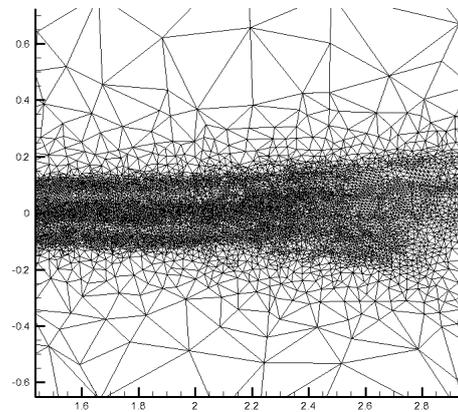
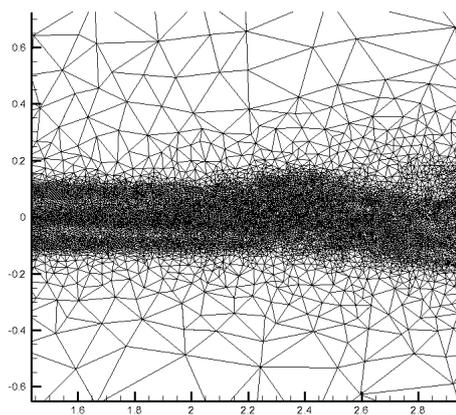
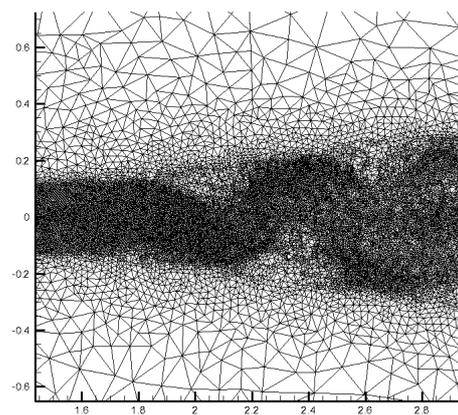
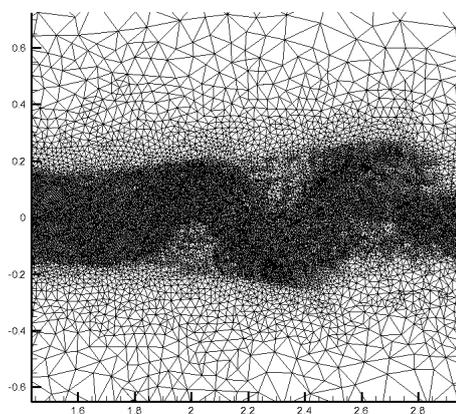
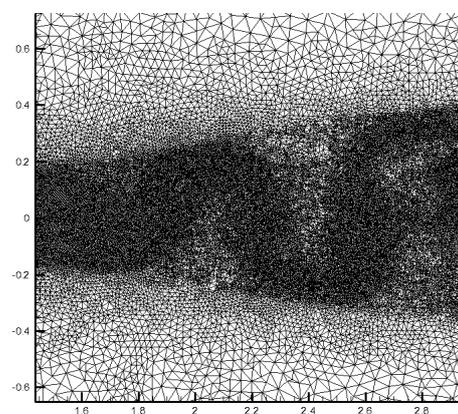
(a) $t = 2.03s$ (b) $t = 5.04s$ (c) $t = 8.26s$ (d) $t = 11.31s$ (e) $t = 14.45s$ (f) $t = 17.80s$

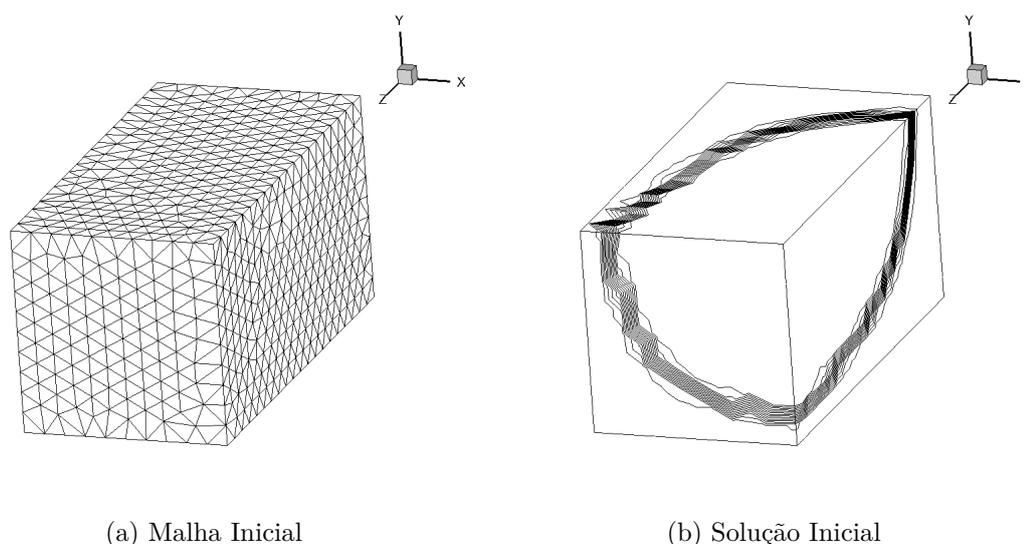
Figura 6.43: Detalhe da malha para diferentes instantes de tempo do exemplo 6.

6.2 Adaptação de Malhas Tridimensionais

Nesta seção são analisadas aplicações da adaptação tridimensional para um caso analítico, escoamentos no entorno de uma esfera, de uma asa, da fuselagem de um avião completo e de um míssil modelo AGM-114 Hellfire. Os dados de performance computacional apresentados são referentes ao cluster Gauss do CESUP/UFRGS, com processadores SGI Altix (AMD Opteron 2.3GHz, 128KB de cache L1 por núcleo, 512KB de cache L2 por núcleo e 12MB de cache L3 por soquete, memória DDR3 1333MHz com 64GB de RAM por unidade).

6.2.1 Exemplo 7: Solução Analítica de Função Trigonométrica

De forma análoga ao estudo previamente apresentado para o caso bidimensional, apresenta-se um caso de estudo onde uma solução analítica é imposta a uma malha inicial. O efeito da escolha do valor do limiar superior e inferior, bem como da alternância de faces é análogo ao problema bidimensional previamente estudado em detalhes, portanto aqui não será investigada a influência de tais parâmetros independentemente. Contudo, inclui-se no presente exemplo a análise dos custos computacionais envolvidos. O problema analisado constitui-se de um domínio $\Omega : \mathbf{x} \in [0,1] \times [0,1] \times [0,2]$ onde impõe-se um campo analítico $u = \arctan[1000x^4y^4z^4 - \frac{1}{256}]$. A malha inicial é composta por 3363 nós e 16420 elementos. A malha inicial bem como a solução inicial são mostradas na Fig. 6.44. A variável utilizada para análise de erro ϕ foi a própria função analítica imposta, ou seja, $\phi = u$. A partir da solução inicial, realizou-se $N_F = 6$ ciclos de adaptação utilizando-se um limiar superior $\eta_S = 1.9$ e um limiar inferior $\eta_I = 0.7$. A Fig. 6.45 mostra as malhas e os campos de soluções ao longo do avanço da adaptação. A Tab. 6.9 indica o número da adaptação com a correspondente quantidade nós, elementos, o ângulo máximo diedral entre todos tetraedros (α_{Dmax}), a percentagem de ângulos diedrais os quais ultrapassam o limite de 160° (α_{D160}), o volume do tetraedro mínimo entre todos os elementos $\|T_K\|_{min}$, a variância do erro das arestas da malha $\sigma^2(E)$ e o custo total do procedimento de adaptação em segundos. O volume do elemento é usado aqui como indicador indireto do ângulo sólido do elemento, uma vez que ângulos sólidos que violem o limite superior de 180° impliquem em um elemento com volume negativo e, por corolário, em uma falha



(a) Malha Inicial

(b) Solução Inicial

Figura 6.44: Malha e solução inicial do exemplo 7.

topológica por degeneração. Cabe também salientar que o procedimento de adaptação não é paralelizado.

Assim como para o caso bidimensional, observa-se uma diminuição de $\sigma^2(E)$ em todas iterações, indicando uma melhora progressiva de equidistribuição de erro sobre as arestas da malha. O percentual α_{D160} , comparando-se à casos bidimensionais, indica que as malhas tridimensionais apresentam maior quantidade de elementos altamente distorcidos, mas apresenta ângulos diedrais extremos α_{Dmax} tão elevados quantos os ângulos extremos bidimensionais. O tempo computacional gasto para a adaptação CPU mostra que há uma proporção aproximadamente linear do número de nós (ou de elementos) na malha com o custo envolvido. Isto indica que o algoritmo está fortemente otimizado seqüencialmente. Caso contrário, isto é, se houvesse dependência altamente não-linear com o número de nós (ou de elementos) da malha, este seria um forte indicador de que grande parte do esforço computacional esteja sendo desperdiçado por ineficiência de programação, sendo cada vez mais visível à medida que o tamanho da malha aumenta.

Pode-se visualizar na Fig. 6.46 a suavidade da função para uma dada isosuperfície constante obtida com a solução final em detrimento àquela obtida com a malha inicial. A Fig. 6.47 mostra em detalhe uma porção da malha adaptada final, destacando-se a propriedade anisotrópica da malha adaptada e a Fig. 6.48 mostra uma porção interna da malha, segmentada para visualização do mesmo efeito no interior do domínio.

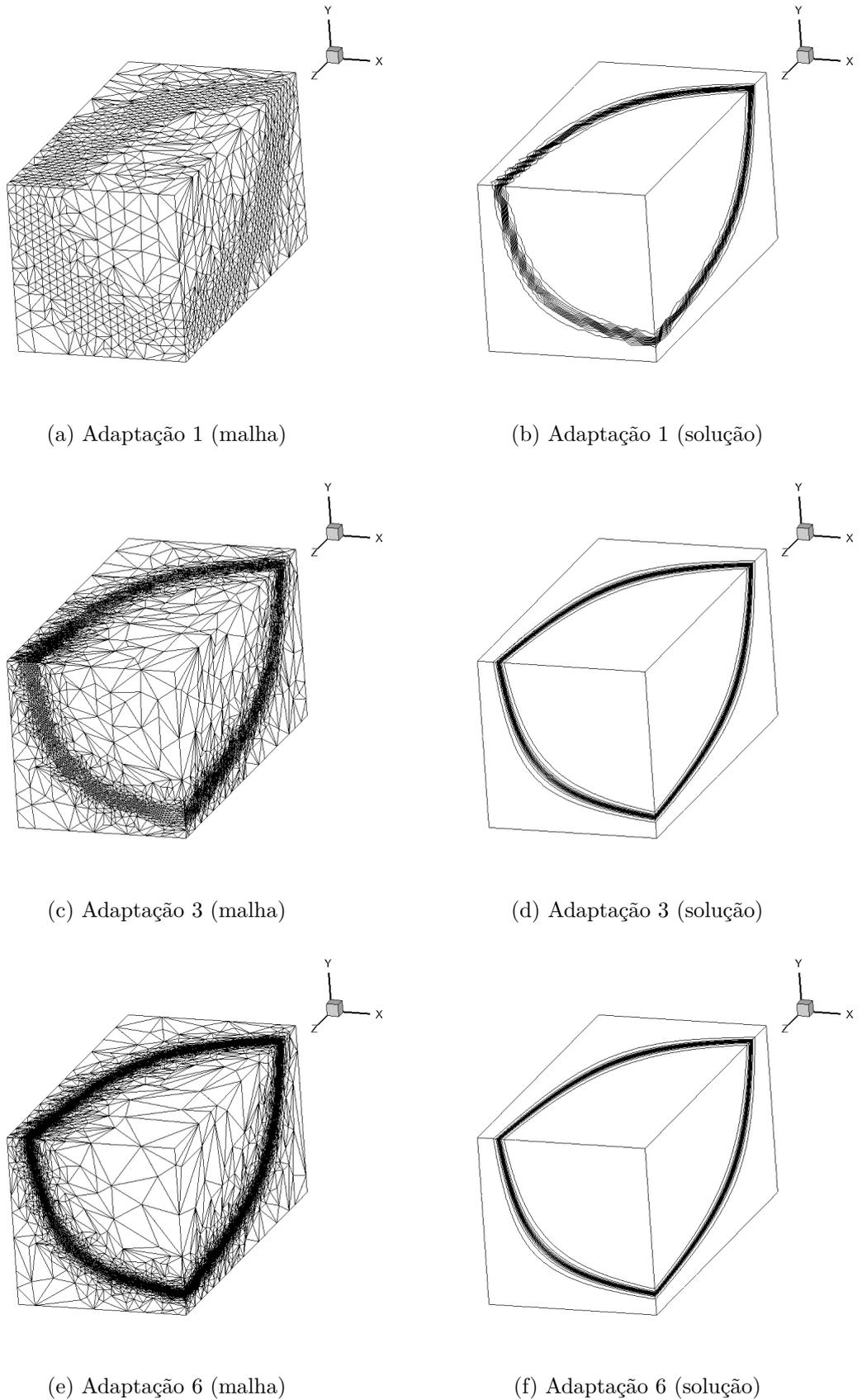


Figura 6.45: Malha e isocontornos da solução ao longo das iterações de adaptação do exemplo 7.

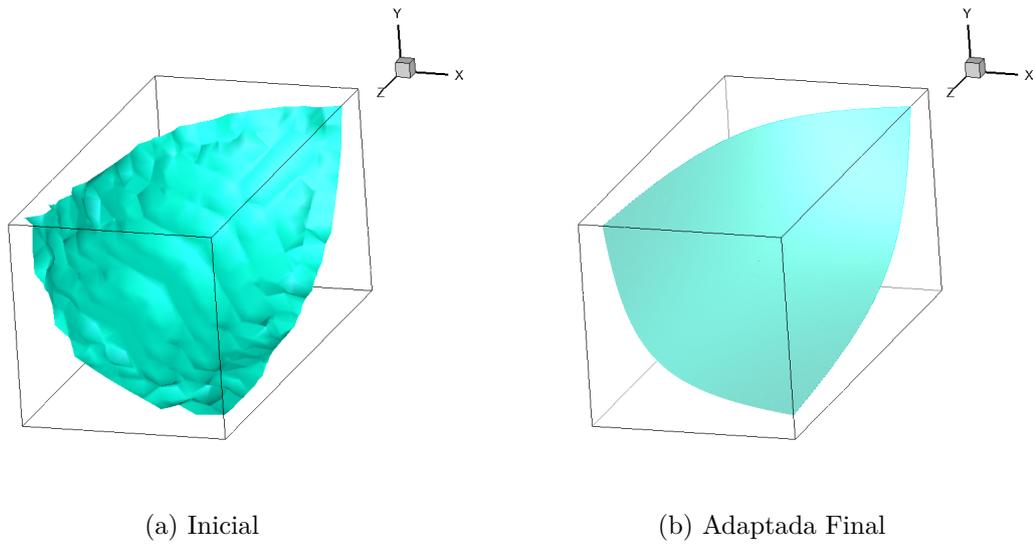


Figura 6.46: Isosuperfícies de $u = -0.597$ para a solução inicial e final do exemplo 7.

Tabela 6.9: Dados das iterações de adaptação do exemplo 7

Adaptação	Nós	Elementos	α_{Dmax}	α_{D160}	$\ T_K\ _{min}$	$\sigma^2(E)$	CPU
0	3363	16420	162.34	1.015E-4	2.734E-4	7.530E-2	-
1	8941	48688	162.34	3.080E-5	1.979E-6	5.867E-2	0.75
2	30430	174370	173.14	3.440E-4	2.771E-7	3.807E-2	2.36
3	101402	590103	173.51	8.233E-4	3.559E-8	2.228E-2	8.13
4	255583	1487259	174.63	1.779E-3	4.448E-9	1.360E-2	21.00
5	540737	3130761	176.65	3.525E-3	7.187E-10	8.308E-3	45.86
6	936498	5373351	178.02	5.915E-3	1.086E-10	5.392E-3	83.61

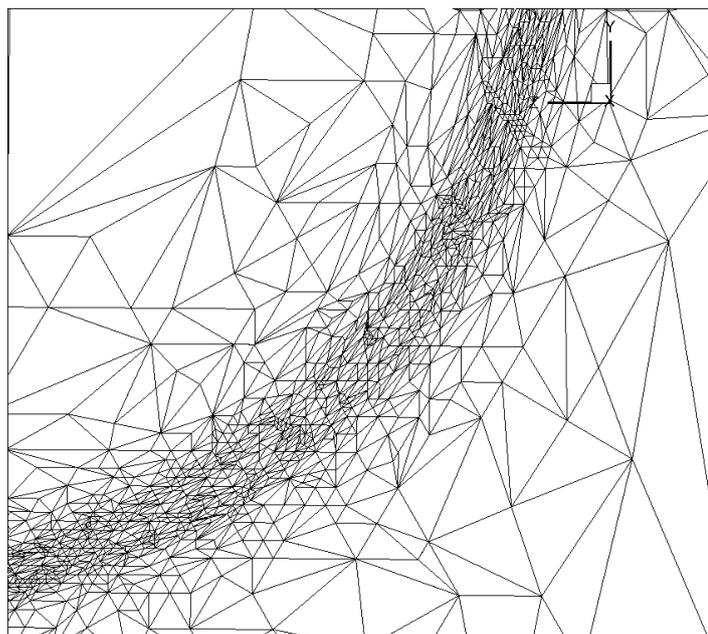


Figura 6.47: Detalhe aumentado de uma porção da malha adaptada final do exemplo 7.

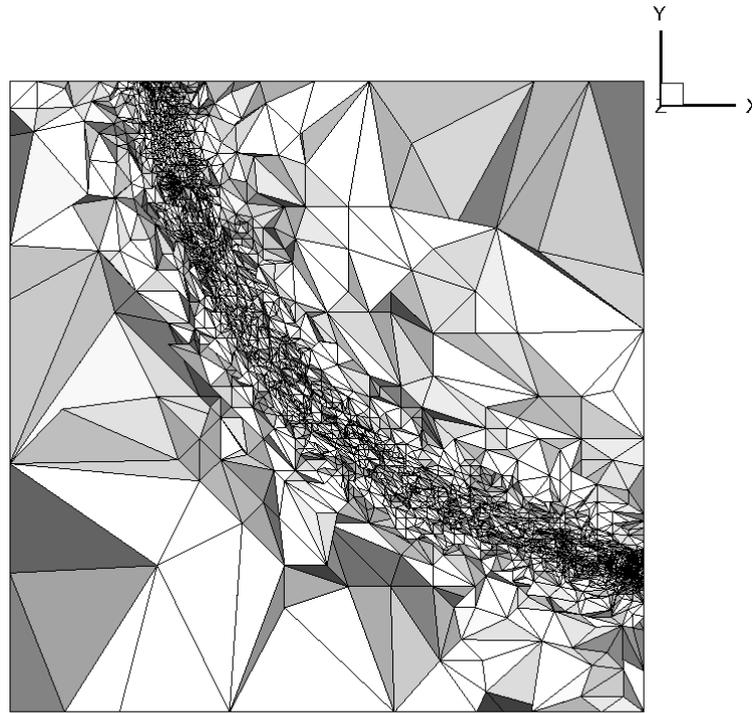


Figura 6.48: Detalhe em corte (plano em $z = 1$) de uma porção da malha adaptada final do exemplo 7.

6.2.2 Exemplo 8: Análise de Desempenho Computacional

O objetivo deste exemplo é de analisar o desempenho do código CBS tridimensional implementado quanto à eficiência de paralelização. Estuda-se 3 malhas distintas, para um problema genérico com domínio $\Omega : \mathbf{x} \in [0,1] \times [0,1] \times [0,3]$ (simulando um canal com escoamento interno). Os dados das malhas são indicados na Tab. 6.10. As malhas utilizadas são apresentadas na Fig. 6.49. O problema é analisado com diferentes número de processadores (*threads*) para cada uma das malhas, obtendo o tempo requerido para executar um único incremento de tempo em cada uma das configurações. O valor do tempo t em segundos (para cada incremento de tempo) apresentado é obtido com uma média simples calculada com 100 incrementos realizados (não incluem-se nos tempos computacionais os tempos requeridos para leitura, preparação ou escrita de dados). Os resultados são mostrados na Tab. 6.11. Com base nestes resultados, observa-se que tanto o *Speedup* S_p quanto a eficiência de paralelização E_p aumentam com o tamanho do problema. Entretanto, a cada duplicação de número de *threads* reduz-se cerca de 20% de eficiência da paralelização. Parte destas perdas é inerente ao fato de porções do código as quais

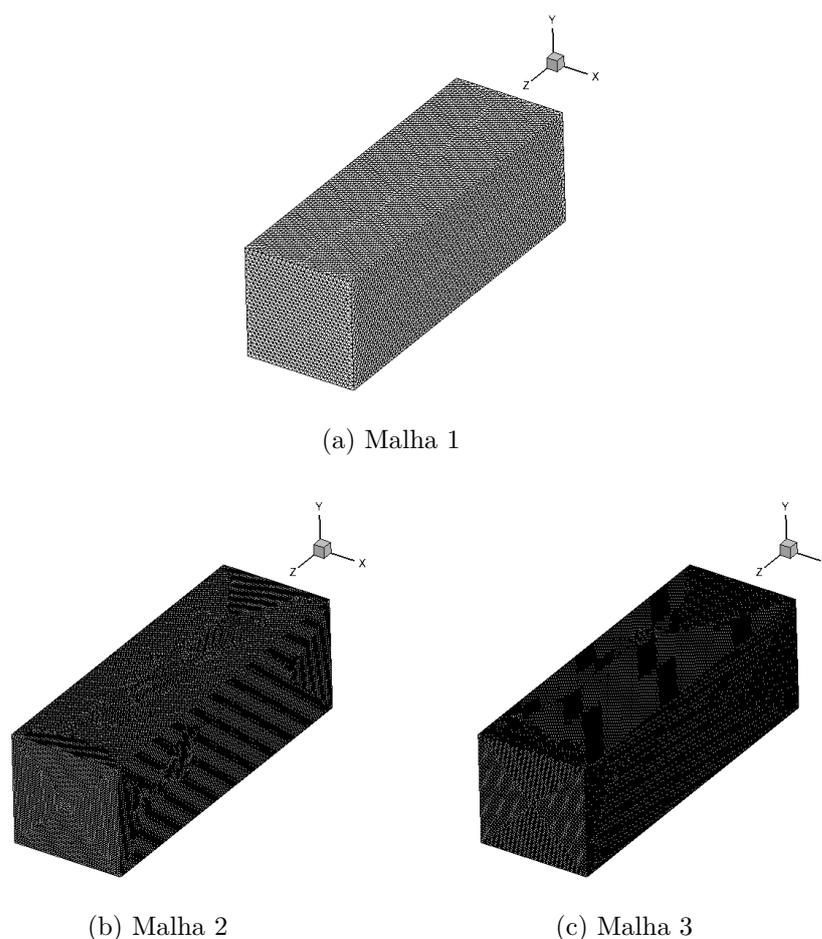


Figura 6.49: Malhas utilizadas no exemplo 8.

não podem ser paralelizadas e outras devido à sincronizações e perdas por distribuição de tarefas para *threads*. É importante ressaltar também que o número de *threads* indicado refere-se ao número máximo disponível utilizado. Assim, cada *loop* do código seleciona automaticamente quantos destes são requeridos de acordo com a carga computacional envolvida. Devido à heterogeneidade do número de variáveis (nós, elementos, faces, contornos sólidos, etc.) resulta menos eficiente a utilização de um número muito grande de *threads* para um carga de trabalho muito pequena, ou que requeira muitas sincronizações. A Tab. 6.12 mostra este efeito para a malha 1 quando não é escalonado o número de *threads* de acordo com carga de trabalho ao longo do código, ou seja, quando utiliza-se o número máximo de *threads* disponíveis em todos os *loops* do código, em contraponto à utilização de um número adequado individual em cada *loop* de acordo com a carga de trabalho requerida pelo mesmo). Embora ocorram pequenos ganhos para um número pequeno de *threads*, para valores maiores ocorrem grandes disparidades e reduções de eficiência da paralelização.

Tabela 6.10: Dados das malhas utilizadas no exemplo 8

Malha	Nós	Elementos
1	98230	547400
2	657575	3765176
3	1067259	6140577

Tabela 6.11: Performance computacional do exemplo 8

Malha	<i>Threads</i>	t	S_p	E_p
1	1	2.64	1.00	1.00
1	2	1.62	1.63	0.81
1	4	1.05	2.51	0.62
1	8	0.64	4.09	0.51
1	12	0.54	4.88	0.40
1	18	0.43	6.15	0.34
1	24	0.35	7.53	0.31
2	1	20.14	1.00	1.00
2	2	12.93	1.55	0.78
2	4	7.66	2.62	0.65
2	8	5.13	3.92	0.49
2	12	4.18	4.82	0.40
2	18	2.91	6.91	0.38
2	24	2.34	8.60	0.35
3	1	36.16	1.00	1.00
3	2	23.14	1.56	0.78
3	4	14.21	2.54	0.63
3	8	9.20	3.92	0.49
3	12	7.45	4.85	0.40
3	18	5.10	7.08	0.39
3	24	4.01	9.01	0.37

Tabela 6.12: Performance computacional sem escalonamento de tarefas do exemplo 8

Malha	<i>Threads</i>	t	S_p	E_p
1	1	2.64	1.00	1.00
1	2	1.60	1.65	0.82
1	4	1.01	2.62	0.65
1	8	0.65	4.07	0.50
1	12	0.53	4.98	0.41
1	18	0.51	5.21	0.28
1	24	0.47	5.58	0.23

6.2.3 Exemplo 9: Escoamento Supersônico Não-Viscoso Sobre Esfera

Este exemplo analisa um escoamento supersônico sobre uma esfera. Dados experimentais de escoamentos compressíveis no entorno de esferas para diferentes número de Mach e Reynolds vêm sendo publicados desde meados do século XVIII. Uma revisão publicada por Miller e Bailey (1979) estuda estes experimentos do século XVIII e XIX para diferentes número de Mach ($0.2 \leq M \leq 2.0$) com $Re \approx 10^7$. A análise mencionada no artigo revela que os dados experimentais obtidos naquela época estão de acordo com resultados modernos. O arrasto de uma esfera para escoamentos compressíveis possui ampla variação em função do número de Mach e Reynolds. O arrasto decresce muito lentamente em escoamentos com $M \geq 2$ a partir de $Re \geq 10^6$. Para $M = 2$, particularmente, o escoamento apresenta um região de separação tridimensional na onda de choque frontal e um comportamento misto subsônico e supersônico na região posterior à linha sônica ($M = 1$). A pressão de estagnação neste caso é muito superior aos escoamentos subsônicos. Além disto, uma segunda onda de choque é formada na região posterior à esfera, a qual possui grande contribuição sobre o arrasto da mesma. Nestas condições, a hipótese de escoamento não-viscoso é utilizado para simular o caso. A Fig. 6.50 indica a geometria do problema. Devido à simetria, apenas a quarta parte do domínio é analisado. Os dados utilizados para o escoamento são:

- $M_\infty = 2.0$
- $u_{1\infty} = 1.0$
- $u_{2\infty} = 0.0$
- $u_{3\infty} = 0.0$
- $CFL = 0.1$
- $C_E = 5.00$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 300K$

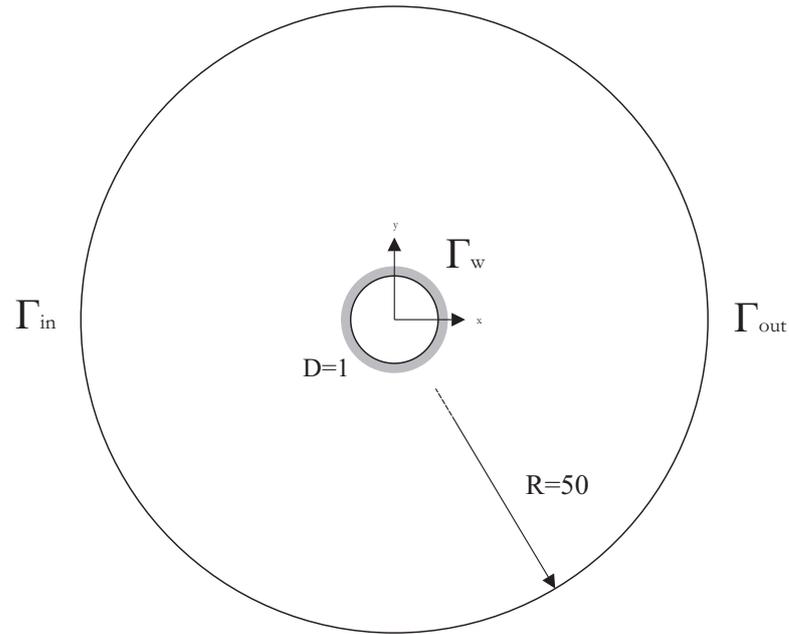
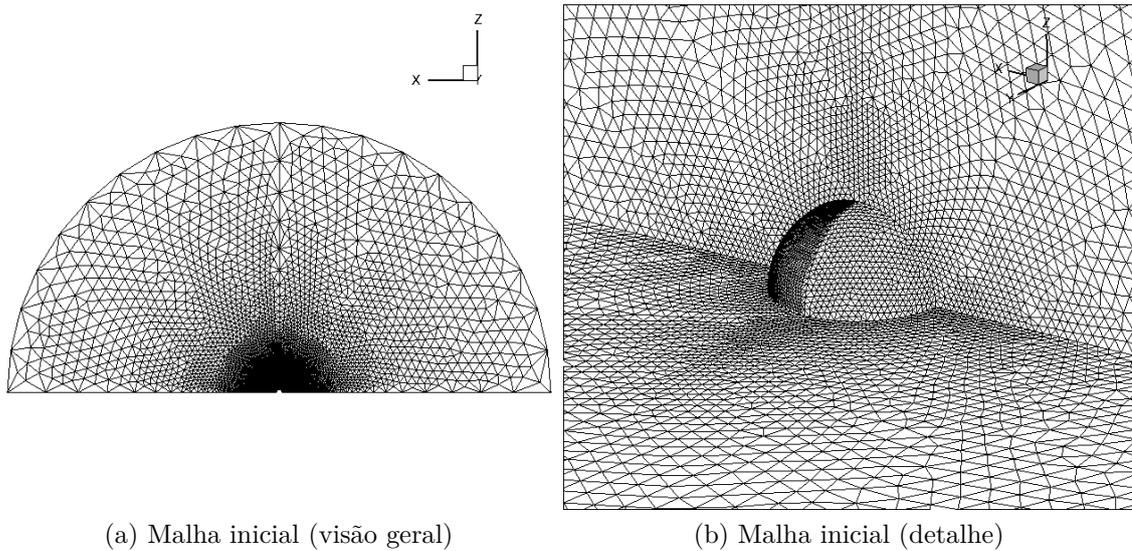


Figura 6.50: Geometria e condições de contorno do exemplo 9.

Os parâmetros de adaptação utilizados são:

- $\phi = \rho$
- $\eta_S = 1.8$
- $\eta_I = 0.1$
- $N_F = 3$
- $F_{\Delta t} = 1500$

A malha inicial utilizada é mostrada na Fig. 6.51. Os resultados obtidos com a malha inicial são apresentados na Fig. 6.52 e os obtidos com a malha adaptada final na Fig. 6.53, no plano xz , para $y = 0$. Na Fig. 6.54 mostra-se 4 diferentes vistas da malha final adaptada no plano yz para diferentes valores de x , evidenciando a captura da onda de choque frontal e posterior, bem como a região no entorno da esfera, a qual possui maior contribuição real sobre o coeficiente de pressão neste caso. A Fig. 6.55 mostra duas isosuperfícies, uma para $\rho = 0.98$ e outra para $M = 1$. A primeira delimita a região em que os efeitos de compressibilidade se iniciam (ou terminam) e a segunda delimita a região sônica. A Tab. 6.13 mostra os dados da malha ao longo das iterações de adaptação e a Tab. 6.14 mostra os coeficientes aerodinâmicos obtidos comparados com



(a) Malha inicial (visão geral)

(b) Malha inicial (detalhe)

Figura 6.51: Malha inicial do exemplo 8.

Tabela 6.13: Dados das iterações de adaptação do exemplo 9

Adaptação	Nós	Elementos	α_{Dmax}	α_{D160}	$\ T_K\ _{min}$	$\sigma^2(E)$
0	84656	463457	178.24	9.062E-5	8.348E-9	4.829E-3
1	290114	1659283	178.88	4.841E-5	1.043E-9	1.879E-3
2	891759	5164125	179.14	1.796E-4	3.940E-10	1.999E-3
3	2844982	16623729	179.50	5.304E-4	6.430E-11	2.435E-3

outras referências. Observa-se que os resultados obtidos possuem boa concordância com os apresentados pelas referências. Por fim, as Fig. 6.56 e Fig. 6.57 mostram a convergência do problema e a variação do coeficiente de arrasto ao longo das iterações, respectivamente. Embora a solução inicial, em termos de arrasto, já possua uma boa resolução, a física do escoamento, contemplando a propagação das ondas de choque corretamente não fica tão bem representada. Através da adaptação de malhas é possível obter uma resolução mais realista, onde as ondas de choque propaguem-se em uma região bastante confinada, sem o espalhamento gerado pela difusão artificial.

Tabela 6.14: Coeficientes aerodinâmicos do exemplo 9

Fonte	C_D
Presente Trabalho (solução inicial)	1.051
Presente Trabalho (solução adaptada final)	1.025
(Nazarov e Hoffman, 2012) (escoamento não-difusivo)	1.008
(Bailey e Hiatt, 1971) (túnel de vento com $Re = 10^6$)	0.990
(Spearman e Brawell, 1993) (túnel de vento com $Re = 10^6$)	1.035

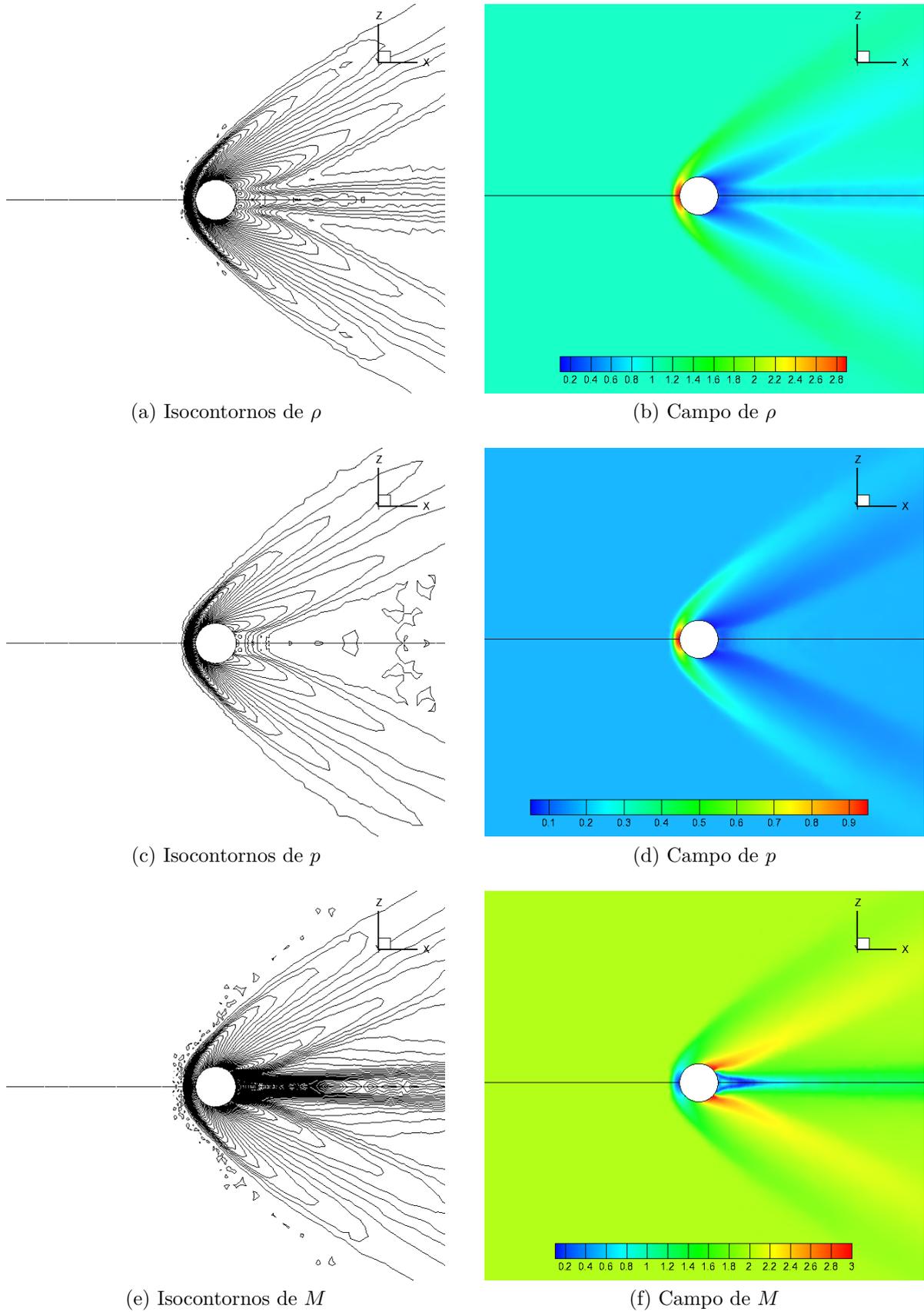


Figura 6.52: Solução inicial no plano xz com $y = 0$ do exemplo 8.

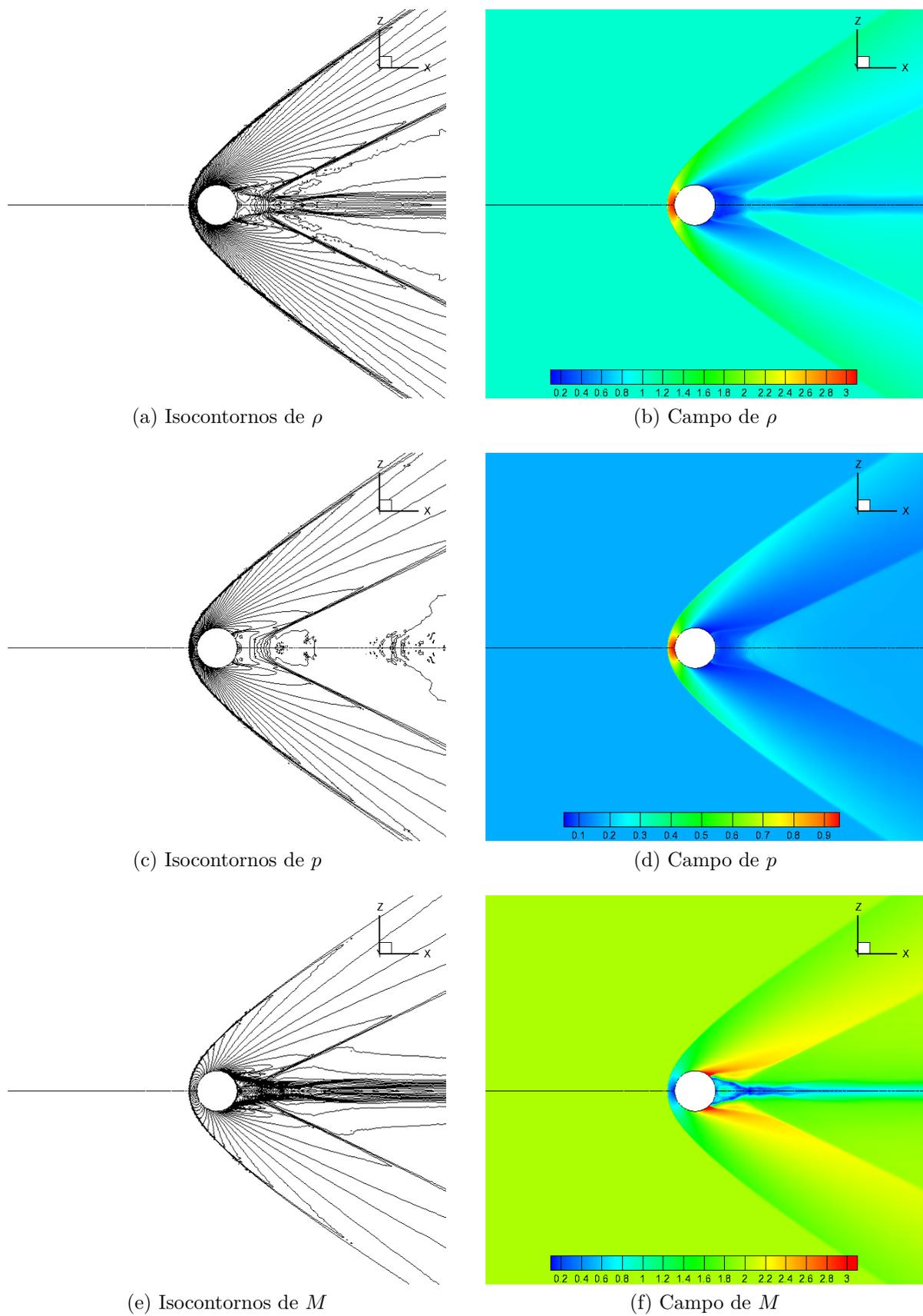


Figura 6.53: Solução final adaptada no plano xz com $y = 0$ do exemplo 8.

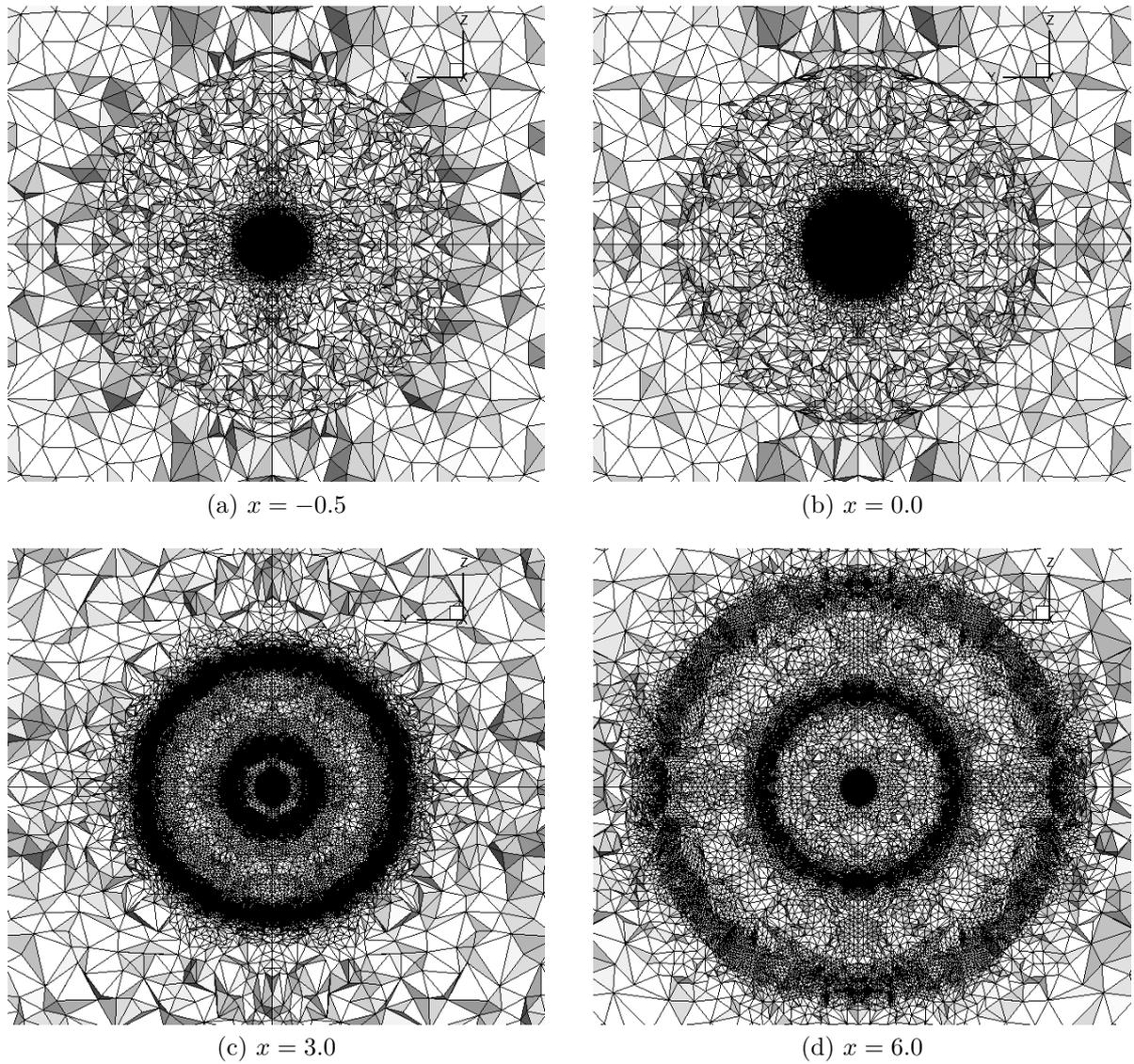


Figura 6.54: Malha da solução adaptada final no plano yz do exemplo 8.

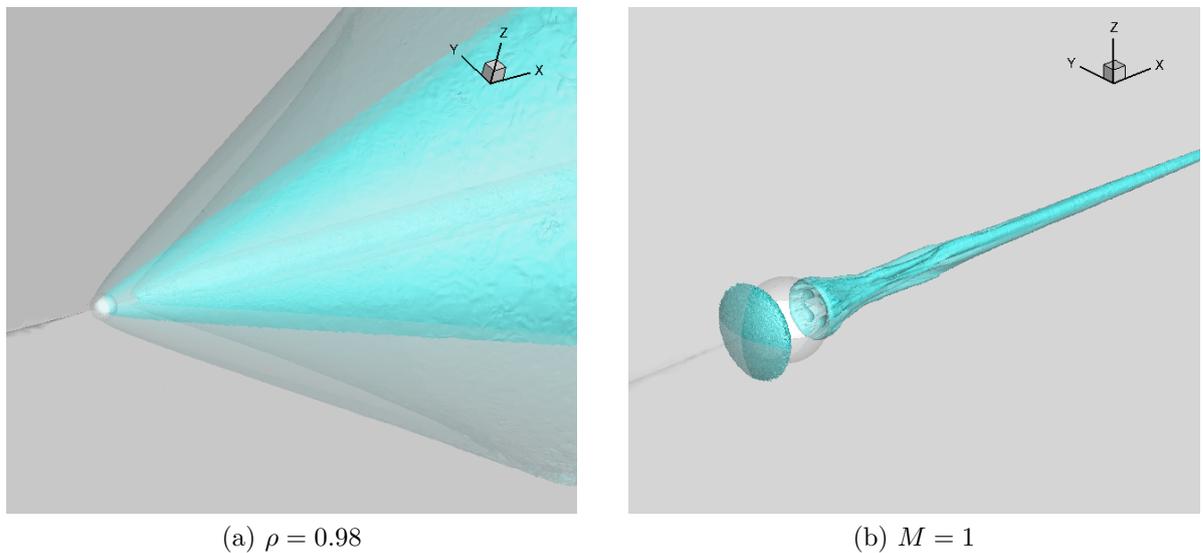


Figura 6.55: Isosuperfícies da solução adaptada final do exemplo 8.

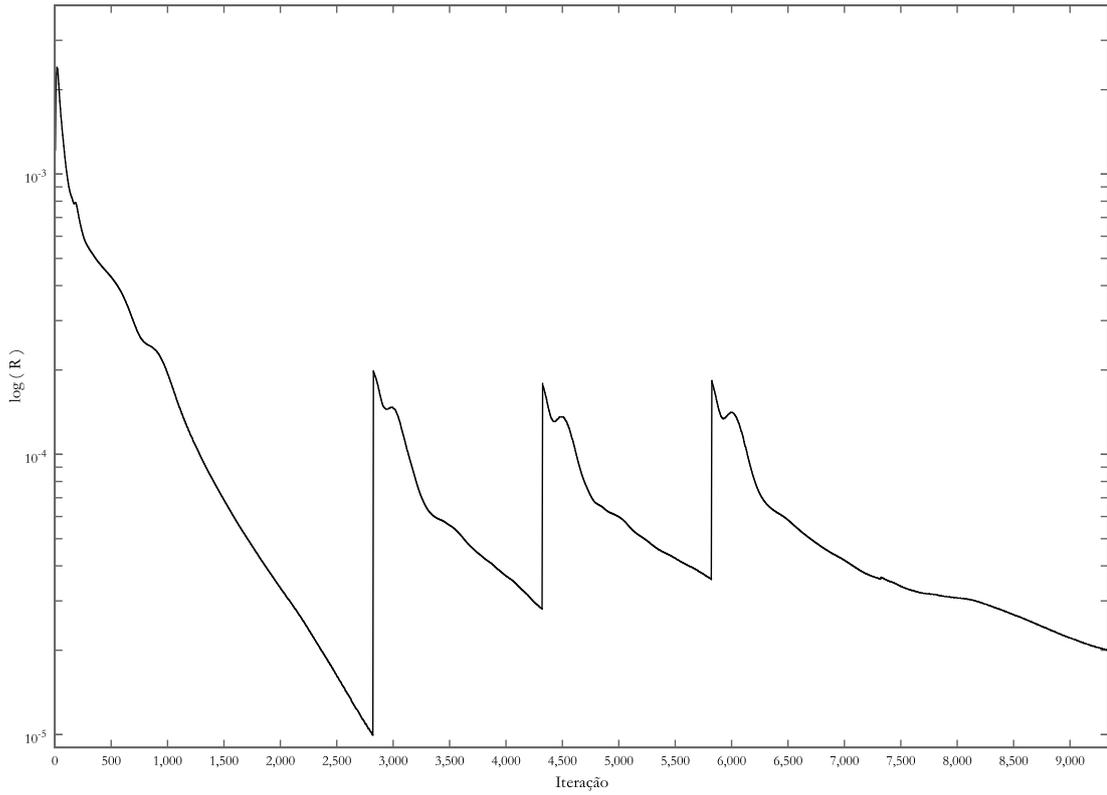
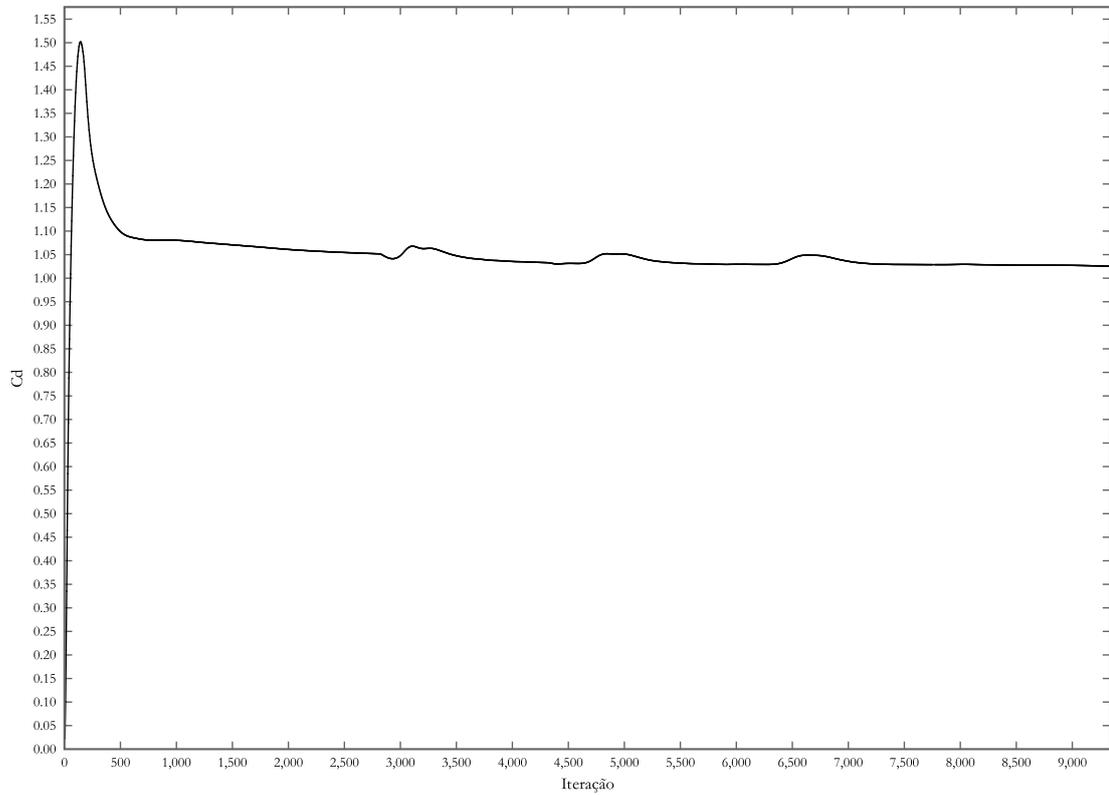


Figura 6.56: Convergência do exemplo 9.

Figura 6.57: Coeficiente de arrasto C_D ao longo das iterações do exemplo 9.

6.2.4 Exemplo 10: Escoamento Transônico Não-Viscoso Sobre Asa Onera M6

A asa Onera M6 é um caso de estudo clássico de validação para um código de dinâmica de fluidos computacional (CFD) devido à geometria simples combinada com um escoamento complexo (regiões supersônicas, subsônicas e presença de choques). Foi analisada experimentalmente em túnel de vento de forma detalhada em diversos regimes transônicos e variados ângulos de ataque por Schmitt e Charpin (1979). A geometria em detalhes da asa é apresentada na Fig. 6.58, onde AR indica a razão de aspecto, λ a razão de esbelte e as outras dimensões estão indicadas abaixo. A área de referência utilizada é $\frac{S}{2}$ e corresponde à área da asa projetada no eixo perpendicular ao da envergadura da asa. O domínio computacional e condições de contorno utilizadas são apresentados na Fig. 6.59. A Fig. 6.60 mostra a malha inicial do problema. Os dados utilizados para o escoamento são:

- $M_\infty = 0.8395$
- $u_{1\infty} = 0.998574$
- $u_{2\infty} = 0.053381$
- $u_{3\infty} = 0.0$
- $CFL = 0.075$
- $C_E = 7.00$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 273.15K$

Os parâmetros de adaptação utilizados são:

- $\phi = \rho$
- $\eta_S = 1.5$
- $\eta_I = 0.15$

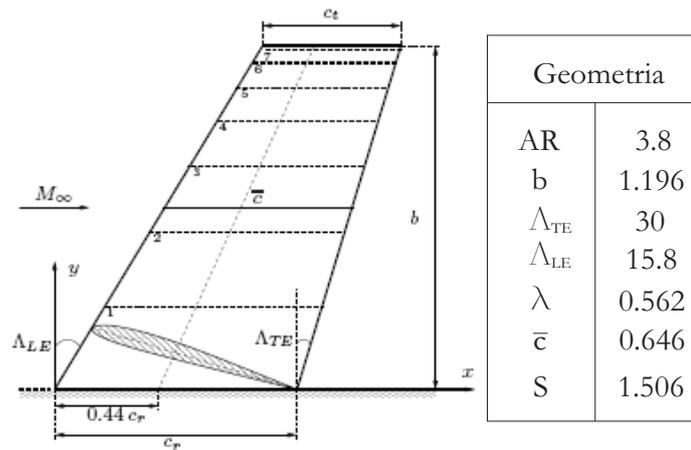


Figura 6.58: Geometria detalhada da asa Onera M6 do exemplo 10.

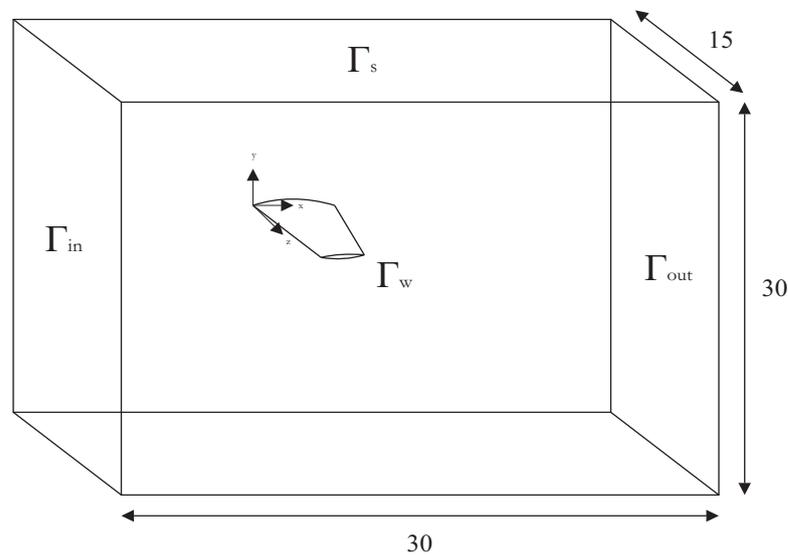


Figura 6.59: Condições de contorno do exemplo 10.

- $N_F = 2$
- $F_{\Delta t} = 1500$

Os resultados obtidos com a malha inicial são apresentados na Fig. 6.61 e os obtidos com a malha adaptada final na Fig. 6.62, onde as isolinhas apresentadas situam-se no plano xy , para três diferentes valores de z . Na Fig. 6.63 mostram-se 4 diferentes vistas da malha final adaptada no plano xy para diferentes valores de z , onde observa-se a captura da onda de choque e o refinamento mais acentuado no bordo de ataque e de fuga. A Fig. 6.64 mostra duas isosuperfícies, uma para $\rho = 0.98$ e outra para $M = 1$, a qual define a superfície sônica do escoamento. A Tab. 6.15 mostra os dados da malha ao longo das iterações de adaptação e a Tab. 6.16 mostra os coeficientes aerodinâmicos obtidos

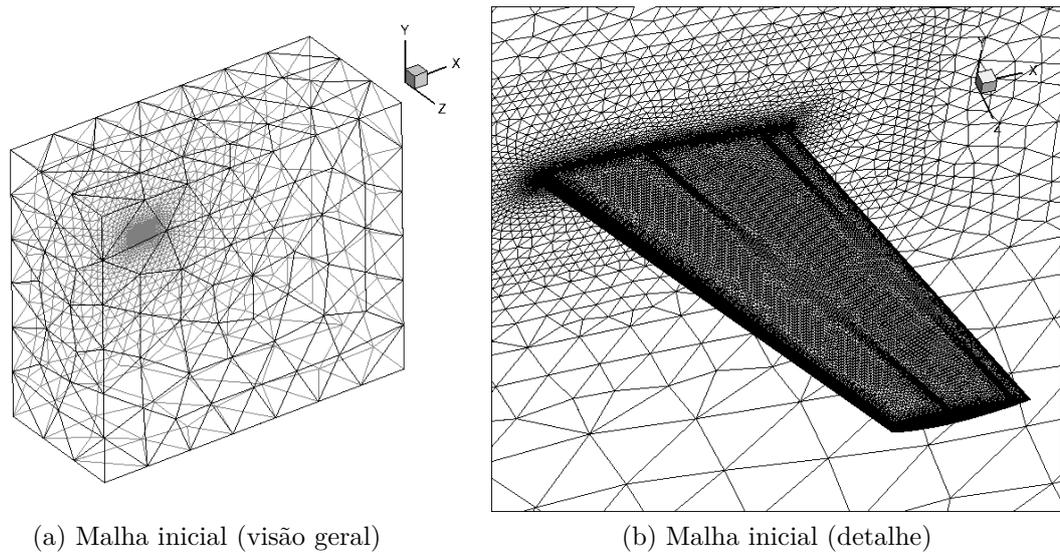


Figura 6.60: Malha inicial do exemplo 10.

Tabela 6.15: Dados das iterações de adaptação do exemplo 10

Adaptação	Nós	Elementos	α_{Dmax}	α_{D160}	$\ T_K\ _{min}$	$\sigma^2(E)$
0	234823	1274973	176.73	3.660E-6	3.179E-10	2.262E-3
1	737837	4141770	176.73	6.518E-6	3.974E-11	1.270E-3
2	2641704	15125961	177.66	8.652E-5	4.967E-12	8.268E-4

comparados com outras referências. Na Fig. 6.65 apresenta-se o coeficiente de pressão para diferentes seções da asa. Observa-se que a solução adaptada é mais precisa que a inicial quando comparada com a solução experimental sobreposta. Na solução adaptada o coeficiente de pressão atinge os picos com precisão, além de melhorar a região no entorno do choque, tornado mais visível a descontinuidade. Na Fig. 6.66 apresenta-se o resíduo ao longo das iterações do problema e na Fig. 6.67 e Fig. 6.68 pode-se observar o coeficiente de arrasto e sustentação, respectivamente. É visível que o o coeficiente de sustentação do problema possui convergência oscilatória assintótica, cuja oscilação é reduzida após a adaptação.

Tabela 6.16: Coeficientes aerodinâmicos do exemplo 10

Fonte	C_L	C_D
Presente Trabalho (solução inicial)	0.2727	0.0153
Presente Trabalho (solução adaptada final)	0.2554	0.0114
(Bassi et al., 2006)	0.2609	0.0118
(Mavriplis, 1991)	0.2923	0.0131
(van Dam, 1999)	0.2827	0.0114

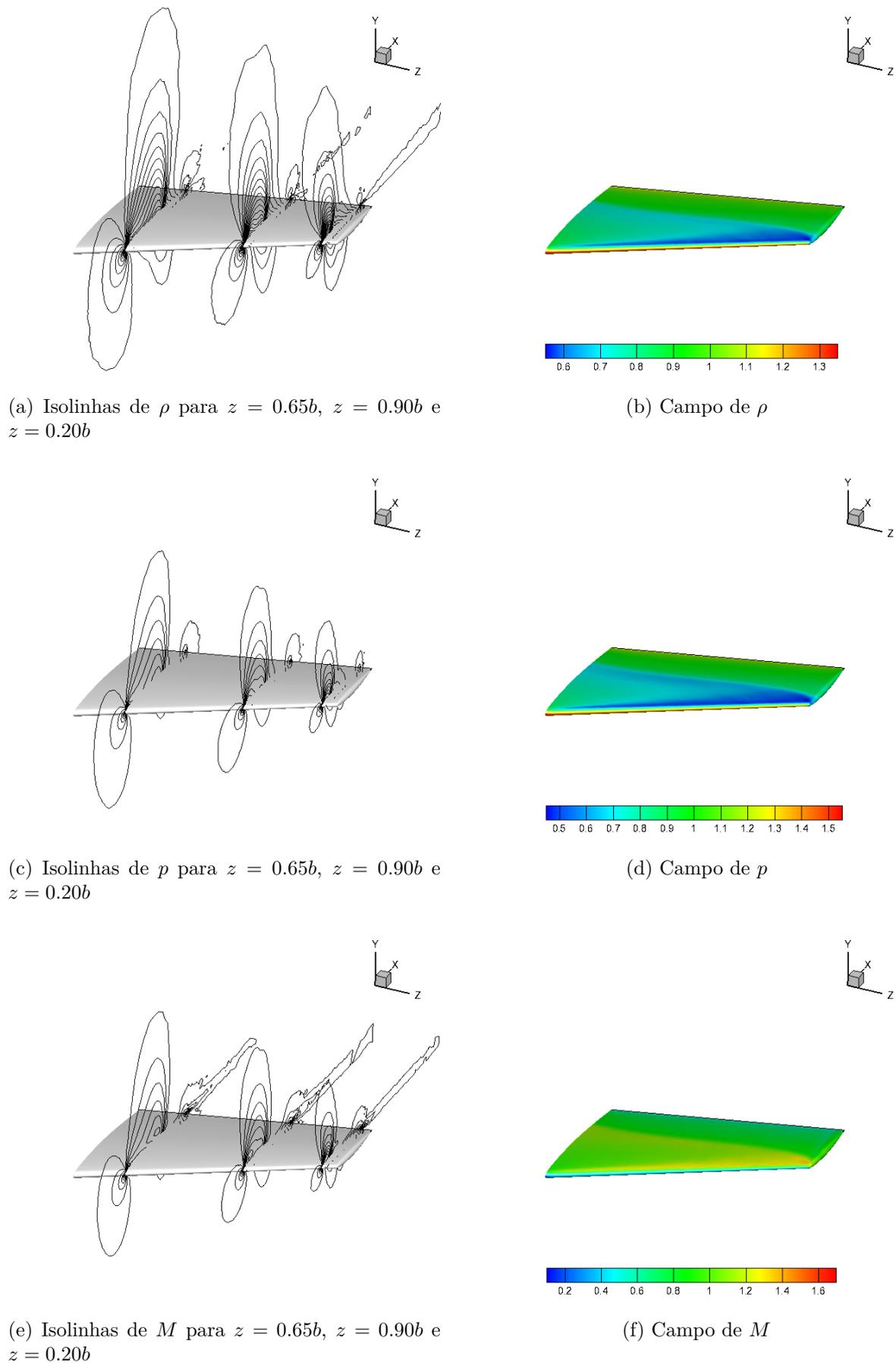


Figura 6.61: Solução inicial do exemplo 10.

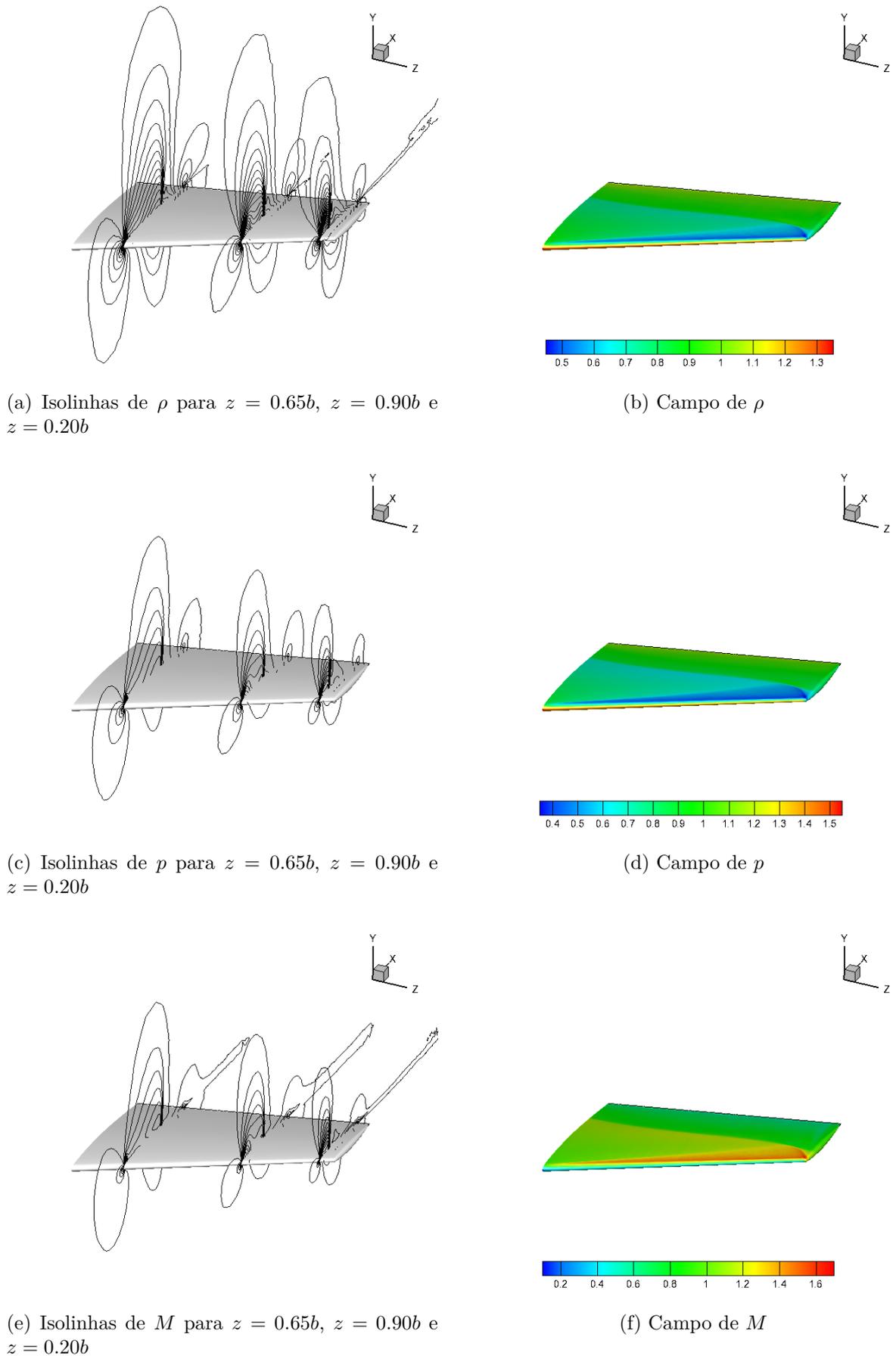


Figura 6.62: Solução final adaptada do exemplo 10.

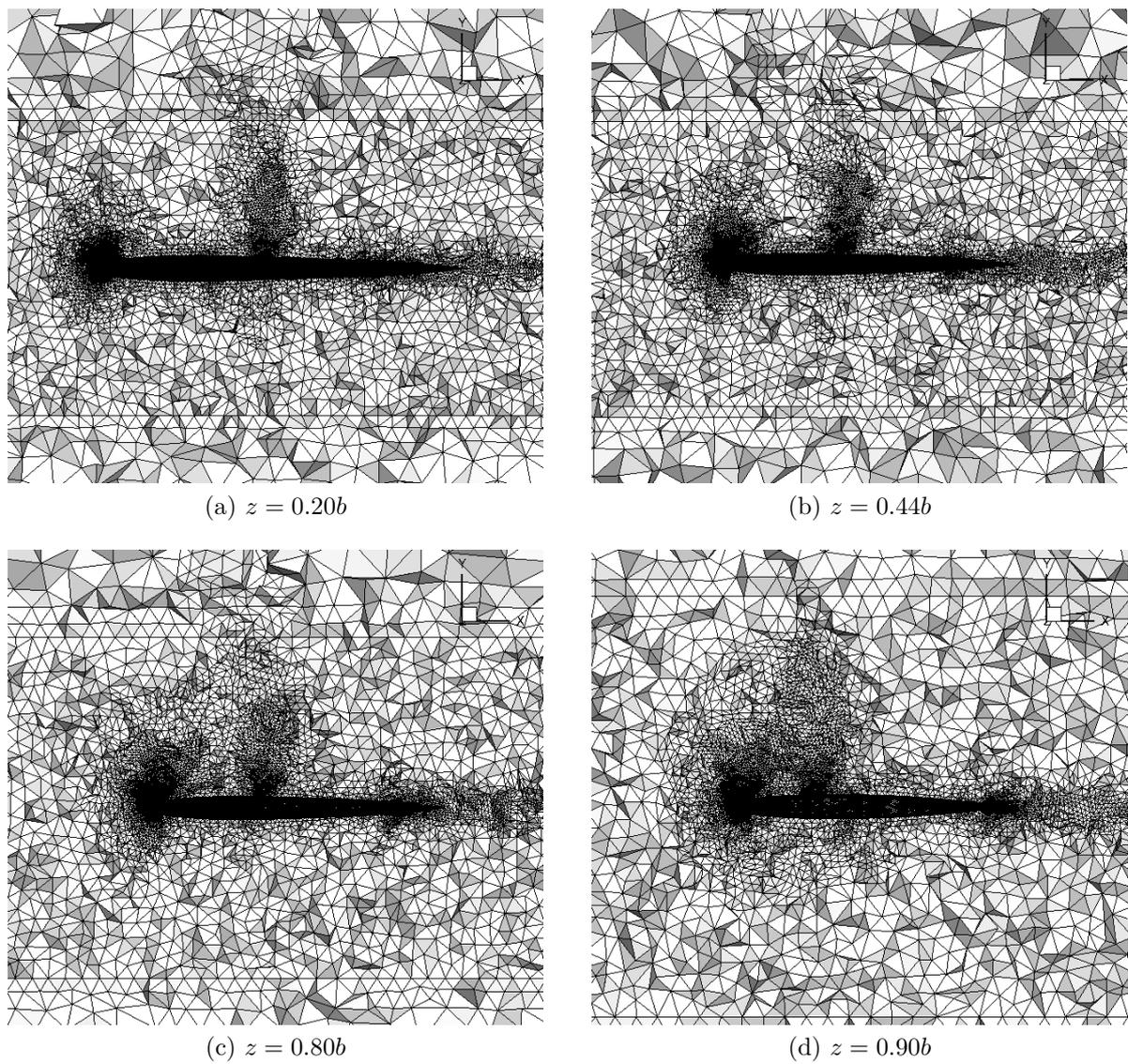


Figura 6.63: Malha da solução adaptada final no plano xy do exemplo 10.

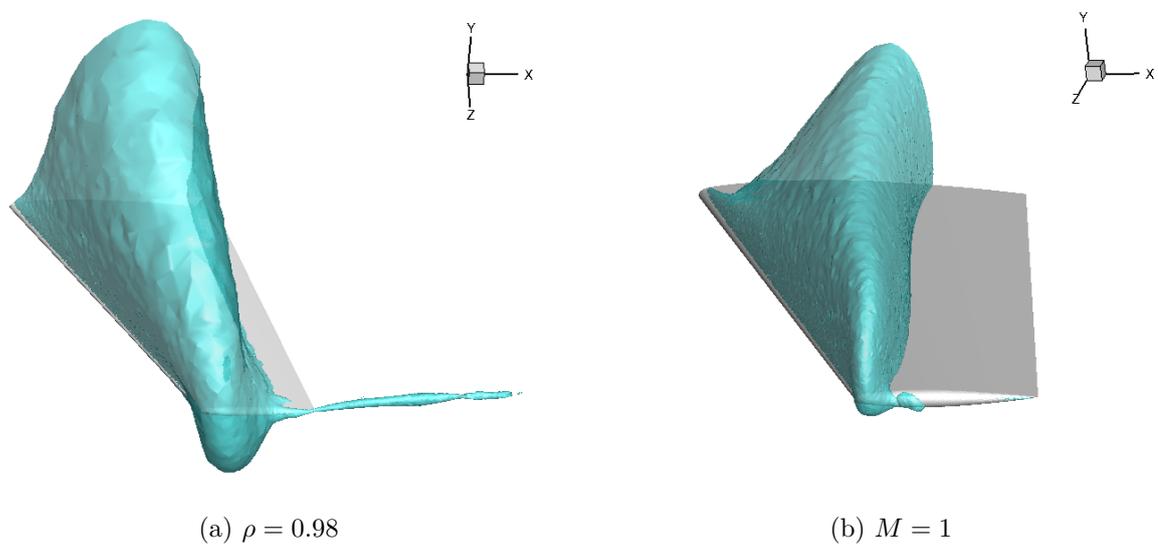


Figura 6.64: Isosuperfícies da solução adaptada final do exemplo 10.

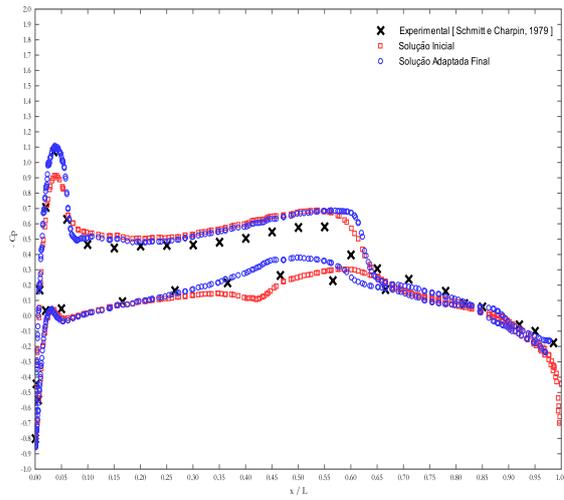
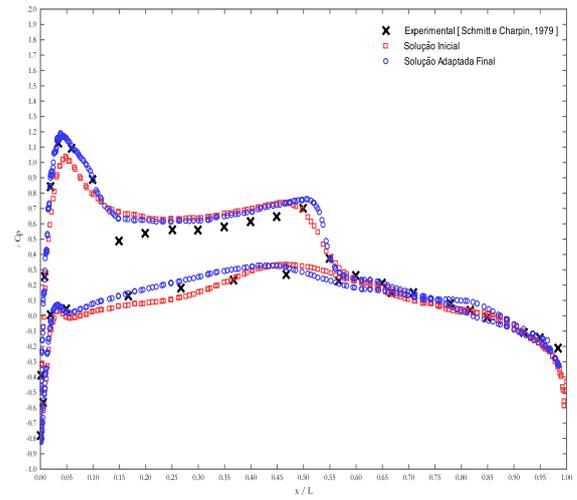
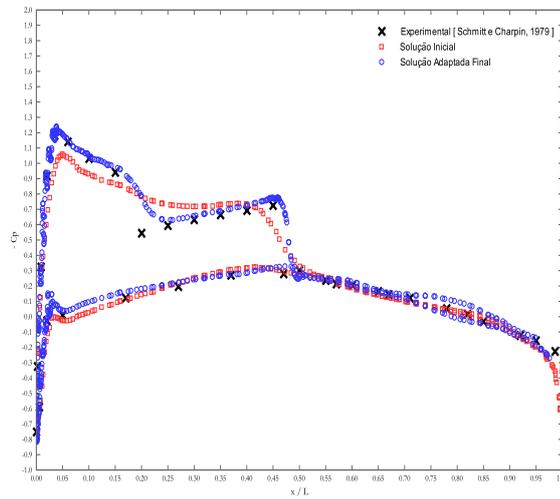
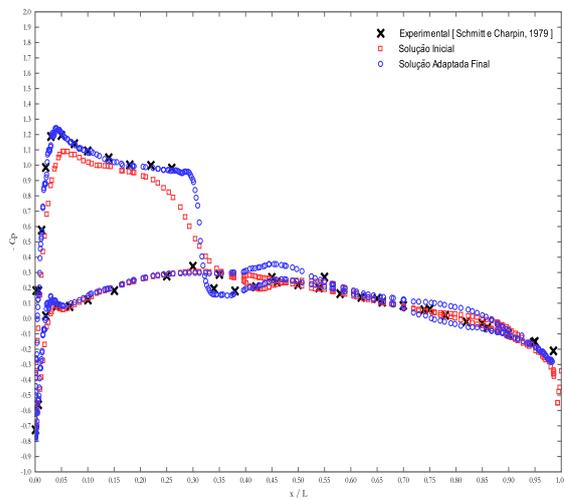
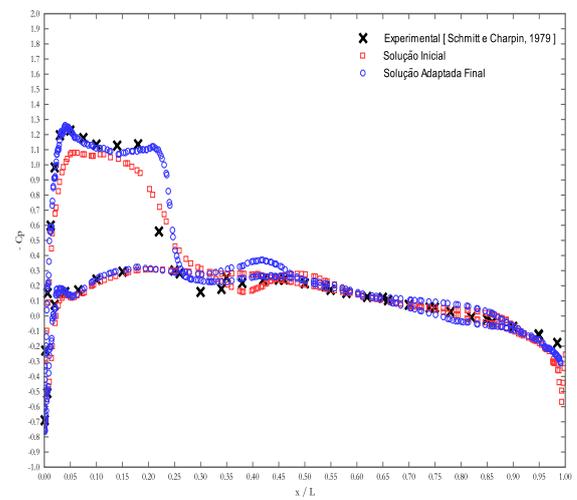
(a) $z = 0.20b$ (b) $z = 0.44b$ (c) $z = 0.65b$ (d) $z = 0.90b$ (e) $z = 0.95b$

Figura 6.65: Coeficientes de pressão C_P para diferentes seções da asa do exemplo 10.

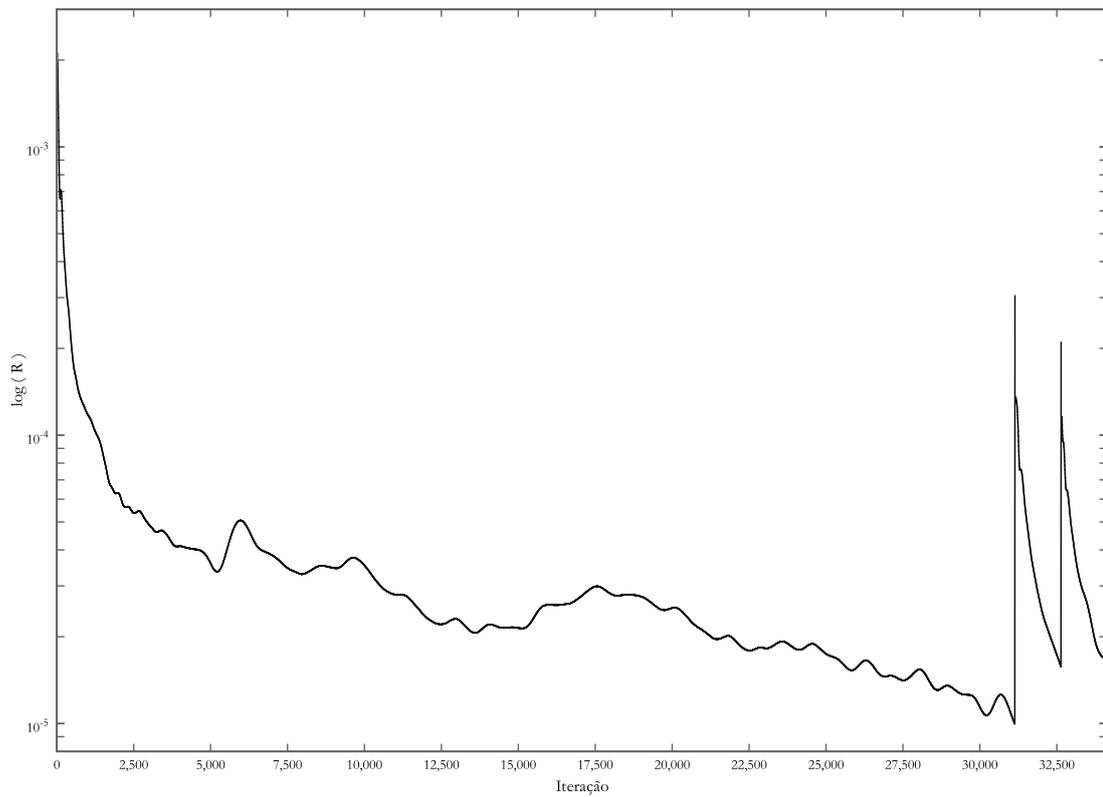
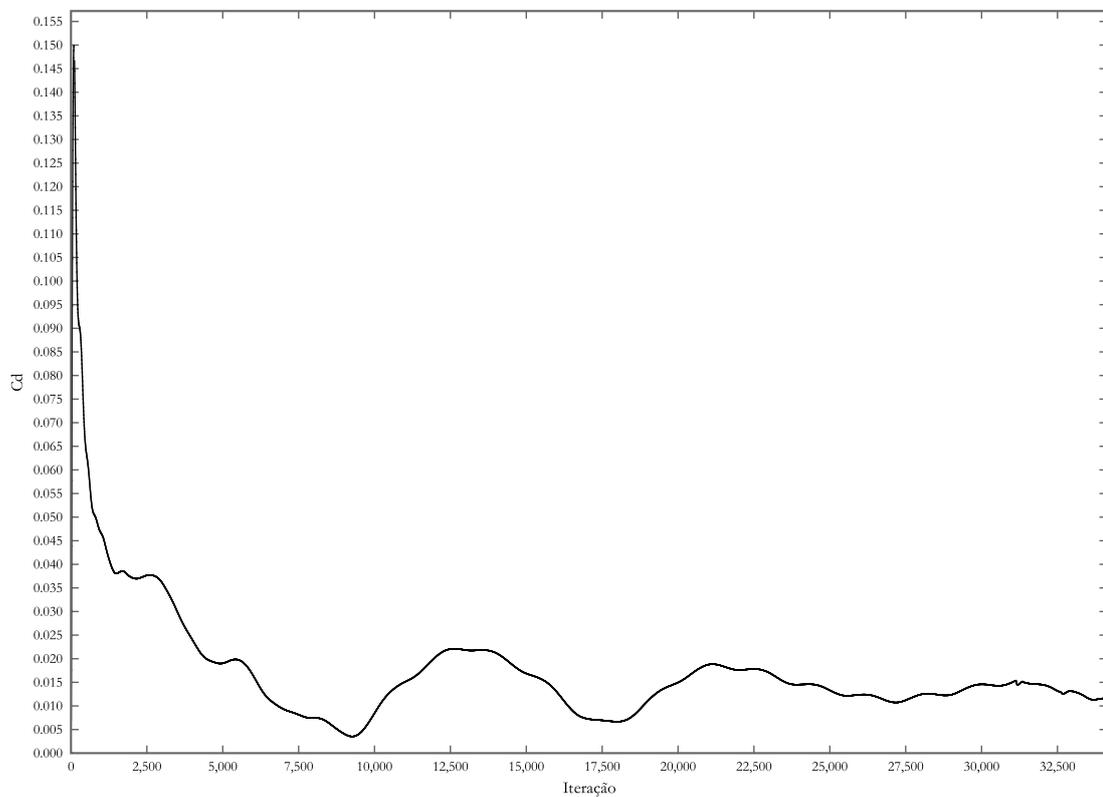


Figura 6.66: Convergência do exemplo 10.

Figura 6.67: Coeficiente de arrasto C_D ao longo das iterações do exemplo 10.

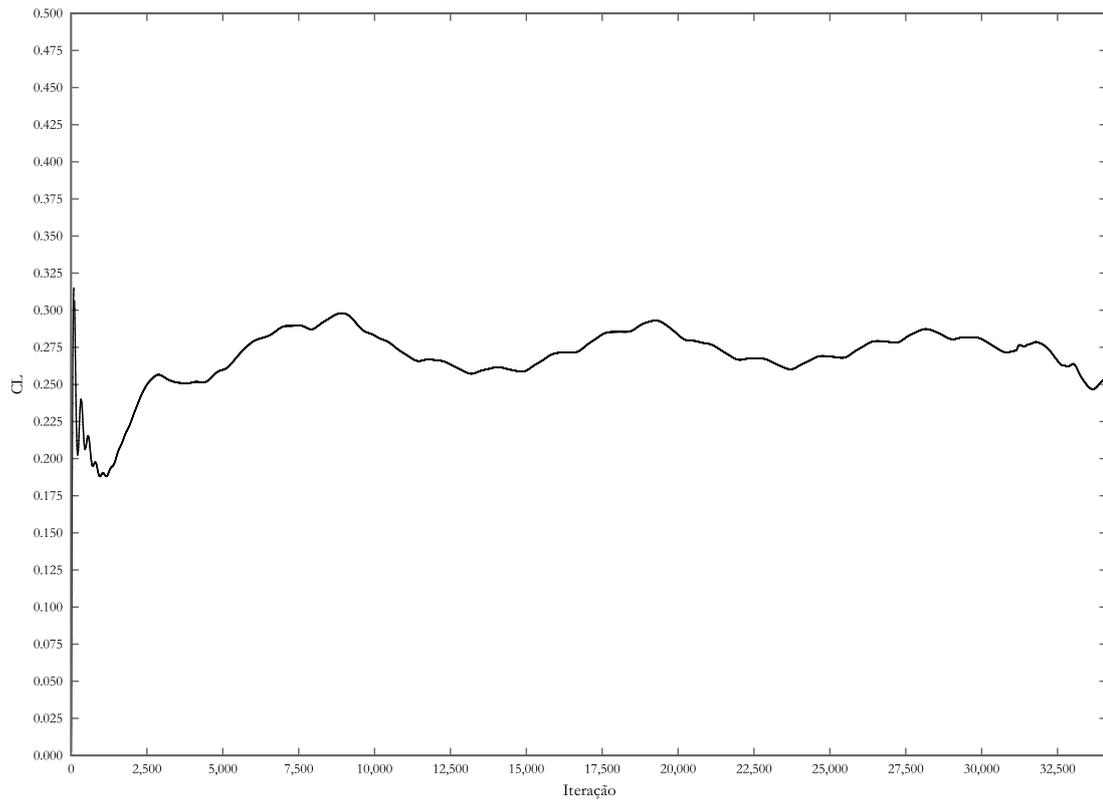


Figura 6.68: Coeficiente de sustentação C_L ao longo das iterações do exemplo 10.

6.2.5 Exemplo 11: escoamento Transônico Não-Viscoso Sobre Avião

Este exemplo é um dos estudos propostos pelo 5th AIAA CFD *Drag Prediction Workshop*, o qual consiste em verificar um dado valor de sustentação para uma geometria de um avião comercial em regime transônico e turbulento. Devido à falta de modelo de turbulência no presente código implementado, o problema é simulado considerando-se o fluido como não-viscoso. Os dados utilizados para o escoamento são:

- $M_\infty = 0.85$
- $u_{1\infty} = -1.0$
- $u_{2\infty} = 0.0$
- $u_{3\infty} = 0.0$
- $CFL = 0.1$
- $C_E = 0.25$

Tabela 6.17: Coeficientes aerodinâmicos do exemplo 11

Fonte	C_L
Presente Trabalho (solução inicial)	0.515
Presente Trabalho (solução adaptada final)	0.549
(Vassberg et al., 2008) (sustentação calibrada para o modelo turbulento)	0.500

Tabela 6.18: Dados das iterações de adaptação do exemplo 11

Adaptação	Nós	Elementos	α_{Dmax}	α_{D160}	$\ T_K\ _{min}$	$\sigma^2(E)$
0	1415648	7889998	177.80	1.079E-5	2.149E-3	8.012E-4
1	3904068	22319872	178.30	7.407E-6	2.687E-4	5.361E-4
2	7883201	45291276	179.54	5.354E-5	3.315E-5	4.738E-4

- $R_T = 1 \times 10^{-6}$
- $T_\infty = 280K$

Os parâmetros de adaptação utilizados são:

- $\phi = \rho$
- $\eta_S = 1.5$
- $\eta_I = 0.25$
- $N_F = 2$
- $F_{\Delta t} = 1500$

As figuras Fig. 6.69, Fig. 6.70 e Fig. 6.71 mostram a malha inicial empregada. Os detalhes da geometria são encontrados em Vassberg et al. (2008). Na Fig. 6.72 são apresentados as isolinhas e campos da solução obtidas com a malha final adaptada. A Fig. 6.73 mostra a malha final adaptada. A Fig. 6.74 mostra o campo de pressão e a Fig. 6.75 o campo de Mach da solução adaptada final em perspectiva. Na Fig. 6.76 apresenta-se o histórico de convergência do problema e na Fig. 6.77 o coeficiente de sustentação ao longo das iterações. A Tab. 6.17 mostra o coeficiente de sustentação obtido e a Tab. 6.18 os dados da malha. Embora haja certa divergência dos resultados obtidos, deve-se lembrar que o problema original comparado é viscoso e altamente turbulento (sendo $Re = 40 \times 10^6$).

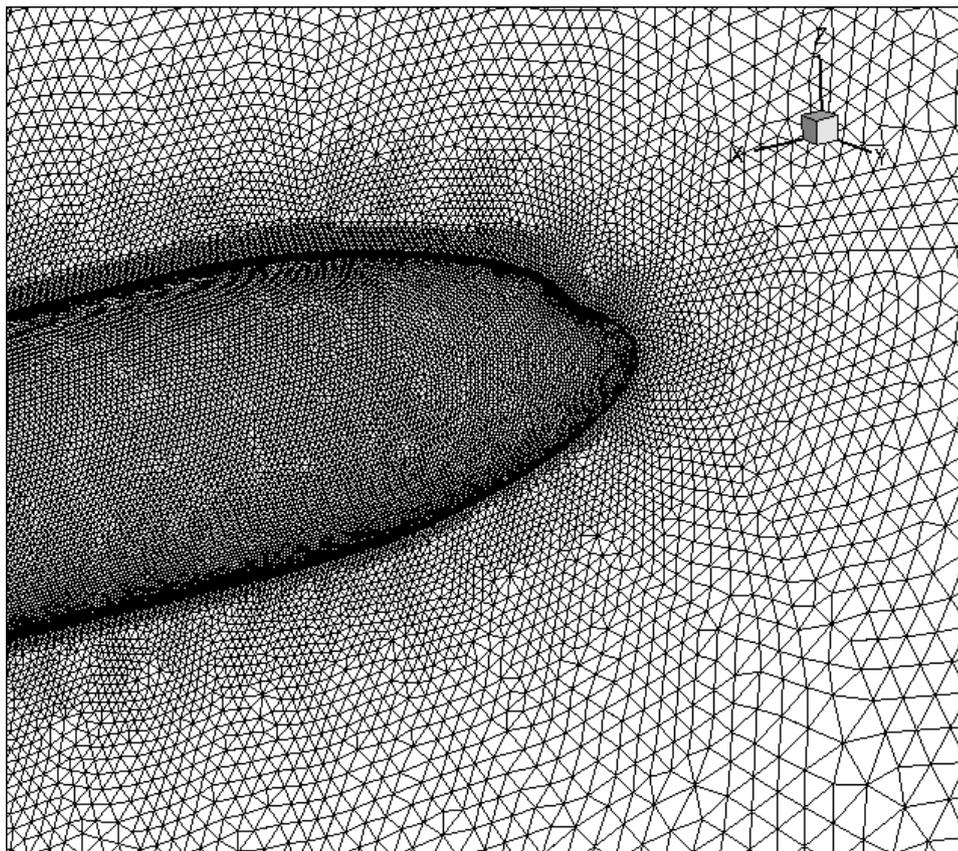


Figura 6.69: Malha Inicial do exemplo 11.

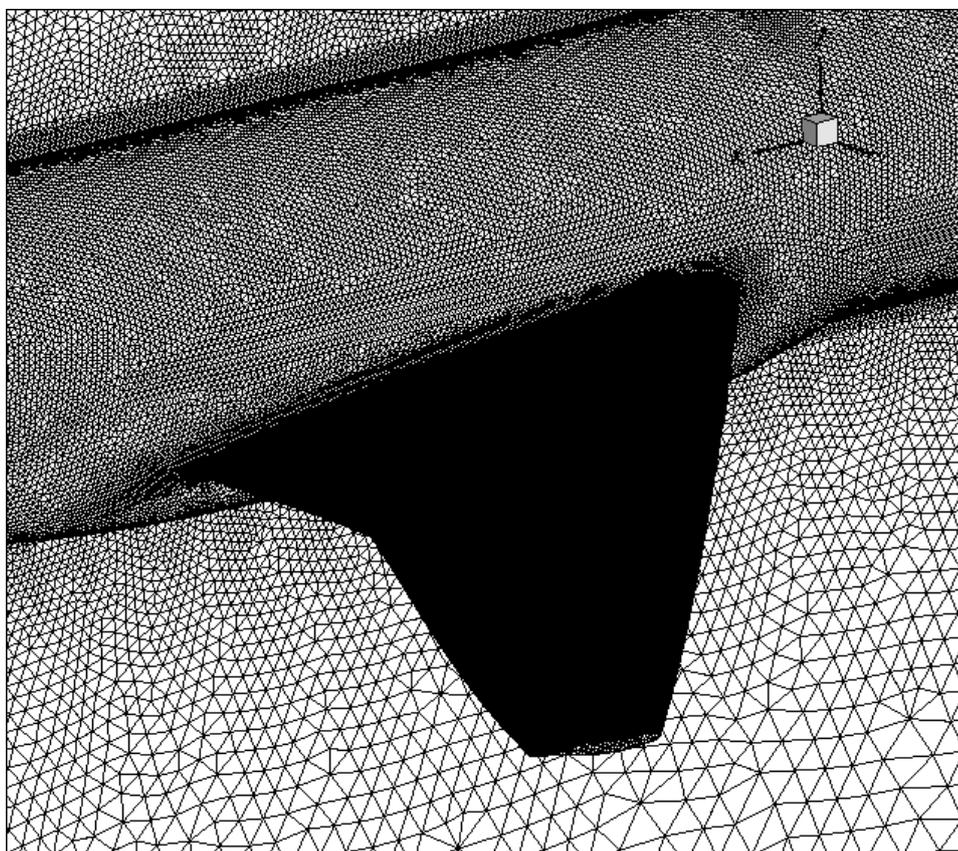


Figura 6.70: Malha Inicial do exemplo 11.

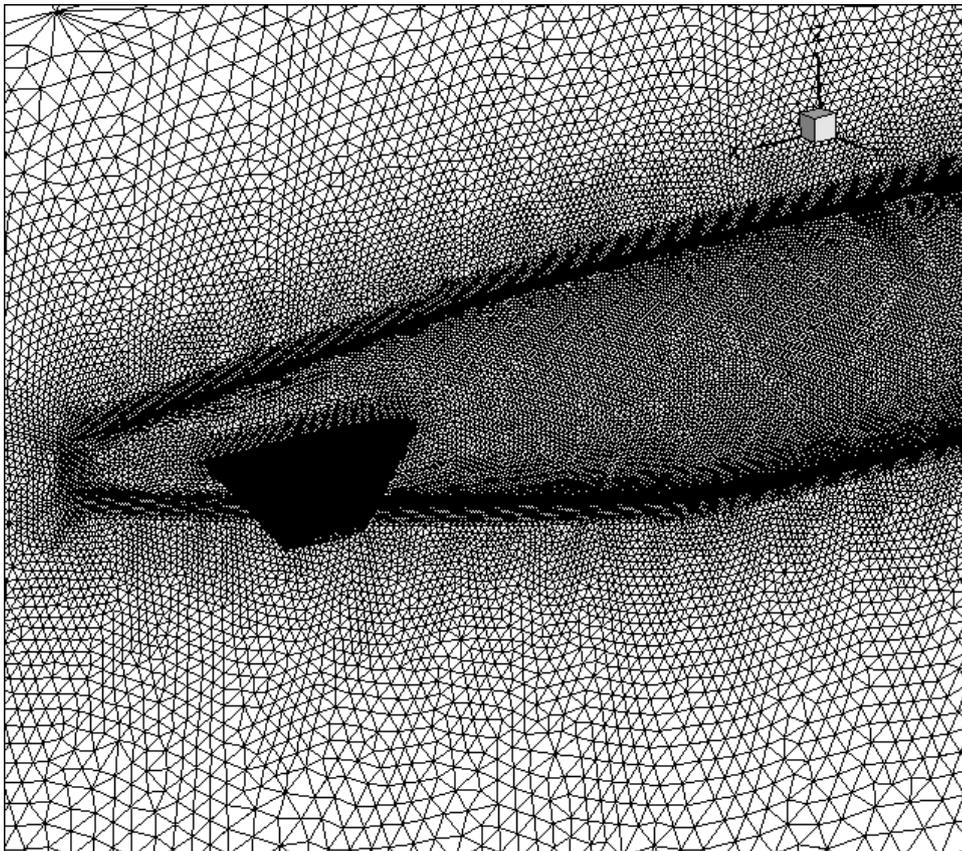


Figura 6.71: Malha Inicial do exemplo 11.

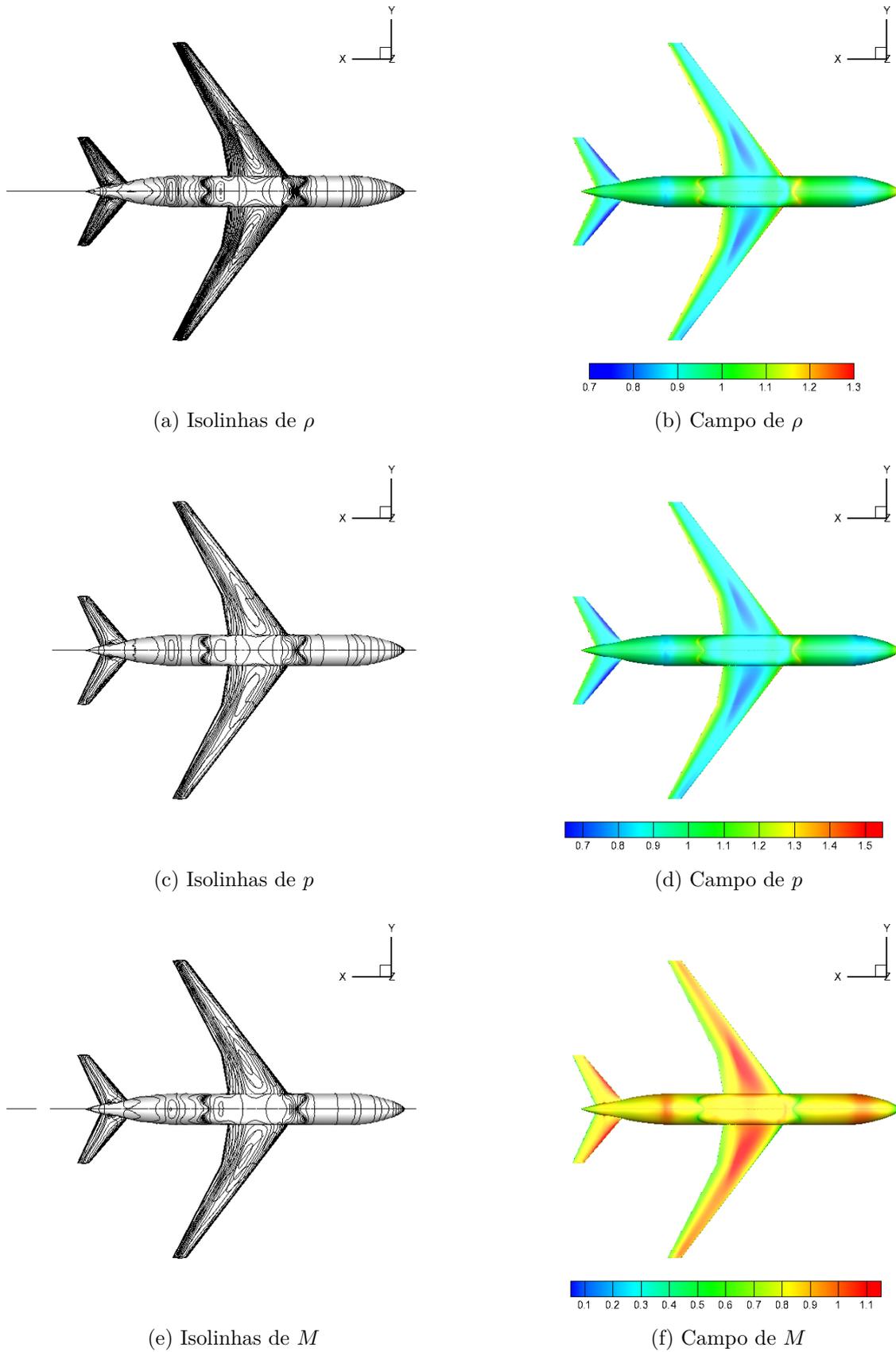
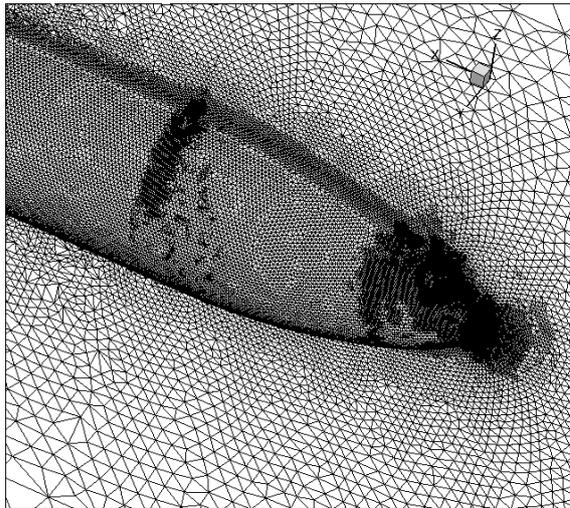
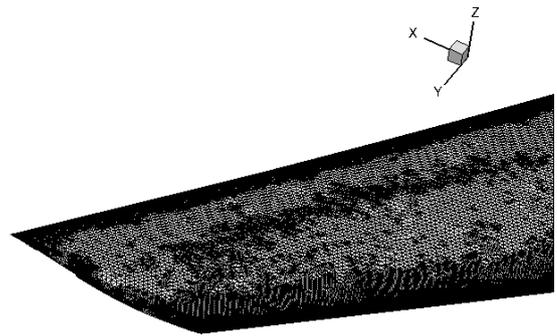


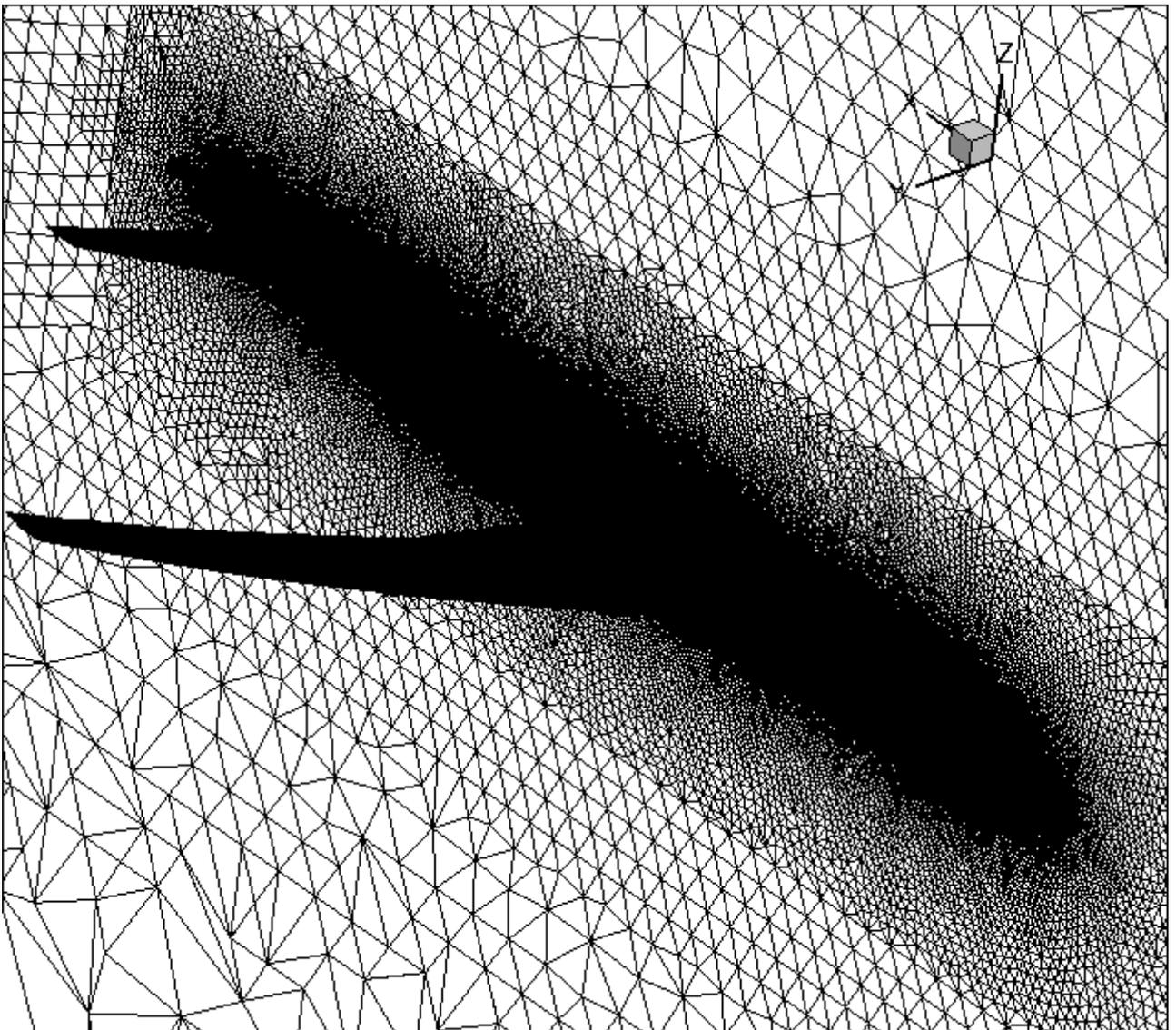
Figura 6.72: Solução adaptada final no plano xy do exemplo 11.



(a) Detalhe da região frontal



(b) Detalhe da asa



(c) Visão geral

Figura 6.73: Malha adaptada final do exemplo 11.

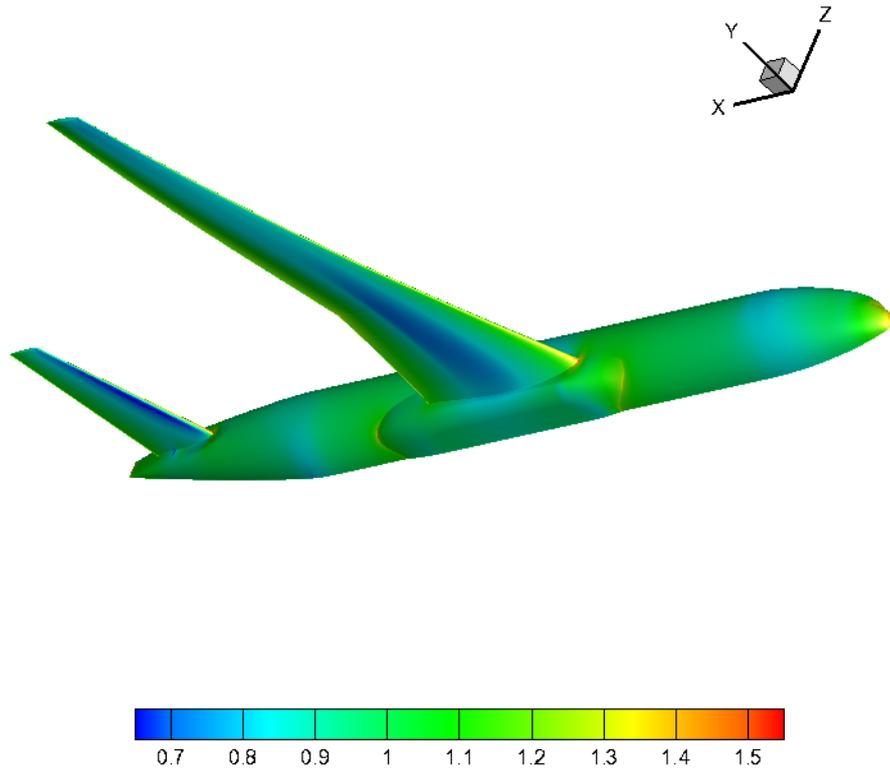


Figura 6.74: Campo de p (vista em perspectiva) do exemplo 11.

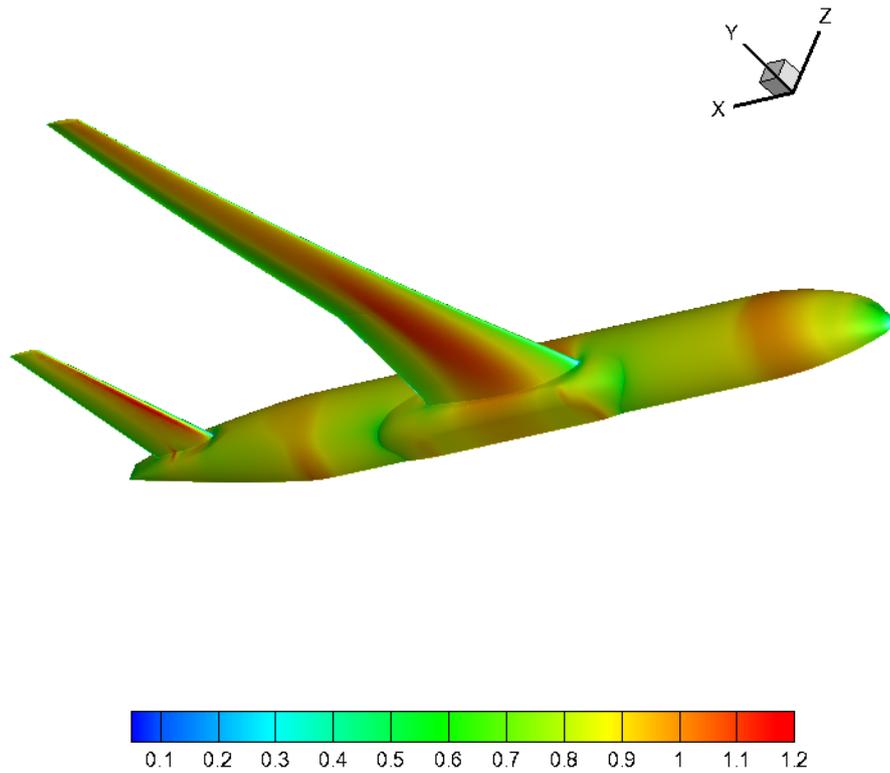


Figura 6.75: Campo de M (vista em perspectiva) do exemplo 11.

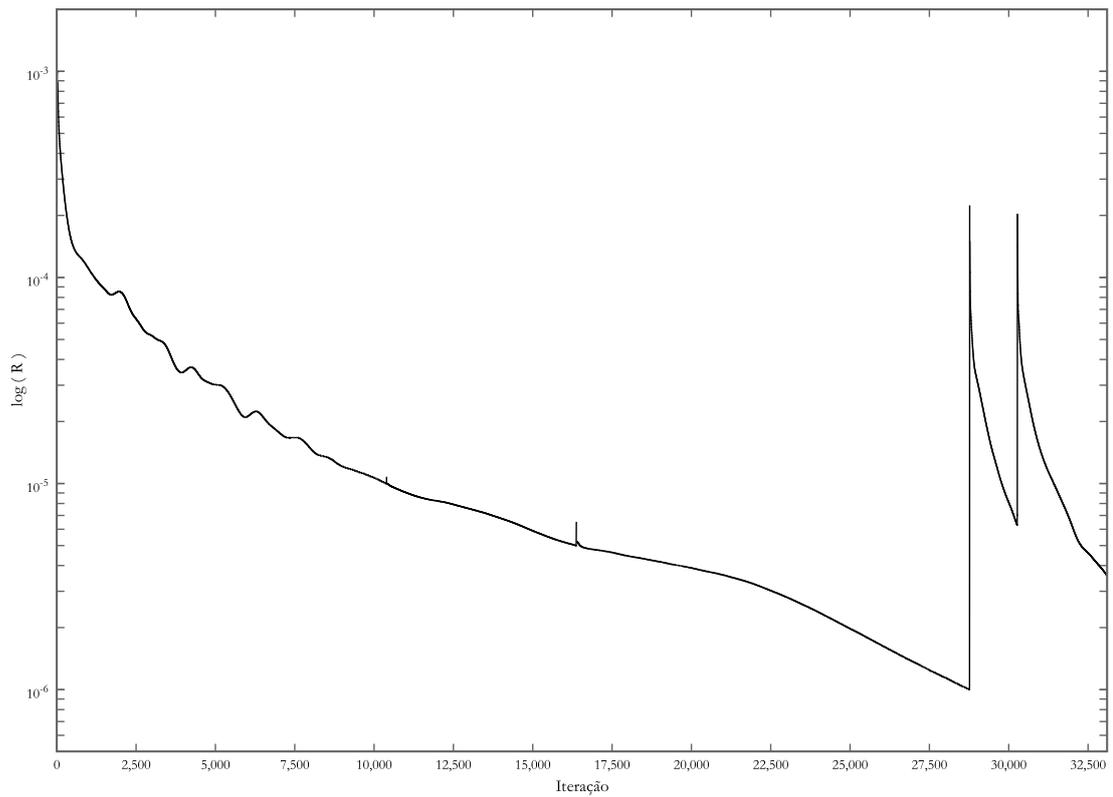
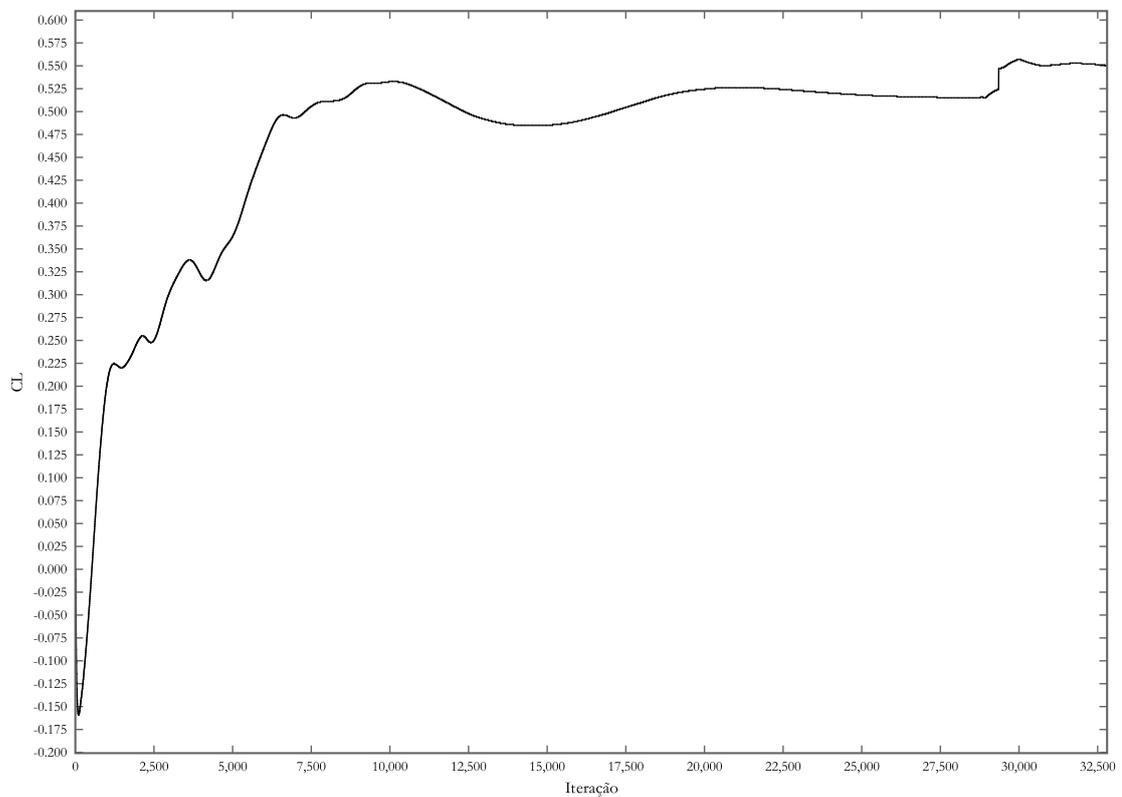


Figura 6.76: Convergência do exemplo 11.

Figura 6.77: Coeficiente de sustentação C_L ao longo das iterações do exemplo 11.

6.2.6 Exemplo 12: Escoamento Supersônico Não-Viscoso Sobre Míssil AGM-114 Hellfire

Do ponto de vista aerodinâmico, o míssil Hellfire é um míssil pequeno comparado com outros modelos AGM como o AGM-86 (míssil de alto alcance), o AGM-88 (míssil anti-radar de alta velocidade) e o AGM-65 (míssil de alta precisão e pequenas distâncias). A característica principal do AGM-114 é a versatilidade de uso. O míssil não possui asas, principalmente devido à velocidade supersônica do mesmo. Estretando, o AGM-114 faz uso de 4 canards e 4 profundores, os quais garantem grande capacidade de manobras aéreas em elevados ângulos de ataque. O uso de 4 canards dispostos simetricamente permite manobras não-rotacionais (como um movimento cartesiano), o qual não é possível com o uso de apenas 3 ou 2 canards, onde o movimento é basicamente de rotação (Fleeman, 2001). A ponteira em forma de calota esférica possui performance aerodinâmica inferior às de ponteira cônica, entretanto é mais eficiente eletromagneticamente e ópticamente. Um ponto considerável sobre o arrasto total do míssil é a saliência da região de engate do míssil. Tipicamente, estas saliências produzem um chamado arrasto parasita que pode atingir parcelas consideráveis do arrasto total do míssil (Gregorek, 1970). Entre as possíveis formas de redução deste arrasto parasita situa-se o aumento do comprimento da região do engate e uso de seção vazada. Outras soluções são possíveis. No caso do AGM-114, a solução utilizada foi o uso de uma espécie de nariz de vento, uma ponteira cônica, a qual protege os engates a jusante do escoamento, reduzindo o arrasto. O presente exemplo consiste em um estudo detalhado das características aerodinâmicas do míssil AGM-114 Hellfire. O estudo é conduzido considerando uma geometria aproximada do míssil, dos canards e de seus engates. O nariz de vento não é incluído no estudo. Entre as dificuldades do exemplo, pode-se citar a complexidade do escoamento, devido à presença dos canards e engates, formando-se múltiplas ondas de choque tridimensionais de formato não simétrico e não cônico, as quais interagem entre si formando regiões complexas, criando a necessidade de uma discretização espacial suficiente que acompanhe estas regiões de gradientes elevados. A geometria real do míssil e a aproximação utilizada são mostradas na Fig. 6.78. O domínio computacional utilizado é de formato esférico e é esquematizado na Fig. 6.79 em conjunto com as condições de contorno utilizadas. A Fig. 6.80 mostra a malha inicial empregada para a simulação. Os dados utilizados para o escoamento são:

- $M_\infty = 1.2$
- $u_{1\infty} = 0.0$
- $u_{2\infty} = 0.0$
- $u_{3\infty} = -1.0$
- $CFL = 0.1$
- $C_E = 5.00$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 280K$

Os parâmetros de adaptação utilizados são:

- $\phi = \rho$
- $\eta_S = 1.5$
- $\eta_I = 0.1$
- $N_F = 3$
- $F_{\Delta t} = 1500$

Assim, simulou-se o míssil próximo de sua velocidade máxima de operação e ângulo de ataque nulo. O problema é simulado sem simetria. Os resultados obtidos com a malha inicial são apresentados na Fig. 6.81 e os obtidos com a malha adaptada final na Fig. 6.82, no plano yz . A Fig. 6.83 mostra os campos de solução para a solução adaptada final sobre a superfície do míssil e em perspectiva.

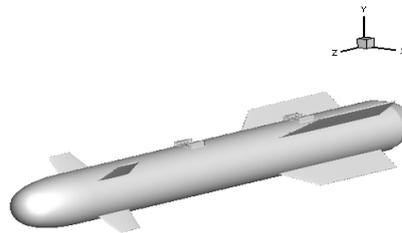
A Fig. 6.84 mostra uma sucessão de cortes do campo de Mach para a solução adaptada final. Os cortes são apresentados segundo um eixo k na direção do escoamento com origem na ponta do míssil. Na Fig. 6.85 tem-se algumas visualizações da malha final adaptada. Na Fig. 6.86 são mostradas várias isosuperfícies para diferentes número de Mach. É possível visualizar a região sônica e outras regiões, mostrando a complexidade espacial

Tabela 6.19: Dados das iterações de adaptação do exemplo 12

Adaptação	Nós	Elementos	α_{Dmax}	α_{D160}	$\ T_K\ _{min}$	$\sigma^2(E)$
0	510407	2747055	176.79	3.064E-6	3.580E-3	5.772E-3
1	2040074	9897207	177.05	4.487E-5	1.642E-4	3.278E-3
2	5318766	30608633	178.31	9.790E-5	5.281E-4	2.440E-3
3	19860325	115496405	178.31	3.047E-4	6.763E-5	1.261E-3



(a) Geometria real



(b) Geometria aproximada utilizada

Figura 6.78: Míssil AGM-114 Hellfire do exemplo 12.

das regiões supersônicas. A Fig. 6.87 mostra uma predição da região de interferência de arrasto produzida pelo escoamento em um míssil que possui 4 canards retos dispostos simetricamente. Esta região assemelha-se muito com a obtida, como é mostrada na região de $k = 0.19L$ para os canards frontais. Por fim, o histórico de convergência ao longo das iterações é mostrado na Fig. 6.88 e do coeficiente de arrasto na Fig. 6.89, onde a área de referência adotada corresponde à área projetada no plano xz . Os dados da malha ao longo das iterações são apresentados na Tab. 6.20.

A consideração no nariz de vento na região frontal certamente reduziria o arrasto elevado, bem como amenizaria as intensidades das ondas de choque nas regiões dos engates, da mesma forma que o primeiro engate já suaviza os efeitos para o segundo, absorvendo a maior parte da pressão. Tal estudo mais detalhado é um passo seguinte natural para este pequeno estudo apresentado.

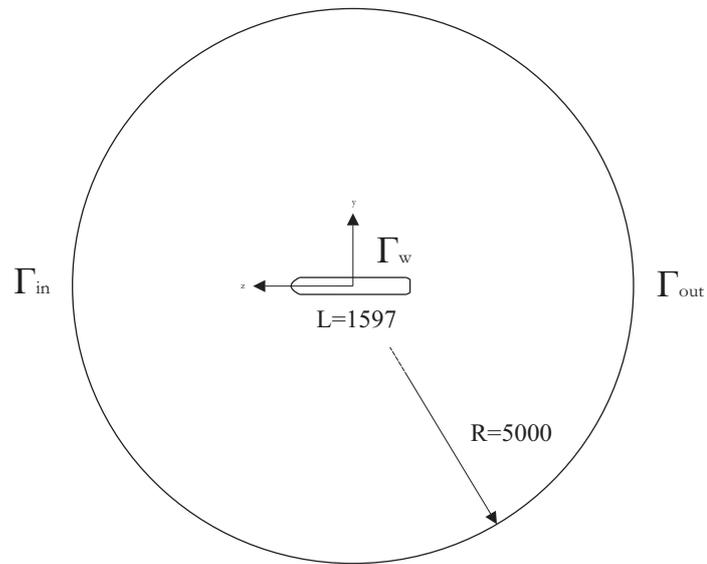


Figura 6.79: Condições de contorno do exemplo 12.

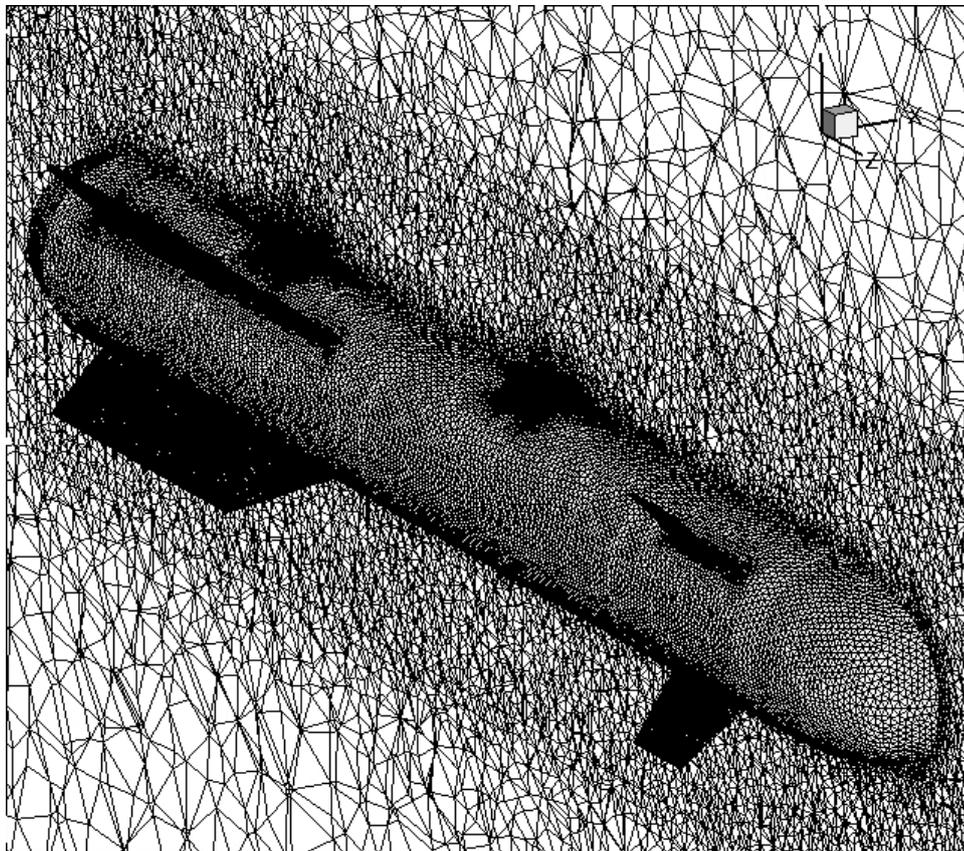


Figura 6.80: Malha Inicial do exemplo 12.

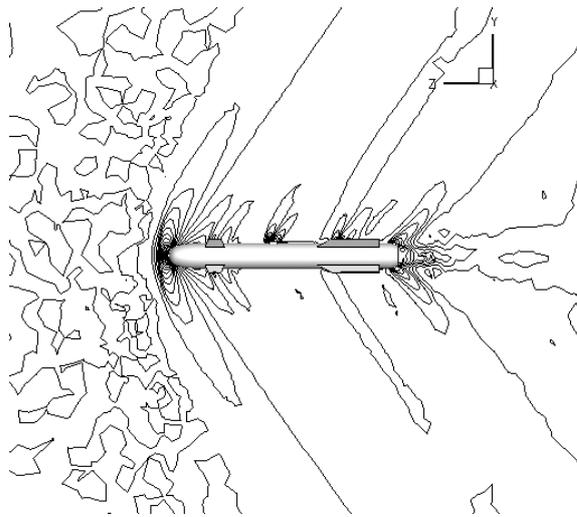
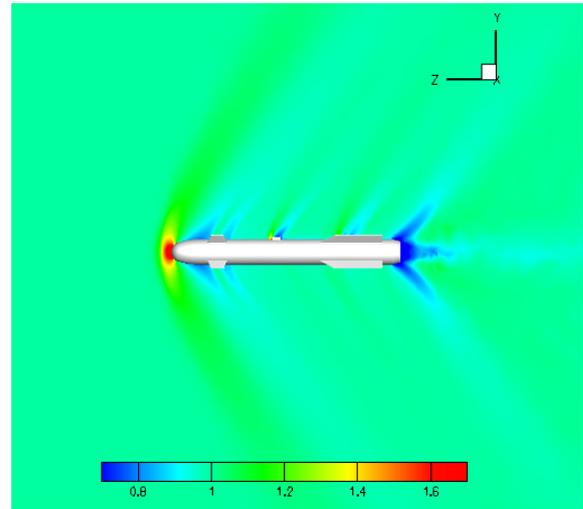
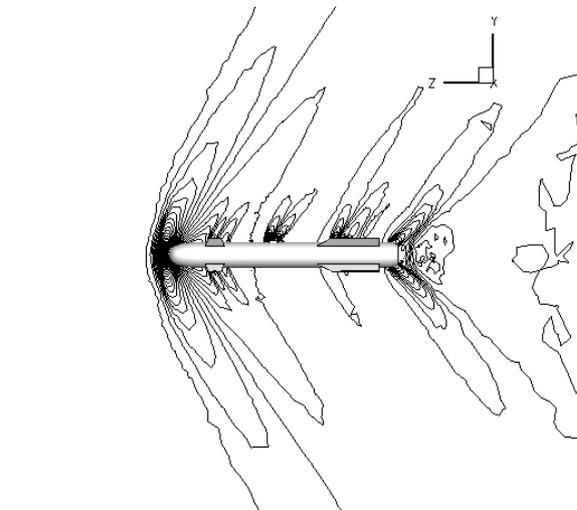
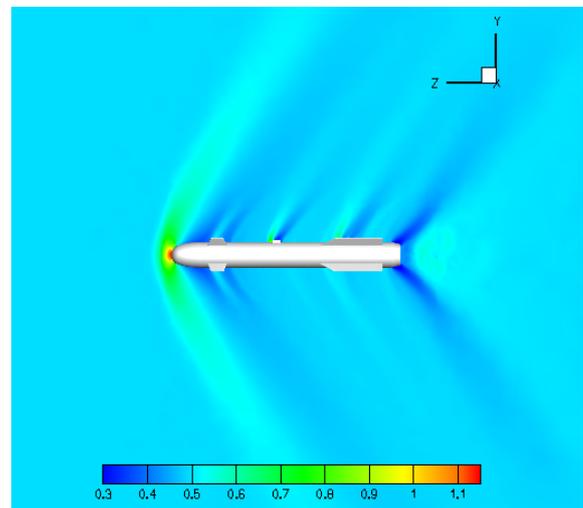
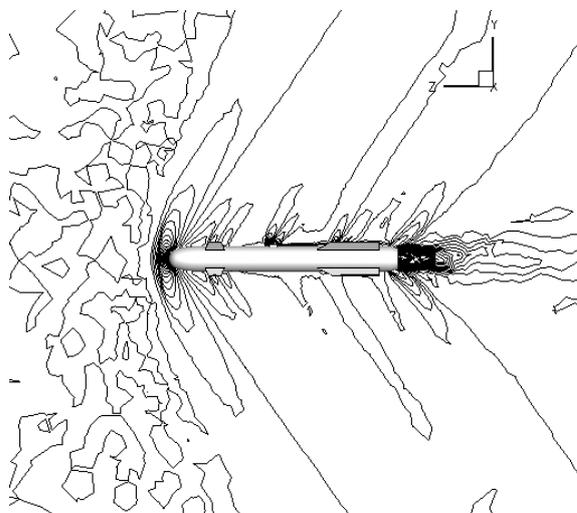
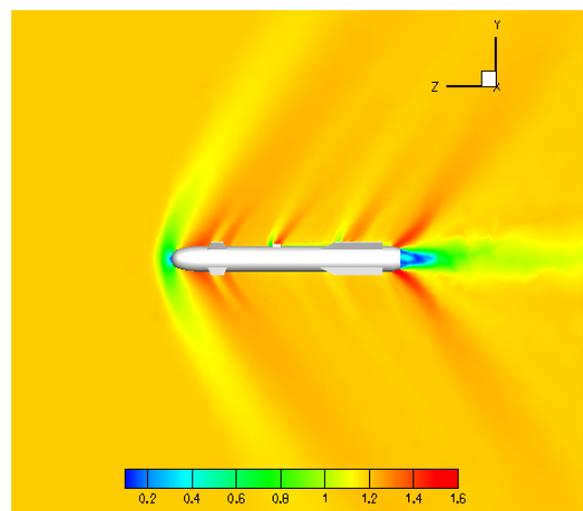
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.81: Solução inicial no plano yz com $x = 0$ do exemplo 12.

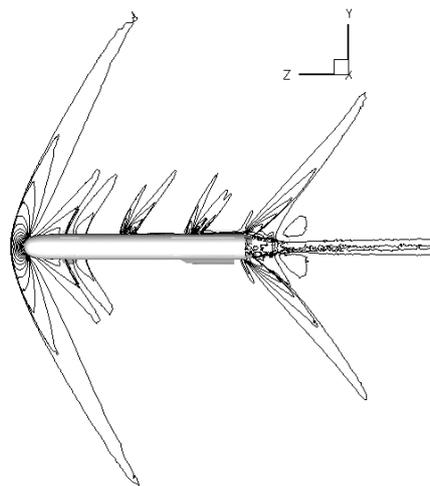
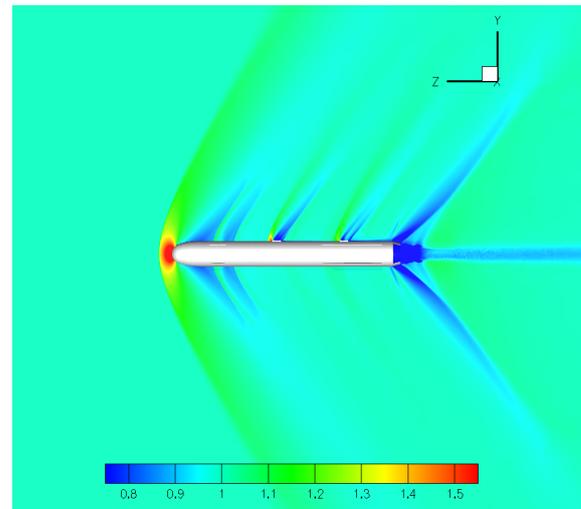
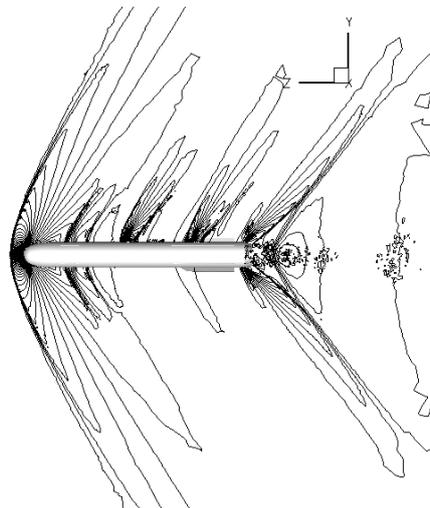
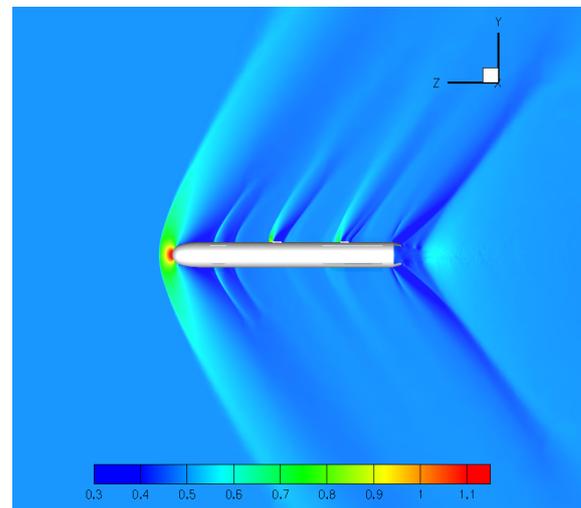
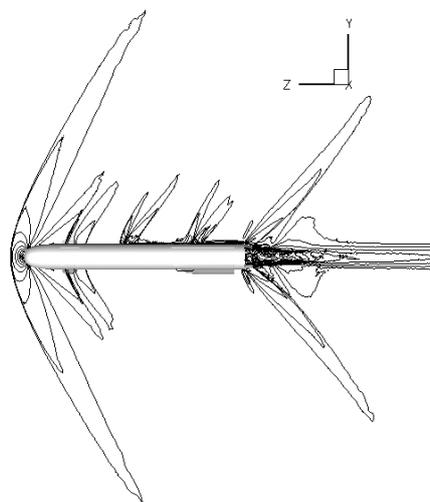
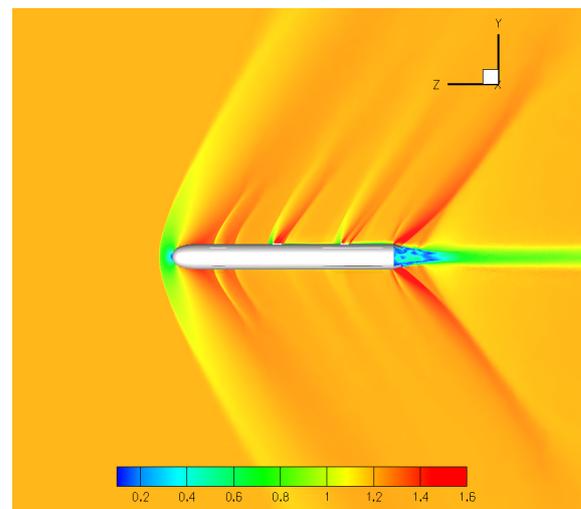
(a) Isocontornos de ρ (b) Campo de ρ (c) Isocontornos de p (d) Campo de p (e) Isocontornos de M (f) Campo de M

Figura 6.82: Solução adaptada final no plano yz com $x = 0$ do exemplo 12.

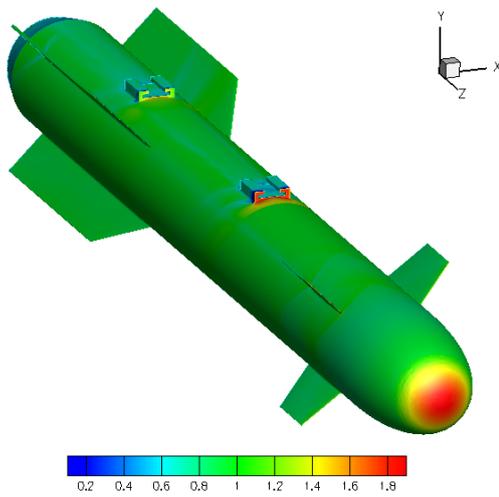
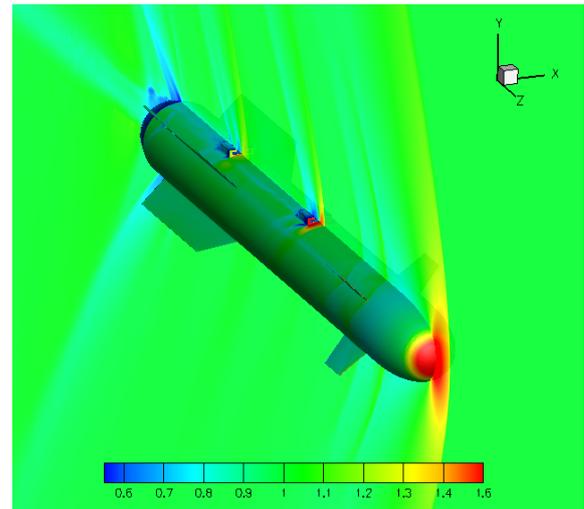
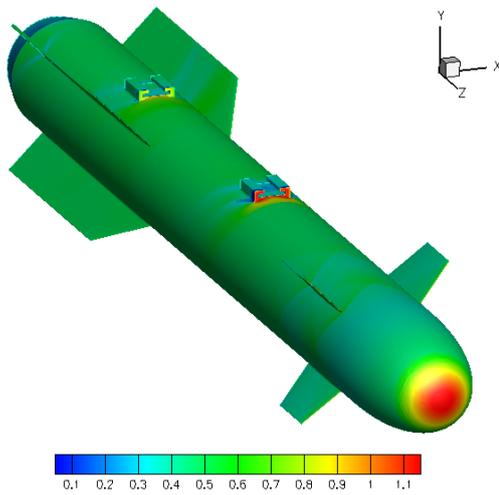
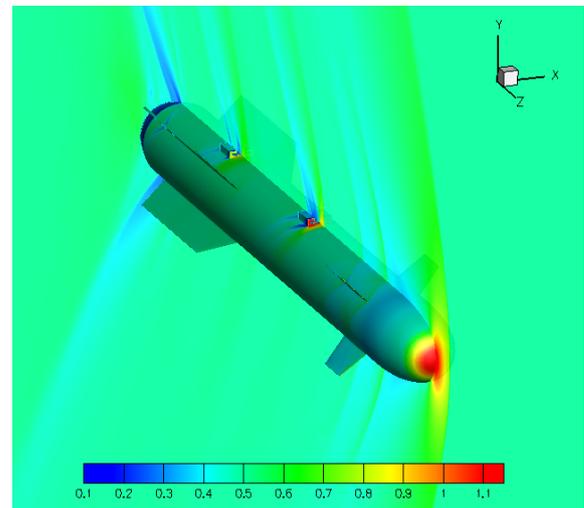
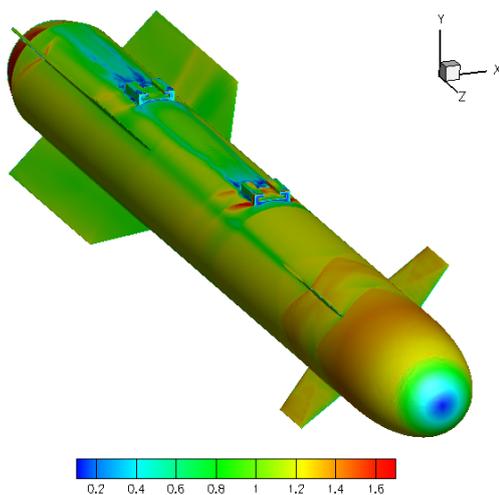
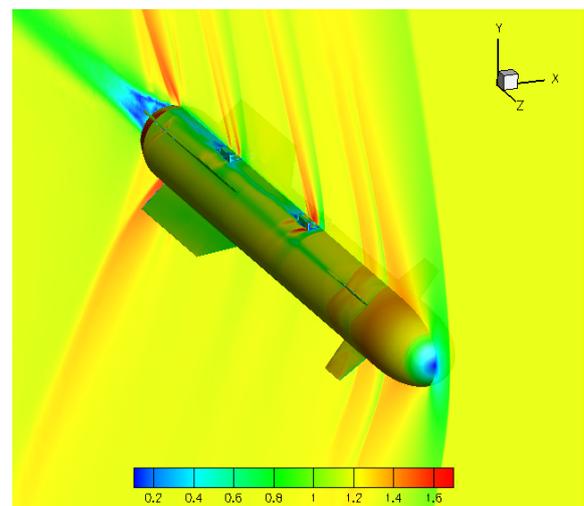
(a) Campos de ρ sobre o míssil(b) Campo de ρ em perspectiva(c) Campo de p sobre o míssil(d) Campo de p em perspectiva(e) Campo de M sobre o míssil(f) Campo de M em perspectiva

Figura 6.83: Campos da solução adaptada final sobre o míssil e em perspectiva do exemplo 12.

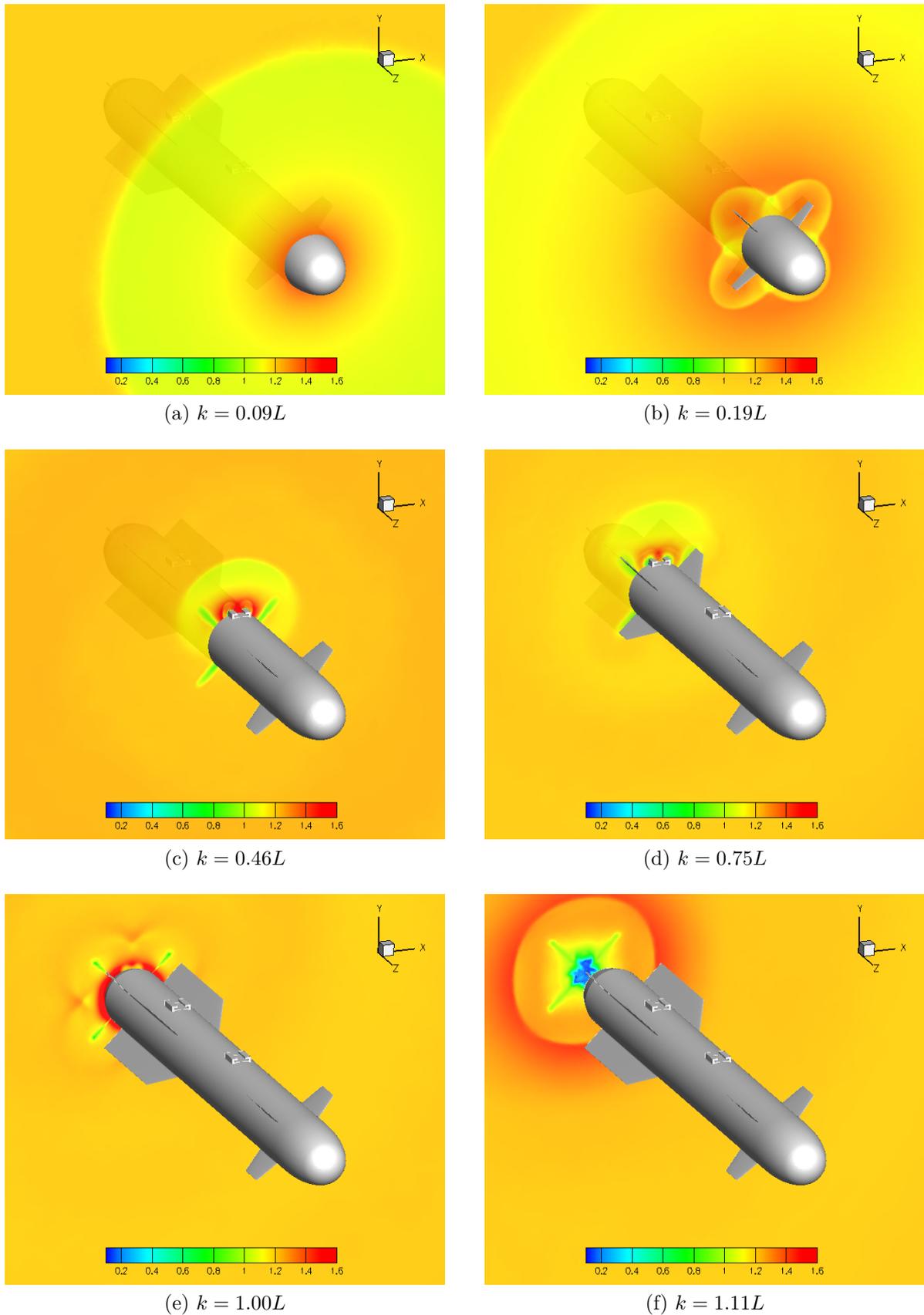
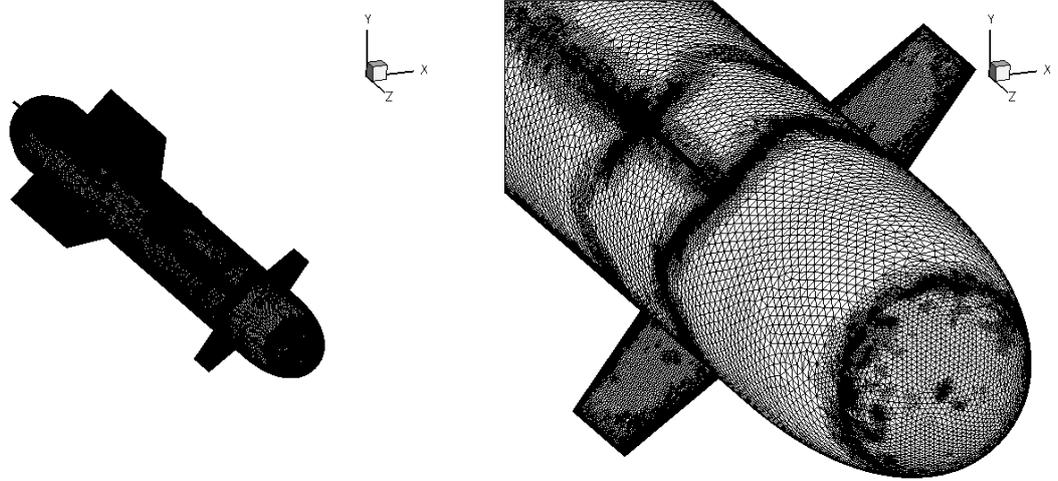
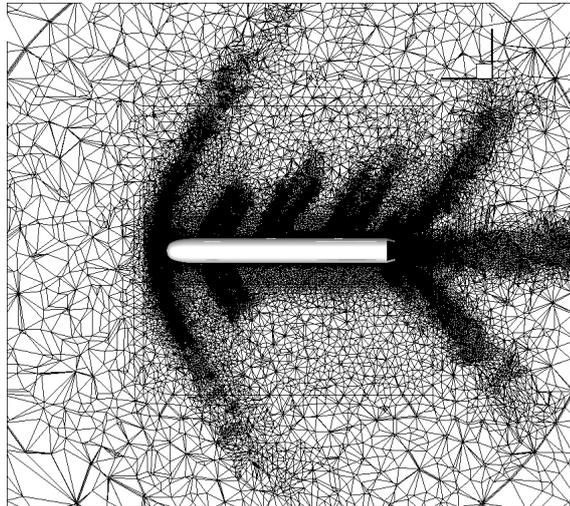
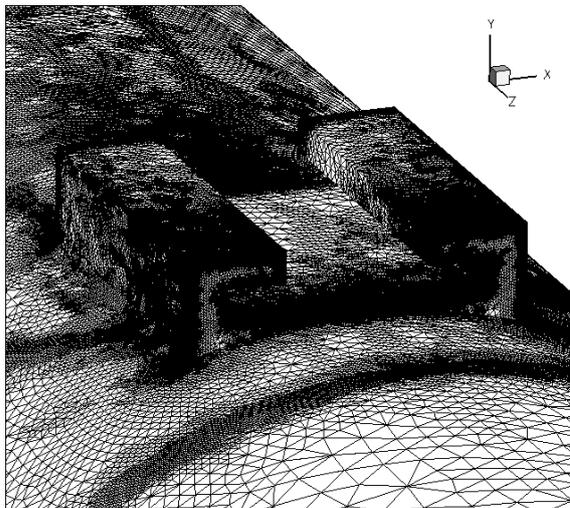


Figura 6.84: Cortes no plano xy para o campo de M do exemplo 12.

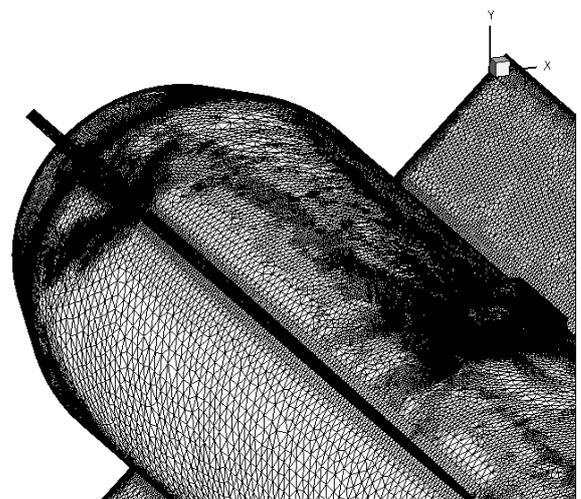


(a) Visão geral

(b) Detalhe da região frontal

(c) Vista lateral no plano yz , com $x = 0$ 

(d) Detalhe da região do engate



(e) Detalhe da região final

Figura 6.85: Malha final adaptada do exemplo 12.

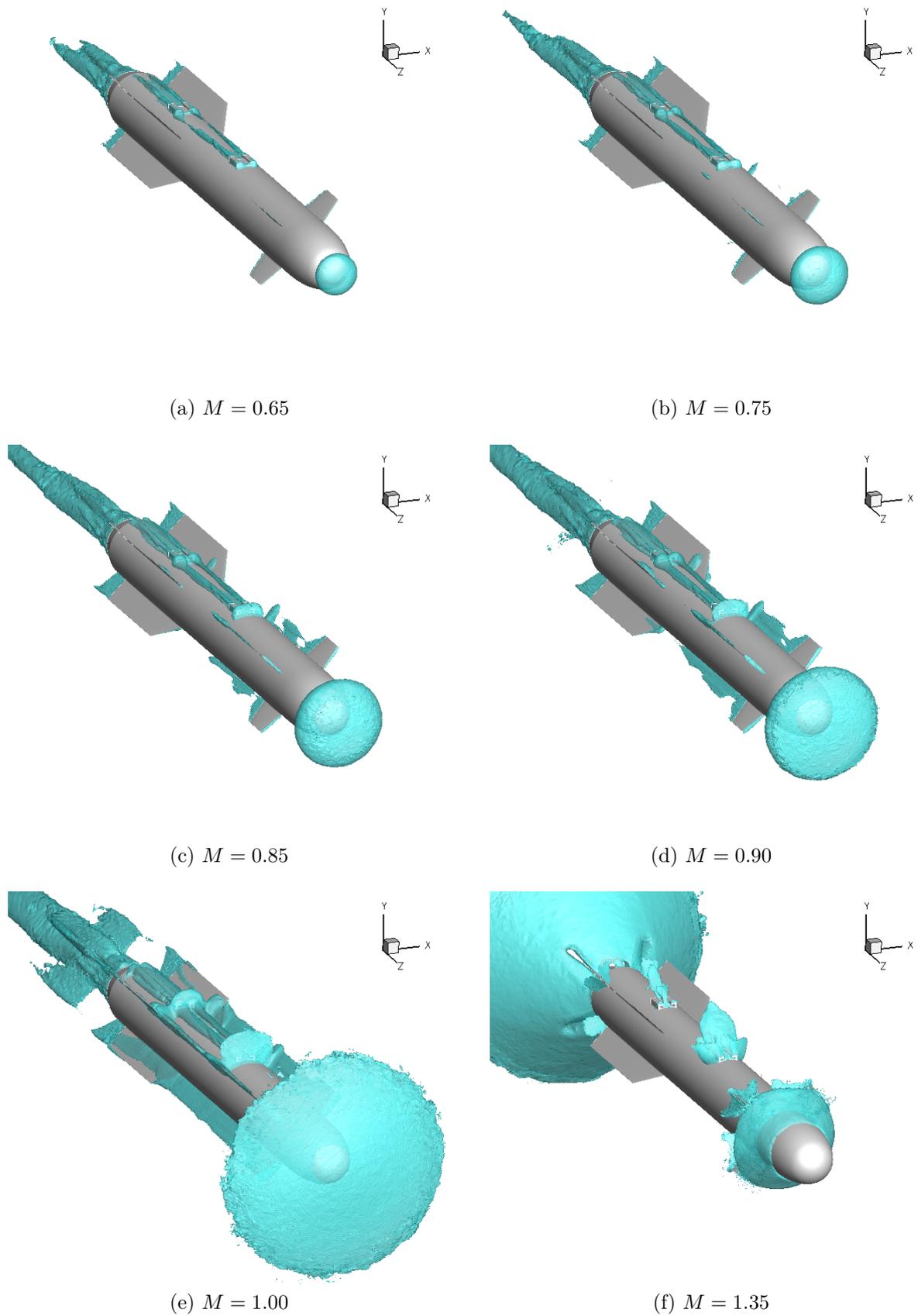


Figura 6.86: Isosuperfícies para de M do exemplo 12.

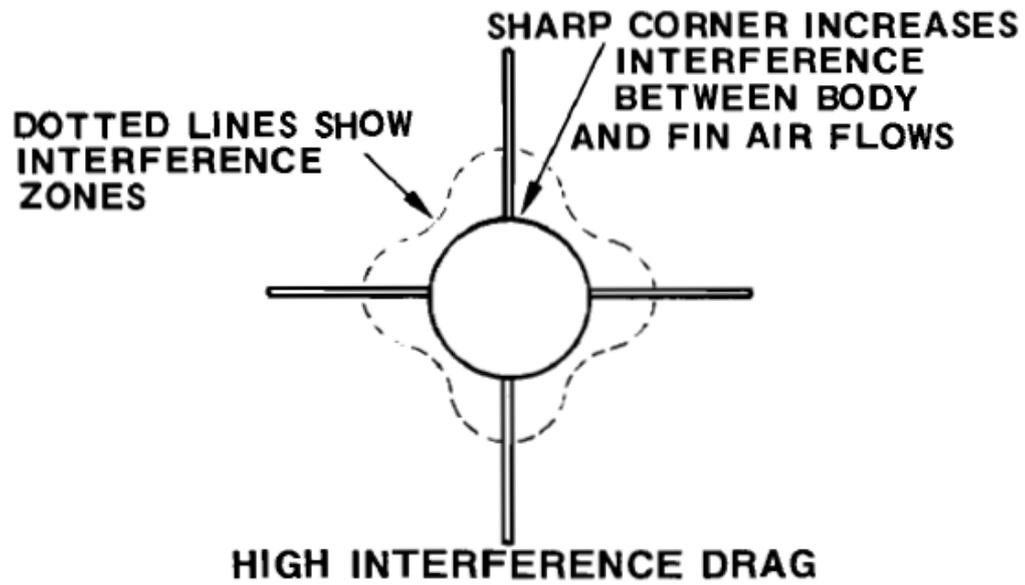


Figura 6.87: Regiões de interferência do arrasto para um míssil com 4 canards dispostos simetricamente (Gregorek, 1970).

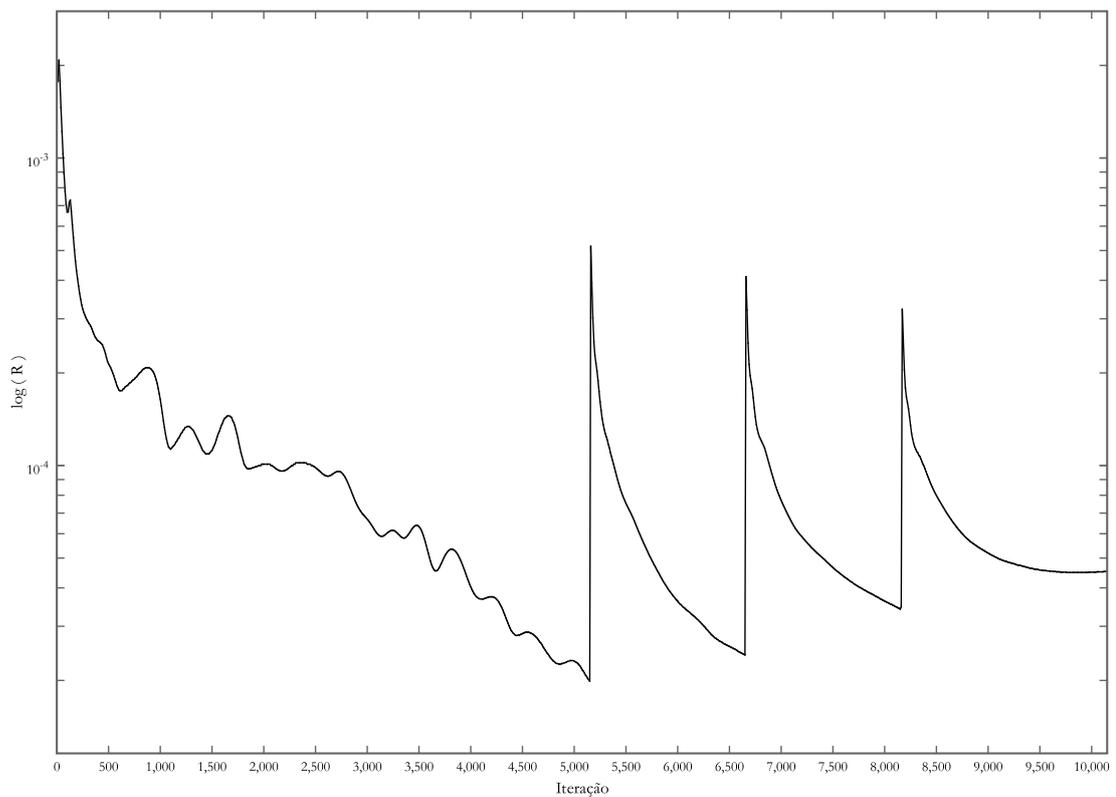


Figura 6.88: Convergência do exemplo 12.

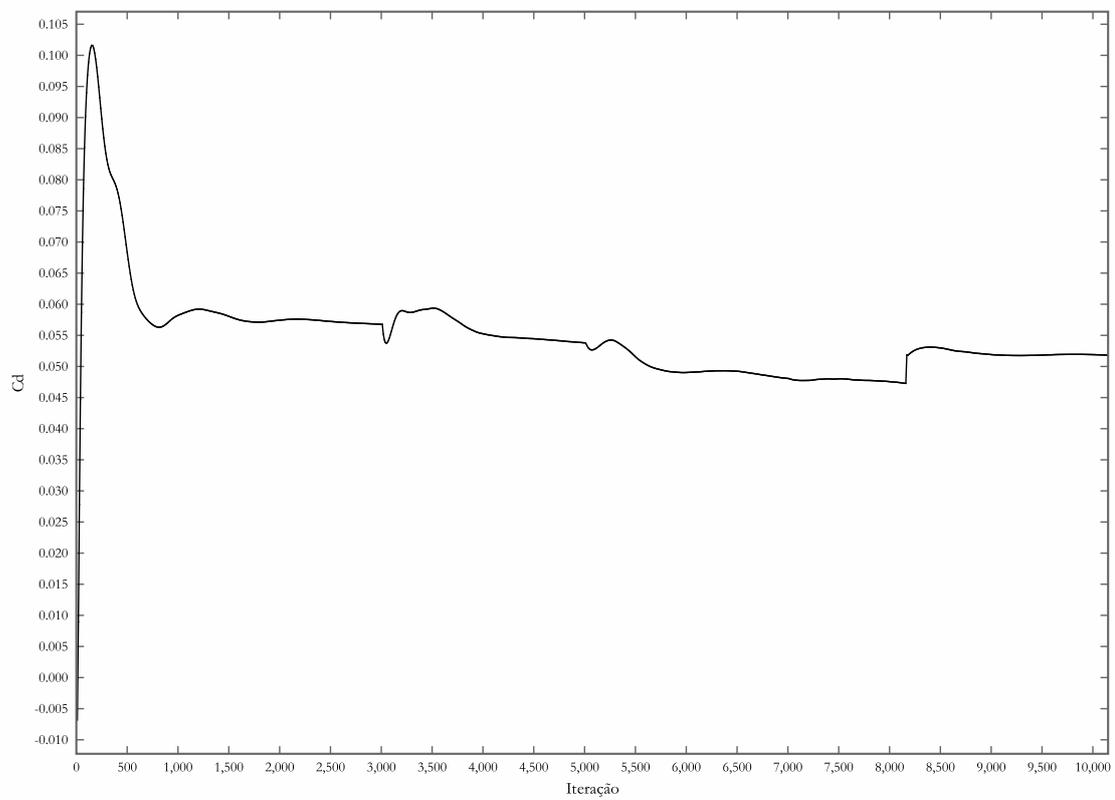


Figura 6.89: Coeficiente de arrasto C_D ao longo das iterações do exemplo 12.

6.2.7 Exemplo 13: Escoamento Supersônico Não-Viscoso Sobre Avião Lockheed SR-71

Também conhecido como *Blackbird*, o SR-71 é um avião de reconhecimento estratégico de longo alcance desenvolvido ao final da década de 60. Do ponto de vista aerodinâmico, o avião atinge velocidade sustentada acima de Mach 3 numa altitude superior à de 25km. Esta alta performance aerodinâmica para um avião ainda é detentora de recorde mundial até o presente momento. Entre as principais características aerodinâmicas presentes no SR-71 que permitem tal performance, pode-se citar o formato único do avião, sendo achatado no plano médio e não possuindo asas ou partes controladoras de vôo móveis; as turbinas de alta performance que utilizam da própria velocidade elevada desenvolvida para o funcionamento e também o material utilizado na fuselagem, de titânio, capaz de suportar as elevadas temperaturas e efeitos de fadiga termomecânica as quais são muito severas nesta faixa de velocidade sustentada, sendo impossível o uso de ligas metálicas e compósitos tradicionais utilizados em aeronáutica. O presente exemplo consiste em uma análise qualitativa do escoamento desenvolvido sobre um modelo geométrico aproximado do SR-71. A geometria real e o modelo utilizado são apresentados na Fig. 6.90. Foi simulado o avião em velocidade de Mach 3 com ângulo de ataque de 5° . Os dados empregados no escoamento são:

- $M_\infty = 3.0$
- $u_{1\infty} = 9.9619$
- $u_{2\infty} = 8.7155E - 2$
- $u_{3\infty} = 0.0$
- $CFL = 0.1$
- $C_E = 0.25$
- $R_T = 1 \times 10^{-5}$
- $T_\infty = 277K$

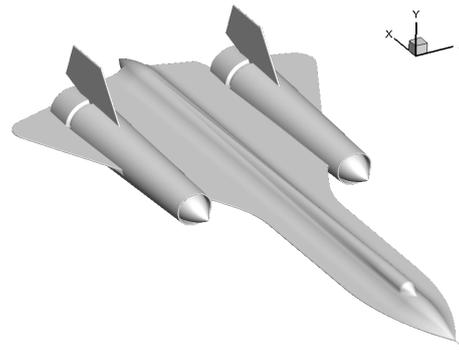
Os parâmetros de adaptação utilizados são:

Tabela 6.20: Dados das iterações de adaptação do exemplo 13

Adaptação	Nós	Elementos	α_{Dmax}	α_{D160}	$\ T_K\ _{min}$	$\sigma^2(E)$
0	696852	3316878	178.47	1.558E-3	5.180E-10	1.066E-2
1	1911234	9354508	179.18	2.441E-3	6.475E-11	7.978E-3
2	5469568	29150890	179.36	4.099E-3	8.094E-12	9.545E-3
3	19775680	111018517	179.65	6.266E-3	1.901E-12	8.265E-3



(a) Geometria real



(b) Geometria aproximada utilizada

Figura 6.90: Avião SR-71 do exemplo 13.

- $\phi = \rho$
- $\eta_S = 1.6$
- $\eta_I = 0.5$
- $N_F = 3$
- $F_{\Delta t} = 1500$

O problema é simulado sem simetria. A malha inicial empregada é mostrada na Fig. 6.91 e malha final adaptada na Fig. 6.92. A Fig. 6.93 mostra os campos de solução obtidos com a malha adaptada final em duas diferentes vistas. É possível observar que os efeitos de compressibilidade e mudança de velocidade são mais acentuados na ponteira e nas regiões no entorno das turbinas. O resto da fuselagem apresenta pouca variação no plano principal. O formato achatado do avião é o principal responsável por esta característica aerodinâmica relativa à configuração do escoamento sobre o mesmo. Por fim, os dados da malha ao longo das iterações são apresentados na Tab. 6.17.

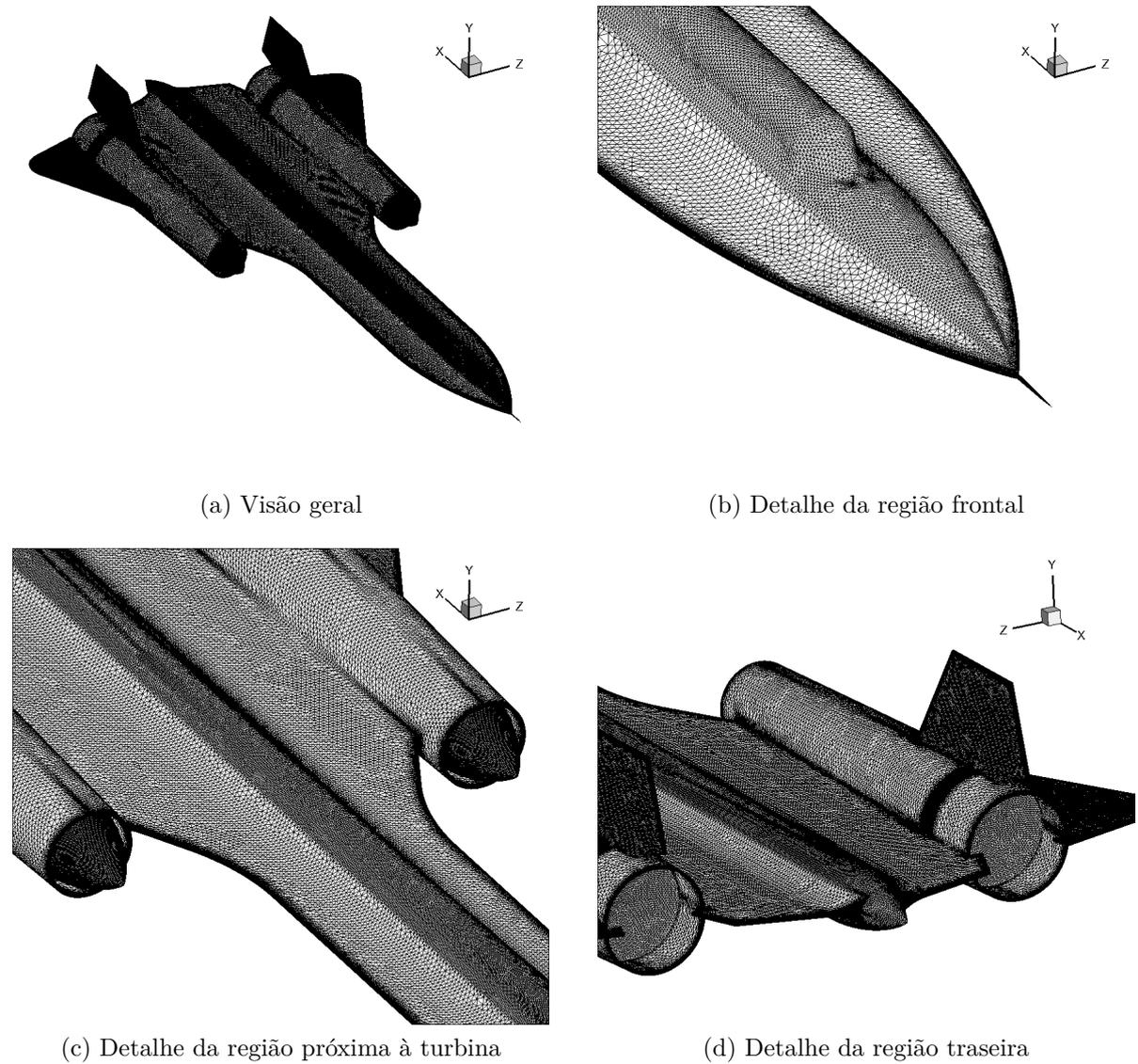


Figura 6.91: Malha inicial do exemplo 13.

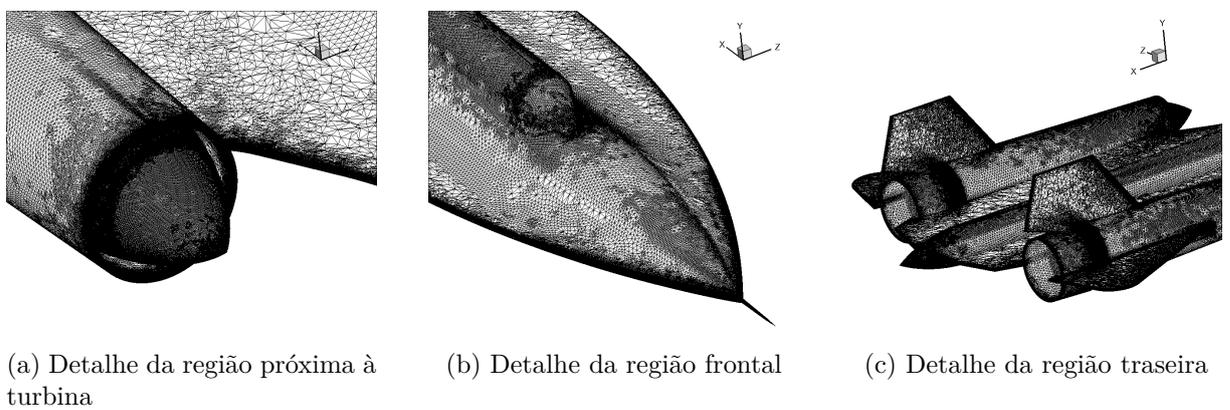


Figura 6.92: Malha adaptada final do exemplo 13.

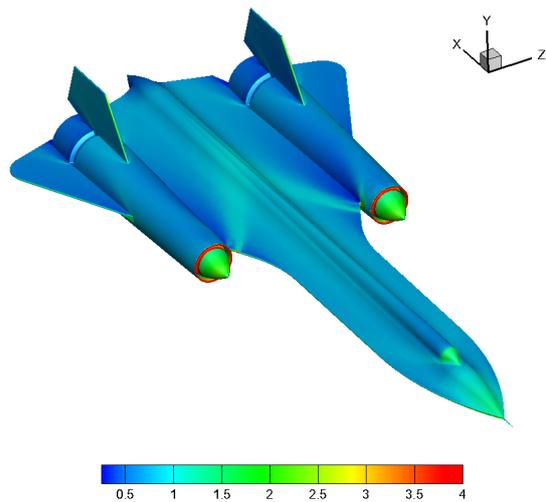
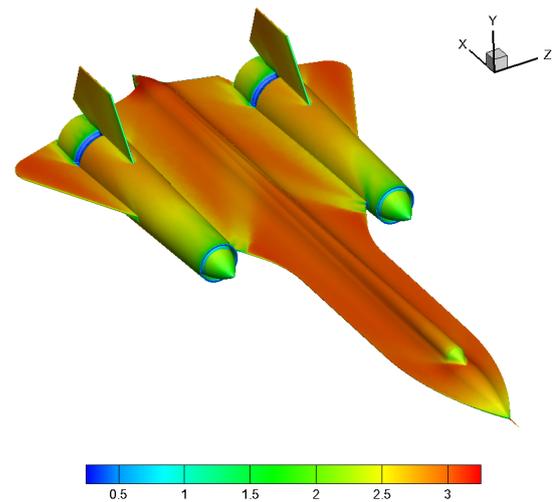
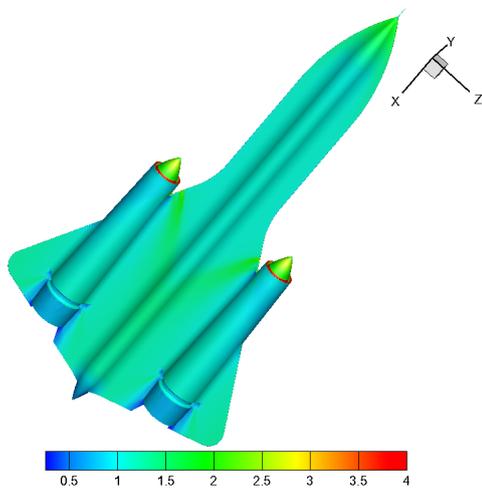
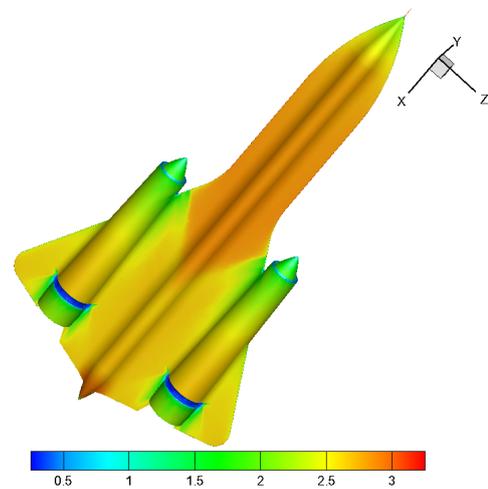
(a) Campo de ρ sobre o avião em vista superior(b) Campo de M sobre o avião em vista superior(c) Campo de ρ sobre o avião em vista inferior(d) Campos de M sobre o avião em vista inferior

Figura 6.93: Campo da solução adaptada final sobre o avião do exemplo 13.

Capítulo 7

Conclusões e Sugestões Para Trabalhos Futuros

A sùmula dos objetivos e resultados obtidos ao longo deste trabalho sao resumidos neste capítulo, assim como sugestões para trabalhos futuros.

7.1 Conclusões

Foi realizada a implementaçao de um codigo para simulaçao de escoamentos compressíveís tridimensionais utilizando o metodo CBS e a implementaçao de um algoritmo de adaptaçao anisotropica de malhas contendo elementos triangulares e tetraédricos lineares, bem como o acoplamento de tais implementaçoes. O algoritmo tridimensional e o bidimensional foram testados e validados para diversos estudos, incluindo regimes supersonicos, transonicos e subsonicos para casos viscosos e nao viscosos e para varias configuraçoes geométricas. Em todos os casos estudados, o algoritmo apresenta boa capacidade de predicao de soluçao em comparaçao com a literatura presente e estudos experimentais. A presente formulaçao adotada em conjunto com a metodologia de passos de tempo locais para escoamentos estacionários resultou em grande aceleraçao da convergência dos problemas. Para problemas de baixa compressibilidade, o metodo CBS apresenta performance superior quando comparado ao metodo de Taylor-Galerkin, sem apresentar oscilaçoes espúrias no entorno das regiões de estagnaçao. Convém comentar também que em muitos casos, embora o problema tenha sido convergido para um regime permanente, o coeficiente de arrasto ou sustentaçao nao acompanhou esta convergência

na mesma velocidade. Nestes casos, as simulações deveriam ter sido executadas por mais passos de tempo ou com uma tolerância menor de convergência para que se obtenha uma convergência mais adequada dos coeficiente aerodinâmicos.

A adaptação anisotrópica implementada para malhas com elementos triangulares lineares e tetraédricos lineares apresenta boa capacidade de inferência local sobre as regiões de gradientes elevados, bem como possibilidade de controle do nível de refinamento e desrefinamento através dos parâmetros que definem o limiar superior e inferior de erro aceitável. Entretanto, a implementação única de adaptação-h sem movimento de malhas (adaptação-r) reduz fortemente a capacidade de distorção local das malhas, pois o posicionamento dos nós é feito de forma discreta no primeiro caso ao passo que no segundo tem-se uma possibilidade de manejo quase que contínuo destas posições. Comprova-se também que as grandes distorções introduzidas na malha não comprometem a solução e convergência do método dos elementos finitos. Também a estimativa de erro implementada baseada na métrica Riemanniana apresenta boa capacidade de predição de direções de anisotropia do erro e de sua medida de intensidade. Tanto a estimativa de erro quanto a adaptação apresentaram boa performance computacional, com baixos custos envolvidos. Em especial, a capacidade de desrefinar qualquer região do domínio, não estando limitado à uma malha inicial, resulta em grande aplicabilidade principalmente em problemas transientes.

O acoplamento da adaptação de malhas com o código CBS apresentou resultados de qualidade superior aos obtidos com uma solução inicial do problema. Para todos os casos estudados obteve-se melhoras significativas da resolução do escoamento bem como predição de coeficientes aerodinâmicos. Para o problema transiente estudado, a maleabilidade da densidade da malha se torna evidente, onde a adaptação consegue acompanhar o desenvolvimento do escoamento ao longo do tempo.

A paralelização para computadores com memória compartilhada implementada para o código CBS tridimensional resultou em ganhos consideráveis, sem os quais tais simulações tornam-se inviáveis devido à grande quantidade de dados envolvidos. Entretanto, tal performance carece de maior otimização ainda, a fim de possibilitar a utilização de malhas robustas em menores intervalos de tempo.

Por fim, os diversos estudos realizados utilizando adaptação de malhas possibilitaram analisar diversas características dos escoamentos as quais são observáveis apenas quando a resolução da malha é suficiente. Em geometrias mais complexas, como a do míssil estudado, tal resolução da malha é de difícil obtenção sem adaptação local de malhas devido às diversas irregularidades da localidade das regiões que possuem elevados gradientes. Como mostrado para o escoamento viscoso sobre o aerofólio NACA 0012, até mesmo regiões que possam parecer estar bem representadas e de boa resolução em uma solução inicial não adaptada podem ter grandes variações quando comparadas ao resultado obtido com a malha adaptada.

7.2 Sugestões Para Trabalhos Futuros

Dentre as diversas sugestões pra trabalhos futuros citam-se:

Quanto ao código CBS:

- Para possibilitar simular escoamentos viscosos em regimes com Reynolds mais altos com precisão, deveria-se implementar um modelo de turbulência acoplado;
- Calcular o tamanho característico dos elementos na direções das linhas de corrente;
- Possibilitar simular escoamentos não-isentrópicos, considerando reações químicas, excitação vibracional das moléculas e desequilíbrio térmico, possibilitando simular escoamentos hipersônicos;
- Implementar o código baseado em arestas ao invés de baseado em elementos para aumentar eficiência computacional;
- Implementar a técnica de *Multigrid* para acelerar a convergência de problemas estacionários;
- Implementar a compressibilidade artificial ou metodologia análoga para possibilitar simular escoamentos incompressíveis;
- Incorporar a elasticidade dos sólidos no escoamento, possibilitando a análise

aeroelástica acoplada do problema através de um modelo de interação fluido-estrutura;

- Implementar a incorporação de efeitos acoplados de ondas curtas tais como as provenientes de eletromagnetismo.

Quanto à adaptação de malhas e análise de erros:

- Implementar a adaptação-r conjuntamente;
- Implementar metodologias que melhorem equidistribuição do erro e promovam maiores distorções na malha;
- Acoplar ferramentas CAGD (*Computed-Aided Geometric Design*) que descrevam parametricamente a geometria do problema, de forma que o refinamento possa aumentar a suavidade da geometria original;
- Estudar a possibilidade de intersecção de diversas métricas, possibilitando a análise de erro baseada de múltiplas variáveis;
- Implementar a paralelização da adaptação utilizando diversas frentes de trabalho que não se interpoem no domínio.

Quanto à paralelização:

- Otimizar mais o código paralelizado em OMP;
- Implementar o código em MPI em conjunto com o OMP, possibilitando maiores ganhos computacionais.

Referências Bibliográficas

- Babuska, I. e Aziz, A. K. (1976). On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, Vol. 13, No. 2, pp. 214–226.
- Bailey, A. B. e Hiatt, J. (1971). Free-flight measurements of sphere drag at subsonic, transonic, supersonic, and hypersonic speeds for continuum, transition, and near-free-molecular flow conditions. Rep. aedc-tr-70-291, Arnold Engng. Development Ctr., Arnold Air Force Station, TN, Rep., Massachusetts.
- Bassi, F., Crivellini, A., Pietro, D. A. D., e Rebay, S. (2006). A high-order discontinuous galerkin solver for 3d aerodynamic turbulent flows. Em *Proceedings of the European conference on Computational Fluid Dynamics*, Eccomas CFD, pgs. 1–41.
- Bono, G. (2008). *Simulação Numérica de Escoamentos em Diferentes Regimes Utilizando o Método dos Elementos Finitos*. PhD thesis, Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- Bottasso, C. L. (2004). Anisotropic mesh adaption by metric-driven optimization. *International Journal for Numerical Methods in Engineering*, Vol. 60, No. 3, pp. 597–639.
- Bourguet, R., Braza, M., Dervieux, A., e Sévrain, A. (2007). Transitions features in transonic flow around a naca0012 airfoil by navier-stokes and low-order modelling. Em *West-East High Speed Flow Field Conference, 2007*, 19-22 November of 2007, pgs. 1–13.
- Brandts, J., Hannukainen, A., Korotov, S., e Krížek, M. (2011). On the angle condition in the finite element method. *SeMA Journal (Boletín de la Sociedad Española de Matemática Aplicada)*, Vol. 56, No. 1, pp. 81–95.
- Brandts, J., Korotov, S., e Krížek, M. (2008). On the equivalence of regularity criteria for triangular and tetrahedral finite element partitions. *Computers and Mathematics with Applications*, Vol. 55, No. 10, pp. 2227–2233.
- Carmo, M. (1976). *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1st edition.
- Castro-Díaz, M. J., Hecht, F., Mohammadi, B., e Pironneau, O. (1997). Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 4, pp. 475–49.
- Chapman, B., Jost, G., e van van der Pas, R. (2007). *Using OpenMP: Portable Shared Memory Parallel Programming*. MIT Press, 1st edition.
-

-
- Chapman, S. (2007). *Fortran 95/2003 for Scientists & Engineers*. McGraw-Hill, 3rd edition.
- Choi, C. e Merkle, C. L. (1985). Application of time-iterative scheme to incompressible flow. *AIAA Journal*, Vol. 23, No. 10, pp. 1518–1524.
- Courant, R., Friedrichs, K., e Lewy, H. (1928). On the partial difference equations of mathematical physics. *Mathematische Annalen*, Vol. 100, No. 1, pp. 32–71.
- Cutler, B., Dorsey, J., e McMillan, L. (2004). Simplification and improvement of tetrahedral models for simulation. Em *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pgs. 93–102.
- Dolejší, V. e Felcman, J. (2004). Anisotropic mesh adaptation for numerical solution of boundary value problems. *Numerical Methods for Partial Differential Equations*, Vol. 20, No. 4, pp. 576–608.
- Fleeman, E. L. (2001). *Tactical Missile Design*. American Institute of Aeronautics and Astronautics (AIAA) Series, 1st edition.
- Freitag, L. A. e Ollivier-Gooch, C. (1997). Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 1, pp. 3979–4002.
- Gregorek, G. M. (1970). Aerodynamic drag of model rockets. Tr-11 model rocket technical report, Estes Industries. Inc., Penrose, Colorado.
- Gruau, C. (2004). *Génération de Métriques Pour Adaptation Anisotrope de Maillages, Applications à la Mise en Forme des Matériaux*. PhD thesis, École Nationale Supérieure des Mines de Paris.
- Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M., e Vallet, M. G. (2000). Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver independent cfd. part i: General principles. *International Journal for Numerical Methods in Fluids*, Vol. 32, No. 6, pp. 725–744.
- Hager, G. e Wellein, G. (2011). *Introduction to High Performance Computing for Scientists and Engineers*. Chapman & Hall, 1st edition.
- Hirsch, C. (1988). *Numerical Computation of Internal and External Flows. Volume 1, Fundamentals of Numerical Discretization*. Wiley, 1st edition.
- Katz, A. (2009). *Meshless Methods for Computational Fluid Dynamics*. PhD thesis, Stanford University.
- Křízek, M. (1991). On semiregular families of triangulations and linear interpolation. *Applied Mathematics*, Vol. 36, No. 1, pp. 223–232.
- Křízek, M. (1992). On the maximum angle condition for linear tetrahedral elements. *SIAM Journal on Numerical Analysis*, Vol. 29, No. 1, pp. 513–520.
- Löhner, R. (2008). *Applied CFD Techniques. An Introduction Based on Finite Element Methods*. Wiley, 2th edition.

- Loiselle, A. (2008). *Adaptation de Maillage Anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides: Application à la prédiction haute-fidélité du bang sonique*. PhD thesis, Université Pierre et Marie Curie, Paris VI.
- Loseille, A. e Alauzet, F. (2011). Continuous mesh framework - part i: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, pp. 38–60.
- Malvern, L. (1969). *Introduction to the Mechanics of a Continuous Medium*. Prentice Hall, 1st edition.
- Masuero, J. R. (2009). *Computação Paralela na Análise de Problemas de Engenharia Utilizando o Método dos Elementos Finitos*. PhD thesis, Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- Mavriplis, D. J. (1991). Three dimensional unstructured multigrid for the euler equations. Nasa contractor report no. nas1-18605; icase report no. 91-41, NASA, Hampton, Virginia.
- Miller, D. G. e Bailey, A. B. (1979). Sphere drag at mach numbers from 0.3 to 20 at reynolds numbers approaching 10^7 . *Journal of Fluid Mechanics*, Vol. 93, No. 3, pp. 449–464.
- Nazarov, M. e Hoffman, J. (2012). An adaptative finite element method for inviscid compressible flow. *International Journal for Numerical Methods in Fluids*, Vol. 64, No. 10, pp. 1102–1128.
- Nithiarasu, P., Codina, R., e Zienkiewicz, O. C. (2006). The characteristic-based split (cbs) scheme - a unified approach to fluid dynamics. *International Journal for Numerical Methods in Engineering*, Vol. 66, No. 10, pp. 1514–1546.
- Nithiarasu, P., Zienkiewicz, O. C., Satya-Sai, B. V. K., Morgan, K., e Vázquez, M. (1998). Shock capturing viscosities for the general fluid mechanics algorithm. *International Journal for Numerical Methods in Fluids*, Vol. 28, No. 9, pp. 1325–1353.
- Pain, C. C., Umpleby, A. P., Oliveira, C. R. E., e Goddard, A. J. H. (2001). Tetrahedral mesh optimization and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 29, pp. 3371–3796.
- Ruprecht, D. e Müller, H. (1994). A scheme for edge-based adaptive tetrahedron subdivision. Technical report, Universität Dortmund, Dortmund, Germany.
- Schlichting, H. (1979). *Boundary-Layer Theory*. McGraw-Hill, 6th edition.
- Schmitt, V. e Charpin, F. (1979). Pressure distributions on the onera-m6-wing at transonic mach numbers. Report of the fluid dynamics panel working group 04, AGARD, AR 138.
- Spearman, M. L. e Brawell, D. O. (1993). Aerodynamics of a sphere and an oblate spheroid for mach numbers from 0.6 to 10.5 including some effects of test conditions. Nasa technical memorandum im-109016, NASA, Hampton, Virginia.

- Tam, A., Ait-Ali-Yahia, D., Robichaud, M. P., Moore, M., Kozel, V., e Habashi, W. G. (2000). Anisotropic mesh adaptation for 3d flows on structured and unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, No. 4, pp. 1025–1230.
- Teixeira, P. R. F. (2001). *Simulação Numérica da Interação de Escoamentos Tridimensionais de Fluidos Compressíveis e Incompressíveis e Estruturas Deformáveis Usando o Método dos Elementos Finitos*. PhD thesis, Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- Thomas, C. G. e Nithiarasu, P. (2005). Influences of element size and variable smoothing on inviscid compressible flow solution. *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 15, No. 5, pp. 420–428.
- Trotts, I. J., Hamann, B., Joy, K. I., e Wiley, D. F. (1998). Simplification of tetrahedral meshes. Em *Proceedings of the conference on Visualization '98, VIS '98*, pgs. 287–295.
- van Dam, C. P. (1999). Recent experience with different methods of drag prediction. *Progress in Aerospace Sciences*, Vol. 35, No. 1, pp. 751–798.
- Vassberg, J. C., DeHaan, M. A., Rivers, S. M., e Wahls, R. A. (2008). Development of a common research model for applied cfd validation studies. Em *Proceedings of the AIAA Applied Aerodynamics Conference*, AIAA Paper 2008-6919, pgs. 1–22.
- White, F. W. (1991). *Viscous Fluid Flow*. McGraw-Hill, 2th edition.
- Zienkiewicz, O. C., Morgan, K., e Satya, B. V. K. S. (1995). A general algorithm for compressible and incompressible flow - part ii: Tests on the explicit form. *International Journal for Numerical Methods in Fluids*, Vol. 20, No. 1, pp. 887–913.
- Zienkiewicz, O. C., Taylor, R. L., e Nithiarasu, P. (2011). *The Finite Element Method For Fluid Dynamics*. Butterworth-Heinemann, 6th edition.
- Zlámal, M. (1968). On the finite element method. *Numerische Mathematik*, Vol. 12, No. 1, pp. 394–409.