

JOÃO MARCELO VEDOVOTO

**MODELAGEM MATEMÁTICA E SIMULAÇÃO NUMÉRICA
DE ESCOAMENTOS INCOMPRESSÍVEIS SOBRE
GEOMETRIAS COMPLEXAS TRIDIMENSIONAIS
UTILIZANDO O MÉTODO DA FRONTEIRA IMERSA**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

JOÃO MARCELO VEDOVOTO

**MODELAGEM MATEMÁTICA E SIMULAÇÃO NUMÉRICA DE
ESCOAMENTOS INCOMPRESSÍVEIS SOBRE GEOMETRIAS
COMPLEXAS TRIDIMENSIONAIS UTILIZANDO O MÉTODO DA
FRONTEIRA IMERSA**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **MESTRE EM ENGENHARIA MECÂNICA**.

Área de Concentração: Transferência de Calor e Mecânica dos Fluidos.

Orientador: Prof. Dr. Aristeu da Silveira Neto

**UBERLÂNDIA – MG
2007**

Dados Internacionais de Catalogação na Publicação (CIP)

V416m Vedovoto, João Marcelo, 1981-
Modelagem matemática e simulação numérica de escoamentos incompressíveis sobre geometrias complexas tridimensionais utilizando o Método da Fronteira Imersa / João Marcelo Vedovoto. - 2007.
125 f. : il.

Orientador: Aristeu da Silveira Neto.

Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Mecânica.
Inclui bibliografia.

1. Dinâmica dos fluidos - Teses. 2. Simulação (Computadores) - Teses.
I. Silveira Neto, Aristeu da. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

CDU: 532.51

Agradecimentos

Ao meu orientador e amigo Aristeu, por ser sempre um exemplo, tanto como ser humano, como profissional.

À minha noiva Karina, pelo apoio e pelo enorme companheirismo, mesmo nos momentos em que eu estava mais distante. Obrigado pela sua paciência, pelo seu amor e por sempre acreditar em mim.

Aos meus amigos, Felipe, Sigeo, Leonardo, Tiago, Elie, Gustavo, Millena, Priscila, Professor Gilmar e a todos os outros membros do LTCM, que de uma maneira ou de outra foram indispensáveis para a realização deste trabalho.

Aos meus amigos Rubens Campregher e José dos Reis Vieira de Moura Jr, pelas inestimáveis idéias e ajudas quando os problemas pareciam não ter mais solução.

Ao meu pai José Vedovoto e a minha mãe Maria Aparecida, por trabalharem incansavelmente para que eu pudesse sempre me concentrar em meus estudos. Obrigado também pelo exemplo de honestidade, trabalho, integridade que são.

Ao meu irmão Marcos por ser também um exemplo de ser humano e por ser também um exemplo de amor ao conhecimento.

À minha irmã Graciela, que sempre foi e sempre será uma das pessoas mais importantes da minha vida. Obrigado por ser uma pessoa tão iluminada, e sempre uma inspiração para mim.

À Massao, Emília, Suzana e Patrícia, que também sempre me apoiaram e acreditaram em mim.

À Anteia e Frida, que por mais incrível que pareça, foram importantíssimas para o desenvolvimento e conclusão deste mestrado.

Ao meu amigo Alexandre Santana (*in memorian*), meu tio Dino Vedovotto (*in memorian*), e pelo meu avô Francisco (*in memorian*), por, cada um à sua maneira, sempre terem me apoiado em minhas decisões.

À CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior por financiar meus estudos junto ao Programa de Pós-Graduação em Engenharia Mecânica (POSMEC-UFU) onde tive todo suporte necessário para realização dos meus trabalhos.

À Deus e todos os bons espíritos que me acompanham nesta minha jornada, sempre me dando serenidade para discernir entre o certo e o errado, e me dando forças para seguir o caminho do bem.

Vedovoto, J. M., 2007. "Modelagem Matemática e Simulação Numérica de Escoamentos Incompressíveis sobre Geometrias Complexas Tridimensionais Utilizando o Método da fronteira imersa", Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.

Resumo

A necessidade de compreender os mecanismos físicos que estão presentes em escoamentos sobre geometrias complexas é algo imperativo na engenharia moderna. Desta maneira, as ferramentas que possibilitem a análise e o entendimento de tais escoamentos são cada vez mais importantes. Apresenta-se neste trabalho uma ferramenta numérica que possibilita a análise transiente de escoamentos sobre diversas geometrias complexas tridimensionais: o código Fluids-3D. Este código utiliza o Método da fronteira imersa, como metodologia de representação de corpos imersos nos escoamentos. Este método utiliza dois domínios independentes, porém acoplados para solução dos escoamentos sobre geometrias complexas: um domínio lagrangiano para representar a interface sólido / fluido e um domínio euleriano para discretizar o domínio fluido. As equações de Navier-Stokes são resolvidas para o domínio euleriano com uma precisão de segunda ordem no tempo e no espaço, empregando o Método dos volumes finitos em malhas cartesianas. A malha lagrangiana é discretizada por uma malha de elementos triangulares e pode ser importada de geradores de malhas comerciais ou geradores de arquivos para estereolitografia (*.stl - *Standard Tessellation Language*). No Método da fronteira imersa, os efeitos de um corpo imerso ao escoamento são impostos inserindo um termo fonte nas equações de Navier-Stokes. Neste trabalho utiliza-se o Modelo físico virtual para modelagem deste termo fonte. Tal modelo está sendo desenvolvido no LTCM - Laboratório de Transferência de Calor e Massa e Dinâmica dos Fluidos. O código conta ainda com capacidade de processamento paralelo. São apresentados resultados de escoamentos sobre diversos tipos de geometrias complexas, tais como cubos solidários à base do domínio, protótipos de automóveis e de aeronaves, proporcionando ricas análises dos respectivos escoamentos. Outras contribuições importantes do presente trabalho são a preparação do código Fluids-3D para a implementação de uma interface gráfica de usuário, bem como otimização da topologia de paralelização do mesmo.

Palavras Chave: Método de Fronteira Imersa, Modelo físico virtual, Escoamentos sobre Geometrias Complexas, Processamento Paralelo.

Vedovoto, J. M., 2007. "Mathematical modeling and Numerical simulation of incompressible flows over complexes three-dimensional geometries using the Immersed Boundary Method", Master Thesis, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.

Abstract

The necessity of understand the physical mechanisms in flows over complexes geometries are something imperative on modern engineering. In this way, tools that allow the analysis and understanding of such flows are very important as well. In this work, a numerical tool that allows the transient analysis of flows over several types of complexes three-dimensional geometries is presented: the Fluids-3D code. Such code uses the Immersed Boundary Method for representing an immersed body into the flow. The Immersed Boundary method uses two independent domains in the solution of the flows over complex geometries: an eulerian domain, which is discretized using Finite Volume Method over a non-uniform mesh to integrate the Navier-Stokes equations, and a second-order approximation for time and space derivatives; the lagrangian domain is represented by a superficial unstructured mesh, composed by triangles. Such mesh can be imported by commercial mesh generators, or generators of files used in stereolithography (**.stl - Standard Tessellation Language*). In the Immersed Boundary Method, the effects of an immersed body are imposed by a source term in the Navier-Stokes equations. In this work the Virtual Physical Model for modeling this term is used. Such model is in development in LTCM – Laboratory of Heat and Mass Transfer and Fluid Dynamics. The Fluids-3D code also counts with parallel processing capabilities. The results of flows over several kinds of complexes geometries, such as wall mounted cubes, automobile and aircraft prototypes are shown. These simulations have provided rich analysis of the flows. Others important contributions of the present work are the preparations of the code to the implementation of a graphical user interface, and the optimization of the parallelization process of the code as well.

Key-Words: Immersed Boundary Methods, Virtual Physical Model, flow over complexes geometries, parallel processing.

Lista de figuras

Figura 2.1 - Exemplo de malha não estruturada (http://www.nada.kth.se/~mihai/airfoil1.gif acessado dia 28/12/06).....	7
Figura 2.2 - Exemplo de malha que não se adapta ao corpo imerso, Oliveira (2007).	8
Figura 2.3 - Problema que motivou o desenvolvimento do método de fronteira imersa. (OLIVEIRA, 2006).....	9
Figura 2.4 - Caracterização de um corpo por pontos discretos, Vedovoto (2006).	10
Figura 2.5 - Muralha de Adriano – um dos primeiros exemplos de processamento paralelo.	13
Figura 2.6 - Cluster Beowulf LTCM – UFU.	14
Figura 2.7 - Supercomputador Cray (www.cray.com).	14
Figura 2.8 - Cabine de um Supercomputador Blue Gene (http://en.wikipedia.org/wiki/Image:BlueGeneL_cabinet.jpg).	15
Figura 2.9 - Sistemas operacionais utilizados nos maiores supercomputadores da atualidade (Fonte: http://en.wikipedia.org/wiki/Supercomputers).	16
Figura 2.10 - Esquema de montagem de um Cluster Beowulf, retirado de Campregher (2005).	18
Figura 2.11 - Resultados de speed-up para três níveis de refinamentos diferentes (MARINHO, 2004).	21
Figura 2.12 - Resultados da eficiência obtidos em três malhas diferentes (MARINHO, 2004).....	22
Figura 3.1 - Desenho esquemático indicando os vetores \vec{x} e \vec{x}_k (CAMPREGHER, 2005).	24
Figura 4.1 - Volume de controle elementar.	28
Figura 4.2 - Posição de um ponto lagrangiano \vec{x}_k (VEDOVOTO et al. 2006).....	35
Figura 4.3 - Vista detalhada de um elemento triangular (VEDOVOTO et al. 2006).	36
Figura 5.1 - Exemplo de arquivo de nós utilizado na primeira versão do código Fluids-3D.	45
Figura 5.2 - Exemplo de arquivo de conectividades utilizado na primeira versão do código Fluids-3D.	46

Figura 5.3 - Exemplo de arquivo de nós utilizado na versão atual do código Fluids-3D.....	46
Figura 5.4 - Exemplo de arquivo de conectividades utilizado na versão atual do código Fluids-3D.	47
Figura 5.5 - Exemplo de malha triangular gerado a partir de arquivos de nós e conectividades Vedovoto et al. (2006).....	47
Figura 5.6 - Cubo gerado a partir de geradores de malhas comercial.	48
Figura 5.7 - Esfera gerada a partir de gerador de malhas comercial (VEDOVOTO et al. 2006).....	48
Figura 5.8 - Aerofólios com winglets de varias dimensões criados em um gerador de malhas comercial.....	49
Figura 5.9 - Exemplo de máquina de estereolitografia - (http://www.stereolithography.com/stereolithography_images.php/).	50
Figura 5.10 - Exemplo de arma de paint-ball, criada em uma impressora 3D – (http://www.stereolithography.com/stereolithography_images.php).	50
Figura 5.11 - Típico motor de combustão interna criado em uma impressora 3D – (http://www.stereolithography.com/stereolithography_images.php).	51
Figura 5.12 - Trecho de uma turbina eólica representada por um arquivo *.stl.....	51
Figura 5.13 - Trecho de um arquivo *.stl . (http://en.wikipedia.org/wiki/STL_(file_format)).	52
Figura 5.14 - Malha *.stl, gerada sobre um protótipo de automóvel (OLIVEIRA, 2007).....	53
Figura 6.1 - Topologias possíveis para a troca de mensagens. a) domínio original, b) paralelização unidimensional, c) bidimensional e d) tridimensional.	56
Figura 6.2 - Relação entre a diminuição do volume computacional e da área de troca de mensagens para as três topologias possíveis de particionamento (CAMPREGHER, 2005).....	56
Figura 6.3 - Sobreposição necessária para aproximações espaciais de segunda ordem (CAMPREGHER, 2005).....	57
Figura 6.4 - Comunicação entre os subdomínios usando uma topologia unidimensional de divisão (MARINHO, 2004).	58
Figura 6.5 - Representação física de um domínio tridimensional.....	59
Figura 6.6 - Áreas de trocas de informações em uma topologia tridimensional de paralelização.....	60

Figura 6.7 - Pseudo-código indicando a formação de um vetor ϕ	61
Figura 6.8 - Área de troca de informação nas três direções coordenadas.....	62
Figura 7.1 - Visualização de uma malha de elementos triangulares provenientes de um arquivo *.stl.....	66
Figura 7.2 - Malha euleriana utilizada nas simulações dos escoamentos sobre esferas.....	67
Figura 7.3 - Particionamento do domínio euleriano para escoamentos sobre esferas.	68
Figura 7.4 - Estruturas turbilhonares a jusante de uma esfera, importada de geradores de malhas comerciais (Re=1.000).	70
Figura 7.5 - Estruturas turbilhonares a jusante de uma esfera, importada de geradores de malhas livres, *.stl (Re=1.000).....	71
Figura 7.6 - Estruturas turbilhonares geradas a jusante de uma esfera, vistas no plano XY, em t=7,0s.	72
Figura 7.7 - Gráfico do coeficiente de arrasto, C_D , por t^*	73
Figura 7.8 - Gráfico do coeficiente lateral, C_S , por t^*	74
Figura 7.9 - Gráfico do coeficiente de sustentação, C_L , por t^*	75
Figura 7.10 - Malha de elementos triangulares representando um cubo.	76
Figura 7.11 - Malha euleriana utilizada na simulação do escoamento sobre um cubo.	77
Figura 7.12 - Evolução temporal das linhas de corrente para os estágios: (a) t = 0,8 s, (b) t = 1,6 s, (c) t = 2,4 s, (d) t = 4,0 s, (e) t = 8,0 s, (f) t = 8,3 s.	79
Figura 7.13 - Evolução temporal dos coeficientes de arrasto, lateral e de sustentação para o escoamento sobre um cubo, Re=290.....	80
Figura 7.14 - Bolhas de recirculação a jusante de um cubo.....	81
Figura 7.15 - Domínio euleriano para as simulações de um aerofólio NACA-0012, Vedovoto et al. (2006).....	82
Figura 7.16 - Aerofólio discretizado por malha de elementos triangulares. Vedovoto et al. (2006).	83
Figura 7.17 - Evolução de linhas de corrente sobre um aerofólio NACA-0012 nos seguintes tempos físicos: (a) t=0,01s, (b) t=0,1s (c) t=0,5s (d) t=1,0s (e) t=1,5s (f) t=3,0s (g) t=5,0s (h) t=10,0s. Vedovoto et al. (2006).....	84

Figura 7.18 - Planos perpendiculares de linhas de corrente em $t = 10.0$ s . Vedovoto et al. (2006).	86
Figura 7.19 - Vista superior da Figura 7.18. Vedovoto et al. (2006).....	86
Figura 7.20 - Vista frontal dos vórtices da Figura 7.18. Vedovoto et al. (2006).....	87
Figura 7.21 - Resultado qualitativo de vórtices de ponta de asa obtidos por Craft et al. (2006).	87
Figura 7.22 – Isso superfícies de $Q = 0,2$ para $t = 5,6$ segundos físicos.....	88
Figura 7.23 - Esquema de posicionamento de um cubo sobre uma parede plana.	89
Figura 7.24 - Esquema da mudança na direção de vetores utilizados nos cálculos das interpolações.	90
Figura 7.25 - Isosuperfícies de Q obtidos por Hwang e Yang (2004), $Re = 1.000$	91
Figura 7.26 - Isosuperfícies de Q obtidos no presente trabalho, $Re = 1000$	92
Figura 7.27 - Linhas de corrente a $Re = 350$ plano xz , $x = 0,0053$ m, retirado de Hwang e Yang (2004).	92
Figura 7.28 - Linhas de corrente vistas em um plano $z = 0,001$ m, $Re=350$	93
Figura 7.29 - Linhas de corrente tridimensionais do escoamento sobre um cubo solidário a base, $Re= 1.000$	94
Figura 7.30 - Linhas de corrente tridimensionais do escoamento sobre um cubo solidário a base, vista do plano xz , $Re= 1.000$	94
Figura 7.31 - Isosuperfícies de $Q = 75$, coloridas por valores da velocidade u , para $Re = 1.000$	95
Figura 7.32 - Evolução temporal de parâmetros quantitativos para o escoamento sobre um cubo solidário à uma parede.	96
Figura 7.33 - Malha de elementos triangulares utilizada para representar uma estrutura treliçada.	97
Figura 7.34 – Geometria da Figura 7.33 inserida no domínio euleriano.	97
Figura 7.35 - Cubo com arestas arredondadas representado por malhas de elementos triangulares.	99
Figura 7.36 - Malha euleriana para simulação do escoamento sobre uma estrutura treliçada.	100

Figura 7.37 - Indicação de um trecho de torre de plataforma de petróleo, Lima e Silva et al. (2005).....	101
Figura 7.38 - escoamento sobre uma estrutura bidimensional treliçada, Lima e Silva et al. (2005).	102
Figura 7.39 - Evolução de linhas de corrente para escoamento sobre uma estrutura treliçada à $Re = 1.000$, nos seguintes tempos físicos (de cima para baixo): $t = 0,3$ s, $t = 1,4$ s, $t = 1,6$ s, $t = 2,0$ s, $t = 9,5$ s e $t = 10,0$ s.....	103
Figura 7.40 - Planos indicando a presença de bolhas de recirculação, em $t = 2,0$ s e $Re=1.000$	104
Figura 7.41 - Iso superfícies de $Q = 5$, para os seguintes tempos físicos (de cima para baixo, da esquerda para direita): $t = 0,1$ s, $t = 0,5$ s, $t = 0,6$ s, $t = 0,9$ s, $t = 2,5$ s, $t = 3,0$ s, $t = 4,5$ s e $t = 10,0$ s.	105
Figura 7.42 - Comparações quantitativas entre o escoamento sobre uma estrutura treliçada e um cubo de arestas arredondadas, $Re = 1.000$	106
Figura 7.43 - Malha de elementos triangulares representando um protótipo de automóvel (OLIVEIRA, 2007).....	107
Figura 7.44 - Domínio de cálculo para simulação do escoamento sobre um protótipo de automóvel.	108
Figura 7.45 - Detalhe da roda dianteira do protótipo do automóvel.	109
Figura 7.46 - Evolução temporal de isosuperfícies de $Q = 5$ para os seguintes tempos físicos: (a) $t = 0,005$ s, (b) $t = 0,0045$ s, (c) $t = 0,09$ s, (d) $t = 0,8$ s, (e) $t = 2,7$ s e (f) $t = 4,7$ s.....	111
Figura 7.47 - Detalhes das isosuperfícies de $Q = 10$ para $t = 0,6$ s.	112
Figura 7.48 - Linhas de corrente do escoamento sobre um protótipo de automóvel, $t = 2,6$ s.	113
Figura 7.49 - Vista posterior das linhas de corrente a jusante do protótipo de automóvel, $t = 2,6$ s.	114
Figura 7.50 - Detalhe do início de uma estrutura helicoidal a jusante do protótipo de automóvel, $t = 2,6$ s.	115
Figura 7.51 – Evolução de uma estrutura turbilhonar a jusante do protótipo de automóvel. ...	118

Figura 7.52 - Malha de elementos triangulares do protótipo de aeronave (OLIVEIRA, 2007).....	119
Figura 7.53 - Linhas de corrente indicando padrão de escoamento sobre o protótipo de aeronave, para $t = 0,6$ s.....	121
Figura 7.54 - Vista lateral da Figura 7.53.	122

Nomenclatura

Letras Latinas

A :	área superficial do domínio computacional;
A_N :	área superficial do subdomínio particionado;
b :	vetor força de corpo;
C_D :	coeficiente de arrasto;
C_L :	coeficiente de sustentação;
C_S :	coeficiente lateral;
d :	distância entre os centros dos volumes;
dS :	área diferencial;
D :	diâmetro da esfera;
D_i :	função distribuição;
E :	eficiência do processamento paralelo;
i :	nº de células em cada aresta do domínio computacional;
f :	vetor força por unidade de volume, frequência;
F :	vetor força no interior do sistema, força interfacial, fluxo nas faces do volume;
i :	direção cartesiana;
I :	tensor unitário;
k :	índice do ponto lagrangiano;
L :	comprimento característico do objeto imerso;
n :	vetor normal;
N :	nº de subdomínios;
p :	pressão
p_k :	pressão no ponto lagrangiano k ;
q :	termo fonte;
q_ϕ :	termo fonte associado à variável genérica ϕ ;
Re :	número de Reynolds;
S :	speed-up, parte viscosa do tensor tensão, área superficial do volume de controle;

St :	número de Strouhal;
t :	tempo;
T_S :	tempo de processamento serial;
T_N :	tempo de processamento em N processadores;
u :	velocidade na direção x;
U_∞	velocidade U na corrente livre;
v :	velocidade na direção y;
w :	velocidade na direção z;
V :	volume total do domínio computacional;
V_k :	velocidade do ponto lagrangiano;
V_N :	volume do subdomínio particionado;
x :	vetor posição do volume elementar;
x_k :	vetor posição do ponto lagrangiano k;

Letras Gregas

α	parâmetro de função peso, coeficiente de interpolação;
δ	distância entre os centróides dos volumes adjacentes;
Δ	largura do volume finito;
ϕ	variável genérica;
Φ	propriedade extensiva;
Γ^ϕ	coeficiente de difusão para a variável ϕ ;
μ	viscosidade dinâmica;
ρ	densidade;
τ_{ij}	tensor de Reynolds;
ω	vorticidade;
Ω	volume ocupado por uma porção de massa, tensor vorticidade;
Ω_k	volume elementar em torno do ponto lagrangiano k ;

Operadores

D	derivada material, ou substantiva;
Δ	variação;
∂	derivada parcial;
∇	nabla;
\int	integral;
Σ	somatório;
Π	produtório;

Índices

a	advectivo;
d	difusivo;
ent	referente a entrada do domínio;
P	centro do volume de controle;
N, n	ponto e face ao norte do centro do volume de controle;
S, s	ponto e face ao sul do centro do volume de controle;
E, e	ponto e face ao leste do centro do volume de controle;
W, w	ponto e face ao oeste do centro do volume de controle;
T, t	ponto e face superior do centro do volume de controle;
B, b	ponto e face inferior do centro do volume de controle;
i, j	ponto central, componente de tensor;
∞	corrente livre ;
max	máximo;
min	mínimo;

Superíndices

*	grandezas adimensionais, estimativa de propriedade;
H	alta ordem de interpolação;
L	baixa ordem de interpolação;
n	iteração;

t tempo atual;

Siglas

CAD Computer Aided Design;

CFD Computational Fluid Dynamics;

LTCM Laboratório de Transferência de Calor e Massa e Dinâmica dos Fluidos;

MPI Message Passing Interface;

*.STL Standard Tessellation Language;

UFU Universidade Federal de Uberlândia;

Índice

1	Introdução	1
1.1	Objetivos	1
1.2	Metodologia.....	2
2	Revisão Bibliográfica	5
2.1	Métodos numéricos aplicados ao estudo de escoamentos de fluidos	5
2.2	Discretização do domínio computacional.....	5
2.3	O Método da fronteira imersa	7
2.4	Processamento Paralelo	12
2.4.1	Sistemas operacionais.....	15
2.4.2	Computadores paralelos.....	16
2.4.3	Metodologias de paralelização	19
3	Modelagem Matemática.....	23
3.1	Formulação para o domínio fluido.....	23
3.2	Modelo matemático para a interface fluido sólido.....	25
4	Metodologia Numérica	27
4.1	Discretização das equações para o domínio fluido.....	27
4.1.1	O Algoritmo SIMPLEC	30
4.1.2	Interpolação Rhie-Chow	32
4.2	Discretização das equações representativas do domínio lagrangiano	34
4.2.1	O procedimento de distribuição da força lagrangiana	37
5	Metodologia de importação de geometrias complexas e alterações no set-up do programa Fluids-3D	41
5.1	Alterações de forma a facilitar o uso e set-up do código Fluids-3D.....	42
5.2	Melhoramentos realizados com relação à importação de geometrias.....	44
5.2.1	Mudanças na importação de arquivos de nós e conectividades de uma malha triangular	45
5.2.2	Importação de geometrias geradas pelo processo de estereolitografia	49

6	Metodologias de paralelização utilizadas no Fluids-3D	55
6.1	Tipos de paralelização possíveis no código Fluids-3D	55
6.2	Implementação da topologia tridimensional de paralelização no código Fluids-3D.....	58
6.3	Bibliotecas de paralelização.	63
7	Resultados e Discussão	65
7.1	Resultados de re-validação	65
7.1.1	Resultados de validação da importação de geometrias via *.stl	66
7.1.2	Resultados de re-validação de importação de geometrias geradas via geradores de malha comerciais – escoamento sobre um cubo	75
7.2	Escoamentos sobre geometrias complexas	81
7.2.1	Escoamento sobre um aerofólio NACA-0012 – Visualização de vórtices de ponta de asa	81
7.2.2	Escoamento sobre um cubo solidário ao plano $z=0$	88
7.2.3	Escoamento sobre uma estrutura treliçada	96
7.2.4	Escoamento sobre um protótipo de automóvel	107
7.2.5	Escoamento sobre um protótipo de aeronave	119
8	Conclusões e Perspectivas.....	123
9	Referências Bibliográficas	125

CAPÍTULO I

1 Introdução

Nas mais variadas aplicações de engenharia a dinâmica dos fluidos está presente. No entanto, seu estudo, analiticamente, é limitado, devido às complexidades das equações regentes e também a complicações como escoamentos sobre geometrias complexas e/ou móveis, e com grande variação de densidade ou viscosidade.

Para a análise dos escoamentos complexos citados, existem basicamente, duas maneiras de se trabalhar: experimentalmente ou numericamente. Análises experimentais, apesar de também apresentarem suas limitações, também evoluíram muito, com o advento de instrumentos não intrusivos ou de dimensões praticamente microscópicas e de alta precisão. Destaca-se entre esses avanços: modernos anemômetros de fio quente, anemometria *Laser-Doppler* e PIV (*Particle Image Velocimetry*). Essas técnicas são aplicadas a nível acadêmico para estudar e quantificar escoamentos de base, tais como camada limite, escoamentos cisalhantes livres, jatos e esteiras ou mesmo em escoamentos complexos, utilizados na indústria moderna como parte do projeto de aeronaves, veículos, máquinas térmicas, bombas, edificações e etc. Em alguns casos, a realização de um ensaio experimental exige a execução do projeto em si, ou requer uma condição difícil e muitas vezes cara de se obter em laboratório.

Uma grande vertente que vem crescendo a cada dia é a dinâmica dos fluidos computacional, ou CFD, do inglês *Computational Fluid Dynamics*. A dinâmica dos fluidos computacional requer computadores de alta velocidade para, numericamente, resolver as equações não-lineares governantes da dinâmica dos fluidos. Utilizando da modelagem matemática do escoamento sobre, ou através de um objeto, pode-se reduzir o gasto de grandes quantidades de tempo e dinheiro, além de minimizar o risco potencial de um teste experimental ou lançamento de um protótipo de motor de combustão ou aeronave, por exemplo. Condições de operação extremas de temperaturas, pressões, velocidades ou geometrias complexas em geral são de difícil reprodução em laboratório. Aplicando-se modelos e métodos apropriados, resultados confiáveis podem ser obtidos via CFD.

1.1 Objetivos

Este trabalho tem por objetivo a simulação numérica e análise de escoamentos sobre diversas geometrias complexas tridimensionais, utilizando o Método da fronteira imersa, desde

baixos números de Reynolds até Reynolds moderados. Diferentemente da primeira versão do código, desenvolvido no trabalho de Campregher (2005), na qual a importação de geometrias complexas era trabalhosa, e conseqüentemente o estudo de diversos tipos escoamentos sobre geometrias diferentes em tempo hábil era muito limitado, no presente trabalho objetivou-se otimizar a capacidade de importação de geometrias complexas, de forma a facilitar o uso do programa, oferecendo ao usuário, inclusive, uma nova opção de tipo de arquivo a ser importado para representação de geometrias complexas, o tipo *.stl*, utilizado em máquinas de estereolitografia. Esta característica é reforçada ao se utilizar softwares com capacidades de exportar geometrias construídas em programas CAD (*Computer Aided Design*). Tem por objetivo ainda, a otimização da estrutura de paralelização do código utilizado para as simulações, aumentando e estudando as possíveis topologias de paralelização buscando redução no custo computacional inerente às simulações numéricas, e alterar o código de forma a estar apto a aceitar uma interface gráfica visando maior facilidade de uso, tanto acadêmico quanto para aplicações em engenharia.

Estas alterações, na estrutura de paralelização, importação de geometrias e preparação para acoplamento a uma interface gráfica externa exigem mudanças profundas no código fonte, e, por esta razão novas análises qualitativas e quantitativas devem ser feitas de forma a re-validar o mesmo.

Os testes de validação aplicados envolvem escoamentos sobre esferas, ao redor de cubos apoiados em paredes, e canais retangulares. Esta última geometria é utilizada para o caso de escoamentos avaliando topologias de paralelização. Apesar dos corpos imersos escolhidos serem geometricamente simples, estes dão origem a escoamentos extremamente complexos com grande riqueza de detalhes. Outros corpos geometricamente mais complicados como aerofólios e estruturas treliçadas também são estudadas.

1.2 Metodologia

Todas as análises realizadas no presente trabalho provêm de simulações computacionais. Para isto, fez-se necessário o uso e aperfeiçoamento de um código desenvolvido no Laboratório de Transferência de Calor e Massa e Dinâmica dos Fluidos (LTCM-UFU).

Tal código, chamado de Fluids-3D, é baseado na metodologia de Volumes Finitos e resolve as equações de Navier-Stokes transientes e incompressíveis para um domínio cartesiano tridimensional. O programa Fluids-3D ainda utiliza o Método da fronteira imersa para lidar com geometrias imersas no escoamento.

O código utiliza esquemas de segunda ordem para o tempo e espaço, com diferenças centradas para as derivadas espaciais, esquema de avanço temporal *three-time-level* em malha co-localizada. O acoplamento pressão-velocidade é feito pelo algoritmo SIMPLEC, utilizando o solver SOR (*Successive Over Relaxation*), para as velocidades e SIP (*Strong Implicit Procedure*) para a equação de correção da pressão. O código ainda tem capacidade de processamento paralelo.

É utilizado um *cluster Beowulf* de 10 microcomputadores Pentium 4 (2.8 GHz / 1.5Gb RAM), disponível no LTCM. Todo o algoritmo é escrito em FORTRAN 90, utilizando a biblioteca de paralelização MPICH 2, e compilado pelo IFC (*Intel Fortran Compiler*), ambos *softwares* livres para uso acadêmico. A visualização dos resultados é realizada através de gráficos com perfis de velocidade, coeficientes de arrasto, lateral e de sustentação, ou através da plotagem de superfícies tridimensionais de velocidade, pressão, vorticidade e outras grandezas que identificam estruturas turbilhonares.

CAPÍTULO II

2 Revisão Bibliográfica

2.1 *Métodos numéricos aplicados ao estudo de escoamentos de fluidos*

A simulação numérica de escoamentos já atingiu níveis elevados de precisão, podendo ser chamada de experimentação numérica, numa analogia à experimentação física. Porém, para que estas simulações sejam feitas da forma mais adequada possível, é indispensável o conhecimento sobre os métodos numéricos utilizados, suas vantagens e limitações.

Apresenta-se neste capítulo, uma breve revisão sobre métodos numéricos aplicados a escoamentos de fluidos, bem como conceitos do Método da fronteira imersa e Paralelização.

2.2 *Discretização do domínio computacional*

Existem muitas maneiras de realizar a aproximação de um sistema diferencial de equações para um sistema algébrico que contenha tanto as aproximações temporais quanto espaciais. Dentre as mais importantes pode-se citar: Método das Diferenças Finitas, o Método dos volumes finitos, o método dos elementos finitos e métodos espectrais, Versteeg e Malalasekera (1995). Há ainda outros métodos de solução como o método dos elementos de contorno, métodos que utilizam as equações de Lattice-Boltzmann etc, porém a aplicação destes últimos dois métodos é aplicada a classes específicas de problemas, (FERZIGER; PERIC, 2002).

O método das diferenças finitas, um dos primeiros métodos a ser criado, provavelmente por Euler no século 18, é também um dos mais simples de serem implementados. É usado principalmente para escoamentos sobre geometrias simples. Neste método a aproximação das equações diferenciais é feita a partir do uso dos termos da série de Taylor e em malhas regulares, podem ser obtidas altas ordens de resolução.

A desvantagem do método das diferenças finitas reside no fato de que este é um método, a princípio, não conservativo. A princípio, porque cuidados podem ser tomados de forma a garantir a conservação das propriedades (LIMA E SILVA, 2002). Uma outra severa

restrição, é que o uso deste método, é restrita a escoamentos sobre geometrias simples. (FERZIGER; PERIC, 2002).

Tendo como berço as pesquisas em mecânica dos sólidos, o método dos elementos finitos (*Finite Element Method* -FEM) trabalha com funções-base locais, multiplicadas por uma função-teste e, depois, integradas. O resultado das simulações é baseado nos cálculos destas integrais. Esta metodologia apresenta importantes vantagens como a facilidade em ser aplicada a geometrias complexas (desde que sejam empregadas as funções base e teste corretas), a relativamente simples implementação de esquemas de alta ordem e aos muitos trabalhos teóricos já produzidos sobre o tema. Entretanto, o maior desafio ao se adotar esta abordagem é encontrar um conjunto de equações que represente bem o conjunto de equações originais e que seja estável, ou seja, os erros decorrentes da aproximação não sejam acumulativos ao longo do processo de solução (CAMPREGHER, 2005).

De emprego um tanto mais complexo que as técnicas anteriores, os métodos espectrais (*Spectral Methods*) empregam discretizações espaciais das derivadas que são transformadas para o espaço de Fourier (espectral). A maior vantagem deste tipo de método é que a convergência acontece de forma mais rápida do que qualquer outro método espacial de ordem n , para $n < \infty$. Entretanto, a dificuldade em lidar com domínios complexos e condições de contorno, além de requerer passos de tempo muito menores (por questão de estabilidade), impede sua disseminação em simulações de problemas reais de engenharia. Novos avanços tem sido feitos no sentido de utilizar métodos espectrais em problemas não periódicos, utilizando o método da fronteira imersa, mais detalhes sobre esta metodologia podem ser encontrados em Mariano et al. (2006), e Mariano (2007).

O atual trabalho utiliza o método dos volumes finitos como abordagem para discretização das equações diferenciais. Neste método é realizada uma integração das equações diferenciais parciais (EDP) em uma região, ou volume do espaço. Este método está ligado diretamente ao conceito de fluxo entre volumes adjacentes. A quantidade líquida de uma grandeza ϕ , que atravessa as fronteiras do volume de controle Ω por unidade de tempo, é calculada pela integração, sobre estas fronteiras, da diferença entre os fluxos que entram e os que saem de Ω , (FORTUNA, 2000). O resultado desta integração, mais a produção líquida de ϕ no volume é proporcional à variação temporal de ϕ dentro do volume.

A desvantagem do Método dos volumes finitos em relação a Diferenças Finitas, por exemplo, é que aquele método, em discretizações de ordem maior que 2 são mais complicadas de serem desenvolvidas. Este fato deve-se à necessidade do Método dos volumes finitos requerer três níveis de aproximação: interpolação, diferenciação, e integração (FERZIGER; PERIC, 2002).

O código Fluids-3D ainda utiliza o Algoritmo SIMPLEC, para o acoplamento pressão velocidade com interpolação Rhie-Chow. Nos termos advectivos das equações de Navier-Stokes, usa-se ainda uma estratégia conhecida como correção atrasada (*deferred-correction*). Estas peculiaridades serão discutidas mais detalhadamente nos capítulos de metodologia matemática e numérica.

2.3 O Método da fronteira imersa

Uma das grandes dificuldades em se estudar escoamentos sobre geometrias complexas, móveis ou deformáveis é justamente fazer com que estas geometrias sejam identificadas adequadamente.

Duas abordagens básicas são possíveis para simulação de corpos imersos em um escoamento: malhas que se adaptam ao corpo (chamadas de *Body-Fitted Meshes*) e malhas que não se adaptam ao corpo. Os exemplos mais comuns da primeira abordagem são as malhas não estruturadas, como exemplifica a Figura 2.1. No caso da segunda abordagem, um exemplo mostra-se à Figura 2.2. Dá-se uma idéia de dois domínios acoplados, porém com malhas não coincidentes.

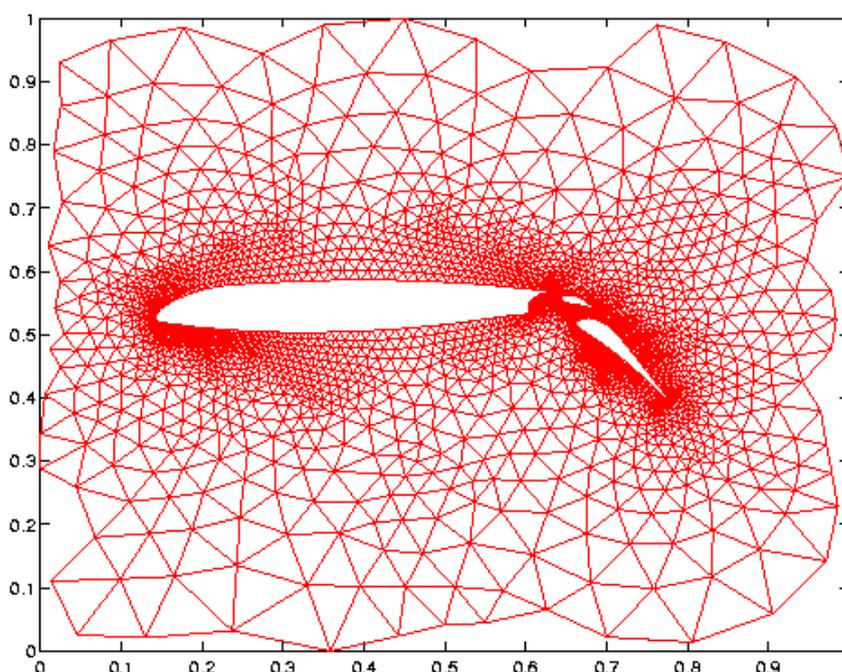


Figura 2.1 - Exemplo de malha não estruturada (<http://www.nada.kth.se/~mihai/airfoil1.gif> acessado dia 28/12/06).

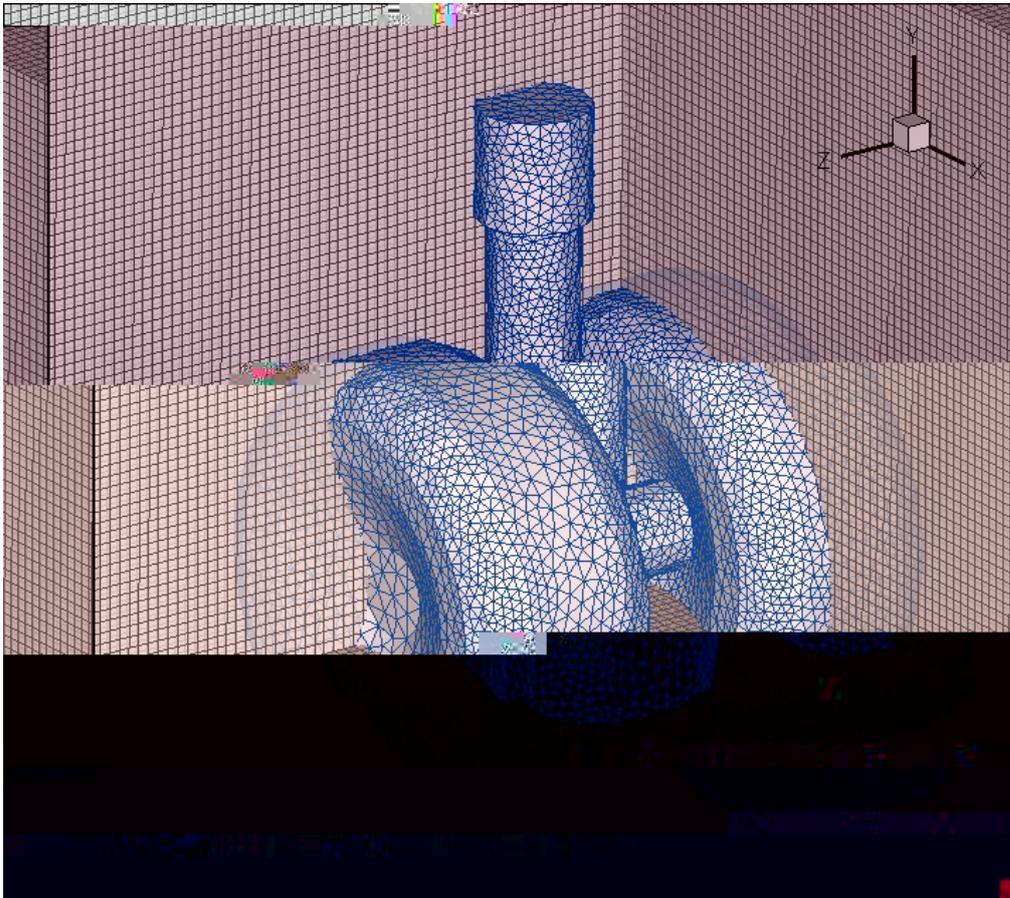


Figura 2.2 - Exemplo de malha que não se adapta ao corpo imerso, Oliveira (2007).

No caso das malhas que não se adaptam ao corpo, a técnica mais utilizada é a dos domínios fictícios (*Fictitious Domains* - FD), Yu (2005). Primeiramente desenvolvida pelos soviéticos, esta técnica é utilizada a mais de 30 anos (GLOWINSKI et al. 1998).

A técnica FD pode ser subdividida em três tipos de modelos: i) modelos que não se baseiam em forças de corpo, ii) modelos que usam forças de corpo e usam multiplicadores de Lagrange Distribuídos (*Distributed Lagrangian Multipliers* – DLM), com o intuito de obter uma pseudo-força de corpo. Este método é muito utilizado em escoamentos contendo particulados (OLIVEIRA, 2006). A terceira técnica, que é a técnica na qual se encontram os Métodos de Fronteira Imersa, são os modelos que utilizam forças de corpo, porém não utilizam DLM.

O método de fronteira imersa (*Immersed Boundary Method* – IB) surgiu como uma alternativa eficiente aos métodos cujas malhas se ajustam às fronteiras (*body-fitted*) para tratamento de problemas envolvendo geometrias complexas, móveis e deformáveis. No método de fronteira imersa o corpo é representado por um campo de forças que, de alguma forma, é

inserido às equações do fluido, fazendo com que o corpo seja modelado indiretamente. O método foi desenvolvido por Peskin (1972) cuja motivação era simular o escoamento de sangue em válvulas cardíacas (Figura 2.3).

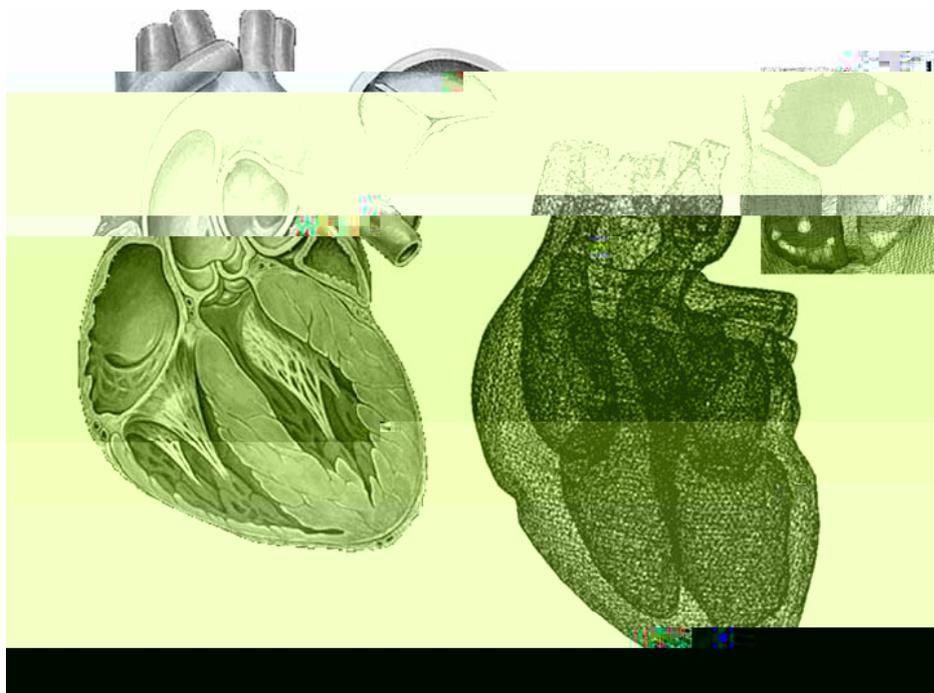


Figura 2.3 - Problema que motivou o desenvolvimento do método de fronteira imersa. (OLIVEIRA, 2006).

Uma das grandes vantagens do Método da fronteira imersa é ser uma metodologia baseada na ação de forças “externas”, tornando-se possível que o domínio de cálculo do escoamento seja representado por malhas extremamente simples (cartesianas), utilizando sem grandes dificuldades, diferenças finitas ou volumes finitos. No caso do corpo imerso, este pode ser representado como um conjunto de pontos discretos, como exemplifica a Figura 2.4.

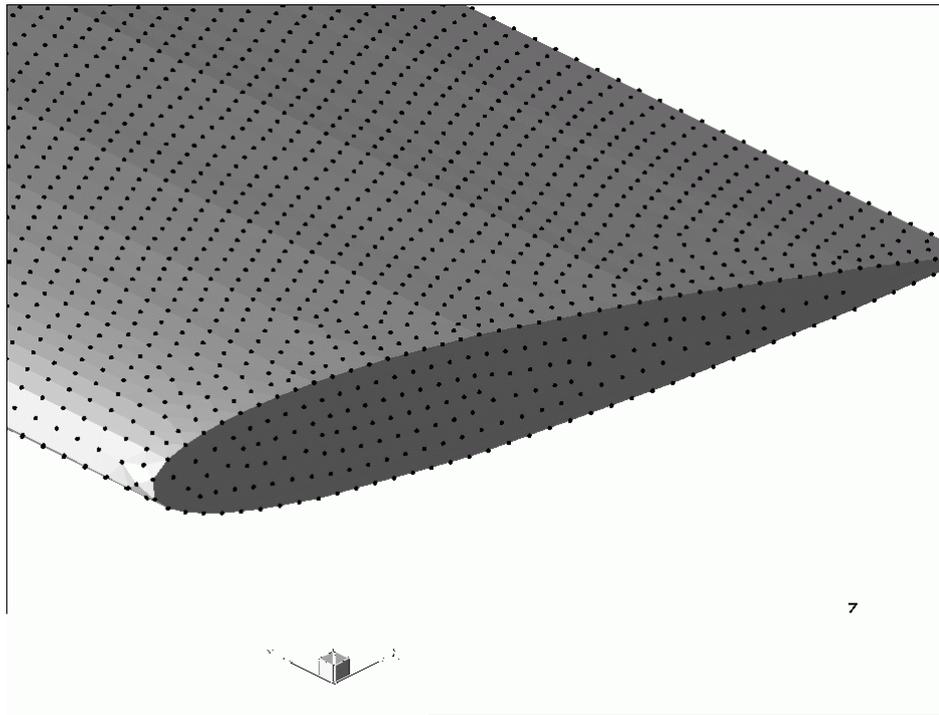


Figura 2.4 - Caracterização de um corpo por pontos discretos, Vedovoto (2006).

Como já citado em parágrafos anteriores, o desenvolvimento do Método da fronteira imersa deveu-se à Charles Peskin e colaboradores, os quais tinham por motivação simular o escoamento, em domínios bidimensionais, do sangue por válvulas cardíacas. De acordo com seus trabalhos (PESKIN, 1972 e PESKIN, 1977), a natureza do termo de força adicional era proveniente da taxa de deformação da fronteira, na qual seus pontos constitutivos eram unidos por forças elásticas. O Método da fronteira imersa foi desenvolvido para simular o escoamento de fluidos viscosos em domínios complexos, onde a fronteira do corpo é representada por uma coleção de pontos discretos. A interação entre o fluido e a fronteira é modelada através de forças de interação que dependem da taxa de deformação da fronteira. Este método é amplamente utilizado em simulações de escoamento de fluidos em domínios complexos, especialmente em aplicações biomédicas, como a simulação do fluxo sanguíneo em válvulas cardíacas.

tecnologia

inodous p snous aroblemças2D s

om

Mohd-Yusof (1997) propôs que o cálculo da força lagrangiana fosse realizado com base na equação da quantidade de movimento do fluido na interface, sem o emprego de constantes que necessitem de ajuste. Este método foi chamado de *direct forcing method*. Entretanto, requer algoritmos complexos, de modo a localizar a geometria no interior do domínio, além de interpolar os valores das propriedades nas partículas de fluido adjacentes usando B-splines, o que encarece a proposta.

Kim et al. (2001) realizaram experimentos com a metodologia de Mohd-Yusof em domínios discretizados por Volumes Finitos. Porém, procuraram empregar interpolações lineares e bilineares para a velocidade na avaliação do campo de força. Os autores incluíram, na equação da continuidade, termos fonte ou sumidouro de massa na tentativa de melhorar a acurácia do método e obter soluções fisicamente mais consistentes. Conseguindo assim impor a condição de não-deslizamento para a fronteira e também a equação da continuidade nas células Eulerianas da interface. Funções de interpolação de segunda ordem lineares e bilineares foram utilizadas para a velocidade.

O modelo de força utilizado no presente trabalho, denominado modelo físico virtual (Virtual Physical Model – VPM), proposto por Lima e Silva et al. (2003), é um modelo de força discreta com imposição indireta da condição de contorno. A força sobre a interface é calculada dinamicamente através das equações de balanço da quantidade de movimento sobre uma partícula de fluido na interface. A força calculada é inserida como termo fonte nas equações de Navier-Stokes. Assim impõe-se, de maneira indireta, a condição de contorno desejada sobre a fronteira. O método tem a capacidade de se auto-ajustar ao escoamento uma vez que a força necessária para frear as partículas de fluido próximas a interface é calculada de maneira automática. Este método vem apresentando bons resultados na simulação de diferentes casos.

Este modelo foi testado em domínios bidimensionais, para diversos problemas práticos de engenharia, além de problemas clássicos em mecânica dos fluidos. Escoamentos ao redor de obstáculos a altos números de Reynolds podem ser encontrados em Oliveira et al. (2004b), escoamentos ao redor de geometrias complexas em Lima e Silva et al. (2005), escoamentos ao redor de obstáculos móveis, objetos em queda livre (constituindo um excelente teste de interação fluido-estrutura) podem ser vistos em Vilaça et al. (2004) e, ainda, escoamentos sobre cilindros de diâmetro variável em Oliveira et al. (2004a). Escoamentos forçados em condutos e cavidades de fundo móvel podem ser encontrados em Arruda (2004) e Arruda et al. (2004).

Campregher (2005), estendeu o Modelo físico virtual para domínios tridimensionais, e simulando escoamentos a baixos números de Reynolds conseguiu ótimos resultados tanto para esferas estáticas imersas quanto com interação fluido estrutura. Neste último caso, o sistema dinâmico escolhido foi composto de uma esfera imersa no escoamento sustentada por

molas. Foi estudado o efeito provocado pela ação do escoamento sobre a dinâmica do sistema e o conseqüente movimento da esfera sobre a geração e emissão de estruturas turbilhonares.

Como aplicação industrial, Padilla (2007), implementou o método da fronteira imersa para simulação de escoamentos transicionais em canais cilíndrico-anulares com excentricidade variável. Estes tipos de escoamentos estão presentes em condutos de perfuração de poços de extração de petróleo.

2.4 Processamento Paralelo

Um dos maiores problemas enfrentados em simulações envolvendo problemas físicos complexos são os recursos computacionais necessários para realizar estas simulações de modo satisfatório. Diante disto o uso de processamento paralelo na dinâmica dos fluidos computacional é, mais que indispensável, inevitável. O uso de computadores paralelos, (conjunto de unidades centrais de processamento, CPUs, que cooperativamente resolvem um problema), é cada vez mais difundido, seja com a utilização de supercomputadores, seja através de clusters de computadores pessoais.

O conceito de processamento paralelo não é algo novo. Um dos primeiros exemplos de processamento paralelo foi a construção da muralha de Adriano em 122 DC (Figura 2.5), a mando do imperador Adriano de Roma, que queria construir uma defesa das invasões de tribos do norte, melhorando a estabilidade econômica e promovendo a paz em seu reino. Na construção da muralha de quase 120 quilômetros, um grande número de legionários (em processamento paralelo entende-se processadores), foi destacado para a tarefa. Cada legionário sabia exatamente o que fazer (recebia ordens explícitas), e agia independentemente dos outros. Um gerenciador central não era necessário, cada legionário somente precisava falar com o legionário à sua esquerda e a direita. Além disto cada legionário levava consigo seus suprimentos de água e comida, e suas próprias ferramentas de trabalho.

Entende-se, deste procedimento adotado pelos legionários, que cada soldado agia como se fosse um processador realmente, uma vez que, cada processador já vem completamente munido de ferramentas a serem utilizadas, como a capacidade de compilação etc, cada computador tem sua própria capacidade de estocagem de dados, além disso, os computadores não necessitam estar ligados diretamente à um nó central (chamado comumente de "root"), pois estes podem comunicar entre si, conectados diretamente uns aos outros, além do fato de que assim como cada legionário tinha suas ordens explícitas, o código computacional tem domínio total do nó em que ele está sendo executado.



Figura 2.5 - Muralha de Adriano – um dos primeiros exemplos de processamento paralelo.

A Figura 2.6, assim como a Figura 2.7 e Figura 2.8 mostram três classes de computadores paralelos. A Figura 2.6 mostra um Cluster tipo Beowulf (a ser explicado mais detalhadamente nos tópicos a seguir). A Figura 2.7 e Figura 2.8, mostram dois supercomputadores, sendo que a última mostra parte do atual computador mais rápido do mundo.

Alguns fatores são importantes quando se fala em processamento paralelo são eles: o sistema operacional, os próprios computadores paralelos e as metodologias de programação. Nos tópicos seguintes serão apresentadas algumas características de cada um dos fatores citados acima.



Figura 2.6 - Cluster Beowulf LTCM – UFU.



Figura 2.7 - Supercomputador Cray (www.cray.com).



Figura 2.8 - Cabine de um Supercomputador Blue Gene (http://en.wikipedia.org/wiki/Image:BlueGeneL_cabinet.jpg).

2.4.1 *Sistemas operacionais*

Um importante fator de desempenho é a escolha de um sistema operacional que seja adequado ao computador paralelo e às aplicações que se desejam realizar com estes computadores. Durante muito tempo teve-se o domínio de sistemas UNIX, porém, nota-se, pela Figura 2.9, que nos últimos anos os sistemas operacionais baseados em LINUX tem ganhado bastante espaço. Este fato é devido ao crescente desenvolvimento de sistemas

LINUX, principalmente no que diz respeito à computação científica. A questão do sistema operacional é um tópico importante, pois ligado diretamente ao desenvolvimento do código paralelo. O código Fluids3D, que é a plataforma de trabalho da presente dissertação, foi desenvolvido para ser executado em LINUX.

Um fator interessante, e comum a praticamente todos os sistemas operacionais em supercomputadores é o fato de que nestes sistemas o uso de interfaces gráficas com usuários nunca foi alvo de grandes desenvolvimentos, pelo menos em computadores operando com sistemas UNIX. Isto deve-se ao fato de que supercomputadores podem chegar a custar milhões de dólares, e o desenvolvimento do sistema operacional deve ser realizado visando extrair o máximo de desempenho para cálculos matemáticos.

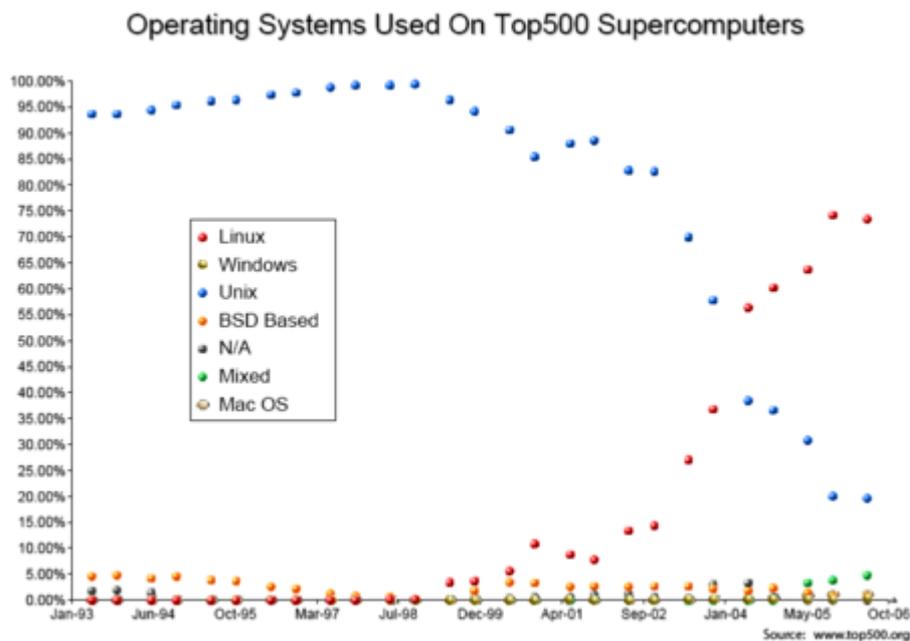


Figura 2.9 - Sistemas operacionais utilizados nos maiores supercomputadores da atualidade (Fonte: <http://en.wikipedia.org/wiki/Supercomputers>).

2.4.2 Computadores paralelos

Durante muito tempo, a única opção disponível aos pesquisadores era o emprego de supercomputadores caros e, freqüentemente, de acesso restrito, devido a estratégias comerciais e/ou reservas de mercado. Felizmente, a evolução dos computadores pessoais

(*Personal Computers* -PC) deu-se a taxas muito maiores do que os computadores de grande porte, estendendo o leque de opções em recursos (CAMPREGHER, 2005).

Esta evolução dos computadores pessoais possibilitou uma nova categoria de computadores: os clusters de computadores pessoais. Montados a partir de máquinas comuns é possível criar uma solução viável para o alto custo de problemas de mecânica dos fluidos, por exemplo, com relativamente baixo custo. Estritamente falando é quase impossível obter baixo custo utilizando um computador paralelo (*overhead*, despesas adicionais com redes, instalações etc...). O baixo custo é derivado de benefícios adicionais, como resolver problemas em menor espaço de tempo, ou que não poderia ser simulado em uma única máquina.

Uma outra grande vantagem deste tipo de cluster é escalabilidade, ou seja, é possível aumentar o número de nós de um cluster simplesmente anexando novos computadores à rede. Um tipo de cluster muito comum, que é formado por computadores pessoais é o chamado Cluster Beowulf.

2.4.2.1 O cluster Beowulf

"Beowulf": poema épico com 3182 linhas, considerado o maior expoente da literatura anglo-saxônica; escrito por um anônimo cristão, provavelmente no século X. Narra as três batalhas do herói, que dá nome a obra, contra o gigante Grendel, a mãe de Grendel e o dragão que guarda um tesouro. Uma alegoria da luta entre o Bem e o Mal, retratando aspectos da cultura germânica na região onde hoje se encontram a Dinamarca e sul da Suécia.

"Beowulf": conjunto de computadores pessoais (PC's), agrupados com o objetivo de obter a menor razão custo/benefício. Para isto são usados componentes de "hardware" disponíveis no mercado, independentes de fornecedores específicos, e baseados em sistemas operacionais e "softwares" gratuitos.

Estas são duas possíveis definições do termo "Beowulf". Existem várias outras, tanto para a versão literária como para a computacional. Em ambas "Beowulf" enfrenta um poderoso adversário, e vence. Enquanto a versão mítica triunfa sobre monstros e dragões, a real apresenta-se como uma alternativa barata à computação de alto desempenho, até então dominada pelo monopólio das grandes fornecedoras.

Thomas Sterling e Don Becker, pesquisadores do Goddard Space Flight Center (NASA), construíram o primeiro "cluster" de PC's em 1994. Uma alternativa barata e eficiente à limitação computacional da época. Mesmo a NASA não poderia fornecer individualmente a cada grupo de pesquisa recursos suficientes para a obtenção de um supercomputador. Assim, sem auxílio financeiro, Sterling teve a idéia de usar processadores de baixo custo (16 PC's com

processadores 486DX4), com um sistema operacional gratuito (Linux) e placas de rede Ethernet. Estas máquinas foram montadas e programadas de forma a possibilitarem a paralelização/divisão das tarefas, buscando atingir-se um poder de processamento equivalente a um supercomputador da época, por uma fração do preço. Este primeiro "cluster" atingia 70 megaflops, ou seja, setenta milhões de operações por segundo, com um custo estimado em 1/10 do valor cobrado pelo mercado para um sistema de desempenho similar. Tal projeto fez tanto sucesso que o termo "Beowulf" foi estendido a todos os "clusters" de PC's que viriam a surgir.

Atualmente, entre as 500 máquinas mais rápidas registradas pelo Top500 (www.top500.org), 28 são "clusters Beowulf", estando o Los Lobos, da Universidade do Novo México em octogésimo lugar, atingindo 237 gigaflops, ou 237 bilhões de operações por segundo.

Um esquema de um cluster de classe *Beowulf* empregado no presente trabalho pode ser visto na Figura 2.10. Atualmente, conta-se com 10 máquinas ligadas por uma rede de 1GBps sendo que, cada uma, conta com um processador Intel® Pentium IV de 2.8GHz, 1.5GBytes de memória DDR (*Double Data Rate*) RAM, 80GBytes de HD e rodam o sistema operacional Linux. O conjunto conta, ainda, com uma unidade KVM (*Keyboard, Video, Mouse*) para auxiliar no gerenciamento do equipamento e No-breaks como fonte ininterrupta de energia.

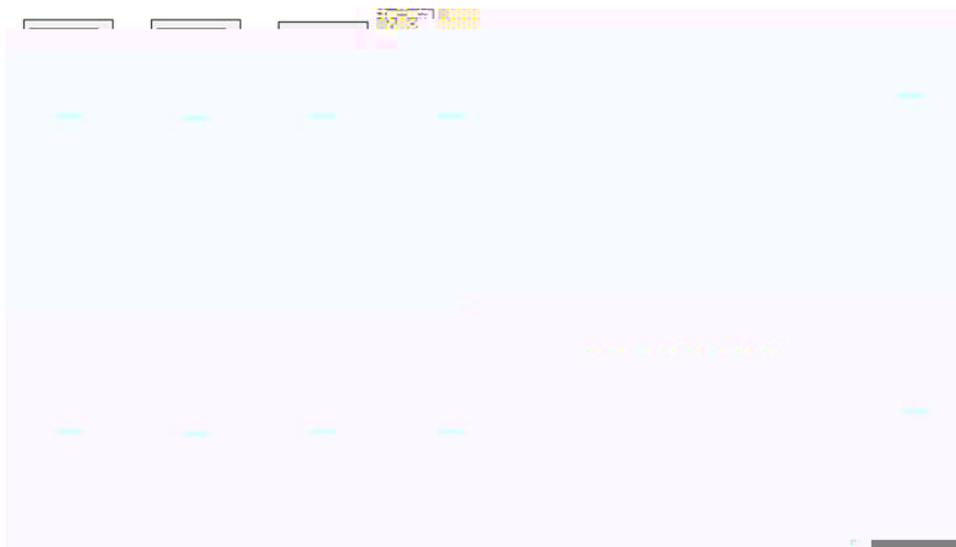


Figura 2.10 - Esquema de montagem de um Cluster Beowulf, retirado de Campregher (2005).

2.4.3 Metodologias de paralelização

Antes de estudar as metodologias de paralelização, é importante comentar sobre a Taxonomia de Flynn., Michael Flynn. Em seus artigos de 1966 e 1972, propôs uma classificação para as arquiteturas de processamento paralelo, criando o que se conhece hoje por Taxonomia de Flynn (FLYNN, 1966).

Segundo o autor, a configuração mais simples possível é a *Single Instruction / Single Data -SISD*, na qual um processamento serial convencional é realizado, por exemplo, em um PC convencional. *Single Instruction* significa que apenas uma instrução é realizada pelo processador por ciclo de *clock* e *Single Data* significa que existe apenas uma única entrada de dados a ser processada durante este mesmo ciclo.

A arquitetura paralela *Single Instruction / Multiple Data -SIMD* pode ser entendida como um paralelismo de dados, onde uma única instrução é executada paralelamente utilizando vários dados, daí a expressão *Multiple Data*. Esta classificação abrange a tecnologia MMX (*MultiMedia eXtension*) de alguns processadores modernos e também os processadores vetoriais do tipo CRAY.

Uma configuração de cunho mais teórico, que seria a *Multiple Instruction / Single Data -MISD*, onde se encaixariam as máquinas capazes de realizar várias instruções (ou seja: *Multiple Data*) sobre um único dado. Não há registro de máquinas operando segundo esta arquitetura. Entretanto, poder-se-ia imaginar um procedimento de quebra de um determinado código criptográfico sendo realizado, ao mesmo tempo, por um grupo de processadores. Cabe salientar que este problema poderia ser resolvido, também, com processamento paralelo em outras arquiteturas (CAMPREGHER, 2005).

Por fim, a arquitetura *Multiple Instruction / Multiple Data -MIMD*, que é a arquitetura utilizada neste trabalho, é caracterizada por ter cada processador agindo independentemente (*Multiple Instructions*) sobre dados diferentes (*Multiple Data*). Os processadores se comunicam usando uma rede que permite compartilhar dados e sincronizar os cálculos. São raros os problemas nos quais não seja necessária nenhuma comunicação ou sincronismo entre os processadores. Um problema que apresente esta característica é denominado "*embarrassingly parallel*".

Além da Taxonomia de Flynn, outro fator importante em computação utilizando computadores paralelos é a performance. Não é uma tarefa fácil determinar a performance de um programa paralelo, devido às diversas configurações de hardware e software possíveis. Entretanto, de forma a padronizar as relações de desempenho entre os diferentes algoritmos paralelizados, algumas definições se fazem necessárias, como, por exemplo, o *speed-up*, a escalabilidade e o desempenho.

O *speed-up* pode ser entendido como a razão entre o tempo de processamento para um único processador e o tempo para N processadores. Entretanto, cabe salientar que todo programa paralelo possui uma parte serial, impossível de ser paralelizada, tais como geração de malha, ou mesmo controle de iterações e avanço no tempo computacional. Esta parcela serial do programa e o *speed-up* estão intimamente ligados pela lei de *Amdahl*, (GUSTAFSON, 1988). Segundo esta lei, se um programa possui uma parte serial (e todo o programa a possui) que representa $1/S$ do tempo total de execução, diz-se que o *speed-up* máximo que pode ser atingido por este mesmo programa é S. A título de exemplo, se um determinado programa possui 5% de seu código serial (o que já é um tanto difícil de se obter) podemos dizer, a partir da lei de *Amdahl*, que o *speed-up* máximo a ser atingido é 20. Entretanto, na maioria dos problemas práticos, esta relação torna-se muito simplista para poder representar todas as variantes do processo, de forma que o *speed-up* pode atingir valores bem maiores que S. Uma expressão matemática prática para o *speed-up* poder ser:

$$S(N) = \frac{T_S}{T_N}, \quad (2.1)$$

onde T_S é o tempo gasto por um programa rodando em um só processador e T_N é o tempo gasto por este mesmo programa rodando em N processadores.

A eficiência (E) pode ser definida como sendo o quanto de *speed-up* é obtido à medida que novos processadores são adicionados. Uma expressão para o seu cálculo é dada por:

$$E(N) = \frac{S(N)}{N}. \quad (2.2)$$

Costuma-se, ainda, empregar o termo desempenho, que nada mais é do que $E(N) \times 100\%$. Pode-se dizer que o valor ideal de eficiência seja 1 ou, então, um desempenho de 100%.

Uma curva característica de *speed-up* em função do número de processadores pode ser vista na Figura 2.11, retirada de Marinho et al. (2004). Nesta figura, a reta a 45° representa o *speed-up* ideal, ou seja, aquele que aumenta na proporção que mais máquinas são adicionadas ao processamento. Na situação apresentada não foi possível observar o comportamento do código para um número maior de processadores. Porém, sabe-se que, à medida que eles são incluídos nos cálculos, a curva apresenta um padrão aproximadamente parabólico. Após o *speed-up* atingir o máximo possível, seu valor começa a regredir, de forma que o acréscimo de novos processadores pode prejudicar o desempenho do código. Na Figura

2.12, os resultados da eficiência do código computacional para o mesmo problema pode ser observada. A reta paralela ao eixo das abscissas e que cruza o eixo de eficiência no valor igual a 1, representa o desempenho de 100%, considerado ideal.

A capacidade que os programas possuem de aumentar seu *speed-up* à medida que mais processadores são adicionados é definida como escalabilidade. Esta propriedade é muito significativa a ponto de se tornar peça de propaganda das empresas de softwares comerciais, ao dizerem que seus produtos possuem a escalabilidade de centenas, ou mesmo milhares, de processadores. A escalabilidade perfeita, ou seja, a capacidade de manter a eficiência constante, não importando o número de processadores que são adicionados, é o "Santo Graal" do processamento paralelo (CAMPREGHER, 2005).

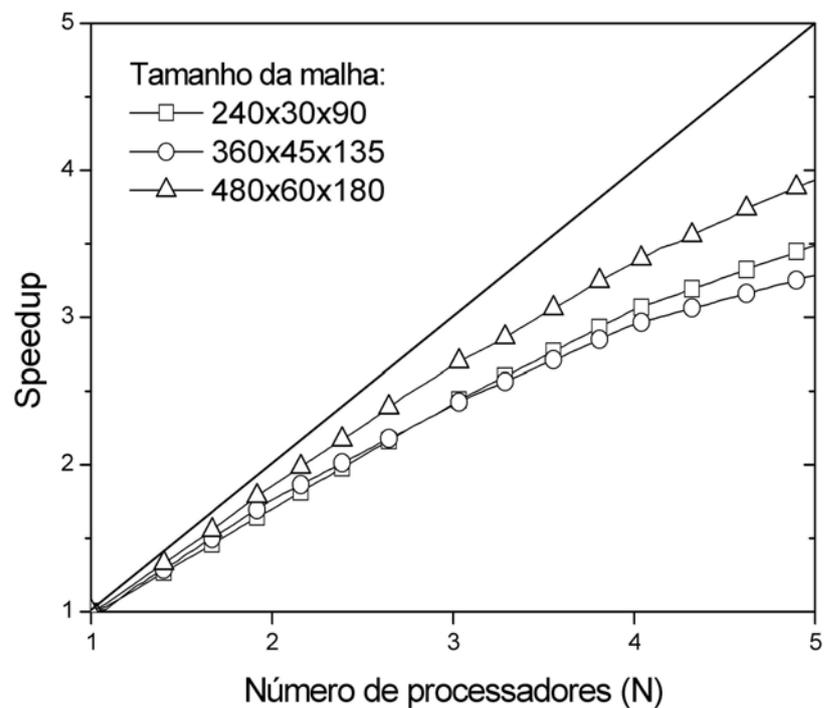


Figura 2.11 - Resultados de *speed-up* para três níveis de refinamentos diferentes (MARINHO, 2004).

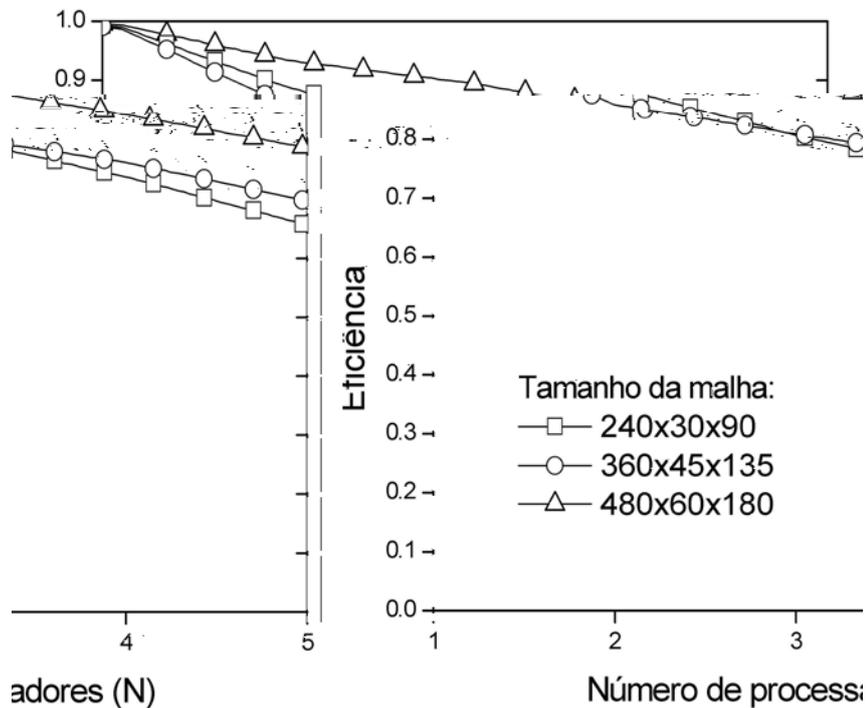


Figura 2.12 - Resultados da eficiência obtidos em três malhas diferentes (MARINHO, 2004).

Segundo Winkelmann (1998), existem basicamente três maneiras de paralelizar um código. A primeira, e mais simples, é paralelizar os *loops* do código. A maioria dos paralelizadores automáticos utiliza esta estratégia em seus algoritmos. Este conceito, no entanto não é indicado no caso de problemas muito grandes, uma vez que os ganhos em *speed-up* são muito limitados. Uma segunda abordagem é paralelizar os processos da solução numérica, por exemplo, se existir a necessidade de multiplicar um vetor por uma matriz qualquer, esta operação é distribuída entre vários processadores. Assim como a primeira opção, este conceito dificilmente é vantajoso em grandes problemas.

A terceira abordagem, que é a utilizada neste trabalho, é conhecida como decomposição de domínios, também denotada como particionamento de domínios. Neste caso o domínio é dividido em uma série de subdomínios, que tem a capacidade de trocar informações como intuito de atualizar suas fronteiras. A solução numérica é obtida em cada subdomínio, mas não de forma independente dos demais. É necessário utilizar informações armazenadas nos outros subdomínios. Este processo aumenta a escalabilidade do código, fazendo com que seja possível utilizar mais processadores de maneira otimizada, tendo, portanto, um *speed-up* maior.

CAPÍTULO III

3 Modelagem Matemática

O Método da fronteira imersa faz uso de dois domínios distintos, porém acoplados, para avaliar escoamentos sobre geometrias complexas. Um domínio euleriano é utilizado para descrever o comportamento do escoamento em si, e um domínio lagrangiano é utilizado para representar a interface imersa. É também estabelecida uma forma de interação entre o fluido e a interface imersa nele, ou seja, as duas formulações são acopladas. Os modelos de fronteira imersa buscam avaliar este acoplamento pela inserção de um termo de força às equações para o domínio fluido. Escoamentos ao redor de geometrias complexas sempre representaram, e ainda representam, sérias dificuldades para os numericistas. Nas discretizações em que se procura ajustar a malha numérica ao objeto de estudo e ao domínio, o problema surge na geração da geometria, nem sempre trivial e propensa a inserir severos erros nos balanços de massa entre seus nós elementares. Nos casos de discretizações em blocos, o acoplamento entre os diversos domínios, se mal construído, pode implicar em sérias inconsistências físicas.

Neste capítulo serão apresentadas tanto a modelagem matemática para o domínio euleriano, quanto para o domínio lagrangiano, bem como a obtenção das propriedades requeridas pelo Modelo físico virtual.

3.1 Formulação para o domínio fluido

As equações de Navier-Stokes são resolvidas em todo o domínio de cálculo. Estas equações podem ser escritas na forma tensorial para escoamentos isotérmicos e incompressíveis, como:

$$\frac{\partial(u_i)}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_i} \left[\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \frac{f_i}{\rho}, \quad (3.1)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (3.2)$$

onde ρ e ν são respectivamente a massa específica e a viscosidade cinemática, propriedades que caracterizam o fluido. As características do escoamento são representadas por: p , o campo de pressão, u_i as componentes do vetor velocidade e f_i as componentes do campo de força que atua sobre o escoamento.

No método da fronteira imersa o termo f_i , que é o termo responsável por fazer o escoamento sentir o efeito de um corpo imerso deve existir apenas nos pontos eulerianos coincidentes com a interface. Para todos os outros o valor deste termo deve ser nulo. Esta característica é alcançada com a utilização da função Delta de Dirac (

3.2. Modelo matemático para a interface fluido sólido

O cálculo da densidade de força lagrangiana é feito utilizando-se o Modelo físico virtual (*Virtual Physical Model* -VPM) proposto por Lima e Silva et al. (2003). Esse modelo avalia dinamicamente a força que o fluido exerce sobre a superfície sólida imersa no escoamento. A força lagrangiana $\vec{F}(\vec{x}, t)$ é avaliada fazendo-se um balanço de forças sobre uma partícula de fluido que se encontra junto à interface sólido-fluido, utilizando as próprias equações de Navier-Stokes. Assim a densidade de força lagrangiana pode ser dada por:

$$\vec{F}_i(\vec{x}_k, t) = \vec{F}_{acc}(\vec{x}_k, t) + \vec{F}_{inert}(\vec{x}_k, t) + \vec{F}_{visc}(\vec{x}_k, t) + \vec{F}_{press}(\vec{x}_k, t). \quad (3.4)$$

Os termos do lado direito da Eq.(3.4) são aqui respectivamente denominados por: força de aceleração, força inercial, força viscosa e força de pressão, os quais são definidos pelas equações de (3.5) a (3.8), escritas aqui na forma tensorial:

$$\vec{F}_{acc} = \rho \frac{\partial (u_{ki})}{\partial t}, \quad (3.5)$$

$$\vec{F}_{inert} = \rho \frac{\partial}{\partial x_{kj}} (u_{ki} u_{kj}), \quad (3.6)$$

$$\vec{F}_{visc} = -\frac{\partial}{\partial x_{kj}} \left[\nu \left(\frac{\partial u_{ki}}{\partial x_{kj}} + \frac{\partial u_{kj}}{\partial x_{ki}} \right) \right], \quad (3.7)$$

$$\vec{F}_{press} = \frac{\partial (p_{kj})}{\partial x_{kj}}. \quad (3.8)$$

Uma vez analisadas as forças interfaciais (provenientes do campo lagrangiano), deve-se promover o acoplamento entre as formulações representativas do domínio fluido (provenientes do campo euleriano). Este acoplamento é feito pelo processo de distribuição da força lagrangiana para o domínio euleriano e pela interpolação das velocidades e pressão eulerianas para a malha lagrangiana.

Mais detalhes sobre a modelagem e implementação do Método da fronteira imersa serão apresentados no capítulo de metodologia numérica.

CAPÍTULO IV

4 Metodologia Numérica

Nesta seção, descreve-se os procedimentos numéricos empregados na solução das equações apresentadas no capítulo anterior. Como consequência do processo de discretização resultam sistemas lineares cuja solução leva à aplicação de *solvers* para a obtenção do valor das incógnitas. Evidentemente, a transformação de um problema originalmente proposto num domínio contínuo, para uma solução possível e baseado num domínio discreto acarreta, invariavelmente, perdas na precisão da solução. Este fenômeno é conhecido como erro de discretização. Dessa forma, a busca por técnicas que minimizem o erro de discretização é uma atividade incessante por parte dos pesquisadores, demandando um equilíbrio entre custo computacional e precisão desejada, Campregher (2005).

4.1 Discretização das equações para o domínio fluido

De uma forma resumida, o código computacional da presente dissertação é descrito como: implícito, de segunda ordem no tempo e espaço, utilizando arranjo co-localizado de variáveis e algoritmo SIMPLEC com interpolação de Rhie-Chow.

A seguir é apresentada a discretização de uma equação de transporte por volumes finitos para uma variável genérica ϕ em um volume de controle elementar, Figura 4.1.

A integração de uma equação de transporte no tempo e espaço, no interior de um volume elementar como o mostrado na figura 4.1 produz (CAMPREGHER, 2005):

$$\begin{aligned} & \left(\frac{3\phi_p^n - 4\phi_p^{n-1} + \phi_p^{n-2}}{2\Delta t} \right) \Delta x \Delta y \Delta z + (\rho_e u_e \phi_e - \rho_w u_w \phi_w)^n \Delta y \Delta z + (\rho_n u_n \phi_n - \rho_s u_s \phi_s)^n \Delta x \Delta z + \\ & (\rho_t u_t \phi_t - \rho_b u_b \phi_b)^n \Delta x \Delta y = \left[\left(\Gamma^\phi \frac{\partial \phi}{\partial x} \right)_e - \left(\Gamma^\phi \frac{\partial \phi}{\partial x} \right)_w \right]^n \Delta y \Delta z + \quad (4.1) \\ & \left[\left(\Gamma^\phi \frac{\partial \phi}{\partial y} \right)_n - \left(\Gamma^\phi \frac{\partial \phi}{\partial y} \right)_s \right]^n \Delta x \Delta z + \left[\left(\Gamma^\phi \frac{\partial \phi}{\partial z} \right)_t - \left(\Gamma^\phi \frac{\partial \phi}{\partial z} \right)_b \right]^n \Delta x \Delta y + q_\phi \Delta x \Delta y \Delta z \end{aligned}$$

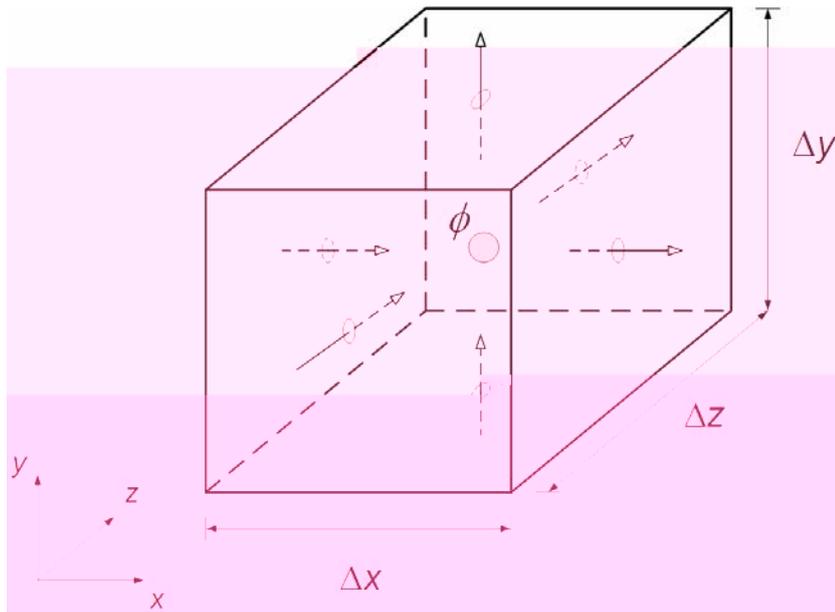


Figura 4.1 - Volume de controle elementar.

O primeiro termo do lado esquerdo da Eq. (4.1) representa a discretização do termo transiente pelo esquema *three-time level* (MUZAFERIJA; PERIC, 1997), um esquema implícito de segunda ordem que não acarreta maiores complicações numéricas ou de implementação, com passo de tempo pré-definido. Porém, ressalta-se que as informações no instante de tempo $n-2$ só são obtidas na segunda iteração. A estratégia utilizada neste trabalho é ajustar uma parábola pelos instantes de tempo t^n , t^{n-1} e t^{n-2} , separados entre si por intervalo de tempo Δt . Campregher (2005).

As aproximações para os valores do transporte de ϕ em cada uma das faces i do volume elementar da figura 4.1, pelos fluxos $F_i = F_{a,i} + F_{d,i} = \rho u_i \phi_i - \Gamma_i \left(\frac{\partial \phi}{\partial x} \right)_i$, têm um papel fundamental na consistência e estabilidade do algoritmo numérico utilizado. Os índices *e* (*east*), *w* (*west*), *n* (*north*), *s* (*south*), *t* (*top*), *b* (*bottom*) denotam as faces compartilhadas pelos volumes vizinhos (cujos centróides recebem índices maiúsculos) com o volume elementar central P. O termo $F_{a,i} = \rho u_i \phi_i$ representa a parcela advectiva e o termo $F_{d,i} = -\Gamma_i \left(\frac{\partial \phi}{\partial x} \right)_i$, a parcela difusiva do fluxo da propriedade ϕ que atravessa a face do volume elementar.

As interpolações das derivadas de primeira ordem são aproximadas por diferenças centradas, porém o valor de ϕ nas faces requer uma análise mais detalhada, pois tratando-se

do transporte de quantidade de movimento, origina termos não lineares responsáveis pela dificuldade matemática em resolver as equações de Navier-Stokes.

Esquemas de baixa ordem, em geral, injetam uma forte difusão numérica na solução numérica, estabilizando-a, mas podem gerar resultados sem consistência física. Já esquemas de alta ordem, principalmente os de diferença centrada, são conhecidos por gerar menor difusão, mas, em principalmente em altos números de Reynolds, podem produzir oscilações numéricas. Estas oscilações podem ser eliminadas com o uso de um modelo de turbulência adequado (SILVA, 2004). Esquemas que procuram mesclar as qualidades de esquemas de baixa e alta ordem são uma alternativa para se obter uma aproximação precisa e livre de oscilações numéricas.

O presente trabalho adota a estratégia conhecida como correção atrasada (*deferred-correction*) (FERZIGER; PERIC, 2002), que interpola o fluxo advectivo em uma face qualquer i , no instante de tempo n como:

$$\left(F_{a,i}\right)^n = \left(F_{a,i}^L\right)^n + \lambda \left(F_{a,i}^H - F_{a,i}^L\right)^{n-1}, \quad (4.2)$$

onde os índices L e H significam termos de baixa e alta ordem respectivamente, e o coeficiente λ permite uma combinação entre os termos. Na convergência os termos de baixa ordem se anulam, restando o termo de alta ordem $F_{a,i}^H$. A aproximação de baixa ordem empregada é o esquema *up-wind* e, para a componente de alta ordem de diferenças centradas.

Os termos no instante de tempo anterior, denotados pelo índice $n-1$ são adicionados ao termo fonte, de forma que na convergência, os termos de baixa ordem se cancelam, restando apenas o termo de alta ordem quando $\lambda = 1$. Com esse esquema, é possível ter uma aproximação de segunda ordem com as facilidades numéricas de um esquema de primeira ordem.

Com as aproximações acima descritas, a equação de balanço de quantidade de movimento apresentada no capítulo anterior pode ser integrada no tempo e espaço, resultando:

$$A_P \phi_P = A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + A_T \phi_T + A_B \phi_B + B_P, \quad (4.3)$$

onde:

$$A_I = \max(-Flux_i, 0) + Diff \quad I = E, N, T \quad i = e, n, t, \quad (4.4)$$

$$A_i = \max(Flux_i, 0) + Diff \quad I = W, S, B \quad i = w, s, b, \quad (4.5)$$

$$Flux_i = \rho_i u_i S_i \quad i = e, w, n, s, t, b, \quad (4.6)$$

$$Diff_i = \frac{\mu_i S_i}{\delta_i} \quad i = e, w, n, s, t, b, \quad (4.7)$$

$$A_p = \sum A_i + B_p \Delta V \quad I = E, W, N, S, T, B, \quad (4.8)$$

$$B_p = q_\phi \Delta V, \quad (4.9)$$

sendo: S_i a área da face do volume transversal ao escoamento, $\Delta V = \Delta x \Delta y \Delta z$, δ_i é a distância entre os centróides adjacentes a face i e B_p contém todas as componentes do termo fonte. Os índices maiúsculos se referem aos centróides dos volumes elementares. Os termos que compõem os coeficiente A_i foram calculados com base nos valores obtidos no instante de tempo anterior ($n-1$). O gradiente de pressão está incluído no termo fonte q_ϕ e é calculado por diferenças centradas tomando como referência o nó central P . A equação (4.3) aplicada a todos os volumes do domínio juntamente com as condições de contorno origina um sistema linear que ao ser resolvido fornecer valores atualizados para a variável ϕ .

Formadas por quatro incógnitas (u , w , v e p), as equações de Navier-Stokes não apresentam uma equação de transporte para a pressão, sendo necessário adotar alguma tática de fechamento do sistema de equações de forma a englobar as três componentes de velocidade juntamente com a pressão. No presente trabalho isso é feito através do algoritmo SIMPLEC descrito na próxima seção, sendo este método um preditor-corretor, ou seja, o método se baseia em uma velocidade estimada que é corrigida pela pressão de forma a satisfazer a continuidade.

4.1.1 O Algoritmo SIMPLEC

O algoritmo SIMPLEC aqui utilizado foi proposto por Van Doormal e Raithby (1984) e será descrito através da reescrita da Eq. (4.3) para o transporte de uma componente U_i^* da velocidade estimada, separando-se o gradiente de pressão do termo fonte B :

$$A_P U_{i,P}^* - \sum_I A_I U_{i,I}^* = B^{n-1} - \nabla P_{i,P}^* \Delta V, \quad (4.10)$$

onde o índice I representa todos os vizinhos ao nó central P , o sobrescrito $n-1$ denota a variável conhecida no instante de tempo anterior. O subscrito P no gradiente de pressão indica que o mesmo foi calculado com relação a esse nó. O índice $*$ indica que se trata de uma aproximação, pois ainda não se sabe o valor do campo de pressão (no instante n) que satisfaça à conservação da massa. Entretanto, se o campo correto de velocidade fosse imposto, a equação assumiria a forma:

$$A_P U_{i,P} - \sum_I A_I U_{i,I} = B^{n-1} - \nabla P_{i,P} \Delta V. \quad (4.11)$$

Subtraindo-se a Eq. (4.10) de (4.11), tem-se:

$$A_P U'_{i,P} - \sum_I A_I U'_{i,I} = B^{n-1} - \nabla P'_{i,P} \Delta V, \quad (4.12)$$

onde $U' = U - U^*$ e $P' = P - P^*$ são as correções necessárias a serem aplicadas às variáveis estimadas.

No método SIMPLEC a correção das velocidades é dada por:

$$U'_{i,P} = \frac{\sum_I A_I U'_{i,I}}{\sum_I A_I}, \quad (4.13)$$

que inserido na Eq. (4.12) fornece:

$$U_{i,P} = U_{i,P}^* - d_P^{U_i} \nabla P'_{i,P}, \quad (4.14)$$

sendo:

$$d_P^{U_i} = \frac{\Delta V}{A_P - \sum_I A_I}. \quad (4.15)$$

A equação (4.14) permite efetuar as correções nos campos de velocidade uma vez conhecido o campo de correção de pressão P' , cuja equação discretizada é obtida aplicando as condições de conservação da massa às componentes de $U_{i,P}$, resultando em:

$$A_P P'_P = A_E P'_E + A_W P'_W + A_N P'_N + A_S P'_S + A_T P'_T + A_B P'_B + B_P \quad (4.16)$$

onde:

$$A_P = \sum A_I, \quad (4.17)$$

$$A_I = \frac{\rho d_i^{U_i} S_i}{\delta_i}, \quad I = E, W, N, S, T, B \quad i = e, w, n, s, t, b \quad (4.18)$$

$$B_P = \nabla \cdot u_I^* \quad I = 1, 2, 3 \quad (4.19)$$

onde δ_i é a distância, no sentido de i , entre o centróide do volume central P e o do seu vizinho I e o termo fonte B o divergente do campo estimado de velocidades.

4.1.2 Interpolação Rhie-Chow

O uso de malhas co-localizadas tende a provocar oscilações numéricas no campo de velocidades, devidas aos campos de pressão do tipo *checkerboard* (PATANKAR, 1980). As equações de balanço de quantidade de movimento são basicamente as mesmas para arranjos co-localizados e deslocados. Entretanto, para o arranjo co-localizado, as velocidades que são necessárias nas faces devem ser interpoladas a partir do centro do volume de controle. A função de interpolação mais usada é a interpolação de Rhie-Chow (RHIE; CHOW, 1983).

Rearranjando a Eq. (4.10):

$$A_P U_{i,P}^* + \nabla P_{i,P}^* \Delta V_P = B_P^{n-1} + \left(\sum_I A_I U_{i,I}^* \right)_P. \quad (4.20)$$

Aplicando-se a mesma equação para o ponto nodal E , tem-se:

$$A_E U_{i,E}^* + \nabla P_{i,E}^* \Delta V_E = B_E^{n-1} + \left(\sum_I A_I U_{i,I}^* \right)_E. \quad (4.21)$$

E para uma face e entre os pontos nodais P e E :

$$A_e U_{i,e}^* + \nabla P_{i,e}^* \Delta V_e = B_e^{n-1} + \left(\sum_I A_I U_{i,I}^* \right)_e. \quad (4.22)$$

Considerando-se que o lado esquerdo da Eq. (4.22) pode ser aproximado, na interface e , por uma interpolação linear (termos representados por uma barra superior) a partir das Eqs. (4.20) e (4.21), segue-se que:

$$\overline{A_e U_{i,e}^* + \nabla P_{i,e}^* \Delta V_e} = \overline{B_e^{n-1} + \left(\sum_I A_I U_{i,I}^* \right)_e} = \overline{A_e U_{i,e}^*} + \overline{\nabla P_{i,e}^* \Delta V_e}. \quad (4.23)$$

Rearranjando-se a equação acima e assumindo que $A_e \approx \overline{A_e}$:

$$U_{i,e} = \overline{U_{i,e}^*} - \overline{d_e^{U_i}} \left(\overline{\nabla P_{i,e}^*} - \nabla P_{i,e}^* \right), \quad (4.24)$$

onde:

$$\overline{U_{i,e}^*} = \alpha U_{i,P}^* + (1 - \alpha) U_{i,E}^*, \quad (4.25)$$

$$\overline{\nabla P_{i,e}^*} = \alpha \nabla P_{i,P}^* + (1 - \alpha) \nabla P_{i,E}^*, \quad (4.26)$$

$$\nabla P_{i,e}^* = \frac{P_E - P_P}{\delta x_e}, \quad (4.27)$$

$$\nabla P_{i,P}^* = \frac{P_E - P_W}{\delta x_e + \delta x_w}, \quad (4.28)$$

$$\nabla P_{i,E}^* = \frac{P_{EE} - P_P}{\delta x_{ee} + \delta x_e}, \quad (4.29)$$

$$\overline{d_e^{U_i}} = \alpha d_p^{U_i} + (1 - \alpha) d_E^{U_i}. \quad (4.30)$$

Percebe-se claramente que α é um coeficiente de interpolação e costuma ser baseado na distância entre os pontos nodais envolvidos. Alguns autores preferem um valor de $\alpha = 0,5$ para a interpolação da pressão e um valor ponderado pela distância para as velocidades. No presente trabalho foi adotada a interpolação ponderada para todas as variáveis.

Agora as velocidades nas faces do volume dependem da pressão nos nós adjacentes, exatamente como no caso de malhas deslocadas, permitindo que o mesmo tipo de acoplamento pressão-velocidade seja usado.

Os sistemas de equações gerados pelas discretizações são resolvidos utilizando dois *solvers* distintos. As equações de quantidade de movimento são resolvidas utilizando o SOR (*Successive Over Relaxation*), enquanto a equação de correção da pressão é resolvida com o SIP (*Strong Implicit Procedure*).

4.2 Discretização das equações representativas do domínio lagrangiano

A Discretização da Eq (3.4) é feita utilizando um sistema de eixos de referência tridimensional, com sua origem localizada no ponto k , como pode ser visto na Figura 4.2. Um polinômio de lagrange é então usado para obter as derivadas espaciais ao longo da coordenada em questão. Seja m o número de pontos utilizados na construção de uma interpolação polinomial de ordem $m-1$, assim o valor da propriedade ϕ ao longo da direção i , em qualquer ponto p é dado por:

$$\phi_i(p) = \sum_m \psi_m(p) \phi_m, \quad (4.31)$$

onde,

$$\psi_m(p)_i = \prod_{n, n \neq m} \left[\frac{x_i(p) - x_i(n)}{x_i(m) - x_i(n)} \right], \quad (4.32)$$

Substituindo os m pontos, de acordo com o estêncil da Figura 4.2 o valor da propriedade ϕ ao longo do eixo x (sobre o qual k , k_1 e k_2 se localizam) obtém-se:

$$\phi_p = \left[\frac{(x_p - x_{k1})(x_p - x_{k2})}{(x_k - x_{k1})(x_k - x_{k2})} \right] \phi_k + \left[\frac{(x_p - x_k)(x_p - x_{k2})}{(x_{k1} - x_k)(x_{k1} - x_{k2})} \right] \phi_{k1} + \left[\frac{(x_p - x_k)(x_p - x_{k1})}{(x_{k2} - x_k)(x_{k2} - x_{k1})} \right] \phi_{k2}. \quad (4.33)$$

Derivando a equação (4.33) na direção x tem-se:

$$\frac{\partial \phi_p}{\partial x} = \left[\frac{(x_p - x_{k1}) + (x_p - x_{k2})}{(x_k - x_{k1})(x_k - x_{k2})} \right] \phi_k + \left[\frac{(x_p - x_k) + (x_p - x_{k2})}{(x_{k1} - x_k)(x_{k1} - x_{k2})} \right] \phi_{k1} + \left[\frac{(x_p - x_k) + (x_p - x_{k1})}{(x_{k2} - x_k)(x_{k2} - x_{k1})} \right] \phi_{k2} \quad (4.34)$$

e a segunda derivada resulta em:

$$\frac{\partial^2 \phi_p}{\partial x^2} = \left[\frac{2\phi_k}{(x_k - x_{k1})(x_k - x_{k2})} \right] + \left[\frac{2\phi_{k1}}{(x_{k1} - x_k)(x_{k1} - x_{k2})} \right] + \left[\frac{2\phi_{k2}}{(x_{k2} - x_k)(x_{k2} - x_{k1})} \right]. \quad (4.35)$$

Das equações acima é possível obter as derivadas espaciais necessárias na equação (3.4), simplesmente substituindo o ponto p , e a variável desejada ϕ .

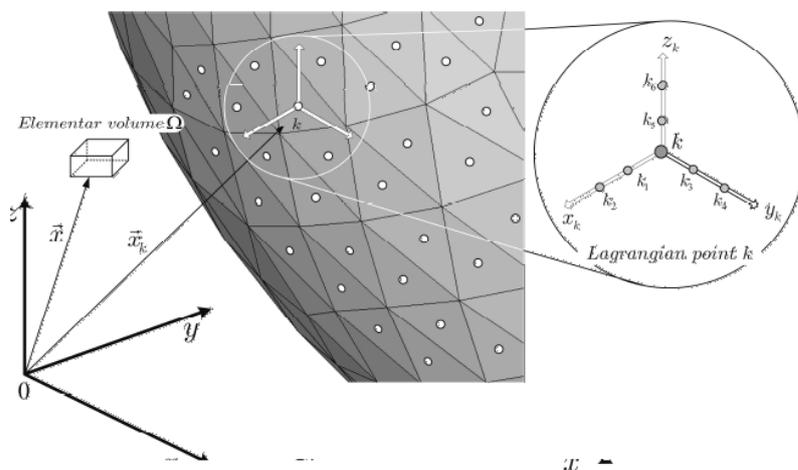


Figura 4.2 - Posição de um ponto lagrangiano \bar{x}_k (VEDOVOTO et al. 2006).

No caso do presente trabalho é utilizada uma malha de elementos triangulares com a finalidade de representar o objeto imerso no escoamento. Uma vista detalhada de um elemento triangular pode ser visto na Figura 4.3. Os lados dos elementos são formados por segmentos designados por S_1 , S_2 e S_3 , entre os vértices P_1 , P_2 e P_3 . Assim tem-se $S_1 = \overline{P_1P_2}$, $S_2 = \overline{P_3P_2}$ e $S_3 = \overline{P_3P_1}$.

A área de superfície do elemento triangular ΔA , pode ser avaliada como:

$$\Delta A_k = \sqrt{S(S - S_1)(S - S_2)(S - S_3)}, \quad (4.36)$$

onde $S = (1/2)(S_1 + S_2 + S_3)$ e ΔS_k é o comprimento médio dos lados do triângulo. É importante notar que cada um dos parâmetros geométricos citados estão associados à um ponto lagrangiano k .

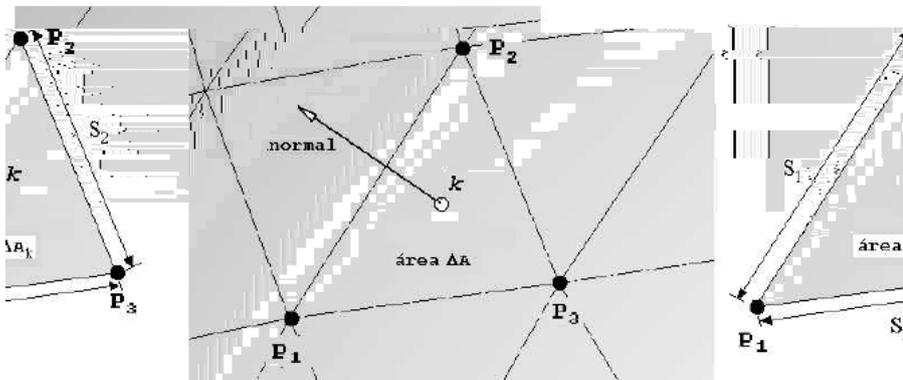


Figura 4.3 - Vista detalhada de um elemento triangular (VEDOVOTO et al. 2006).

Segundo Lima e Silva (2002), é necessário que do tamanho característico de cada malha lagrangiana (ΔS_k) seja aproximadamente igual ao tamanho da malha euleriana na região (Δx), isto porque se a disparidade entre os tamanhos característicos das malhas lagrangiana e euleriana forem muito altas, podem ocorrer problemas como perfuração de linhas de corrente (indicando que a densidade de força gerada pela fronteira imersa não foi o suficiente para “afastar” o fluido da região da fronteira). Pode haver um aumento no coeficiente

de arrasto, por exemplo, devido ao fato de a fronteira imersa ter gerado uma força maior que a força fisicamente correta.

4.2.1 O procedimento de distribuição da força lagrangiana

A forma como \vec{f} é avaliado, determina as diferentes versões das metodologias de fronteira imersa existentes na atualidade. O termo de força interfacial \vec{F} , obtido no domínio lagrangiano (representado doravante por Ω_k), pode ser interpolado para o domínio euleriano (representado por Ω) com o auxílio da função delta de Dirac. Em um espaço N-dimensional esta função é definida como:

$$\vec{f}(\vec{x}) = \int_{R^n} \delta(\vec{x} - \vec{x}_k) \vec{F}(\vec{x}_k) d^n \vec{x}_k . \quad (4.37)$$

Aplicando a equação (4.37) em um volume V do domínio lagrangiano temos:

$$\vec{f}(\vec{x}) = \int_{\Omega_k} \vec{F}(\vec{x}_k) \delta(\vec{x} - \vec{x}_k) d^n \vec{x}_k . \quad (4.38)$$

A função δ tem a seguinte propriedade:

$$\int_{R^n} \delta(\vec{x} - \vec{x}_k) d^n \vec{x} = \begin{cases} 1 & \text{if } \vec{x}_k \in V \\ 0 & \text{if } \vec{x}_k \notin V \end{cases} . \quad (4.39)$$

onde $V \subset \Omega$. Esta função age como o núcleo de uma transformada integral (centrada em \vec{x}_k), que promove a transposição entre os domínios

característica, ponderar o valor de cada força lagrangiana em função da sua distância em relação à cada volume euleriano.

No Modelo físico virtual, para domínios tridimensionais, o campo de força lagrangiano $F_{i,k}$ é distribuído sobre a malha euleriana utilizando a equação (4.40).

$$f_i = \sum_k F_{i,k} D_i \Delta A_k \Delta S_k . \quad (4.40)$$

Neste trabalho, a função distribuição D_i é avaliada como:

$$D_i(x_k) = \prod_i \left\{ \frac{\varphi[(x_k - x_i) / \Delta x_i]}{\Delta x_i} \right\}, \quad (4.41)$$

onde φ é uma função definida como:

$$\varphi(r) = \begin{cases} \tilde{\varphi}(r) & \text{if } \|r\| < 1 \\ \frac{1}{2} - \tilde{\varphi}(2-r) & \text{if } 1 < \|r\| < 2 \\ 0 & \text{if } \|r\| > 2 \end{cases}, \quad (4.42)$$

$$\tilde{\varphi}(r) = \frac{3 - 2\|r\| + \sqrt{1 - 4\|r\| + 4\|r\|^2}}{8}. \quad (4.43)$$

A função distribuição é dividida por unidade de volume, multiplicando esta função por uma área característica (ΔA_k) e por um comprimento característico (ΔS_k), consegue-se obter o valor da densidade de força, que é integrada sobre um volume Ω .

A interface sólido-fluido é gerenciada pelo uso de uma função indicadora I_i , Campregher (2005), construída como:

$$\nabla^2 I_i = \nabla G_i, \quad (4.44)$$

onde a função G é definida como:

$$G_i = \sum D_i \vec{n}_k \Delta A_k , \quad (4.45)$$

e \vec{n}_k é vetor normal ao ponto lagrangiano k.

Após a Discretização da equação (4.44), o sistema algébrico de equações é avaliado pelo algoritmo MSI (SCHNEIDER; ZEDAN, 1981), uma variação do solver SIP. Analisando a equação (4.45), pode-se notar que se a geometria for inserida em uma região não uniforme da malha, esta região da geometria pode ficar deformada, e portanto pode acontecer de o corpo imerso ficar mal representado, levando à incoerências físicas.

De uma forma concisa, pode – se descrever o procedimento empregado no cálculo da força euleriana como:

1. A partir do campo resolvido de variáveis

CAPÍTULO V

5 Metodologia de importação de geometrias complexas e alterações no *set-up* do programa Fluids-3D

Uma das grandes vantagens do Método da fronteira imersa é a facilidade que este método oferece para tratar escoamentos sobre geometrias complexas. Sendo assim, é de fundamental importância que se possa importar geometrias das mais variadas formas e dimensões.

A solução numérica de um problema segue basicamente três etapas: o pré-processamento, o processamento e o pós-processamento. Durante a fase de pré-processamento, são definidas as características gerais do problema, bem como suas condições iniciais e de contorno. Tomando como exemplo o escoamento sobre uma geometria qualquer imersa em um escoamento, na fase do pré-processamento, esta geometria deverá ser construída de tal forma que possa ser lida e identificada pelo código numérico responsável pela solução das equações regentes do escoamento. Se este processo de geração e importação de geometrias e *set-up* do problema não for realizado de maneira conveniente para o usuário, grande parte da funcionalidade do Método da fronteira imersa é perdida. Na fase de processamento (comumente chamado de *solver*), as equações governantes são resolvidas. Nesta fase estão embutidas, em um código computacional, todas as características dos métodos numéricos utilizados e, assim sendo, esta fase é a responsável pela solução do problema em si. Finalmente, com as equações algébricas resolvidas, fornecendo a solução do problema, inicia-se a terceira fase da solução numérica de um problema: o pós-processamento.

Após a solução das equações algébricas, o *solver*, como resultado, libera arquivos que podem chegar a centenas de *megabytes* de dados, arquivos que, para cada ponto discreto do escoamento um conjunto de valores representará as grandezas calculadas, como pressão, velocidades, viscosidade etc... Um arquivo em formato de planilha como este é algo incompreensível. A melhor forma de extrair informações destes dados é utilizando técnicas de visualização científica, ou seja, apresentar de forma gráfica as várias propriedades do escoamento, Fortuna, 2000. O objetivo é facilitar, por imagens, o entendimento dos processos que ocorrem no escoamento.

O objetivo deste capítulo é mostrar as alterações realizadas no código Fluids-3D, de forma a otimizar e facilitar o pré-processamento de um problema numérico de mecânica dos

fluidos, tanto do ponto de vista de *set-up* quanto com relação à importação de geometrias para caracterização de corpos imersos. Apesar de o pré-processamento ser uma etapa que nem sempre é avaliada com o devido cuidado, é nesta fase que pode se dizer se um programa será bem sucedido ou não na tarefa de resolver problemas de mecânica dos fluidos computacional. O código computacional deve ser escrito de forma a facilitar ao máximo a geração e importação de geometrias, assim como qualquer outra variável de *set-up*, como condições de contorno até mesmo o número e em quais máquinas o programa vai ser executado, no caso de processamento paralelo.

Este capítulo é dividido em duas partes: As alterações realizadas no código de forma a facilitar o *set-up* de um problema, e uma segunda parte mostrando alterações na forma de importar geometrias complexas.

5.1 Alterações de forma a facilitar o uso e *set-up* do código Fluids-3D

Sabe-se que quanto maior a necessidade de alterações em linhas de código em um programa a ser compilado, maior a possibilidade de inserir erros ao código, portanto, quanto menos necessidade de alterações internas, melhor. A primeira versão do código Fluids-3D, apesar de funcional, era problemática quanto ao pré-processamento e *set-up* de novas simulações, uma vez que o foco do trabalho era a simulação sobre esferas, estivessem em repouso ou sobre efeito de interação fluido-estrutura, Campregher, 2005. Desta maneira, o *set-up* de um novo caso de simulação não era algo corriqueiro, não importando, portanto quantas alterações internas seriam necessárias para fazer uma nova simulação.

Dado que a ferramenta citada (o código Fluids-3D) mostrou-se uma promissora ferramenta na análise de escoamentos incompressíveis complexos, surgiu a necessidade de que este pudesse ser utilizado por outras pessoas, não importando, inclusive, o nível de conhecimento de programação, processamento paralelo, métodos numéricos do usuário, isto porque, tem-se a intenção de utilizar este programa para fins acadêmicos, como ferramenta de introdução de métodos numéricos para solução de problemas de mecânica dos fluidos. Assim sendo a ferramenta deveria ser preparada de forma a poder ser utilizada sem que houvesse a necessidade de re-compilação, atividade esta que requer um certo domínio do programa, domínio este que exige certo conhecimento, desde processamento paralelo, até utilização do sistema operacional LINUX. Para que isto seja possível uma série de arquivos-texto foi criada. Estes arquivos funcionam como arquivos de suporte e são somente lidos pelo programa executável, eliminando a necessidade de re-compilação do código fonte. São eles:

- Arquivo de posicionamento de sondas (**.prb*): neste arquivo a quantidade e posicionamento tanto de sondas pontuais quanto de sondas lineares e planares são descritas.
- Arquivos de condições de contorno e condições iniciais (**.bnd*): nos dados destes arquivos estão contidas informações tais como, valores de propriedades físicas, passo de tempo utilizado pelo solver, tipos de condição de contorno para as paredes do domínio, velocidades e temperaturas iniciais, etc. Nestes arquivos é que se determinam as características físicas do domínio de cálculo. É também nele que se determina a natureza do arquivo que será utilizado para a caracterização da fronteira imersa (estereolitografia (**.stl*) ou arquivos de nós e conectividades).
- Arquivo de *set-up* da malha euleriana (**.msh*): como já dito anteriormente o código Fluids-3D utiliza malhas cartesianas não uniformes para representar o domínio euleriano. Apesar da geração desta malha em si estar embutida no solver, o *set-up* da malha, ou seja as características geométricas da malha, como tamanho e numero de zonas onde a malha é contraída ou expandida, estão descritas no arquivo **.msh*.
- Arquivos de *set-up* de paralelização (**.procs*): Os dados deste arquivo informam ao programa executável em quantas partes o domínio de cálculo vai ser subdividido, e conseqüentemente a topologia de paralelização (as metodologias de paralelização implementadas serão explicadas no capítulo seguinte). Além disto é neste arquivo que estão as informações do posicionamento da fronteira imersa no interior do domínio. Este posicionamento nada mais é que uma translação nos pontos da malha lagrangiana, de forma a facilitar a criação da mesma.
- Arquivo *machines.linux*: este arquivo, apesar de não ser lido diretamente no programa é importante pois é utilizado para dizer em quais nós do cluster o programa será executado.
- Arquivo *readmain.sup*: assim como o arquivo de *set-up* de paralelização, o arquivo *readmain.sup* é uma das principais alterações realizadas no código. Neste arquivo estão listados os nomes de todos os arquivos que servem de suporte para execução do programa, bem como o endereço computacional de onde o programa será executado. Esta foi uma mudança simples, mas significativa, pois a partir deste arquivo eliminou-se a necessidade constante de re-compilação, tornando o código apto a receber uma interface gráfica externa (atualmente em desenvolvimento, maiores informações podem ser encontradas

em Oliveira, 2007). Além disto esta mudança elimina a necessidade de um

5.2.1 Mudanças na importação de arquivos de nós e conectividades de uma malha triangular

Como já explicado anteriormente, a primeira versão do Fluids-3D visava somente simular esferas de um único diâmetro, portanto, o processo de edição das listas de nós e conectividades só foram feitos uma vez. No caso do presente trabalho a necessidade de criar e simular escoamentos de vários tipos de geometrias complexas exigiu a necessidade de alterações também na leitura destes tipos de arquivos. Fazendo que a alteração dos arquivos de nós e conectividades fosse eliminada, como é atualmente. O processo de importação de geometrias com o uso das listas de nós e conectividades sem a necessidade de alteração acrescentou uma facilidade significativa no *set-up* de um problema de escoamento sobre geometrias complexas.

```

0.000000000000    1.000000000000    0.000000000000
0.000000000000    0.000000000000    0.000000000000
0.000000000000    0.960000000000    0.000000000000
0.000000000000    0.920000000000    0.000000000000
0.000000000000    0.880000000000    0.000000000000
0.000000000000    0.840000000000    0.000000000000
0.000000000000    0.800000000000    0.000000000000
0.000000000000    0.760000000000    0.000000000000
0.000000000000    0.720000000000    0.000000000000
0.000000000000    0.680000000000    0.000000000000
0.000000000000    0.640000000000    0.000000000000
0.000000000000    0.600000000000    0.000000000000
0.000000000000    0.560000000000    0.000000000000
0.000000000000    0.520000000000    0.000000000000
0.000000000000    0.480000000000    0.000000000000
0.000000000000    0.440000000000    0.000000000000
0.000000000000    0.400000000000    0.000000000000
0.000000000000    0.360000000000    0.000000000000
0.000000000000    0.320000000000    0.000000000000
0.000000000000    0.280000000000    0.000000000000
0.000000000000    0.240000000000    0.000000000000
0.000000000000    0.200000000000    0.000000000000
0.000000000000    0.160000000000    0.000000000000
0.000000000000    0.120000000000    0.000000000000
0.000000000000    0.800000000000E-01    0.000000000000
0.000000000000    0.400000000000E-01    0.000000000000
1.000000000000    1.000000000000    0.000000000000
0.960000000000    1.000000000000    0.000000000000
0.920000000000    1.000000000000    0.000000000000
0.880000000000    1.000000000000    0.000000000000
0.840000000000    1.000000000000    0.000000000000

```

Figura 5.1 - Exemplo de arquivo de nós utilizado na primeira versão do código Fluids-3D.


```
LIST ALL SELECTED ELEMENTS. (LIST NODES)
```

ELEM	MAT	TYP	REL	ESY	SEC	NODES			
1	1	1	1	0	1	737	748	743	743
2	1	1	1	0	1	737	738	748	748
3	1	1	1	0	1	736	751	735	735
4	1	1	1	0	1	744	751	735	735
15	1	1	1	0	1	755	52	100	100
16	1	1	1	0	1	53	52	755	755
17	1	1	1	0	1	124	754	76	76
18	1	1	1	0	1	754	27	76	76
19	1	1	1	0	1	28	27	754	754
20	1	1	1	0	1	77	753	123	123
ELEM	MAT	TYP	REL	ESY	SEC	NODES			
21	1	1	1	0	1	753	2	26	26
22	1	1	1	0	1	77	2	753	753
23	1	1	1	0	1	147	752	51	51

Figura 5.4 - Exemplo de arquivo de conectividades utilizado na versão atual do código Fluids-3D.

A importância da não necessidade de alteração destes arquivos é óbvia quando existe a necessidade da geração de uma geometria como a da Figura 5.5, por exemplo. Neste caso, cada um dos arquivos de nós e conectividades tinham mais de 30.000 linhas, ou seja, a edição destes arquivos seria extremamente trabalhosa, custando um tempo precioso.

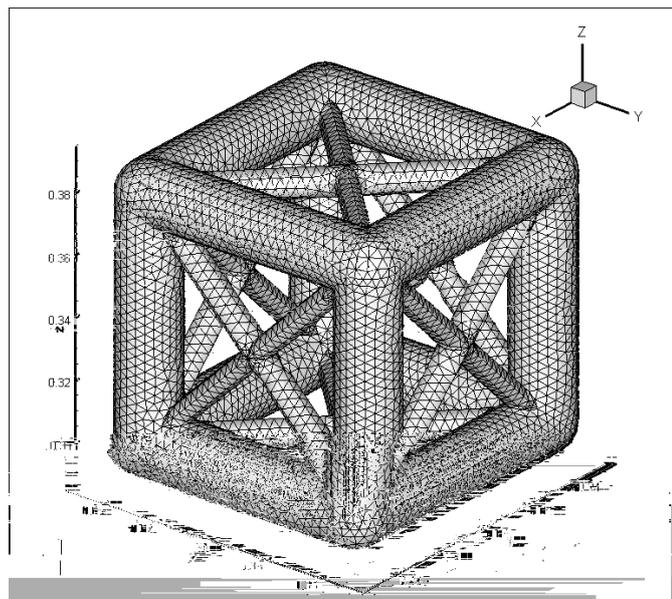


Figura 5.5 - Exemplo de malha triangular gerado a partir de arquivos de nós e conectividades Vedovoto et al. (2006).

Outros exemplos de geometrias geradas por esta metodologia são apresentadas pelas figuras abaixo.

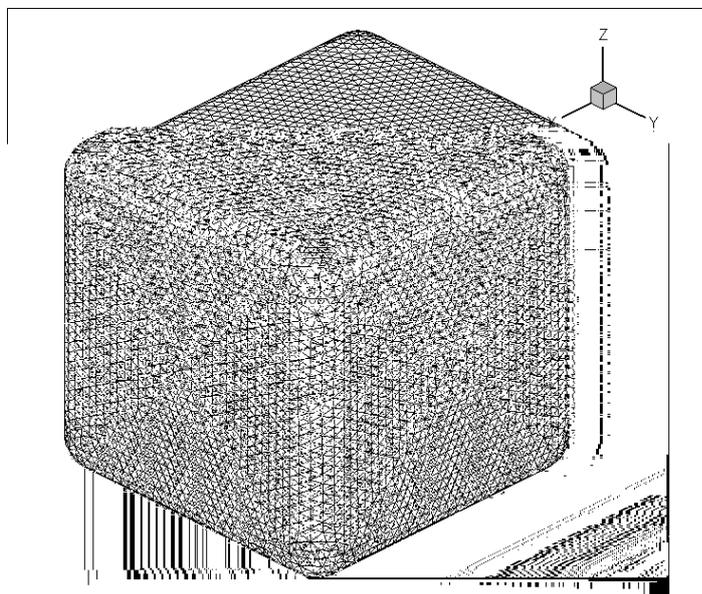


Figura 5.6 - Cubo gerado a partir de geradores de malhas comercial.

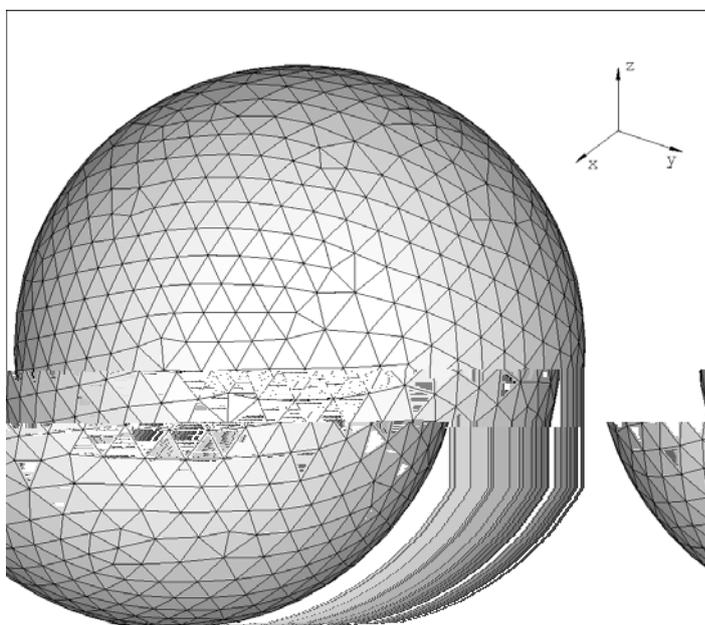


Figura 5.7 - Esfera gerada a partir de gerador de malhas comercial (VEDOVOTO et al. 2006).

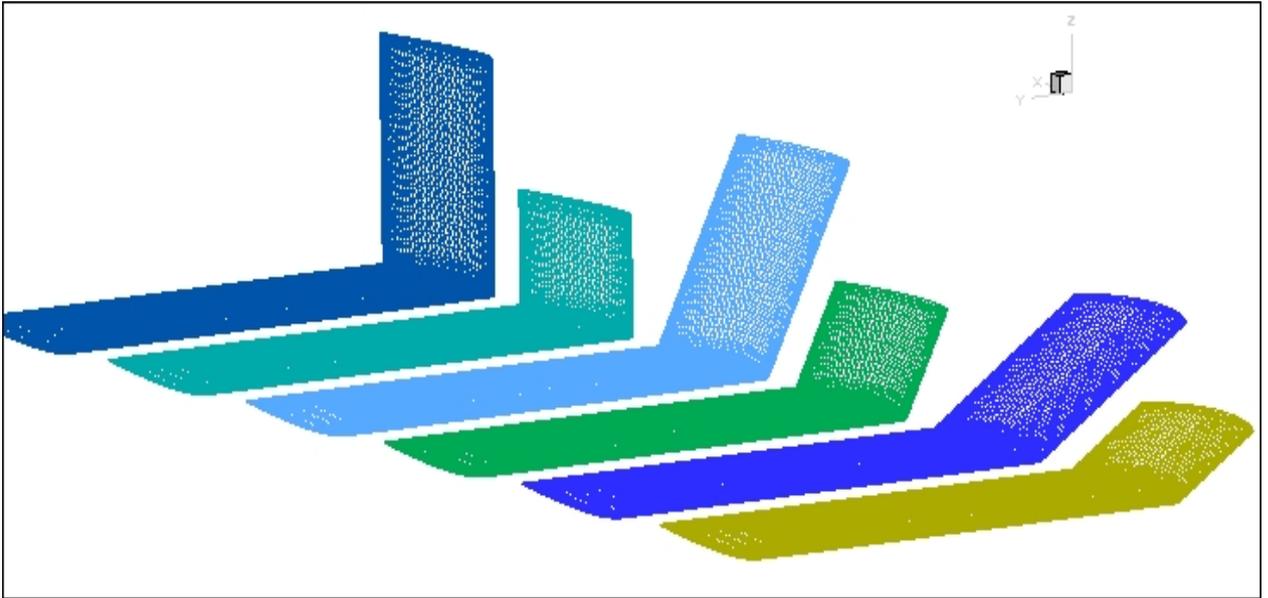


Figura 5.8 - Aerofólios com *winglets* de varias dimensões criados em um gerador de malhas comercial.

Cabe salientar que todas as malhas mostradas neste tópico foram primeiramente desenhadas em um software CAD e depois importadas pelo gerador de malha. Neste trabalho, como gerador de malhas comercial, foi utilizado o software Ansys (www.ansys.com).

5.2.2 Importação de geometrias geradas pelo processo de estereolitografia

Máquinas de estereolitografia (Figura 5.9) são basicamente impressoras 3D, que podem construir qualquer volume a partir de uma série de camadas. Este processo de fabricação utiliza uma resina foto-sensível à raios UV, e um laser para construir peças por camadas. Cada peça (ou parte), é criada camada a camada por um feixe de laser, que, quando em contato com a resina polimérica solidifica-a. Após o término do processo de criação, a peça recém-fabricada, é inserida em um forno de ondas UV, e a peça final é solidificada. A Figura 5.10 e a Figura 5.11 são exemplos de peças fabricadas a partir deste processo.



Figura 5.9 - Exemplo de máquina de estereolitografia - (http://www.stereolithography.com/stereolithography_images.php/).



Figura 5.10 - Exemplo de arma de *paint-ball*, criada em uma impressora 3D – (http://www.stereolithography.com/stereolithography_images.php/).

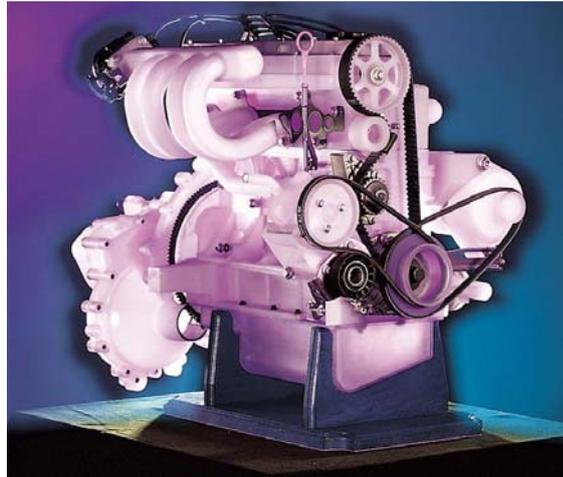


Figura 5.11 - Típico motor de combustão interna criado em uma impressora 3D – (http://www.stereolithography.com/stereolithography_images.php).

Para representação de um desenho a ser impresso em uma impressora 3D o tipo de arquivo utilizado é **.stl* (*Standard Tessellation Language*). Um arquivo **.stl* é uma representação triangular da superfície de uma geometria 3D. Este tipo de arquivo representa um objeto tridimensional criando uma malha de elementos triangulares que envolvem toda a superfície do mesmo, como pode ser visto na Figura 5.12

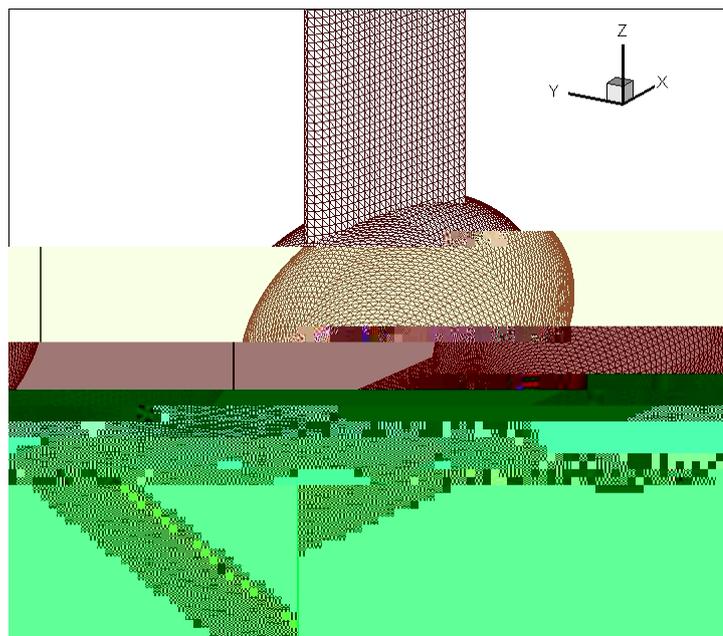


Figura 5.12 - Trecho de uma turbina eólica representada por um arquivo **.stl*.

Para cada triângulo, chamado de *face* no arquivo **.stl*, são dadas informações como a posição dos vértices e a direção do vetor normal à esta face. A Figura 5.13 mostra um pequeno trecho de um arquivo **.stl*.

```
facet normal n1 n2 n3
  outer loop
    vertex v11 v12 v13
    vertex v21 v22 v23
    vertex v31 v32 v33
  endloop
endfacet
```

Figura 5.13 - Trecho de um arquivo **.stl*. ([http://en.wikipedia.org/wiki/STL_\(file_format\)](http://en.wikipedia.org/wiki/STL_(file_format))).

Vale lembrar que em um arquivo **.stl* o trecho de texto mostrado na Figura 5.13, pode ser repetido milhares de vezes. Isto dependerá do refinamento da malha de elementos triangulares que representa uma geometria.

O fato de uma geometria qualquer poder ser representada por uma malha de elementos triangulares em um arquivo **.stl*, juntamente com o fato do código Fluids-3D representar a fronteira imersa a partir de uma malha de elementos triangulares, faz com que o uso de um arquivo **.stl* para a representação de uma geometria complexa no Fluids-3D seja totalmente adequado.

Uma outra grande vantagem do uso de arquivos **.stl*, é que este não necessita de ferramentas proprietárias para sua criação. Diversas ferramentas como o *Gmsh*, *Admesh* e *Meshlab*, por exemplo, são *softwares* livres e, portanto seu uso é gratuito (OLIVEIRA, 2007).

Com o intuito de demonstrar a capacidade do código Fluids-3D em lidar com escoamentos sobre quaisquer geometrias imersas, algumas geometrias, como a da Figura 5.14, criada com o software *Gmsh*, foram realizadas.

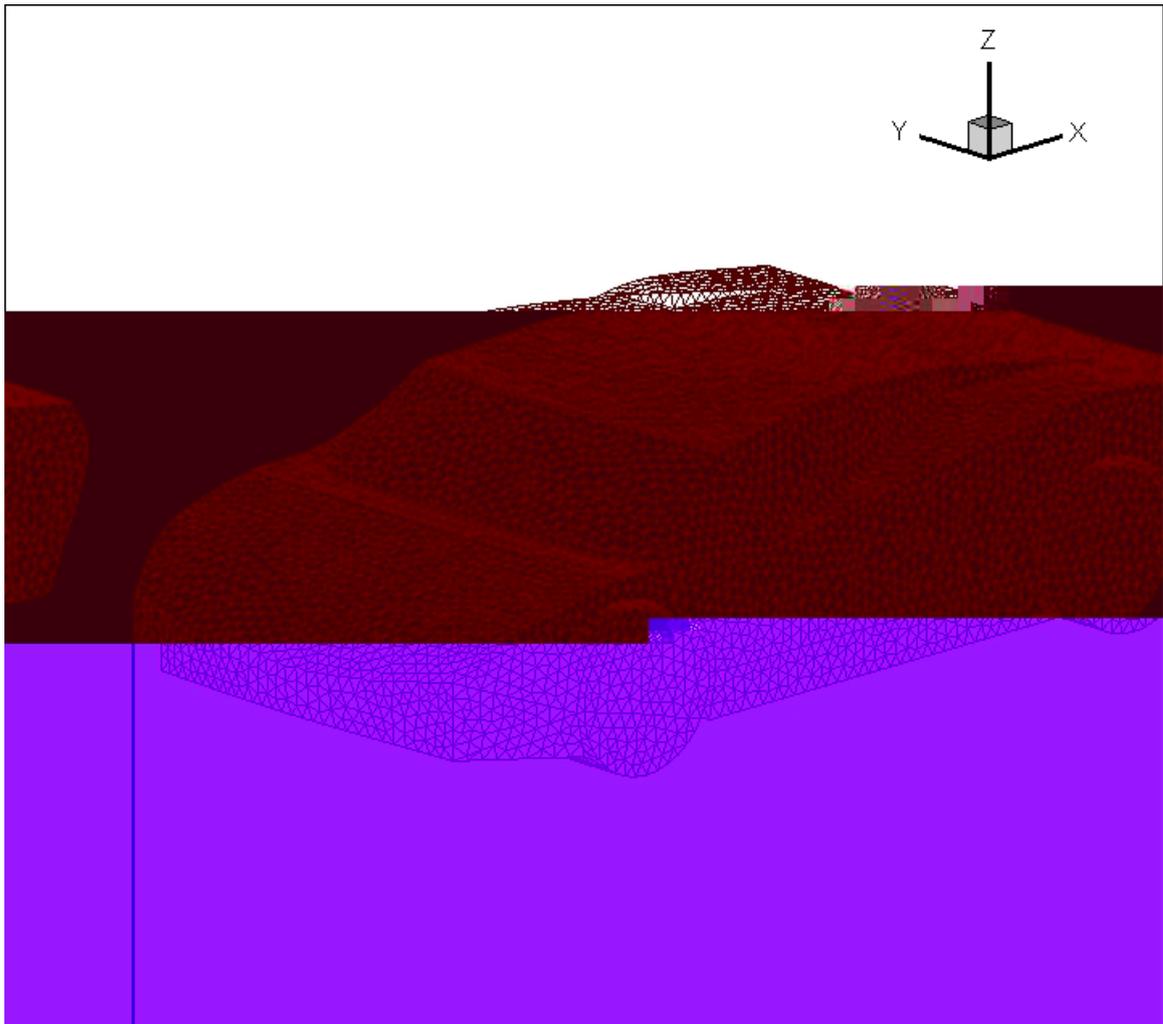


Figura 5.14 - Malha *.stl, gerada sobre um protótipo de automóvel (OLIVEIRA, 2007).

Os resultados de simulações de escoamentos sobre geometrias como a da Figura 5.14 serão apresentados e comentados no capítulo de resultados.

Capítulo VI

6 Metodologias de paralelização utilizadas no Fluids-3D

Como já mencionado em capítulos anteriores o uso de processamento paralelo em mecânica dos fluidos computacional é inevitável. Portanto este é um tópico que merece atenção especial. No presente trabalho a topologia e paralelização do código Fluids-3D foi alterada profundamente, visando tanto uma maior autonomia quanto aos domínios computacionais a serem simulados, e visando conseqüentemente, uma diminuição no tempo de processamento de uma simulação.

6.1 Tipos de paralelização possíveis no código Fluids-3D

Na paralelização escolhida no código Fluids-3D, cada processador executa basicamente o mesmo código, porém sobre uma base de dados diferente. Isto exige que se faça uma comunicação entre os processadores para que as condições de contorno sejam satisfeitas. Neste tipo de paralelização escolhido, o domínio original é subdividido em subdomínios menores, de forma a se particionar a carga de trabalho do modo mais igualitário possível entre os processadores, otimizando o uso do cluster e conseqüentemente melhorando o *speed-up*.

A topologia apresentada pela interface de comunicação entre os subdomínios pode variar de unidimensional a tridimensional, conforme mostrada da esquerda para a direita na Figura 6.1. Na topologia unidimensional, a troca de mensagens é realizada em uma única direção, que é estendida a duas direções na topologia bidimensional e três para o caso tridimensional, sempre relacionado a um volume localizado na interface. A título de exemplo, tome-se o estêncil gerado na interface de troca de mensagens entre os subdomínios: dividindo-os empregando-se a topologia unidimensional, implicaria no intercâmbio de informações ao longo de um único eixo coordenado. No caso bidimensional, os dois eixos do estêncil trocariam mensagens e, por fim, no caso tridimensional, os três eixos sofreriam intercâmbio de dados. A relação de diminuição da área superficial referente à escolha da topologia está apresentada na Figura 6.2. Cabe salientar que o volume computacional no interior de cada subdomínio permanece como sendo $V_i = i^3 / N$ independentemente do tipo de topologia escolhida, Campregher (2005).

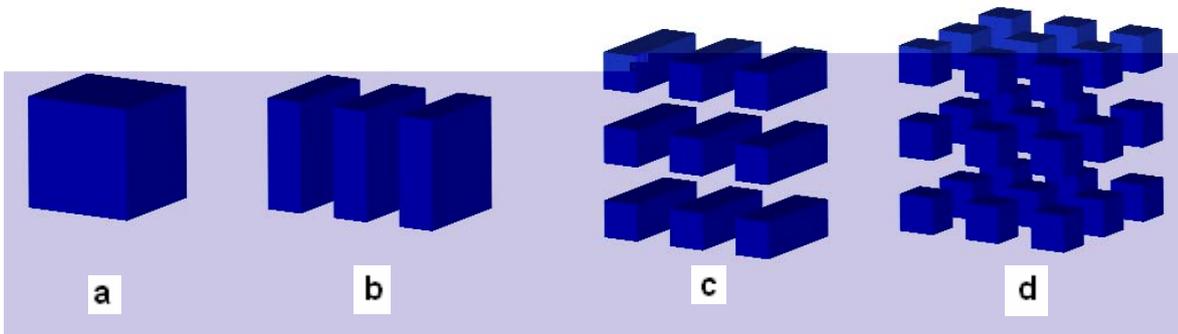


Figura 6.1 - Topologias possíveis para a troca de mensagens. a) domínio original, b) paralelização unidimensional, c) bidimensional e d) tridimensional.

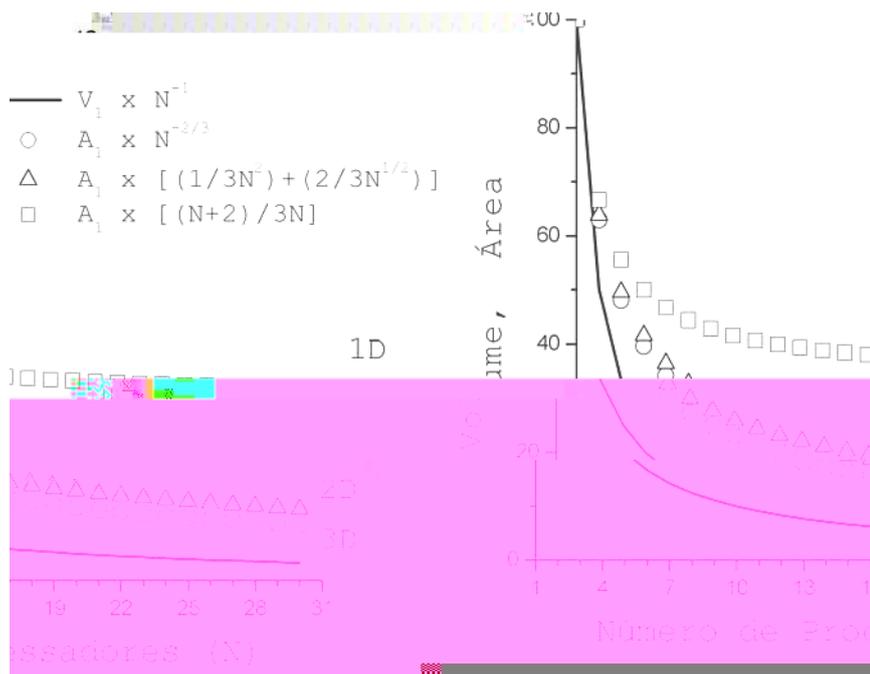


Figura 6.2 - Relação entre a diminuição do volume computacional e da área de troca de mensagens para as três topologias possíveis de particionamento (CAMPREGHER, 2005).

É possível observar, com relação à Figura 6.2, que o particionamento unidimensional atinge a sua capacidade máxima de diminuição da área superficial de troca de mensagens entre os subdomínios adjacentes muito antes das demais topologias. Evidentemente, esta característica, num problema de grande porte, implica num desempenho inferior do cluster. A

primeira versão do código Fluids-3D foi feita a escolha pelo particionamento unidimensional do domínio. Segundo Campregher (2005), esta escolha não comprometeu a performance, do código, principalmente por se empregarem poucos processadores nos cálculos. A vantagem mais clara de se empregar particionamento unidimensional reside na implementação bem mais simplificada do código.

A discretização dos termos espaciais no código Fluids-3D utiliza aproximações de segunda ordem centradas, o que exige um estêncil de três células, sendo uma o ponto de interesse e as outras duas a esquerda e a direita do ponto de interesse. Desta maneira, em uma topologia de paralelização unidimensional, faz-se necessária a sobreposição (*overlapping*) de apenas um plano. Por outro lado, aproximações de ordem mais elevada, exigiriam um número maior de planos e, conseqüentemente, um fluxo maior de troca de mensagens e possível sobrecarga (*overhead*) da rede. Este fato é particularmente complicante na topologia tridimensional de paralelização.

Na Figura 6.3, pode-se observar que os volumes internos ao domínio, como os volumes 2 e $n-1$ que foram calculados e estão atualizados, são copiados aos volumes n e 1, respectivamente, agindo como condição de contorno aos domínios Ω_1 e Ω_2 . Na Figura 6.4 é possível observar, conjuntamente, as sobreposições entre os domínios e a topologia unidimensional empregada

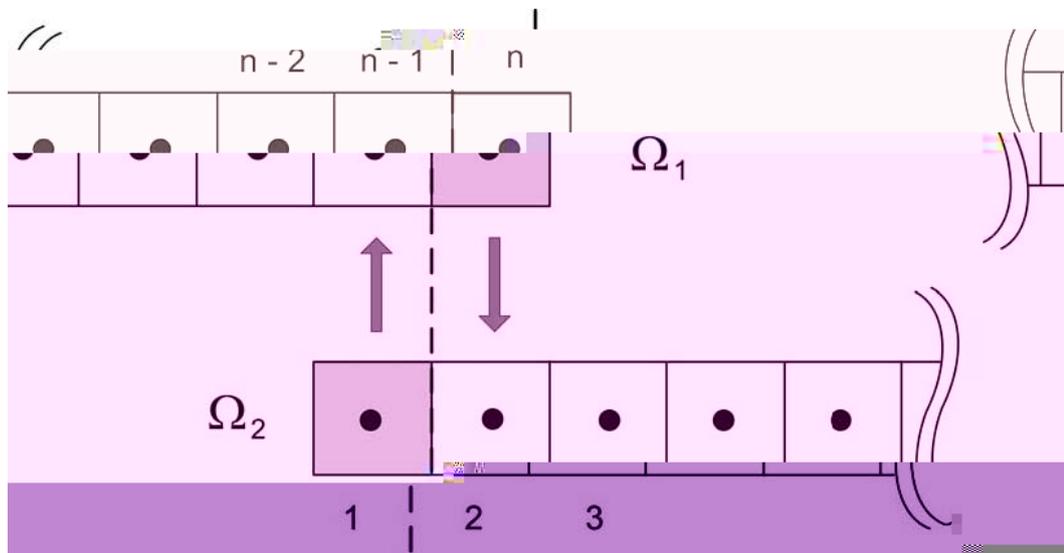


Figura 6.3 - Sobreposição necessária para aproximações espaciais de segunda ordem (CAMPREGHER, 2005).

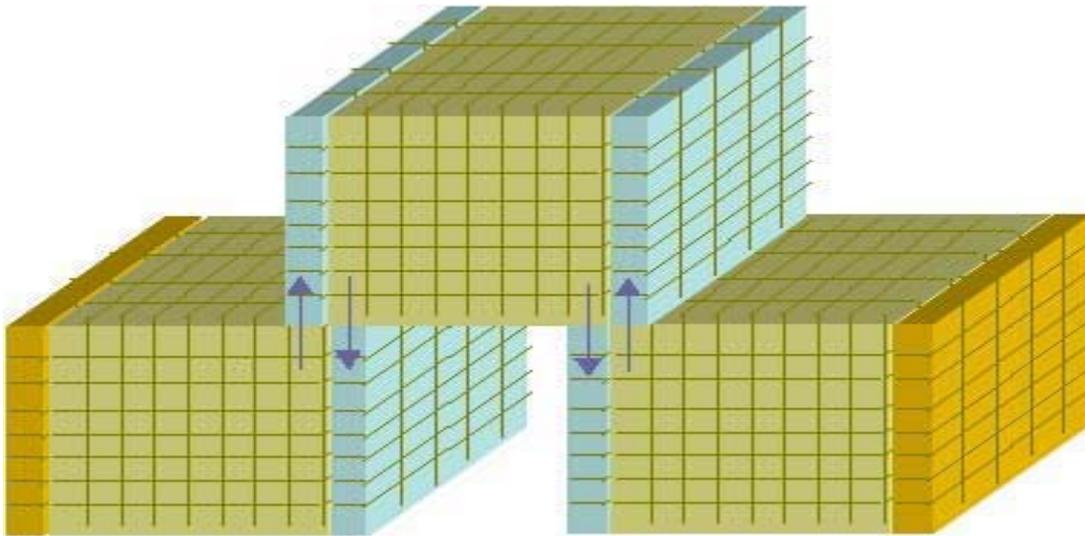


Figura 6.4 - Comunicação entre os subdomínios usando uma topologia unidimensional de divisão (MARINHO, 2004).

6.2 Implementação da topologia tridimensional de paralelização no código Fluids-3D

Existem basicamente duas maneiras de se trabalhar com o armazenamento de dados em um código numérico. A primeira, e mais comum, é trabalhar com vetores multidimensionais, como, por exemplo, uma matriz, designando um campo bidimensional de variáveis como velocidades ou qualquer outra variável de trabalho. Em um caso tridimensional, as declarações das variáveis são semelhantes a um caso bidimensional, porém, a matriz a ser alocada possui uma dimensão a mais. A Figura 6.5 mostra, como seria o formato de uma matriz tridimensional.

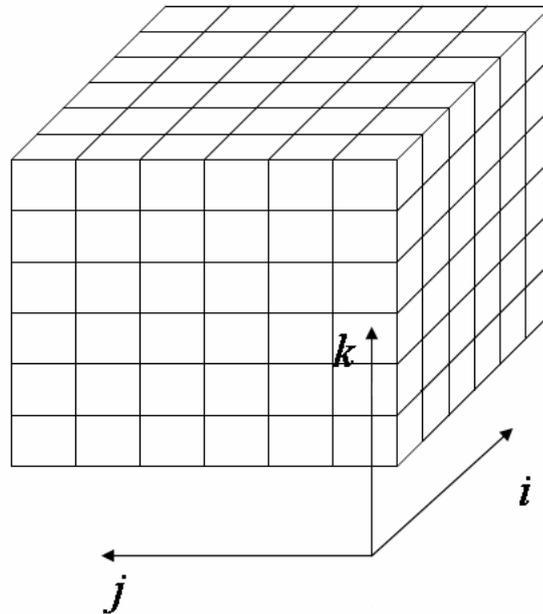


Figura 6.5 - Representação física de um domínio tridimensional.

Uma outra maneira de manipular estes dados é, ao invés de criar matrizes tridimensionais, criar vetores (*arrays* unidimensionais), porém, obviamente com o mesmo número de volumes de uma matriz tridimensional, para o armazenamento e seguindo uma dada lei de formação. No caso do atual trabalho esta lei de formação é dada por:

$$l = (i - 1) \cdot y_{tot} \cdot z_{tot} + (j - 1) \cdot z_{tot} + k, \quad (6.1)$$

onde, l é a posição no vetor, y_{tot} e z_{tot} , são quantidades máximas de volumes nas direções y e z respectivamente, e i , j e k , são contadores que variam de 1 à x_{tot} , y_{tot} e z_{tot} respectivamente, sendo x_{tot} quantidade máxima de volumes na direção x .

Nota-se que, em um armazenamento com o uso de matrizes, uma variável possui três coordenadas para localização, por exemplo, se a variável em questão estiver posicionada onde as dimensões em x , y e z sejam máximas, seu valor é dado por:

$$\phi(x_{tot}, y_{tot}, z_{tot}) = aux, \quad (6.2)$$

onde aux é um valor numérico qualquer.

No caso de um código vetorizado (onde os dados são armazenados em vetores, ou seja, a segunda opção), a mesma variável citada no parágrafo acima seria encontrada por:

$$\phi(l) = aux, \quad (6.3)$$

onde l , aplicando a eq (6.1), é dado por $xtot * ytot * ztot$.

A vantagem de se usar um código vetorizado, é que, uma vez necessário usar o valor armazenado em qualquer local de um vetor, o acesso a tal dado é mais rápido, uma vez que o processador não necessita realizar nenhum cálculo extra para encontrar tal variável. Este cálculo extra é explicado por que atualmente, mesmo declarando variáveis como matrizes, automaticamente, o processador já transforma estas matrizes em vetores. O fato de usar um código vetorizado elimina a necessidade de o processador ficar vetorizando matrizes. Isto é particularmente benéfico em grandes volumes de dados. O código Fluids-3D utiliza desta metodologia. Vale a pena salientar que o termo código vetorizado, utilizado no presente texto diz respeito ao tipo de alocação de dados realizados no código Fluids-3D, não tendo nenhuma relação com os tipos de vetorizações realizados por supercomputadores tipo Crays ou IBM's.

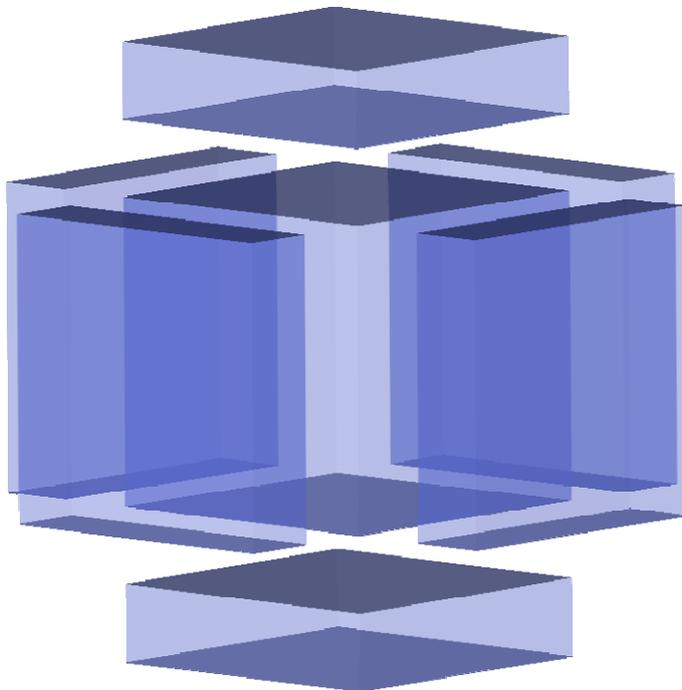


Figura 6.6 - Áreas de trocas de informações em uma topologia tridimensional de paralelização.

Apesar de ser comparativamente mais rápido do que um código matricial, a paralelização tridimensional de um código vetorizado não é algo trivial. Como já citado, a troca de informações é realizada nas fronteiras

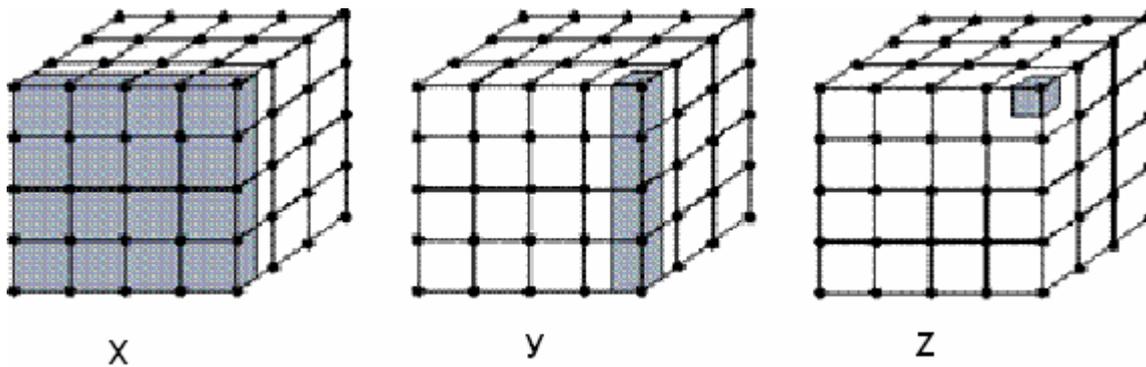


Figura 6.8 - Área de troca de informação nas três direções coordenadas.

O fato de as rotinas de paralelização serem mais eficientes quando mandam maiores volumes de dados inviabiliza a paralelização em y e z , uma vez que o overhead é inevitável. Para ainda sim paralelizar tridimensionalmente um domínio, algumas precauções podem ser tomadas, evitando o *overhead*. Neste trabalho optou-se por criar vetores auxiliares com o tamanho característico da mensagem que se deseja enviar, e então enviá-los utilizando comandos adequados das bibliotecas de paralelização, e no processador de destino “abrir” este conjunto de dados e redirecioná-los às suas posições corretas.

Esta ação de criar vetores auxiliares, apesar de ter um certo custo computacional, foi uma maneira simples de se evitar o overhead na rede, e, assim, paralelizar tridimensionalmente o código Fluids-3D

Duas considerações importantes ainda são necessárias:

- Toda a paralelização citada nesta dissertação diz respeito somente a paralelização do domínio euleriano, e não da fronteira imersa e do modelo físico virtual. No caso da simulação de geometrias complexas a fronteira imersa ainda deve estar confinada em um único processador, porém, com a utilização de topologia tridimensional, uma nova gama de geometrias podem ser simuladas, uma vez que esta metodologia de paralelização, aumenta muito a escalabilidade do código, que por sua vez pode simular domínios maiores.
- Em casos nos quais a razão de aspecto é baixa ou moderada, ou escoamentos que não necessitam de vários milhões de volumes para ficarem bem caracterizadas, a paralelização tridimensional, ou mesmo bidimensional não são indicados. Como quem dita a velocidade da simulação é sempre o processador mais lento (mais requisitado, ou com uma quantidade maior de volumes), o processador que é responsável pelos cálculos lagrangianos é

sempre quem dita a velocidade da simulação. Algumas simulações monitoradas indicam que o processador da fronteira imersa chega a 99% de uso, enquanto os processadores responsáveis pelas outras áreas do domínio não chega, a usar 7% de CPU.

Em todas as simulações realizadas no presente trabalho a razão de aspecto dos domínios eram baixas ou moderadas, assim sendo, não foi necessário utilizar o particionamento bi ou tridimensional. No entanto, o trabalho de paralelização 3D foi de suma importância, uma vez que este proporciona uma maior liberdade de particionamento do domínio euleriano, mesmo que seja em apenas uma direção. Isto é particularmente útil para se trabalhar com malhas refinadas a jusante do objeto imerso.

6.3 Bibliotecas de paralelização.

A comunicação entre os domínios, juntamente com a quantidade de cálculos processados em cada um deles, requer um cuidado especial, pois afeta não somente o desempenho da simulação, mas, também, a integridade dos resultados. Infelizmente, ainda não existe um software que seja capaz de paralelizar automaticamente, de forma eficiente e com segurança, todas as situações possíveis encontradas em processamento paralelo. Programas de maior desempenho ainda necessitam da intervenção do programador para se chegar ao resultado desejado, Campregher (2005).

A responsabilidade de transferência de informações entre os subdomínios em uma simulação fica a cargo das bibliotecas de paralelização, difundidas amplamente na Internet. Entre as mais populares, a PVM (*Parallel Virtual Machine*) e a MPI (*Message Passing Interface*) são as mais utilizadas, sendo esta última a empregada no presente trabalho.

A biblioteca PVM foi desenvolvida em 1989 por um grupo de universidades e centros de pesquisa nos EUA em um projeto chamado *Heterogeneous Network Project*. Como proposta inicial, o objetivo dos pesquisadores era desenvolver uma biblioteca que permitisse a comunicação entre máquinas de natureza heterogênea (diversos tipos de máquinas e processadores) sob o conceito de troca de mensagens (PITANGA, 2002). De forma geral, o sistema PVM consiste de uma aplicação que, uma vez inicializada, é responsável por comunicar-se com as demais máquinas da rede, formando a rede virtual e de um conjunto de bibliotecas próprias (em torno de 38) para realizar a criação e destruição de processos, troca de mensagens e sincronização de tarefas.

Pode-se dizer que a principal característica da PVM é poder construir uma rede heterogênea com eficiência, sendo controlada por uma única máquina virtual. Além disso,

permite criar e destruir processos em tempo de execução, sendo possível remover ou adicionar máquinas durante a simulação. Finalmente, a PVM possui notável tolerância à falhas, pois caso a máquina virtual não encontre comunicação com um nó da rede, ela automaticamente redireciona as tarefas para um outro nó, sem interromper o processo. Entretanto, toda esta facilidade na construção da rede virtual implica num custo alto em termos de sobrecarga de comunicação (overhead), prejudicando o desempenho do código o que tende a piorar, consideravelmente, em sistemas com grande número de processadores.

Na primeira versão do código Fluids-3D foi construída a implementação MPICH (MPICHamaleon) do padrão MPI-2. Nos desenvolvimentos do atual trabalho, a biblioteca escolhida foi a MPICH-2. Desenvolvida a partir de melhoramentos necessários nas primeiras versões do MPI, esta biblioteca chega a ser 30 % mais rápida durante o processo de troca de mensagens, se comparada a versões antigas do MPICH, Gropp et al. (1999). Além de ser considerada uma das únicas versões que combinam portabilidade, interoperabilidade e alta performance, possui ainda capacidade de lidar com máquinas heterogêneas e SMP (*Symmetric Multi Processors*), ou seja, máquinas individuais com dois ou mais processadores. Outras vantagens importantes são a disponibilidade de diversos recursos, que podem ser explorados visando melhorar o rendimento dos códigos e a sua excelente portabilidade. Finalmente, pode ser dito que o MPI apresenta-se como um padrão para processamento paralelo, o que garante constante manutenção e atualização, Gropp et al. (1999).

CAPÍTULO VII

7 Resultados e Discussão

Como já mencionado em capítulos anteriores, as mudanças realizadas no pré-processamento do código Fluids-3D viabilizaram uma maior facilidade de *set-up* e importação de geometrias, facilitando assim, o estudo de escoamentos sobre diversas geometrias. Neste capítulo serão apresentados resultados de re-validação deste código, bem como resultados de aplicações de CFD a escoamentos sobre geometrias complexas.

Existem três possibilidades de condições de contorno nas direções y e z do domínio euleriano, são elas:

- parede: todas as componentes das velocidades (normais e tangenciais) possuem valor nulo;
- fronteira livre: derivadas nulas para os valores das velocidades tangenciais e velocidade nula na direção normal;
- periodicidade: os fluxos de massa e de quantidade de movimento linear que saem de uma superfície, entram pela superfície oposta.

Na entrada do domínio, adota-se um perfil uniforme de velocidade com $u=U_{\infty}$, $v=w=0$. Com relação às condições de contorno na saída do domínio utiliza-se um artifício de correção da componente u no último plano do domínio (em $i=xtot$, onde i representa a posição de um dado plano na direção x), de modo a garantir a conservação de massa global. Este artifício consiste em atribuir um valor equivalente ao valor do penúltimo plano (em $i=xtot-1$) multiplicado por uma razão entre as massas totais que entram e que saem do domínio (VERSTEEG; MALALASEKERA, 1995, CAMPREGHER, 2005).

7.1 Resultados de re-validação

Esta seção é dividida em duas partes. Primeiramente serão apresentados resultados para escoamentos sobre uma esfera estática, com o intuito de validar a sub-rotina de importação de geometrias geradas na forma de arquivos **.stl*. O processo de geração de um arquivo **.stl* pode ser encontrado mais detalhadamente em Oliveira (2007). Com o intuito de re-

validar a importação de geometrias provenientes da importação de malhas geradas em geradores de malhas comerciais, o escoamento sobre um cubo estático será apresentado.

7.1.1 Resultados de validação da importação de geometrias via *.stl

A validação da sub-rotina de leitura de arquivos *.stl foi realizada comparando o escoamento sobre uma esfera, utilizando a primeira versão do código Fluids-3D, que foi amplamente validada em Campregher (2005), e a versão atual do código Fluids-3D. A Figura 7.1 mostra um exemplo de malha de uma esfera representada por uma arquivo *.stl.

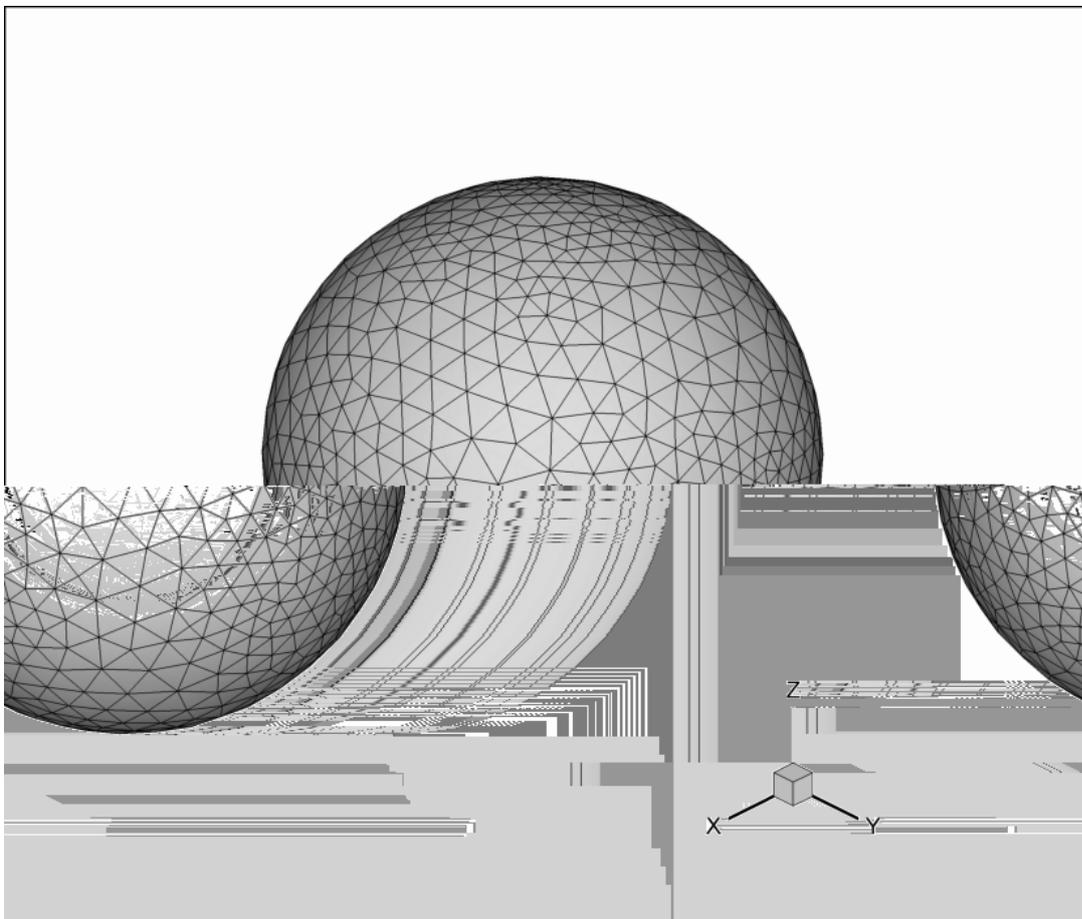


Figura 7.1 - Visualização de uma malha de elementos triangulares provenientes de um arquivo *.stl.

A malha euleriana utilizada em ambos os casos é mostrada na Figura 7.2. Tal malha tem dimensões $1,0 \times 0,68 \times 0,68 \text{ m}$. A esfera está posicionada com seu centro na posição $(x, y,$

$z) = (0,38, 0,34, 0,34)$ m. O domínio euleriano foi discretizado por volumes finitos, como mencionado anteriormente. Com base no diâmetro D da esfera, o domínio possui dimensões de $9D$ à montante, $15D$ à jusante e largura de $17D$ em ambas as direções y e z . A razão de bloqueio, em termos de área foi de $0,27\%$, Campregher (2005). Para estas simulações foram utilizados três processadores em paralelo, como mostrado na Figura 7.3.

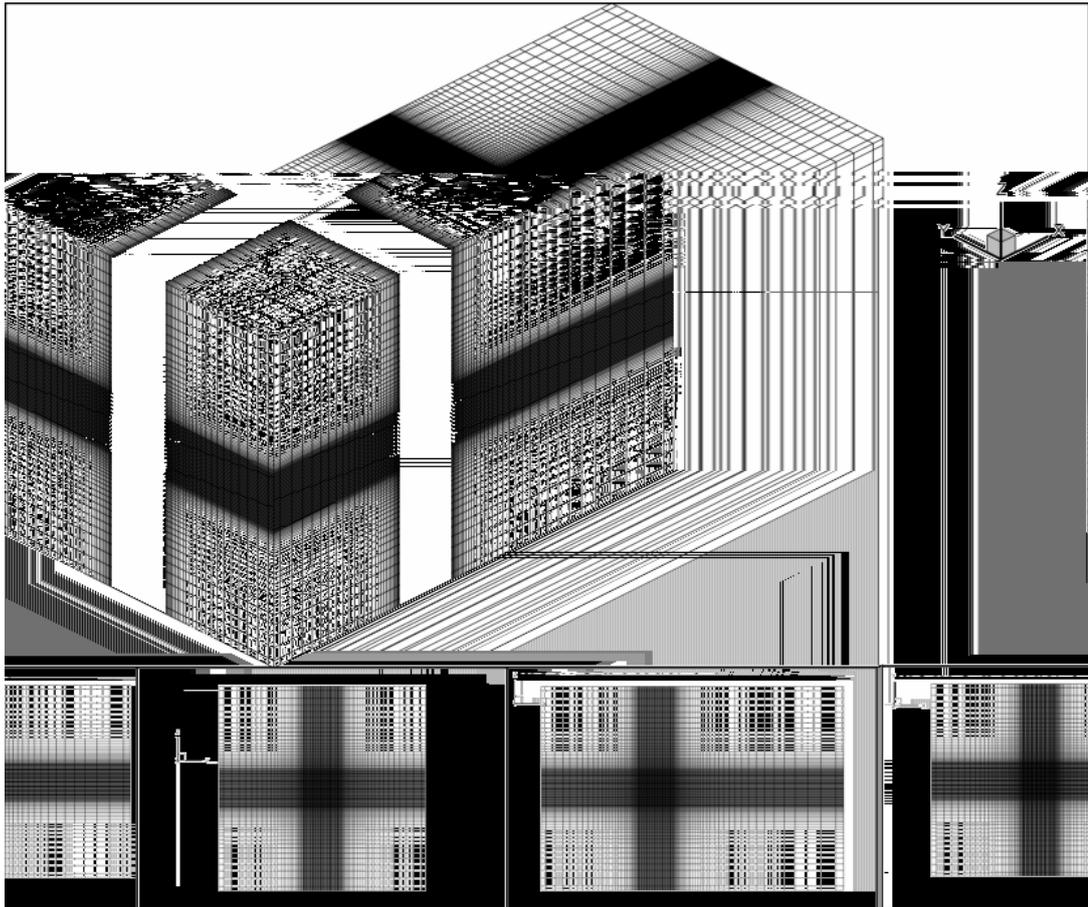


Figura 7.2 - Malha euleriana utilizada nas simulações dos escoamentos sobre esferas.

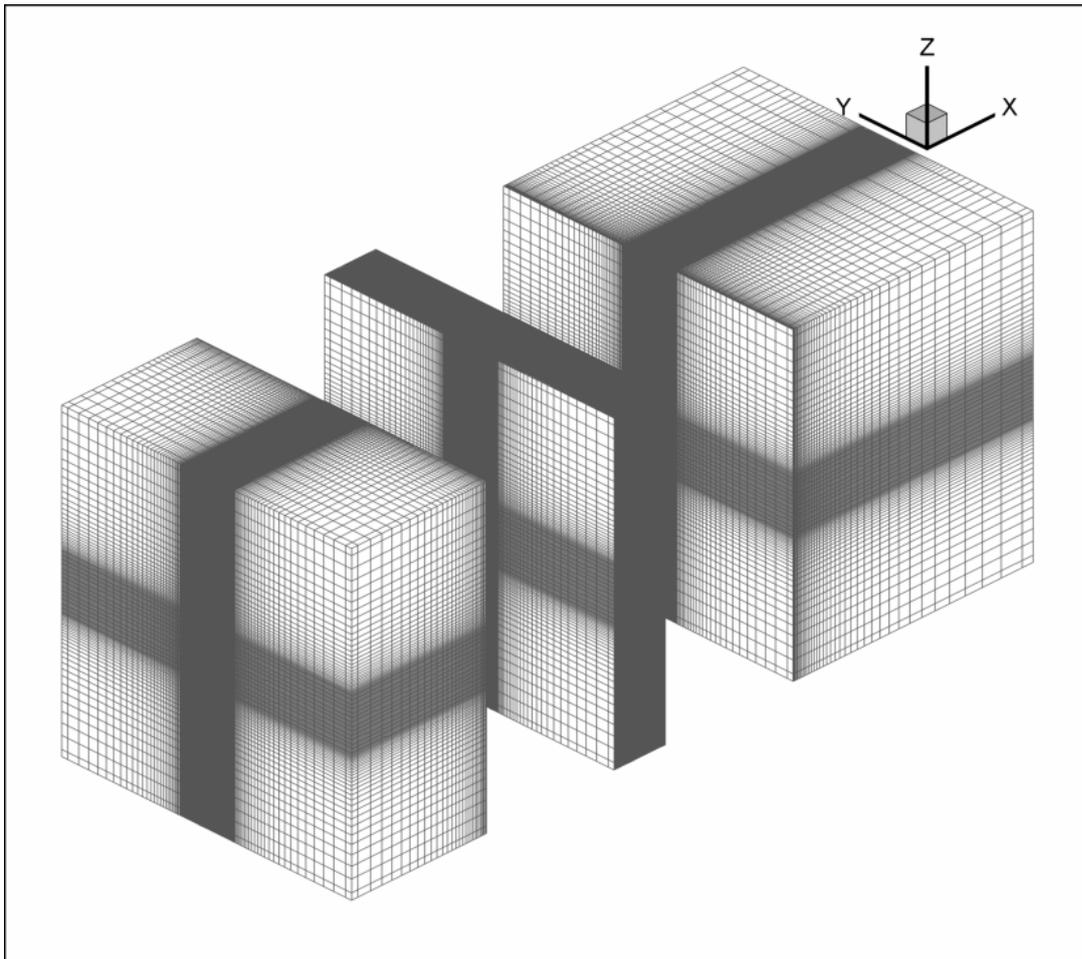


Figura 7.3 - Particionamento do domínio euleriano para escoamentos sobre esferas.

Para um número de Reynolds igual a 1000, são comparadas tanto propriedades qualitativas, como o critério Q (JEONG; HUSSAIN, 1995), quanto valores quantitativos, como valores dos coeficientes de arrasto, lateral e de sustentação.

O critério Q é definido na como:

$$Q = \frac{1}{2} \left[|\Omega|^2 - |S|^2 \right] > 0, \quad (7.1)$$

ou seja, a norma Euclidiana para a qual o tensor vorticidade sobrepuja o tensor deformação, definidos respectivamente como (Campregher, 2005):

$$\bar{\Omega} = \frac{1}{2} \left[\bar{\nabla} \bar{v} - (\bar{\nabla} \bar{v})^T \right], \quad (7.2)$$

$$\bar{\bar{S}} = \frac{1}{2} \left[\bar{\nabla} \bar{V} + (\bar{\nabla} \bar{V})^T \right]. \quad (7.3)$$

Os coeficientes de arrasto, lateral e de sustentação e o número de Strouhal, são definidos respectivamente como:

$$C_D = \frac{F_x}{\frac{1}{2} \rho U_\infty^2 A}, \quad (7.4)$$

$$C_s = \frac{F_y}{\frac{1}{2} \rho U_\infty^2 A}, \quad (7.5)$$

$$C_L = \frac{F_z}{\frac{1}{2} \rho U_\infty^2 A}, \quad (7.6)$$

$$St = \frac{f L}{U_\infty}, \quad (7.7)$$

onde F_x , F_y e F_z , são respectivamente a somatória das forças de cada ponto lagrangiano nas direções x, y e z. No caso da esfera, A é a projeção do objeto imerso na direção transversal ao escoamento ($A = \pi D^2/4$) e D é o diâmetro da esfera, que em todas as simulações foi mantido em 0,04m, U_∞ é a velocidade de entrada do fluido no início do domínio ($x=0$), e L é o comprimento característico do objeto imerso.

A Figura 7.4 abaixo mostra estruturas turbilhonares através da visualização de isosuperfícies $Q = 10$, para o escoamento gerado a partir de uma geometria gerada por um gerador de malhas comercial, para tempos físicos de 5,0 s, 5,5 s, 6,0 s, 6,5 s e 7,0 segundos. Em todas as simulações sobre esferas as condições de contorno do domínio euleriano foram impostas como superfícies livres.

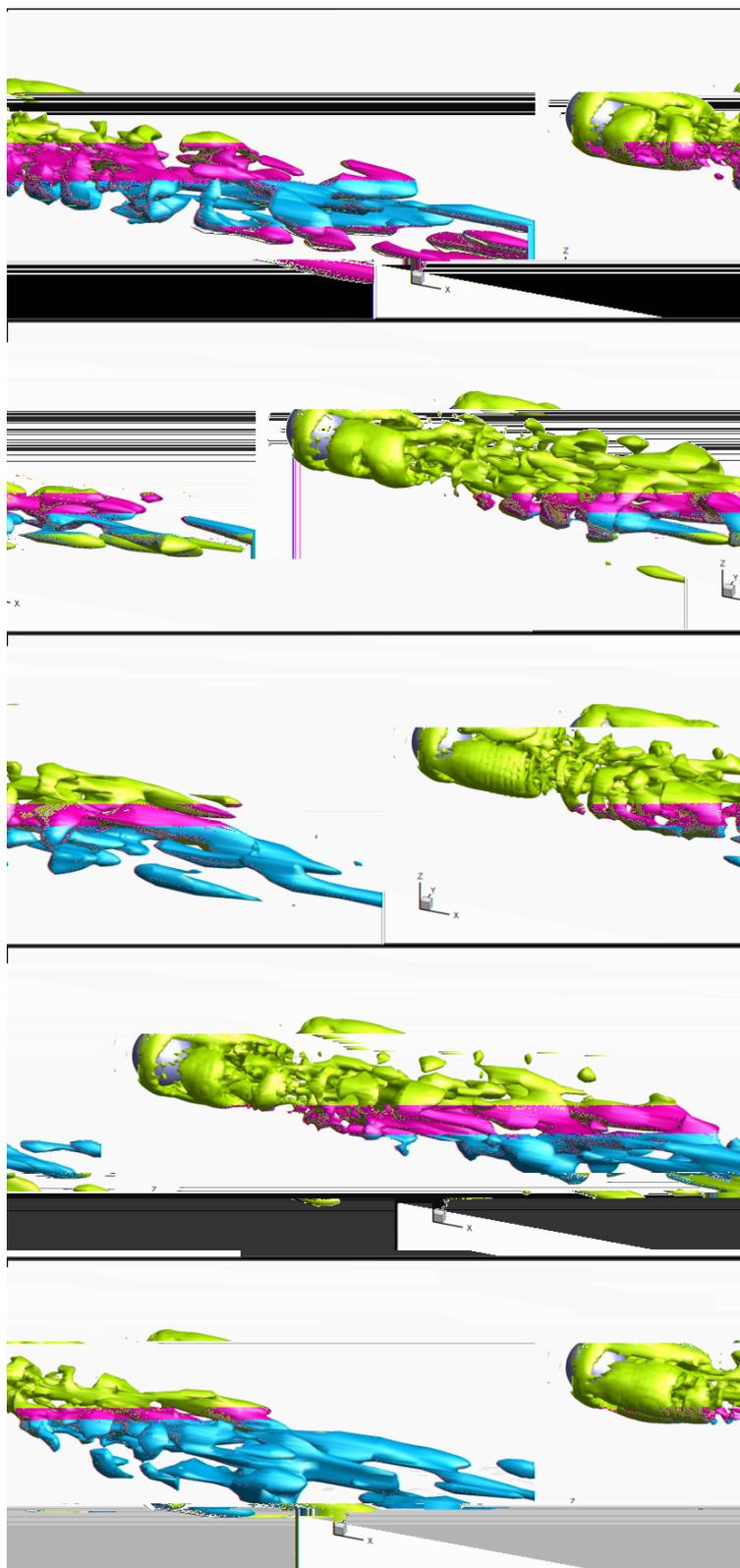


Figura 7.4 - Estruturas turbilhonares a jusante de uma esfera, importada de geradores de malhas comerciais ($Re=1.000$).

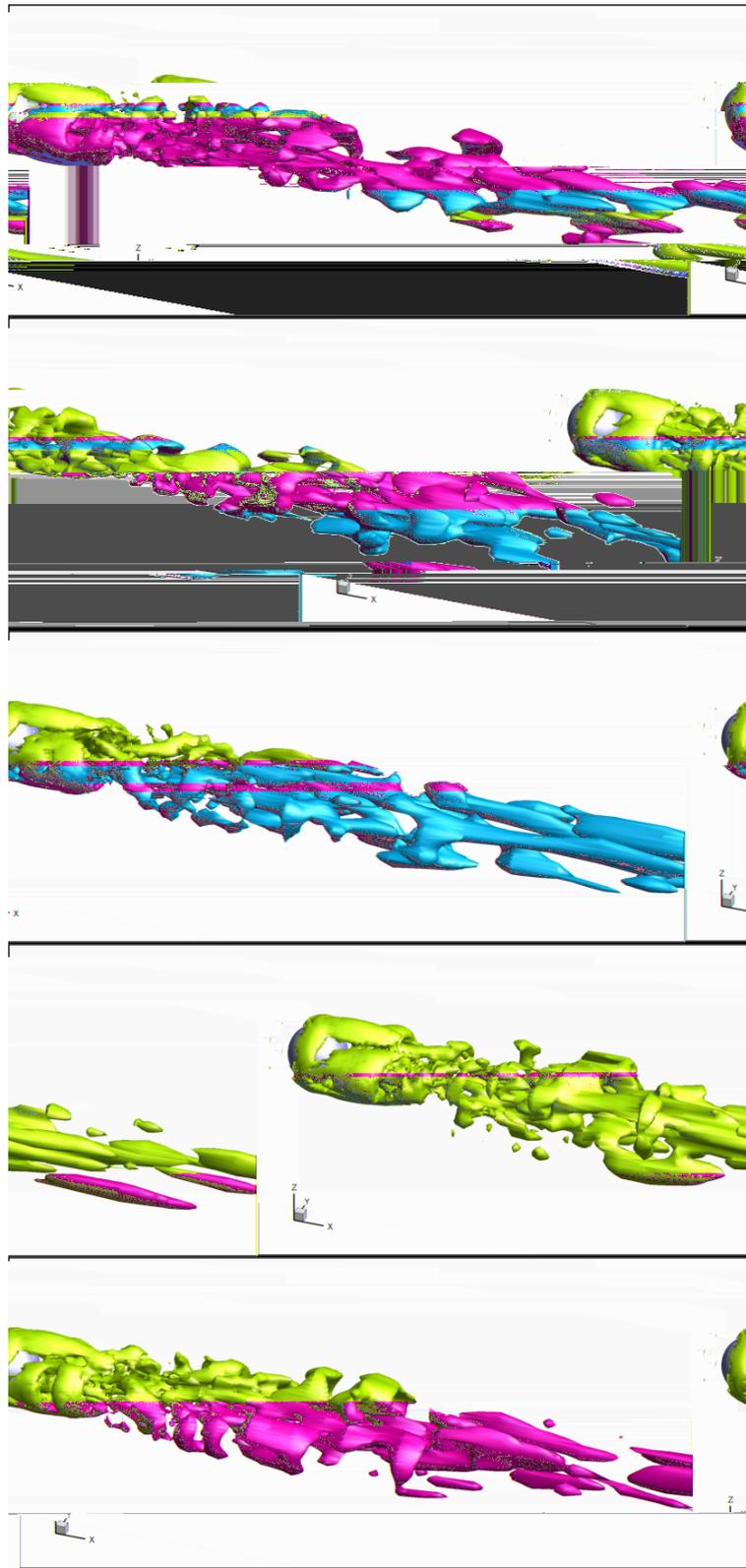


Figura 7.5 - Estruturas turbilhonares a jusante de uma esfera, importada de geradores de malhas livres, *.stl (Re=1.000).

Semelhantemente à Figura 7.4, a Figura 7.5 mostra isosuperfícies de $Q = 10$ para o escoamento a jusante de uma esfera para 5,0 s, 5,5 s, 6,0 s, 6,5 s e 7,0 segundo físicos. Pode-se notar a grande semelhança entre as estruturas turbilhonares existentes. A Figura 7.6 mostra mais claramente esta semelhança.

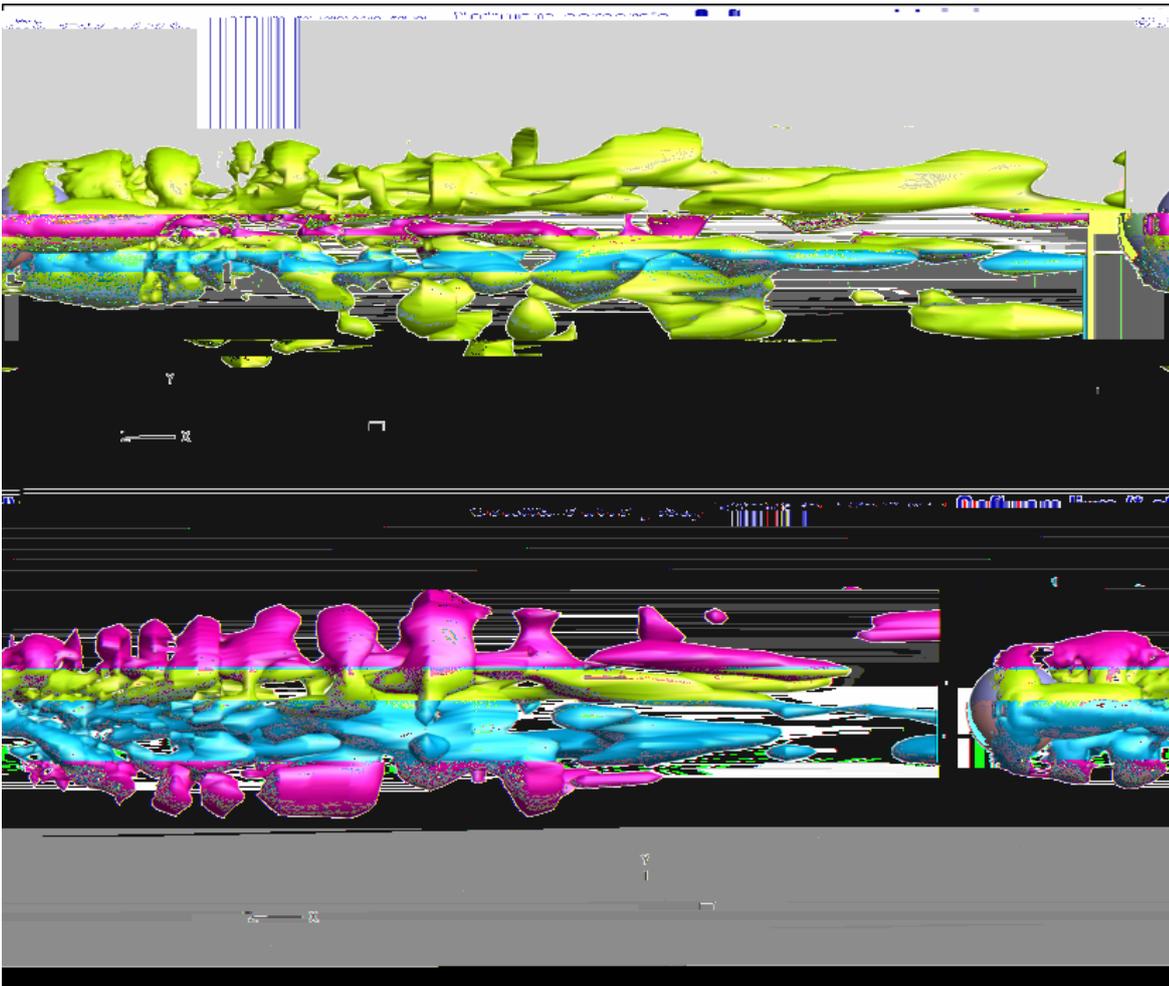


Figura 7.6 - Estruturas turbilhonares geradas a jusante de uma esfera, vistas no plano XY, em $t=7,0s$.

A validação quantitativa, através de propriedades como coeficiente de arrasto, lateral e de sustentação mostraram-se satisfatórias. A comparação dos coeficientes de arrasto pode ser vista na Figura 7.7. Nesta figura nota-se que o coeficiente de arrasto praticamente não se altera com relação ao tipo de gerador de malha.

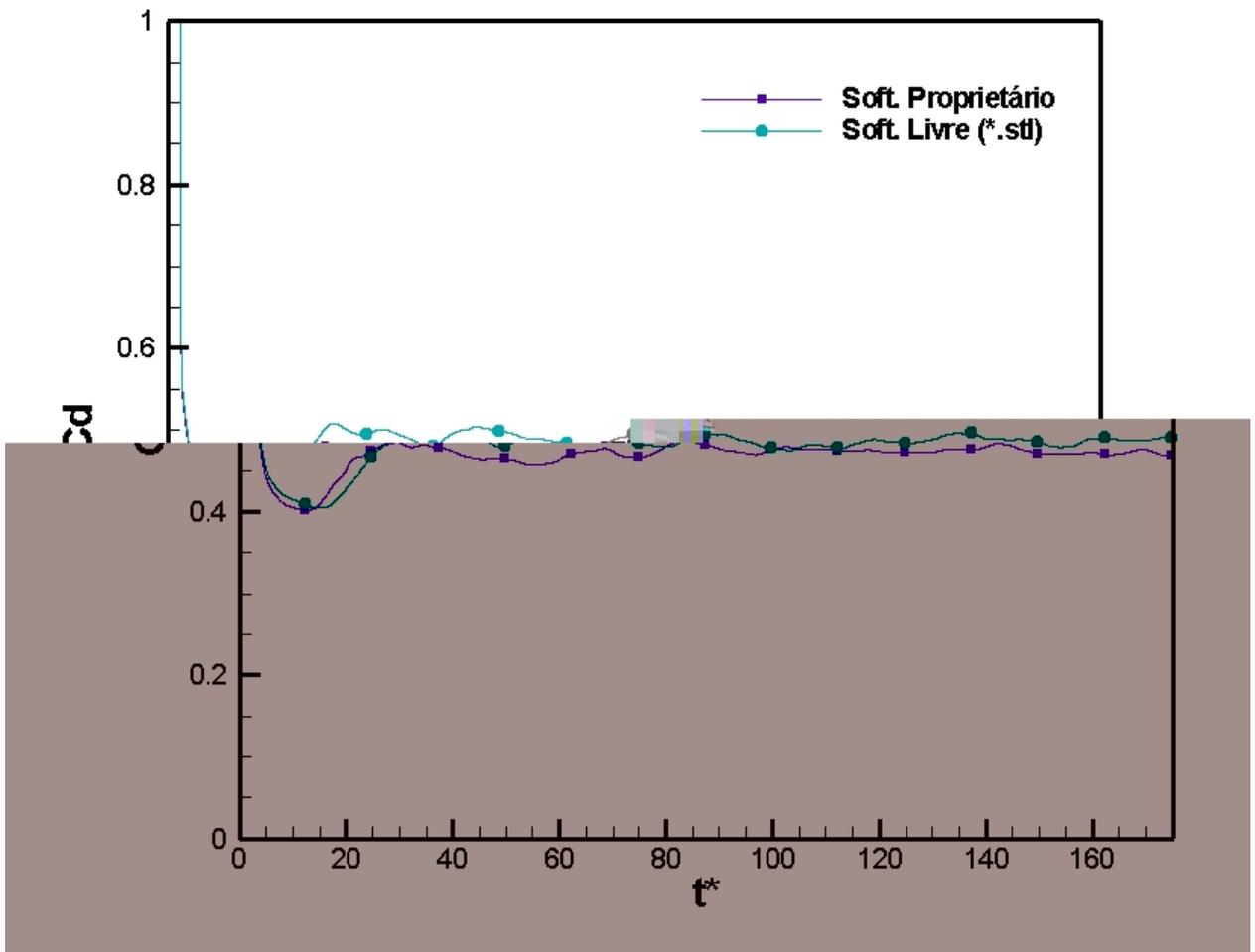


Figura 7.7 - Gráfico do coeficiente de arrasto, C_D , por t^* .

o tempo adimensional t^* é dado por:

$$t^* = \frac{t \cdot U_\infty}{D}, \quad (7.8)$$

onde t é o tempo físico e D é o diâmetro da esfera.

A Figura 7.8 e a Figura 7.9 mostram comparações entre os coeficientes laterais e de sustentação, respectivamente, para a simulação com as duas abordagens de geração de malha. Nota-se mais uma vez a semelhança entre as respectivas curvas. Deve-se salientar que as pequenas variações nas curvas são devidas à quantidade de pontos lagrangianos em cada

esfera. No caso da malha *.stl, existem cerca de 200 pontos a mais que na malha gerada por geradores comerciais, estes pontos excedentes são devidos ao fato de que a ferramenta geradora do arquivo *.stl, o software Gmsh, (OLIVEIRA, 2007), não possui um controle de comprimento característicos de malha (Δx_k) tão eficaz quanto os geradores de malha comerciais. Este fato, apesar de alterar sensivelmente áreas e comprimentos característicos de cada elemento triangular, no entanto, não impediu a validação da importação de geometrias via *.stl.

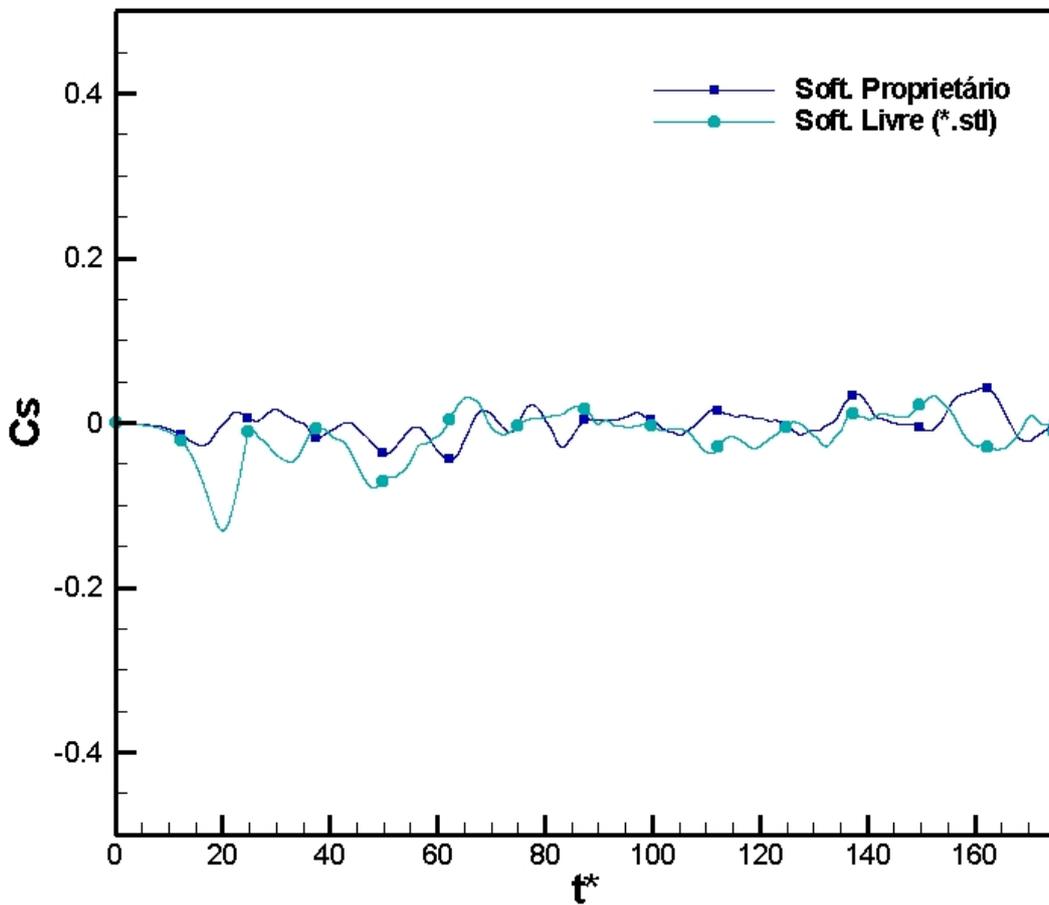


Figura 7.8 - Gráfico do coeficiente lateral, C_S , por t^* .

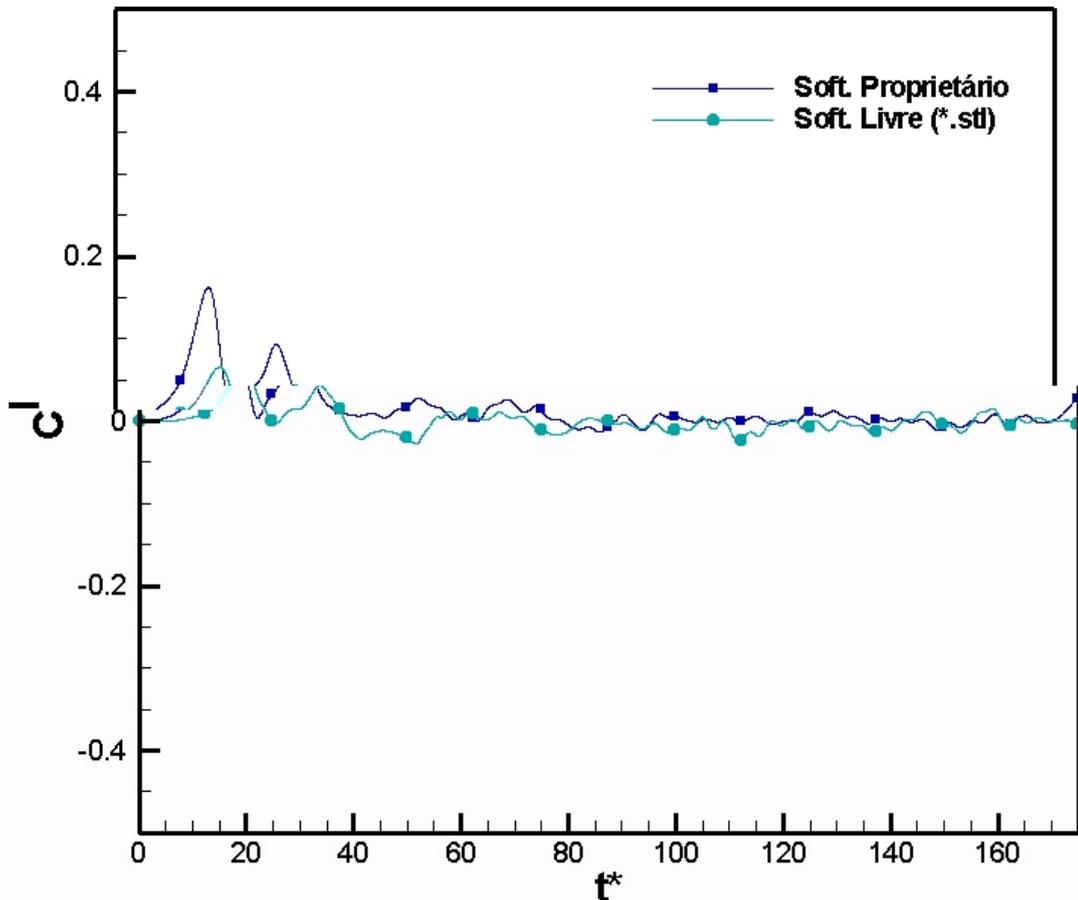


Figura 7.9 - Gráfico do coeficiente de sustentação, C_L , por t^* .

7.1.2 Resultados de re-validação de importação de geometrias geradas via geradores de malha comerciais – escoamento sobre um cubo

Para fins de re-validação da importação de geometrias geradas em geradores de malha comerciais, o escoamento sobre um cubo à um número de Reynolds igual a 290 foi escolhido. A malha do cubo pode ser visualizada através da Figura 7.10. Como já mencionado no capítulo cinco, a importação de novas geometrias, que não as esferas do trabalho de Campregher (2005), era possível, mas inviável para uso comum da ferramenta Fluids-3D. Com o intuito de facilitar a importação de geometrias, a sub-rotina de importação de malhas foi

alterada de forma a poder ler os arquivos de nós e conectividades geradas por um gerador de malhas comerciais, sem que houvesse necessidade de alteração destes arquivos por parte do usuário do código.

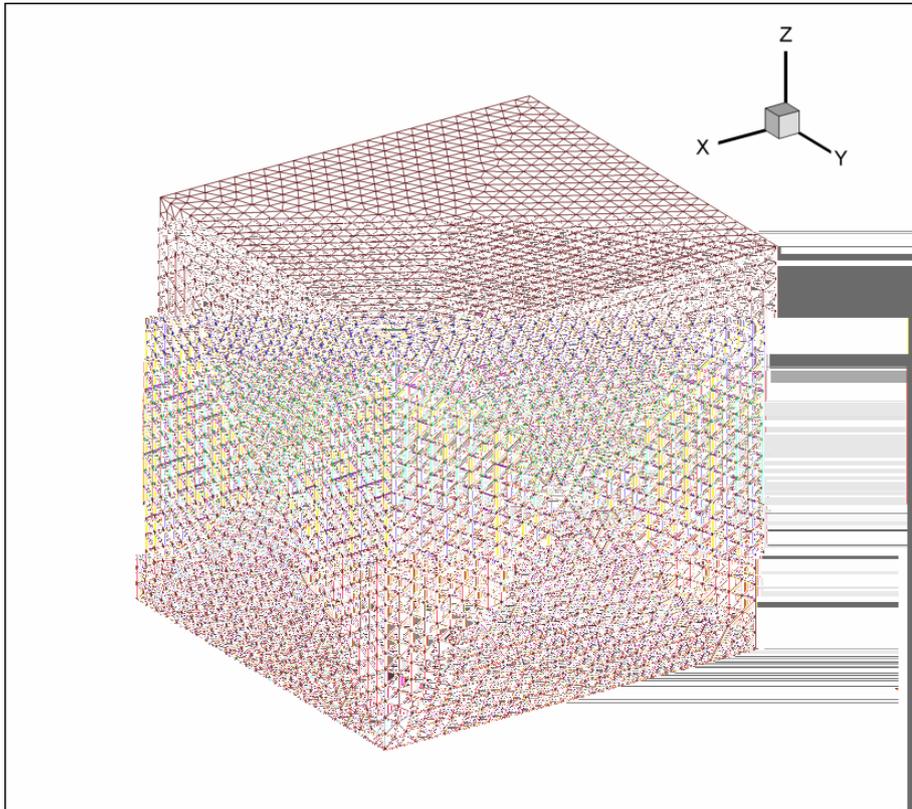


Figura 7.10 - Malha de elementos triangulares representando um cubo.

O Cubo tem dimensões de 0,1 x 0,1 x 0,1 m, e está inserido em um domínio retangular de dimensões 2,8 x 1,4 x 1,4 m. A Figura 7.11 mostra tal domínio na maneira em que foi paralelizado, ou seja em quatro subdomínios. As dimensões do domínio e do cubo foram escolhidas para comparação com o trabalho de Saha (2004), que estudou a tridimensionalização de um escoamento sobre um cubo a baixos números de Reynolds. A malha euleriana utilizada possui aproximadamente 2.100.000 volumes (194x104x104), também das mesmas dimensões da malha utilizada por Saha (2004). Neste caso, as condições de contorno, tanto na direção y quanto na direção z são impostas como parede.

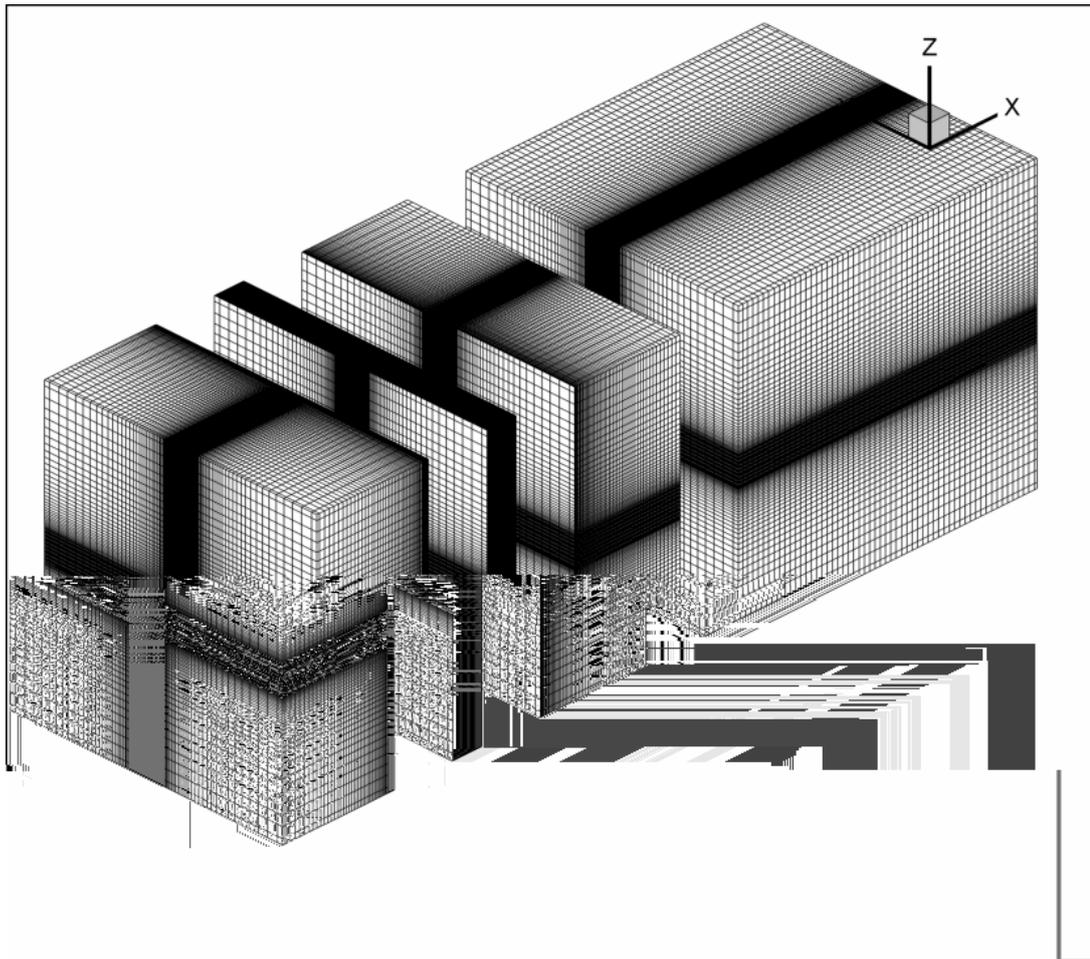
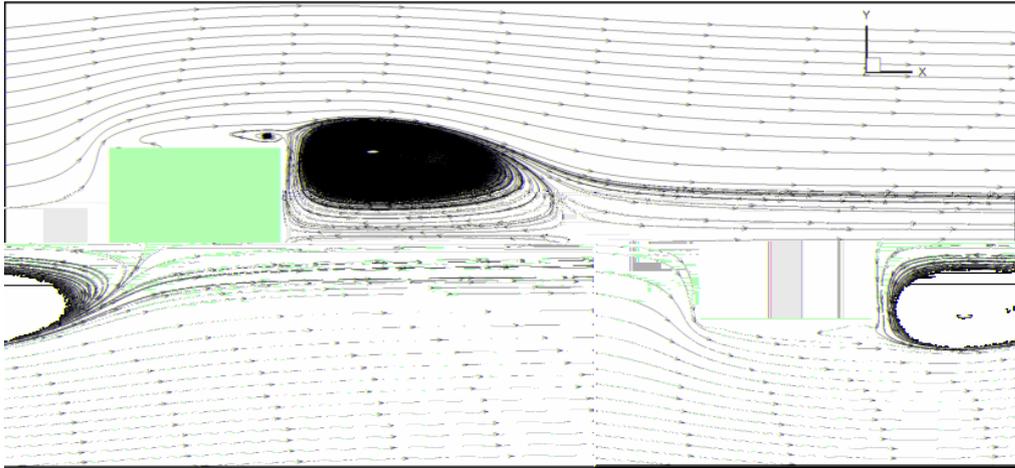
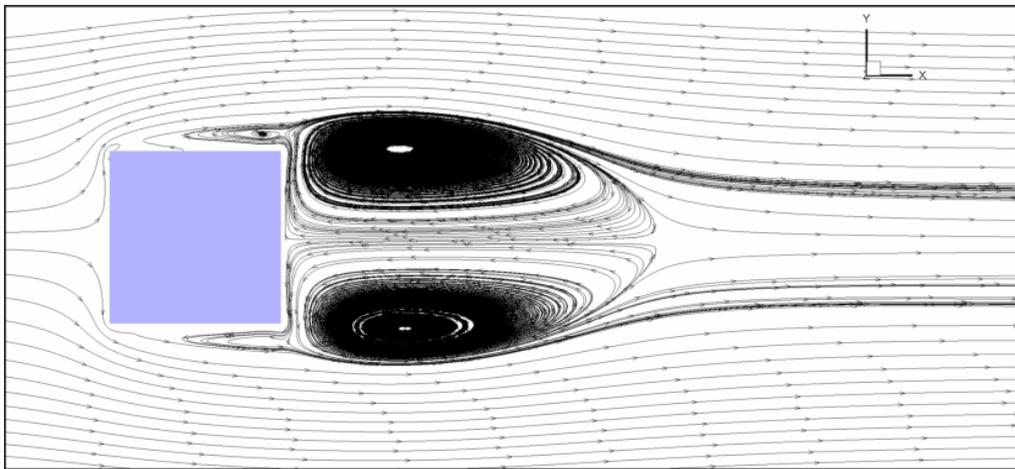


Figura 7.11 - Malha euleriana utilizada na simulação do escoamento sobre um cubo.

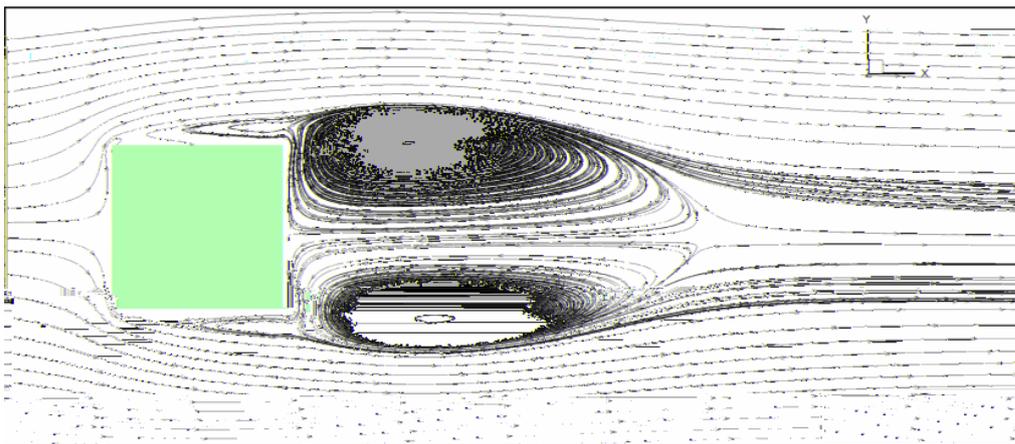
A Figura 7.12 (a)-(f) mostra a evolução temporal para linhas de corrente em um plano xy em $z=0,65$ m do escoamento sobre um cubo a Reynolds 290.



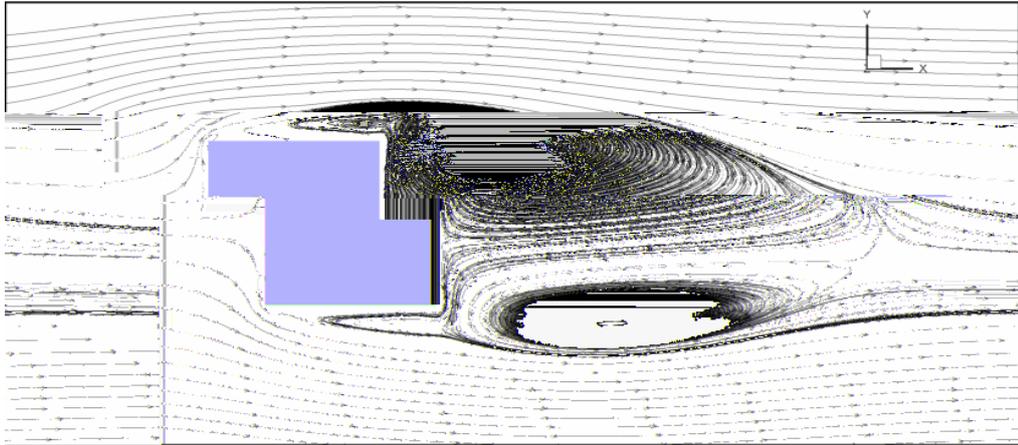
(a)



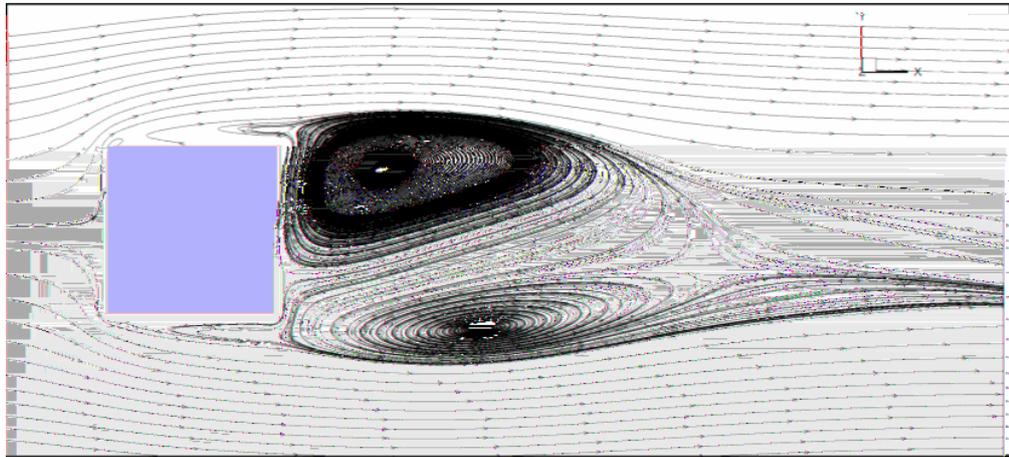
(b)



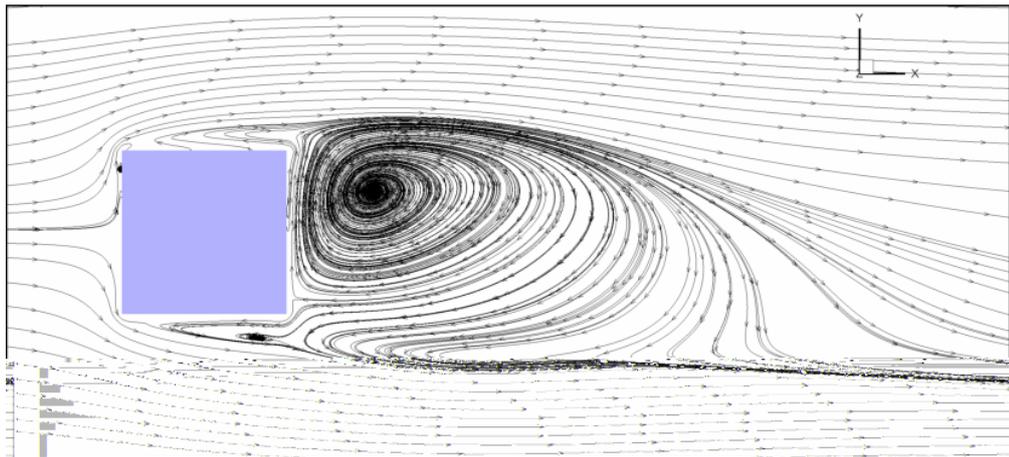
(c)



(d)



(e)



(f)

Figura 7.12 - Evolução temporal das linhas de corrente para os estágios: (a) $t = 0,8$ s, (b) $t = 1,6$ s, (c) $t = 2,4$ s, (d) $t = 4,0$ s, (e) $t = 8,0$ s, (f) $t = 8,3$ s.

Os valores de parâmetros quantitativos como o coeficiente de arrasto, lateral, de sustentação, e número de Strouhal, St , foram satisfatórios, quando comparados com o trabalho de Saha (2004). A tabela 7.1 mostra os valores médios das propriedades citadas acima, e os obtidos por Saha (2004).

	Presente trabalho	Saha, 2004
C_D	0,91	0,79
C_L	0,00	0,00
C_S	0,1	-0,06

Tabela 7.1: Parâmetros quantitativos para escoamento sobre um cubo a $Re=290$.

A evolução temporal dos coeficientes de arrasto, lateral e de sustentação podem ser vistos na Figura 7.13.

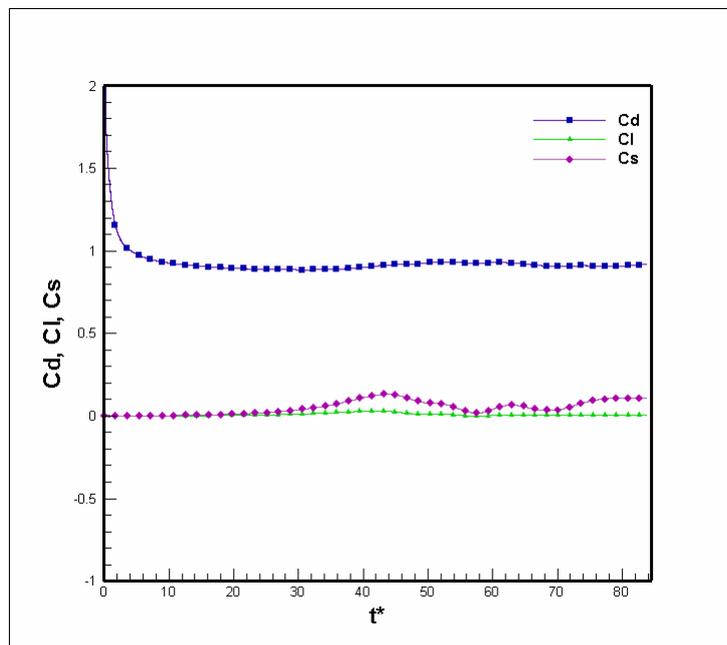


Figura 7.13 - Evolução temporal dos coeficientes de arrasto, lateral e de sustentação para o escoamento sobre um cubo, $Re=290$.

A Figura 7.14 mostra, para 2,5 segundo físicos, a forma média de duas bolhas de recirculação à $Re=290$, com tubos de corrente coloridos pela velocidade u .

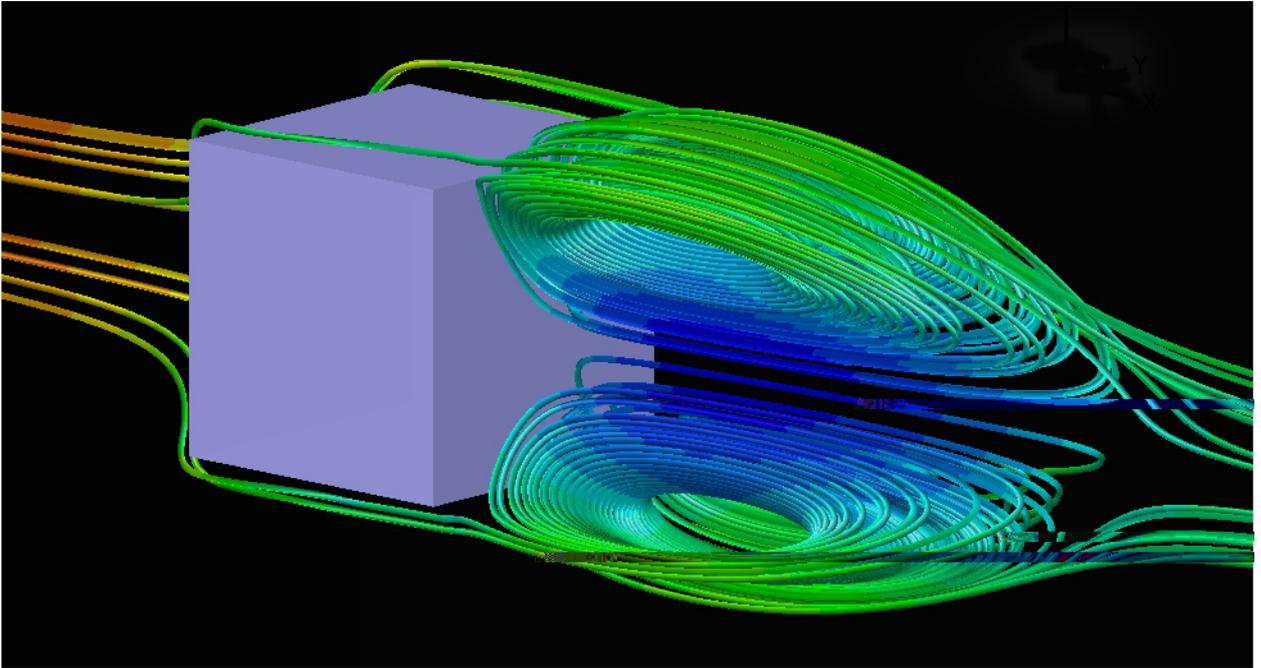


Figura 7.14 - Bolhas de recirculação a jusante de um cubo.

7.2 Escoamentos sobre geometrias complexas

Nesta seção serão apresentados resultados para escoamentos sobre diversas configurações de geometrias, com o intuito de avaliar a robustez do código Fluids-3D.

7.2.1 Escoamento sobre um aerofólio NACA-0012 – Visualização de vórtices de ponta de asa

O código Fluids-3D foi desenvolvido de forma a poder simular geometrias das mais variadas formas e dimensões, no entanto, até o trabalho de Campregher (2005), a configuração de geometria simulada foi somente o escoamento sobre uma esfera. Na presente dissertação objetivou-se avaliar escoamentos sobre diferentes geometrias. A primeira configuração escolhida foi o escoamento sobre um aerofólio NACA-0012, tridimensional, com o intuito de visualizar os vórtices típicos gerados nas extremidades do mesmo, conhecidos comumente como vórtices de ponta de asa (*wing tip vortices*).

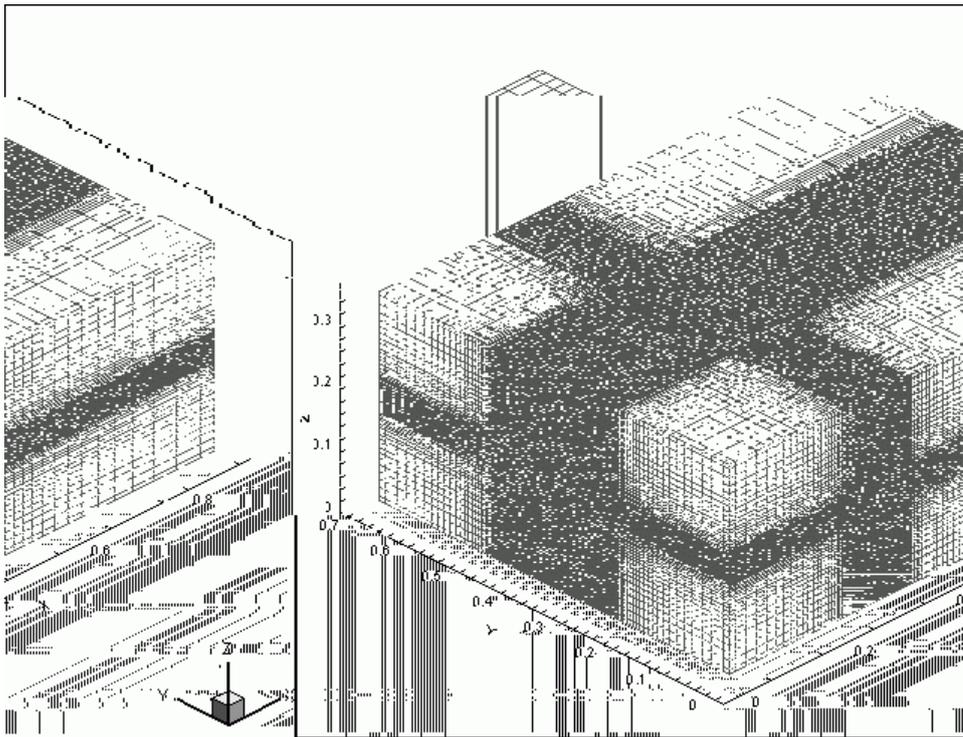


Figura 7.15 - Domínio euleriano para as simulações de um aerofólio NACA-0012, Vedovoto et al. (2006).

A malha euleriana utilizada em todas as simulações pode ser vista na Figura 7.15. Tal domínio (euleriano), possui dimensões: $x = 0,88$ m, $y = 0,68$ m e $z = 0,336$ m, discretizado por uma malha de $148 \times 268 \times 76$ volumes, nas direções x , y e z , respectivamente.

O aerofólio NACA-0012 é representado por uma malha de elementos triangulares, como pode ser visto na Figura 7.16, composta por 15.446 nós e 30.888 elementos. É importante lembrar que o Método da fronteira imersa somente requer a discretização da superfície que representa a interface sólido / fluido. O aerofólio está centralizado em $(x_c, y_c, z_c) = (0,28, 0,34, 0,165)$ m, tendo como comprimentos característicos: corda $c = 0,04$ m e largura equivalente à quatro vezes a corda. Para todas as simulações um ângulo de ataque $\alpha = 10^\circ$ foi escolhido, o número de Reynolds, baseado na corda foi $Re = 10.000$. Como condições de contorno das fronteiras eulerianas, considerou-se superfície livre tanto na direção y quanto na direção z .

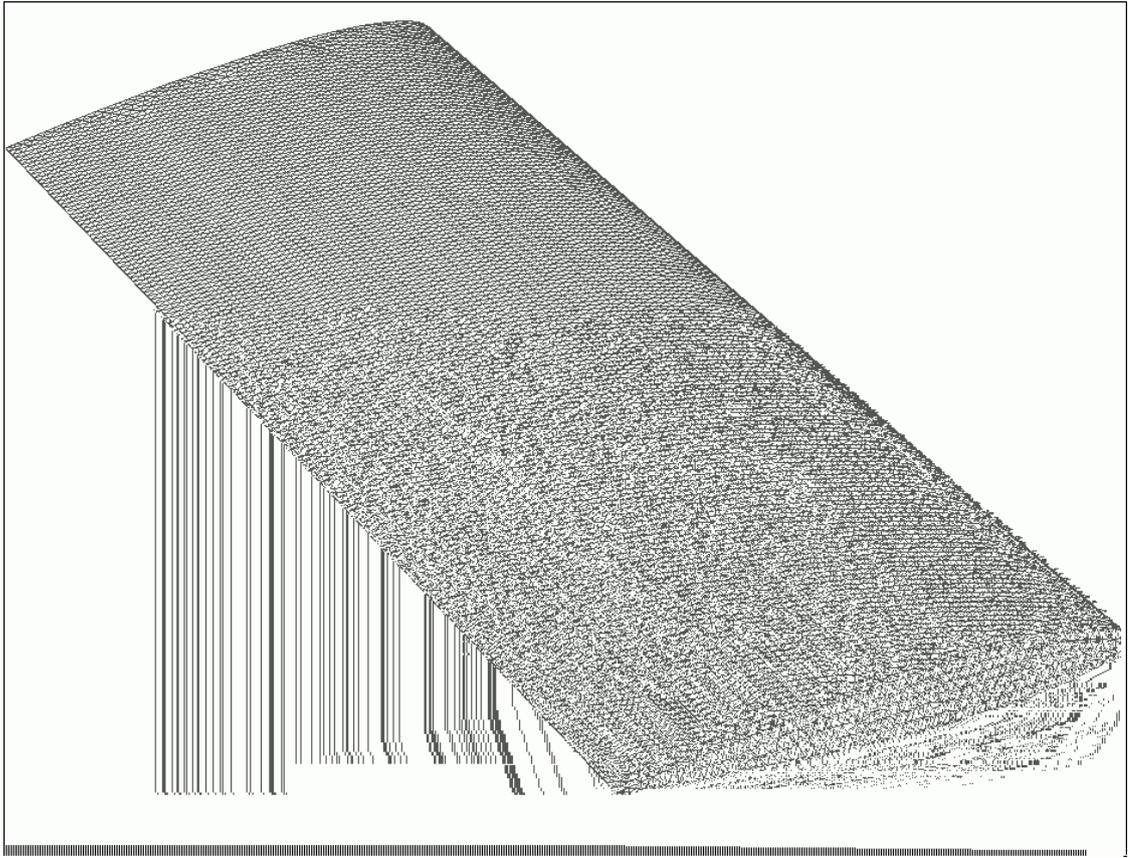
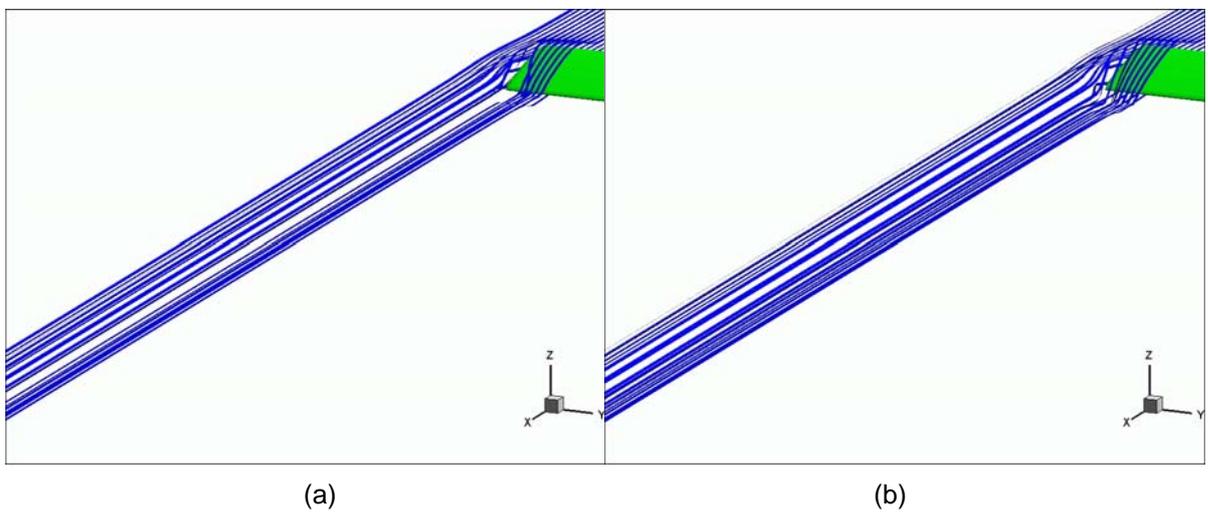


Figura 7.16 - Aerofólio discretizado por malha de elementos triangulares. Vedovoto et al. (2006).

Na Figura 7.17 (a)-(h) pode se notar a formação e evolução de linhas de corrente sobre o aerofólio NACA-0012 mostrado na Figura 7.16. O padrão do escoamento é simétrico na direção y , portanto, somente um lado do aerofólio é mostrado.



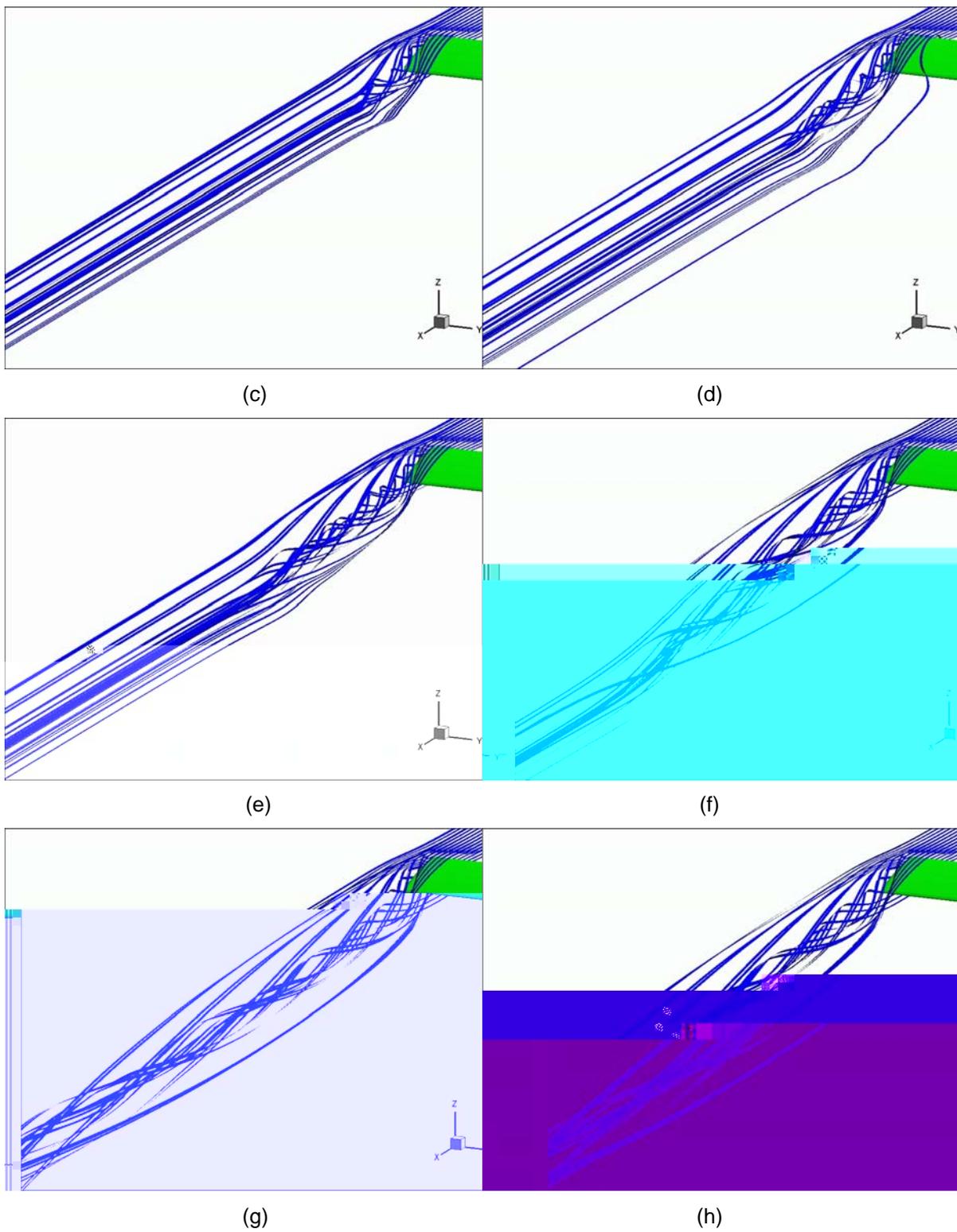


Figura 7.17 - Evolução de linhas de corrente sobre um aerofólio NACA-0012 nos seguintes tempos físicos: (a) $t=0,01s$, (b) $t=0,1s$ (c) $t=0,5s$ (d) $t=1,0s$ (e) $t=1,5s$ (f) $t=3,0s$ (g) $t=5,0s$ (h) $t=10,0s$. Vedovoto et al. (2006).

A norma L_2 fornece uma medida da diferença entre a velocidade do fluido na parede do objeto imerso (u_{fk}), e a própria velocidade da parede do objeto imerso (u_k). A rigor, fisicamente, esta norma deve ser nula, para que a condição de não deslizamento seja satisfeita, porém, tanto experimentalmente, quanto computacionalmente, uma pequena variação é admissível, uma vez que existem erros inerentes à experimentação (impossibilidade de medição da velocidade imediatamente sobre a parede), e às simulações numéricas (interpolação de valores de velocidade para faces ou centros de volumes de controle). O valor da norma L_2 , dada pela equação (7.9), estabeleceu-se em uma média de 10^{-3} , o que é aceitável. Vedovoto et al. 2006.

$$L_2 = \sqrt{\frac{\sum (u_k - u_{fk})^2}{N}}, \quad (7.9)$$

onde N é o número de pontos da fronteira lagrangiana.

A Figura 7.18 mostra três planos de linhas de corrente perpendiculares ao aerofólio. É notável a influencia da ponta do mesmo sobre o escoamento. Pode-se notar também a simetria do escoamento na direção y , e a formação e transporte de uma instabilidade tipo Kelvin-Helmholtz na região central do aerofólio. A Figura 7.19 e Figura 7.20, mostram uma vista superior e vista frontal dos vórtices gerados na ponta do aerofólio. Na Figura 7.20 o vórtice é claramente identificado. Craft et al. (2006), estudaram a formação, evolução e decaimento dos vórtices de ponta de asa. Um resultado qualitativo de tal trabalho pode ser visto na Figura 7.21. Nota-se a semelhança das linhas de corrente com os resultados da presente dissertação.

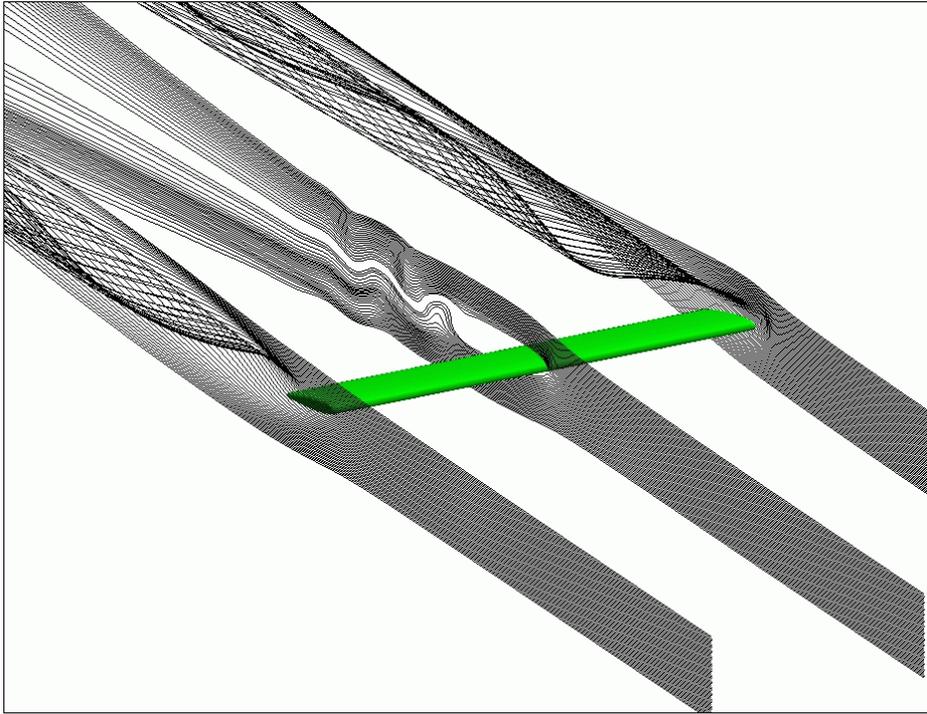


Figura 7.18 - Planos perpendiculares de linhas de corrente em $t = 10.0$ s . Vedovoto et al. (2006).

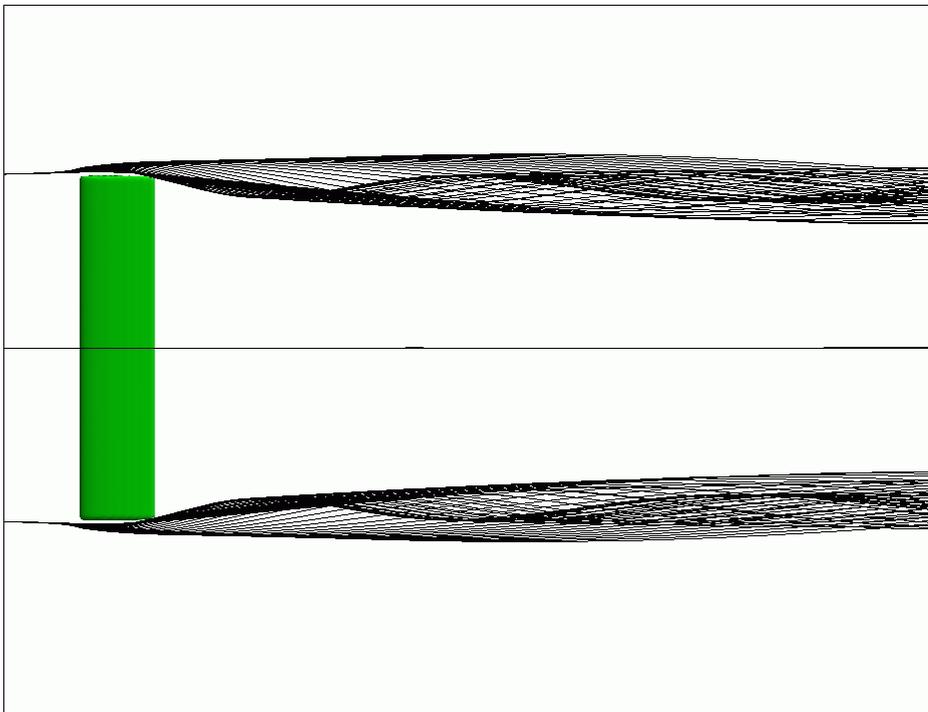


Figura 7.19 - Vista superior da Figura 7.18. Vedovoto et al. (2006).

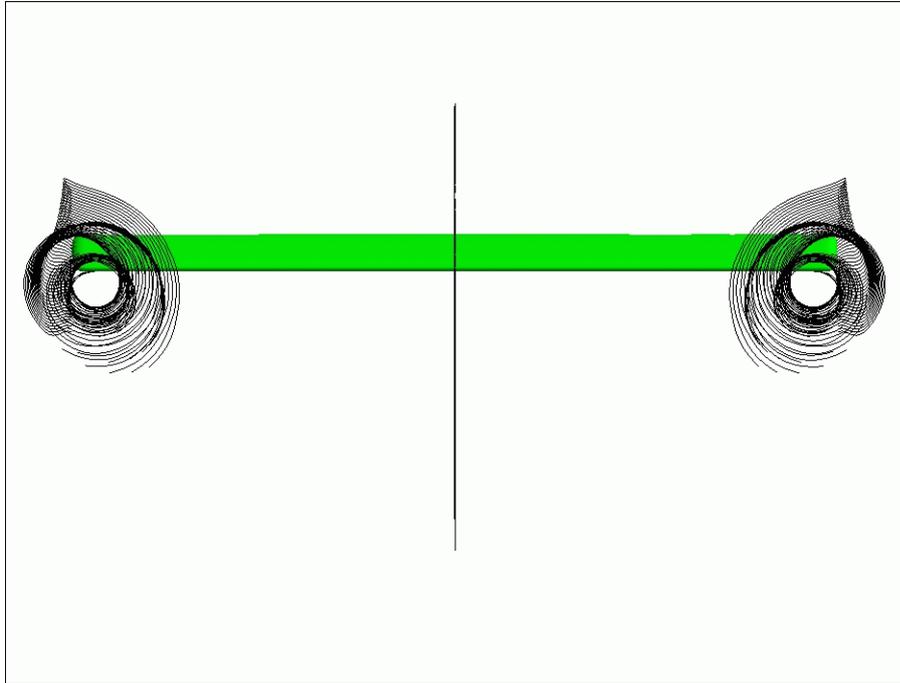


Figura 7.20 - Vista frontal dos vórtices da Figura 7.18. Vedovoto et al. (2006).

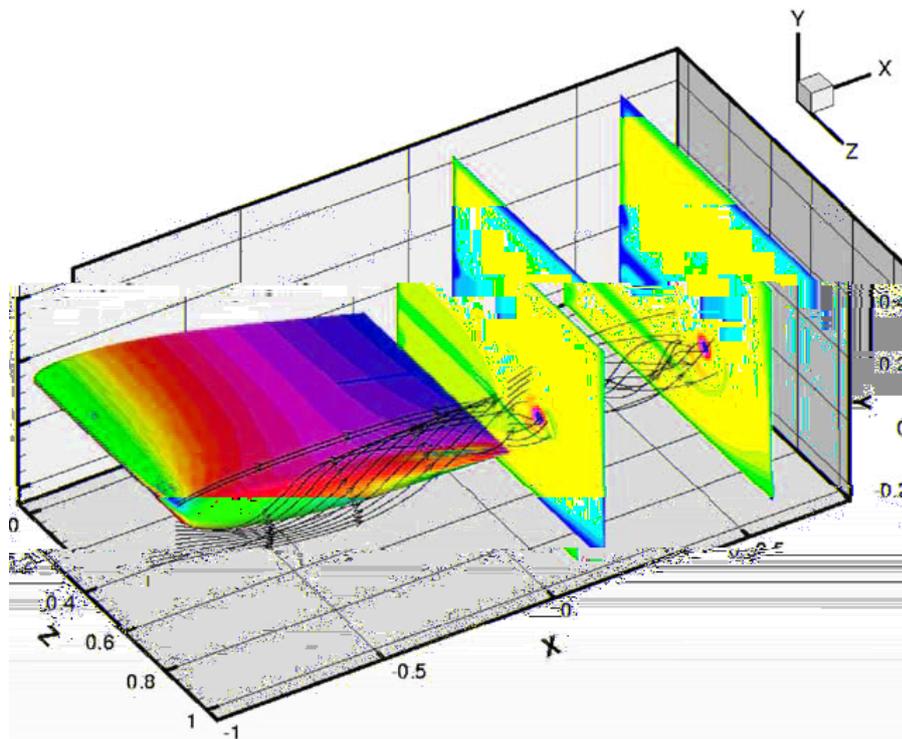


Figura 7.21 - Resultado qualitativo de vórtices de ponta de asa obtidos por Craft et al. (2006).

A visualização do escoamento através de isosuperfícies de $Q = 0,2$, mostra as principais regiões de rotação do escoamento. Na figura é notável a presença das duas estruturas turbilhonares que se desenvolvem à jusante do aerofólio. Este tipo de estrutura é formada sempre que um aerofólio gera sustentação (VEDOVOTO et al. 2006), e é particularmente preocupante em grandes aeronaves. Nestes casos, as duas grandes estruturas turbilhonares mostradas na Figura 7.22 podem se prolongar por centenas de metros, sendo um perigo real para pequenas aeronaves.

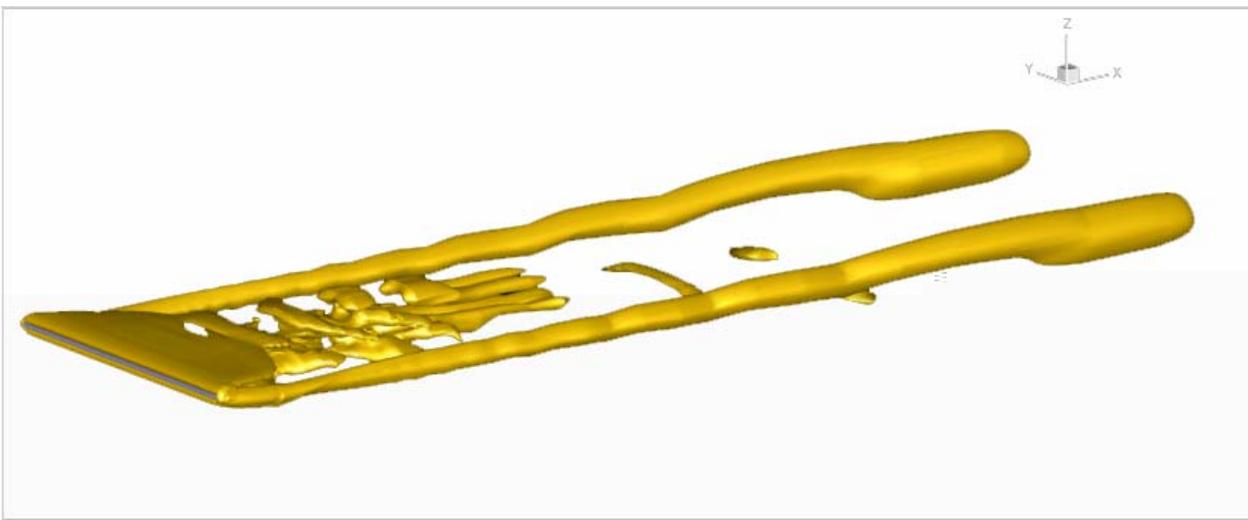


Figura 7.22 – Isso superfícies de $Q = 0,2$ para $t = 5,6$ segundos físicos.

7.2.2 Escoamento sobre um cubo solidário ao plano $z=0$

Esta configuração de escoamento, para a qual o domínio pode ser visualizado na Figura 7.23, apesar de geometricamente simples, promove estruturas altamente complexas no escoamento, como estruturas do tipo grampo de cabelo (*Hairpin vortices*) e ferradura de cavalo (*horseshoes vortices*). Além disso, a escolha de uma geometria que esteja solidária a uma parede do domínio, foi algo que exigiu mudanças no código, para o cálculo de algumas das interpolações necessárias para a avaliação da força lagrangiana.

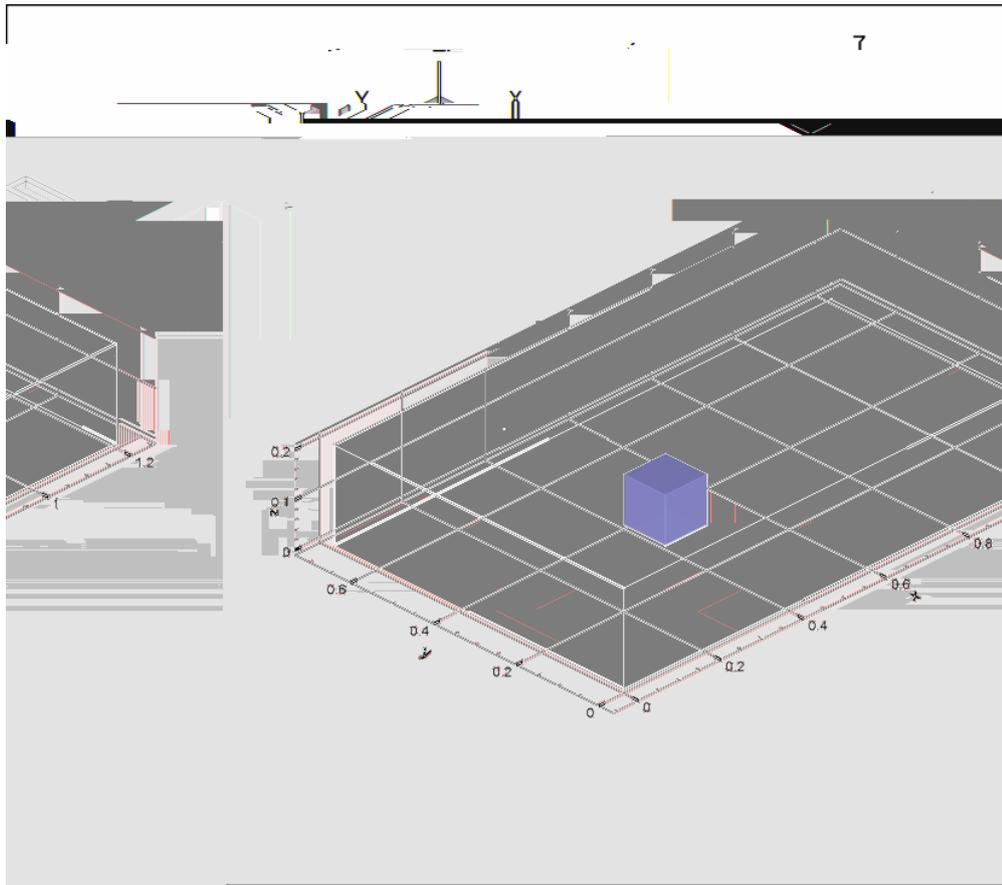


Figura 7.23 - Esquema de posicionamento de um cubo sobre uma parede plana.

Para fins de redução de custo computacional para a avaliação da função indicadora, foi construída uma “caixa” na qual a Fronteira imersa é inserida, Lima e Silva (2002). As dimensões desta caixa devem ser tal que caibam todos os pontos de interpolação, como os indicados na Figura 4.2. A alteração feita, diz respeito a quando pontos de interpolação estariam fora do domínio de cálculo, devido à malha de elementos triangulares da fronteira imersa estar muito próxima, ou solidária à uma parede, como a configuração da geometria mostrada na Figura 7.23. Neste caso, faz-se uma mudança de direção do vetor onde os pontos de interpolação infringiriam as fronteiras do domínio, estes novos vetores, apontariam para a direção oposta à original (Figura 7.24), garantindo que todos pontos necessários para interpolação estejam posicionados dentro do domínio euleriano.

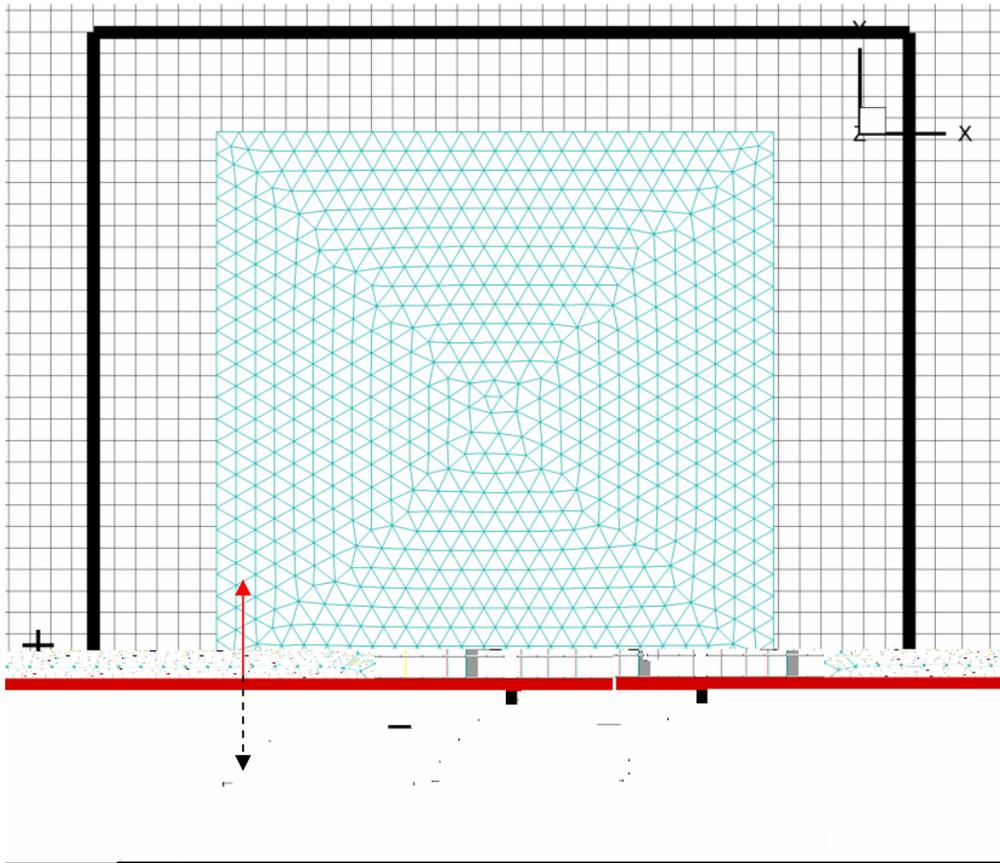


Figura 7.24 - Esquema da mudança na direção de vetores utilizados nos cálculos das interpolações.

A seta pontilhada na região negativa da Figura 7.24 indica o sentido original da posição dos pontos de interpolação das forças lagrangianas, já a seta contínua, na região positiva da figura indica a direção dos pontos de interpolação após a mudança de sentido.

Esta mudança no sentido de alguns vetores para interpolação não representou nenhum problema ao Método da fronteira imersa, uma vez que os resultados obtidos após esta alteração mantiveram-se fisicamente coerentes. Especificamente para o caso do escoamento sobre o cubo solidário a base, uma outra alteração foi necessária. Uma constante *ad hoc* foi adicionada ao termo de aceleração do cálculo da força lagrangiana, desta maneira, para este caso particular, a avaliação da força de aceleração lagrangiana é dada pela equação (7.10):

$$\vec{F}_{acc} = \frac{\rho}{C} \left(\frac{-U_{ki}}{\Delta t} \right) \quad (7.10)$$

onde C é uma constante de valor 0,1, e U_{ki} , é a velocidade do ponto lagrangiano k em um determinado tempo..

Esta mudança foi necessária, pois notou-se durante as simulações, que esta configuração de geometria, por possuir uma razão de bloqueio alta (da ordem de 7,3%), e também devido ao fato de estar em uma região de baixa velocidade, os cálculos da fronteira imersa não geravam uma densidade de forças suficiente para repelir a entrada de fluido para a região da fronteira imersa. Com o uso da constante C , mostrado na eq (7.10), houve uma grande melhora nos resultados.

Hwang e Yang (2004) estudaram o escoamento sobre um cubo solidário a uma parede em $z=0$. As dimensões do domínio computacional, tanto para o trabalho dos autores citados, quanto para as simulações na presente dissertação pode ser visto na Figura 7.23. Este domínio foi dividido em quatro subdomínios para processamento paralelo. Como condições de contorno do domínio euleriano tem-se parede em $z=0$ e superfícies livres em $z=z_{\max}$, $y=0$ e $y=y_{\max}$. Foram realizadas simulações para Reynolds 350 e 1.000. O domínio euleriano têm dimensões 1,2 x 0,7 x 0,2 m e é discretizado em uma malha de 196x136x79 volumes. A Figura 7.25 mostra isosuperfícies de Q obtidos por Hwang e Yan, (2004). Pode-se notar a grande semelhança da Figura 7.25 com a Figura 7.26, obtida nas mesmas condições de simulação, porém com o código Fluids-3D.

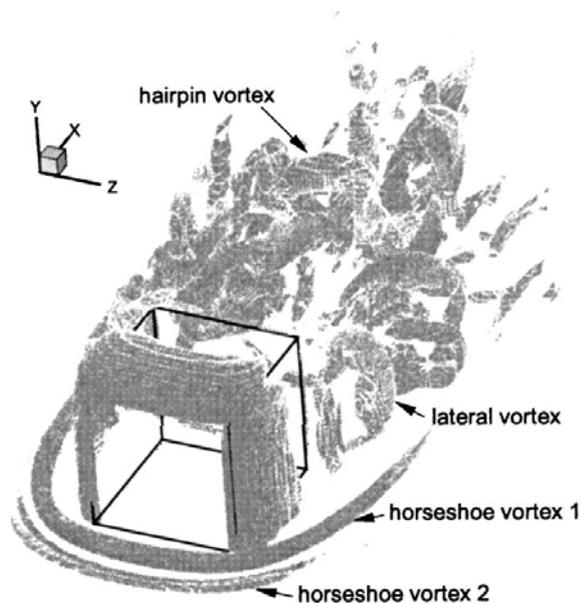


Figura 7.25 - Isosuperfícies de Q obtidos por Hwang e Yang (2004), $Re = 1.000$.

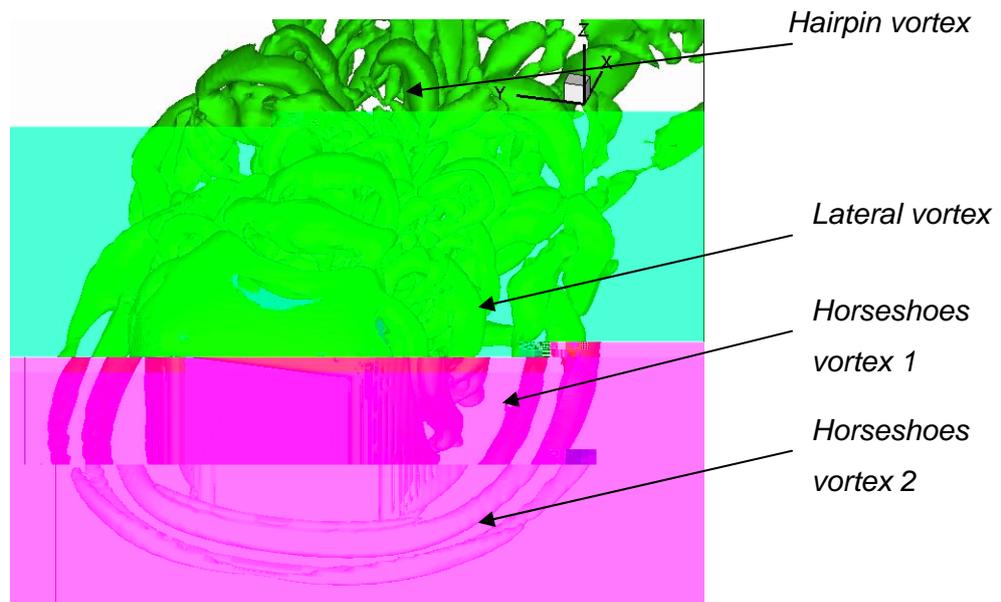


Figura 7.26 - Isosuperfícies de Q obtidos no presente trabalho, $Re = 1000$.

Pode-se notar também a grande semelhança entre a Figura 7.27 e Figura 7.28 , sendo a primeira foi obtida no trabalho de Hwang e Yang (2004), e a ultima obtida no presente trabalho.

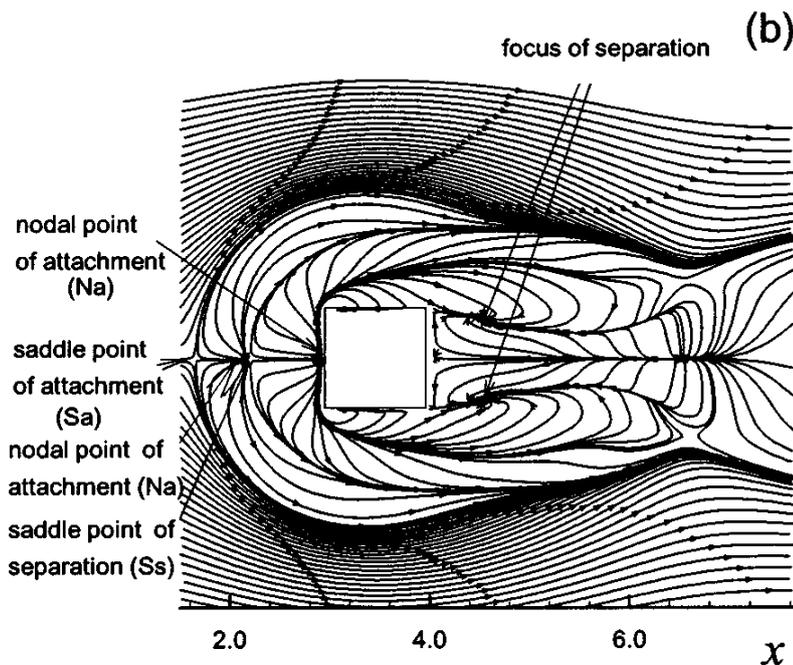


Figura 7.27 - Linhas de corrente a $Re = 350$ plano xz , $x = 0,0053$ m, retirado de Hwang e Yang (2004).

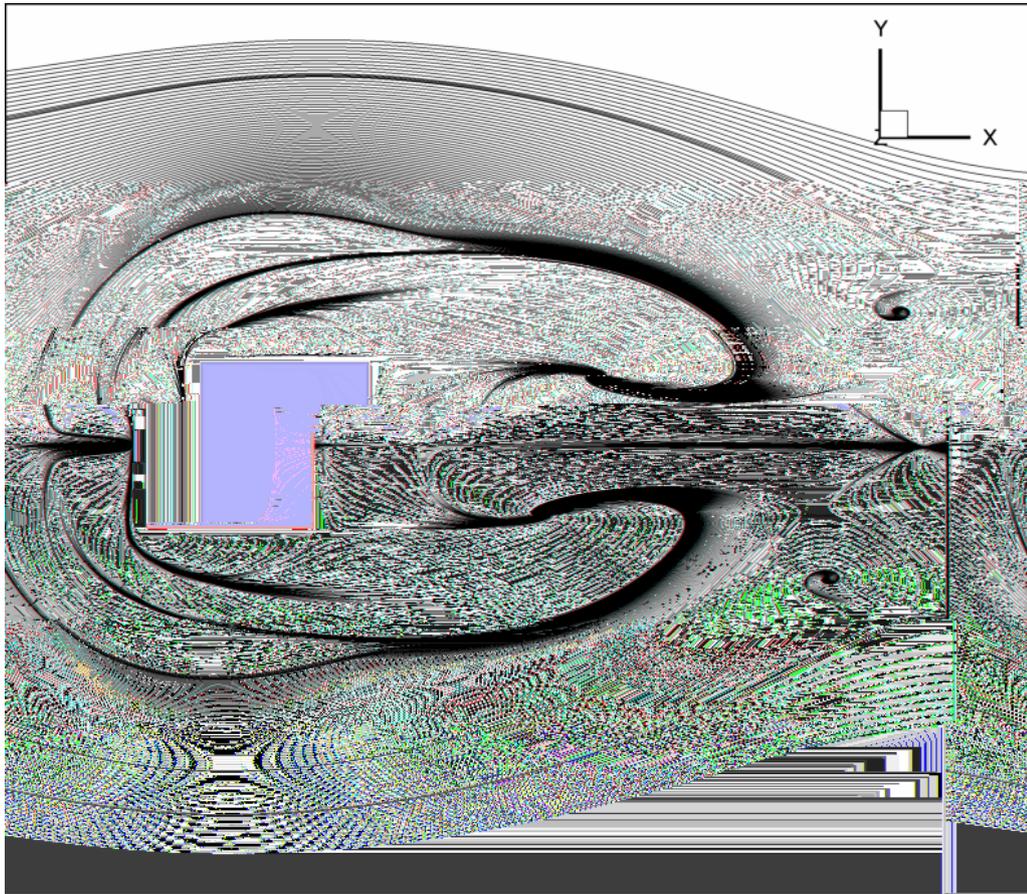


Figura 7.28 - Linhas de corrente vistas em um plano $z = 0,001$ m, $Re=350$.

A Figura 7.29 e a Figura 7.30 mostram linhas de corrente tridimensionais indicando a formação de vórtices tipo ferradura de cavalo em duas vistas distintas, a primeira mostra uma vista tridimensional, e a última mostra uma vista do plano xz . A Figura 7.31 mostra isosuperfícies de $Q = 75$, coloridas por valores da velocidade u , para Reynolds 1.000.

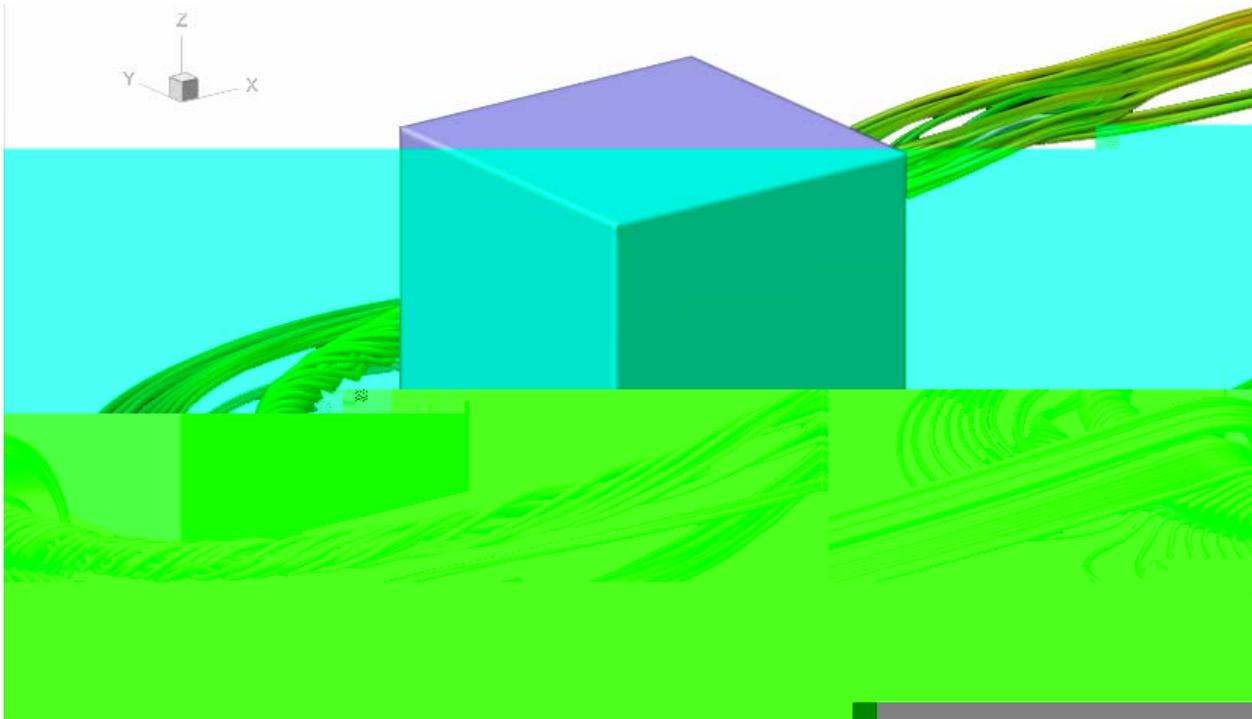


Figura 7.29 - Linhas de corrente tridimensionais do escoamento sobre um cubo solidário a base, $Re= 1.000$.

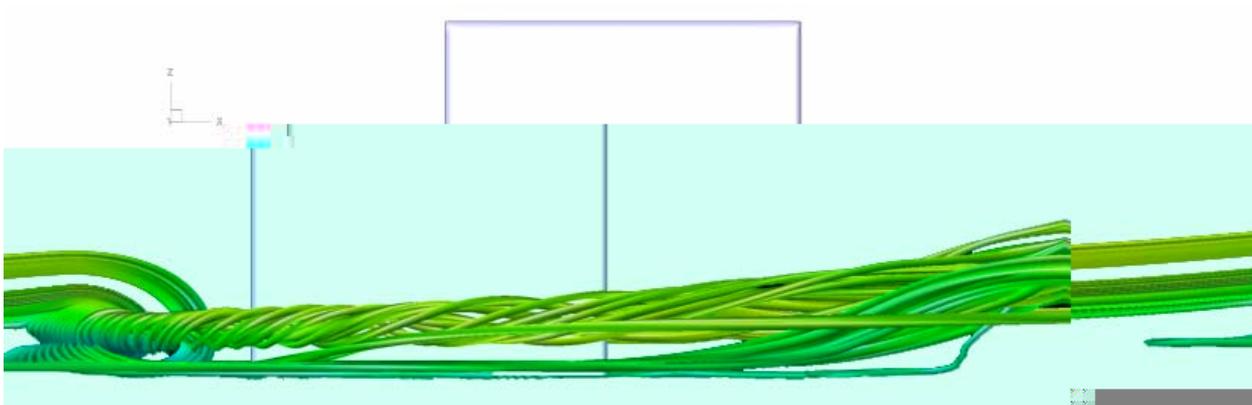


Figura 7.30 - Linhas de corrente tridimensionais do escoamento sobre um cubo solidário a base, vista do plano xz, $Re= 1.000$.

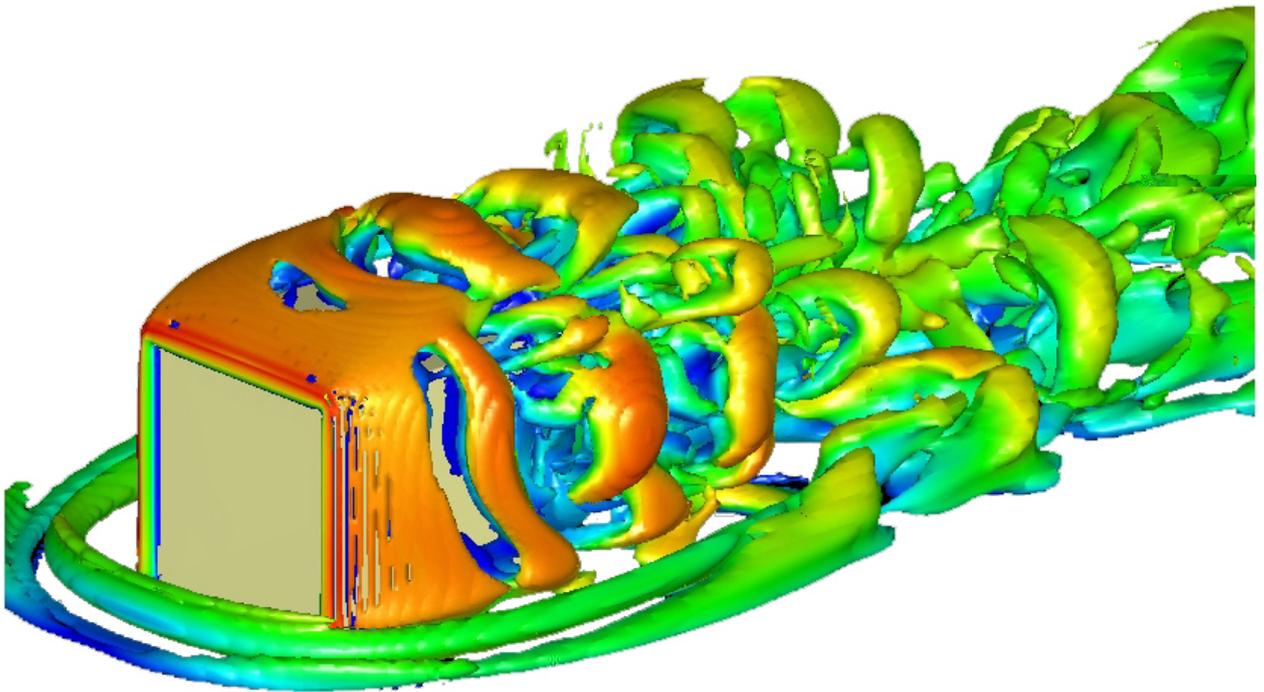


Figura 7.31 - Isosuperfícies de $Q = 75$, coloridas por valores da velocidade u , para $Re = 1.000$.

É feita uma comparação quantitativa entre escoamentos a números de Reynolds 350 e 10.000. Na Figura 7.32, pode-se notar três pares de curvas de evolução temporal. Estes pares de curvas representam, para os dois regimes de escoamento (Reynolds 350 e 1.000), o coeficiente de arrasto, coeficientes lateral e de sustentação, conseqüentemente. Nota-se na Figura 7.32, que existe um leve incremento no valor do coeficiente de arrasto para Reynolds 350, o que é esperado fisicamente. Já para os coeficientes lateral e de sustentação, nota-se que não há diferenças significativas, uma vez que cada par de curvas oscila, respectivamente próximo a um dado valor, ou seja, o coeficiente lateral, tanto para Reynolds 350 quanto para Reynolds 1000, oscila próximo à 0,4, já o coeficiente de sustentação, oscila com valores próximos a zero para ambos as simulações.

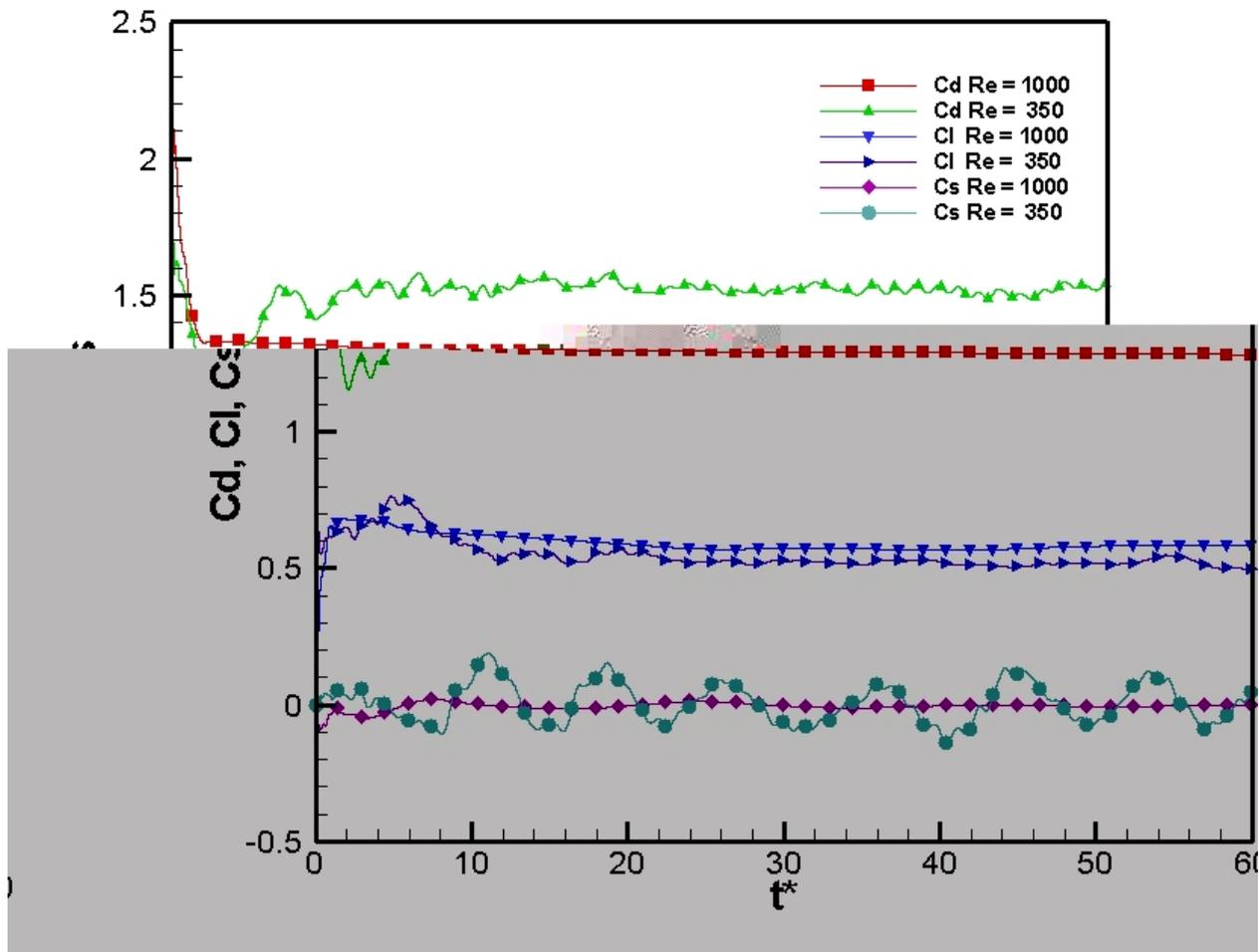


Figura 7.32 - Evolução temporal de parâmetros quantitativos para o escoamento sobre um cubo solidário à uma parede.

7.2.3 Escoamento sobre uma estrutura treliçada

Dando prosseguimento ao objetivo de testar a robustez do código Fluids-3D, e à facilidade de importação de geometrias complexas, foi escolhido um corpo imerso tipo estrutura treliçada, mostrada na Figura 7.33. Esta geometria foi escolhida devido ao fato de possuir várias junções e travamentos, o que a priori, poderiam fazer com que o código divergisse ou

não apresentasse resultados consistentes fisicamente. O modo como esta geometria é inserida com domínio euleriano é mostrado na Figura 7.34.

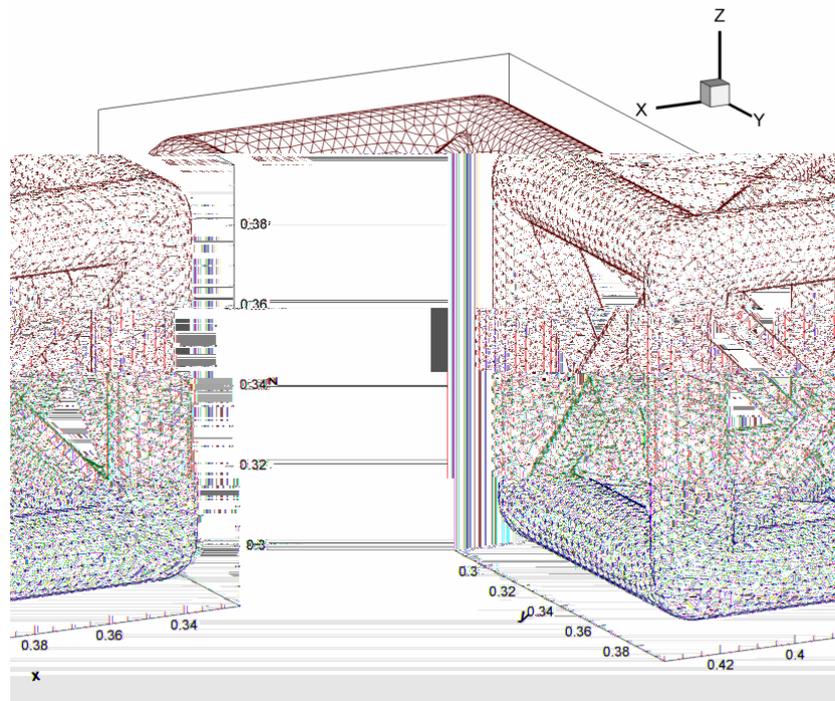


Figura 7.33 - Malha de elementos triangulares utilizada para representar uma estrutura treliçada.

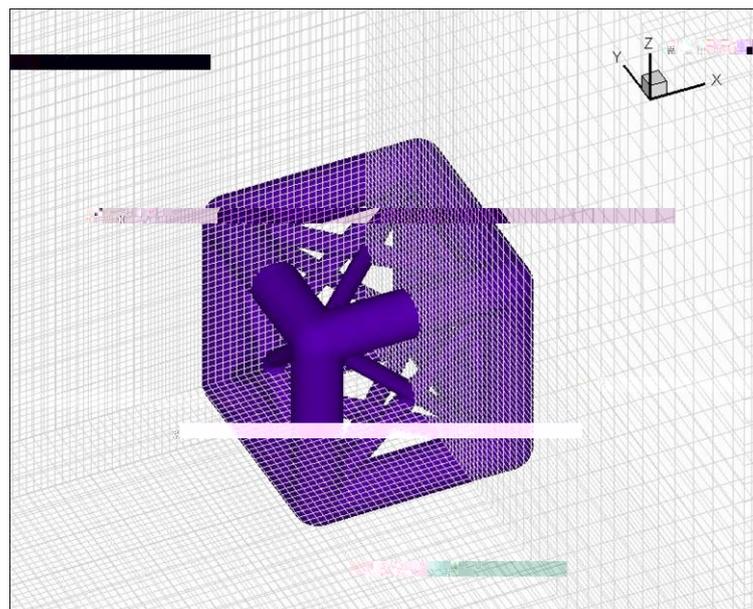


Figura 7.34 – Geometria da Figura 7.33 inserida no domínio euleriano.

O processo de geração de uma malha como a mostrada na Figura 7.33 segue alguns passos básicos, que podem ser listados:

- Desenho da geometria em alguma ferramenta CAD (Computer Aided Design). Usa-se uma ferramenta como esta devido à facilidade de desenho;
- Alguns softwares CAD, já vêm com a opção de exportar a geometria de interesse no formato **.stl*. Caso a ferramenta CAD não ofereça um controle adequado da exportação de um arquivo **.stl*, deve-se exportar a geometria em algum formato que pode ser lido por um gerador de malhas comercial, como a extensão **.sat*, por exemplo. A estrutura treliçada seguiu esta última opção.
- Caso o arquivo **.stl* esteja adequado, simplesmente insere-se o nome deste arquivo em um dos arquivos de *set-up* adequado (os arquivos de *set-up* já foram listados no capítulo 5).
- Caso haja a necessidade do uso de um gerador de malhas comercial, importa-se neste gerador a geometria recém gerada e cria-se uma malha de elementos triangulares. Como gerador de malhas comercial, neste trabalho fez-se o uso do gerador de malha do software Ansys.
- O Ansys possui uma opção de listar todos o nós de uma malha de elementos triangulares, bem como suas conectividades.
- De posse dos arquivos de nós e conectividades, simplesmente insere-se o nome destes arquivos nos arquivos de *set-up* adequados (já listados no capítulo 5).

O processo de criação da geometria em um software CAD, bem como a geração da malha, para uma geometria como a da Figura 7.33, não exige mais que 20 minutos.

O escoamento da estrutura treliçada foi comparado com o escoamento sobre um cubo imerso, na mesma posição e com as mesmas dimensões, e também com arestas arredondadas. Este corpo pode ser visto na Figura 7.35

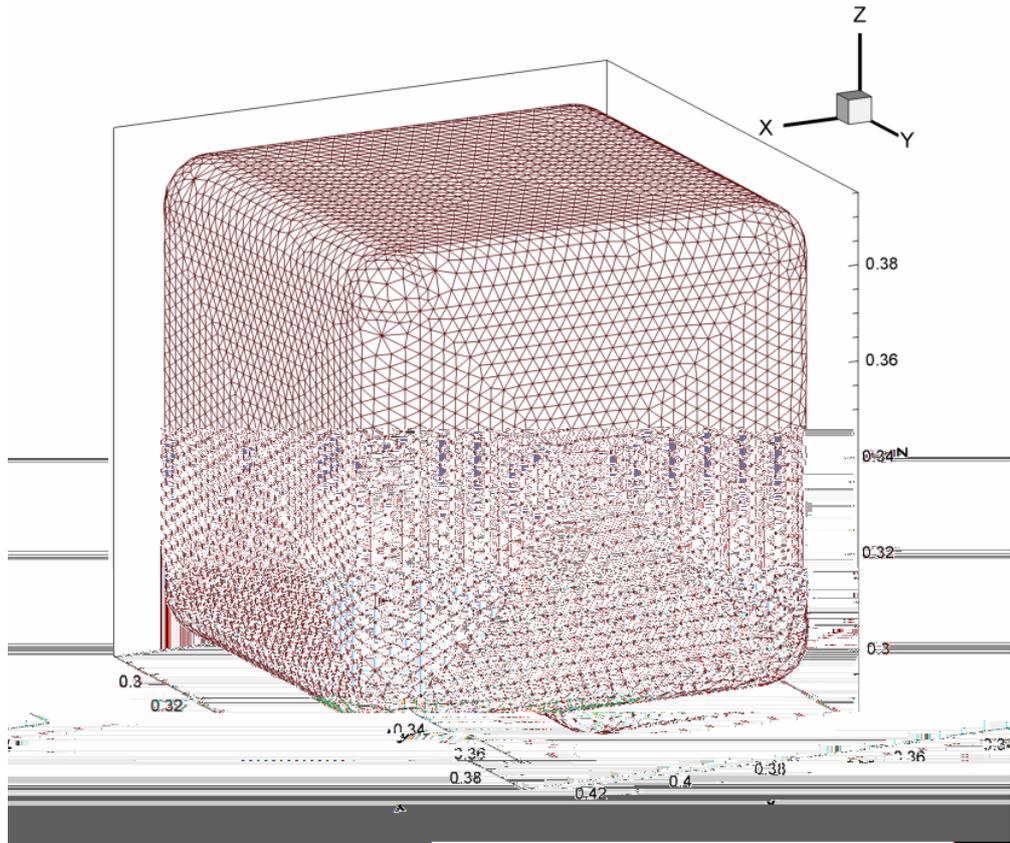


Figura 7.35 - Cubo com arestas arredondadas representado por malhas de elementos triangulares.

O domínio euleriano utilizado para ambas as configurações de geometrias, já subdividido em quatro subdomínios para processamento em paralelo é mostrado na Figura 7.36. Tal domínio têm dimensões máximas de $1,0 \times 0,68 \times 0,68 m$ e é discretizado em uma malha de $166 \times 116 \times 116$ volumes. Vale ressaltar que devido à paralelização da fronteira imersa ainda não ter sido implementada, faz-se com que esta esteja confinada em um único processador, neste caso, o processador que contém o objeto imerso é, na Figura 7.36, o segundo da esquerda para direita.

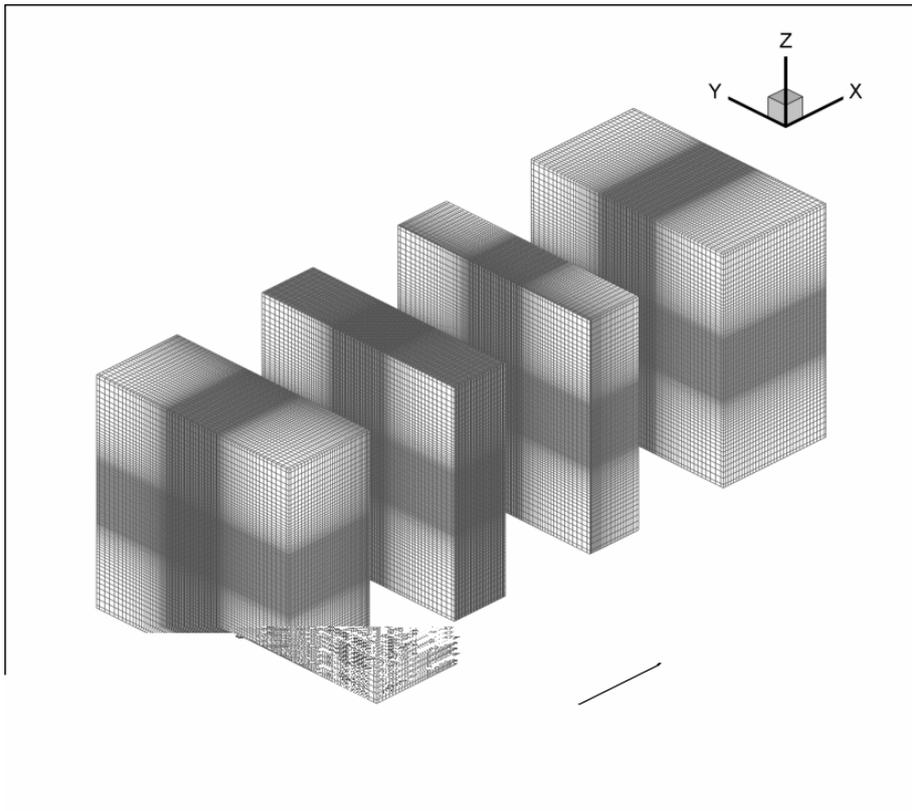


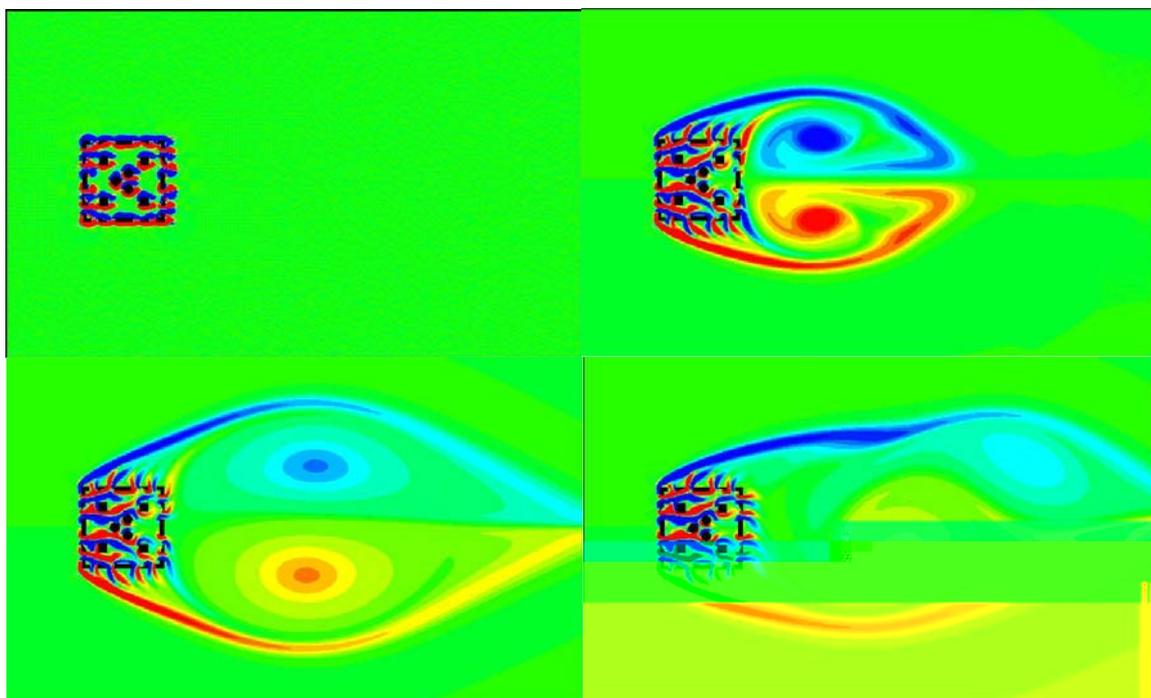
Figura 7.36 - Malha euleriana para simulação do escoamento sobre uma estrutura treliçada.

Como condições de contorno do domínio euleriano tem-se paredes nas faces $y=0$, $y=y_{\max}$, $z=0$ e $z=z_{\max}$. O perfil de velocidades na entrada é uniforme com $u=U_{\infty}$, $v=w=0$. Todas as simulações foram feitas para um número de Reynolds $Re = 1.000$.

Lima e Silva et al. (2005), utilizando o Método da fronteira imersa, simulou o escoamento bidimensional sobre um trecho de uma torre de extração de petróleo (Figura 7.37). O resultado de tal simulação mostrou-se interessante do ponto de vista físico, pois mostrou que ao invés de o escoamento criar pequenas estruturas do tamanho característico dos elementos que compõe a geometria imersa, notou-se a criação de vórtices da mesma ordem de grandeza da estrutura macro. Amostras de tal resultado podem ser visualizadas na Figura 7.38.



Figura 7.37 - Indicação de um trecho de torre de plataforma de petróleo, Lima e Silva et al. (2005).



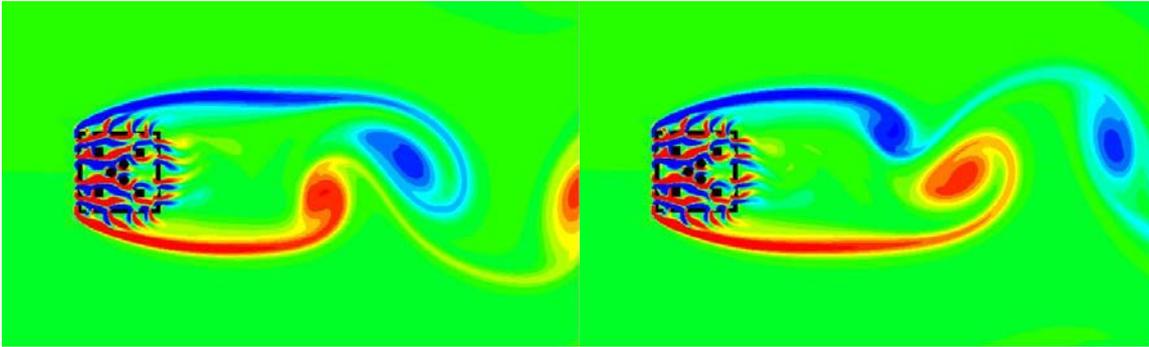
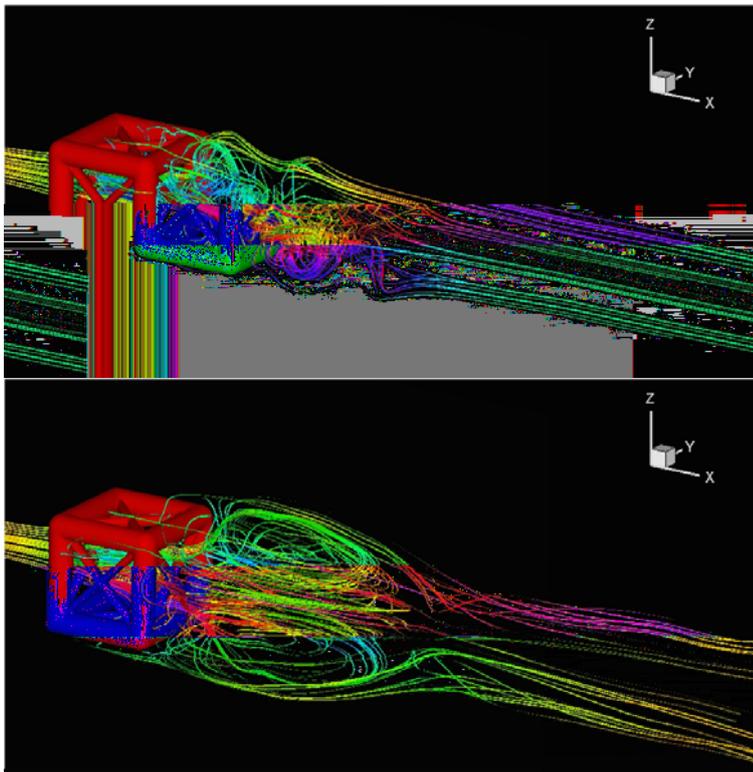


Figura 7.38 - Escoamento sobre uma estrutura bidimensional treliçada, Lima e Silva et al. (2005).

Pode-se notar na Figura 7.39 que recirculações tridimensionais são evidenciadas.. A Figura 7.39 mostra três estágios do escoamento sobre a estrutura treliçada: geração, evolução e evolução para um estado caótico da linha de corrente a jusante da estrutura. Pode se ver a bolha de recirculação mais claramente na Figura 7.40.



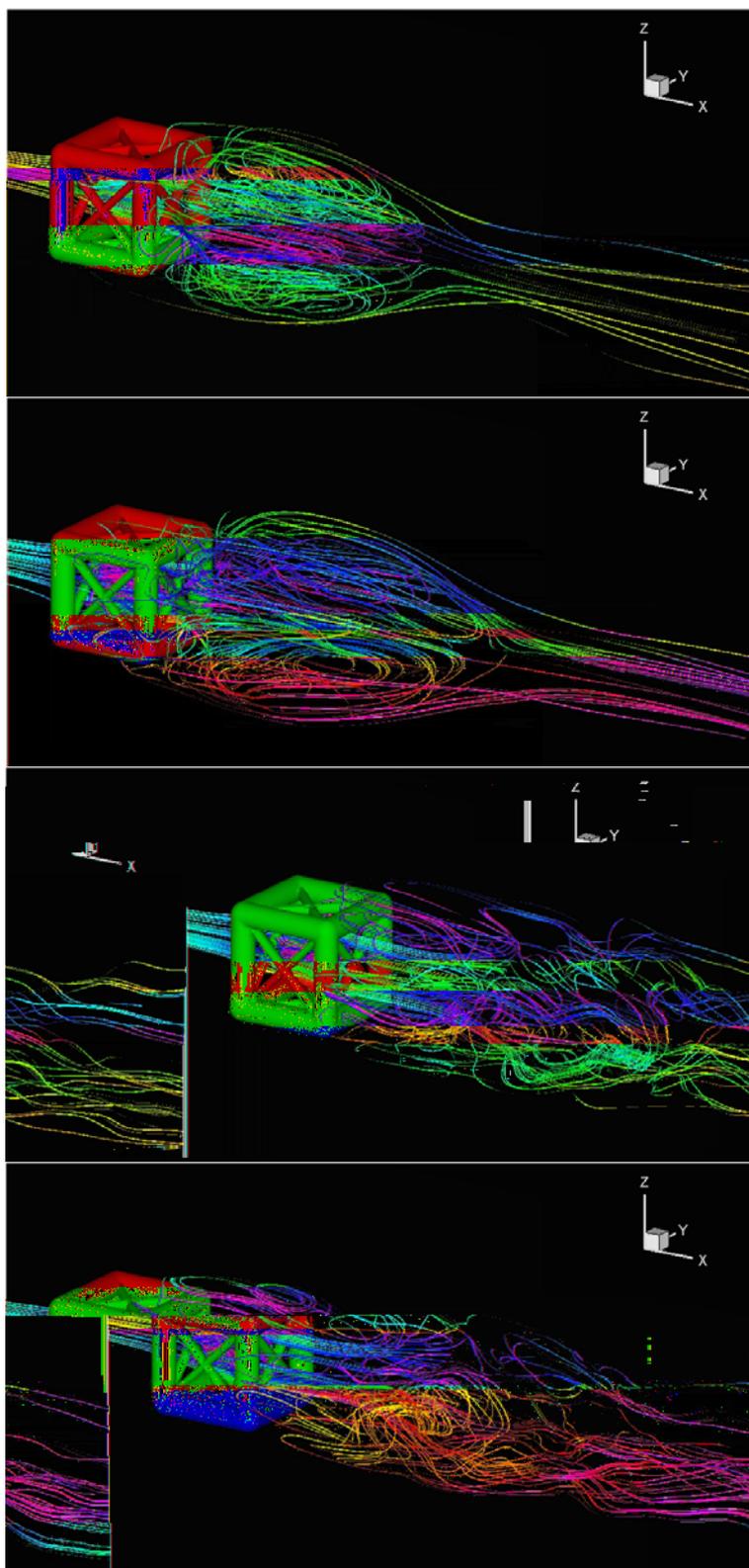


Figura 7.39 - Evolução de linhas de corrente para escoamento sobre uma estrutura treliçada à $Re = 1.000$, nos seguintes tempos físicos (de cima para baixo): $t = 0,3$ s, $t = 1,4$ s, $t = 1,6$ s, $t = 2,0$ s, $t = 9,5$ s e $t = 10,0$ s.



Figura 7.40 - Planos indicando a presença de bolhas de recirculação, em $t = 2,0$ s e $Re=1.000$.

A Figura 7.41 mostra a evolução de isosuperfícies de $Q = 5$, para diversos estágios temporais. Além das recirculações citadas, é possível notar também a presença de estruturas turbilhonares tipo grampo de cabelo, indicativo de consistência física, uma vez que este tipo de estrutura é tipicamente comum em escoamentos sobre corpos rombudos.

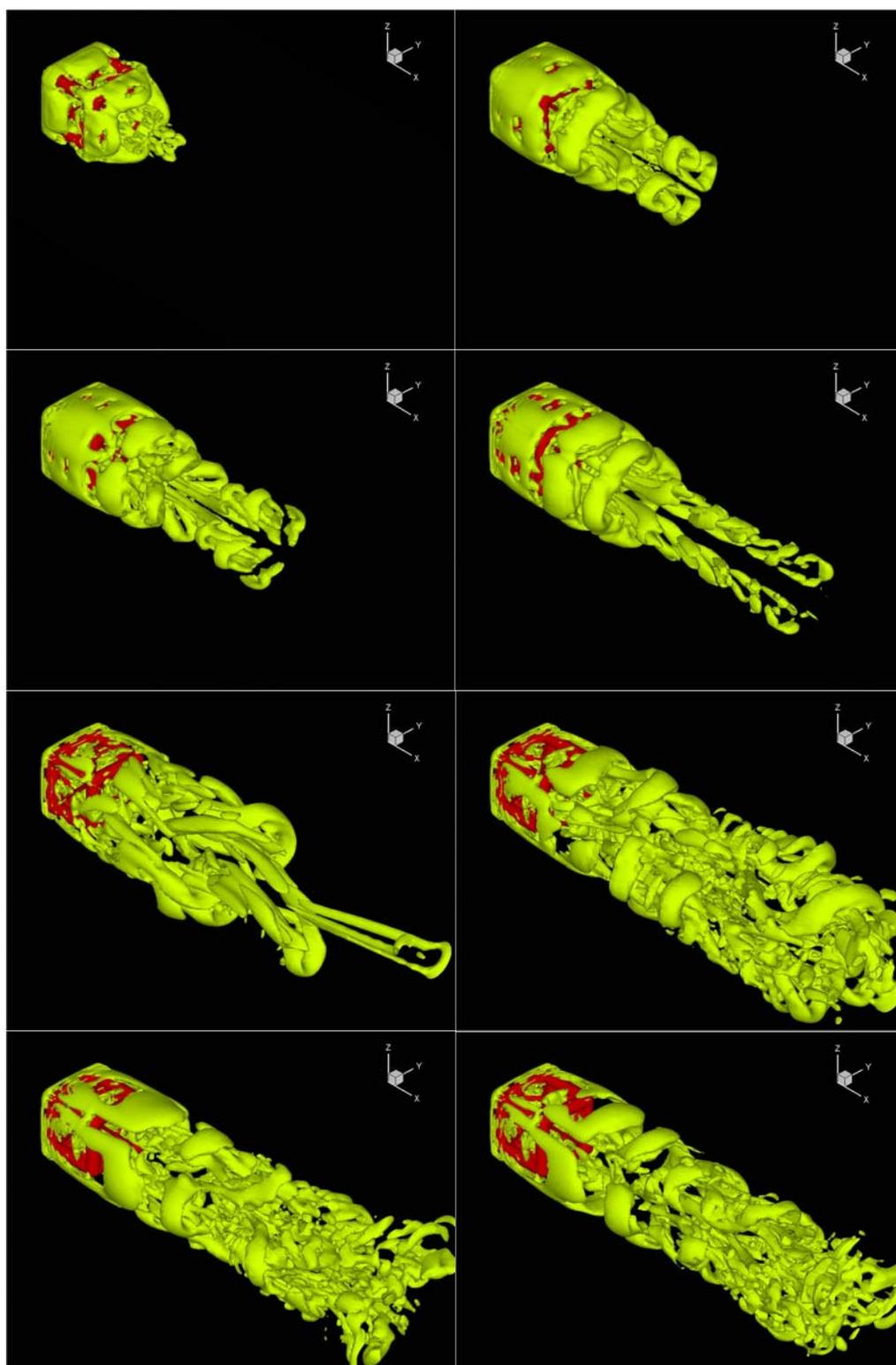


Figura 7.41 - Iso superfícies de $Q = 5$, para os seguintes tempos físicos (de cima para baixo, da esquerda para direita): $t = 0,1$ s, $t = 0,5$ s, $t = 0,6$ s, $t = 0,9$ s, $t = 2,5$ s, $t = 3,0$ s, $t = 4,5$ s e $t = 10,0$ s.

A Figura 7.42 mostra uma comparação quantitativa entre os escoamentos sobre a geometria treliçada e a geometria mostrada na Figura 7.35. Nota-se um fato interessante. O coeficiente de arrasto para a estrutura treliçada é ligeiramente maior que o coeficiente de arrasto do cubo com faces arredondadas. Apesar de intuitivamente, acreditar-se que o coeficiente de arrasto para a estrutura seria menor devido à razão de bloqueio da mesma ser menor, os travamentos que compõem a estrutura perturbam muito mais o escoamento do que um cubo de arestas arredondadas, sendo assim, pode-se dizer que o fato de o coeficiente de arrasto da estrutura treliçada ser maior é consistente fisicamente. Este fato foi observado por Lima e Silva et al. (2005).

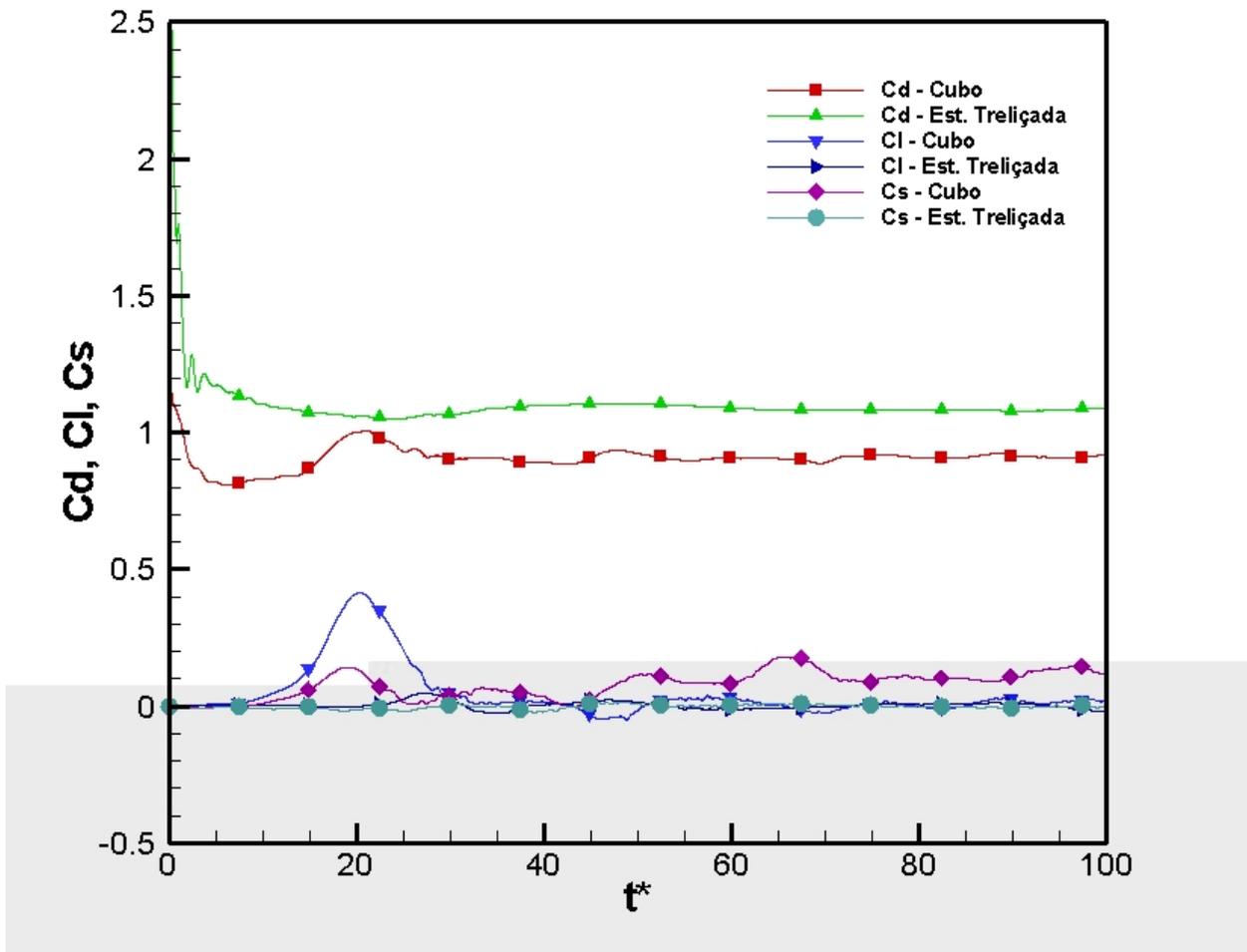


Figura 7.42 - Comparações quantitativas entre o escoamento sobre uma estrutura treliçada e um cubo de arestas arredondadas, $Re = 1.000$.

7.2.4 Escoamento sobre um protótipo de automóvel

Nesta seção serão apresentados resultados de simulações feitas do escoamento sobre um protótipo de automóvel. A geometria do protótipo em questão foi baseada em um automóvel esportivo, o Lamborghini Gallardo. Mais informações sobre a geração do arquivo *.stl que representa tal automóvel pode ser encontrado em Oliveira (2007). A malha de elementos triangulares que representa o protótipo, mostrada na Figura 7.43, é composta por 11.486 nós e 22.953 elementos, e é posicionada no domínio euleriano de acordo com a Figura 7.44.

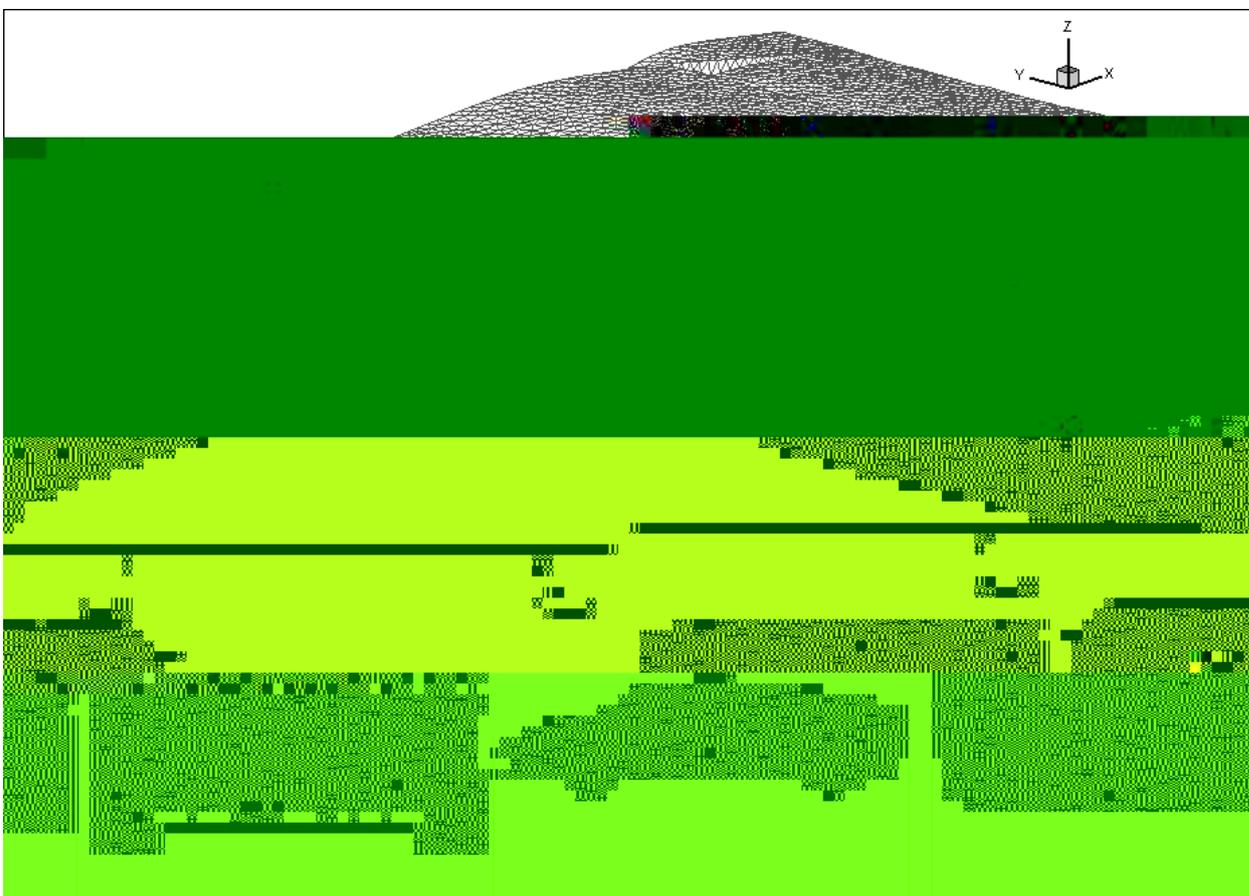


Figura 7.43 - Malha de elementos triangulares representando um protótipo de automóvel (OLIVEIRA, 2007).

Apesar de ser um modelo simplificado de um automóvel, cuidados foram tomados para que o protótipo tivesse características próximas a um automóvel, como a criação de rodas independentes, como mostrado na Figura 7.45. O domínio euleriano possui dimensões tais que

na direção x , a montante tem-se 2,5 vezes o comprimento do protótipo, e a jusante, 6 vezes o seu comprimento. Na direção y , em ambas as laterais, tem-se 1,5 vezes a largura do protótipo e na direção z , tem-se uma altura 3,5 vezes a altura do protótipo. A malha euleriana é composta por $263 \times 86 \times 60$ volumes, subdivididos em quatro processadores. Como condições de contorno do domínio euleriano tem-se parede em $z=0$ e superfícies livres em $z=z_{\max}$, $y=0$ e $y=y_{\max}$. O perfil de velocidades na entrada é uniforme com $u=U_{\infty}$, $v=w=0$. O número de Reynolds é 2.180, baseado no comprimento do objeto imerso.

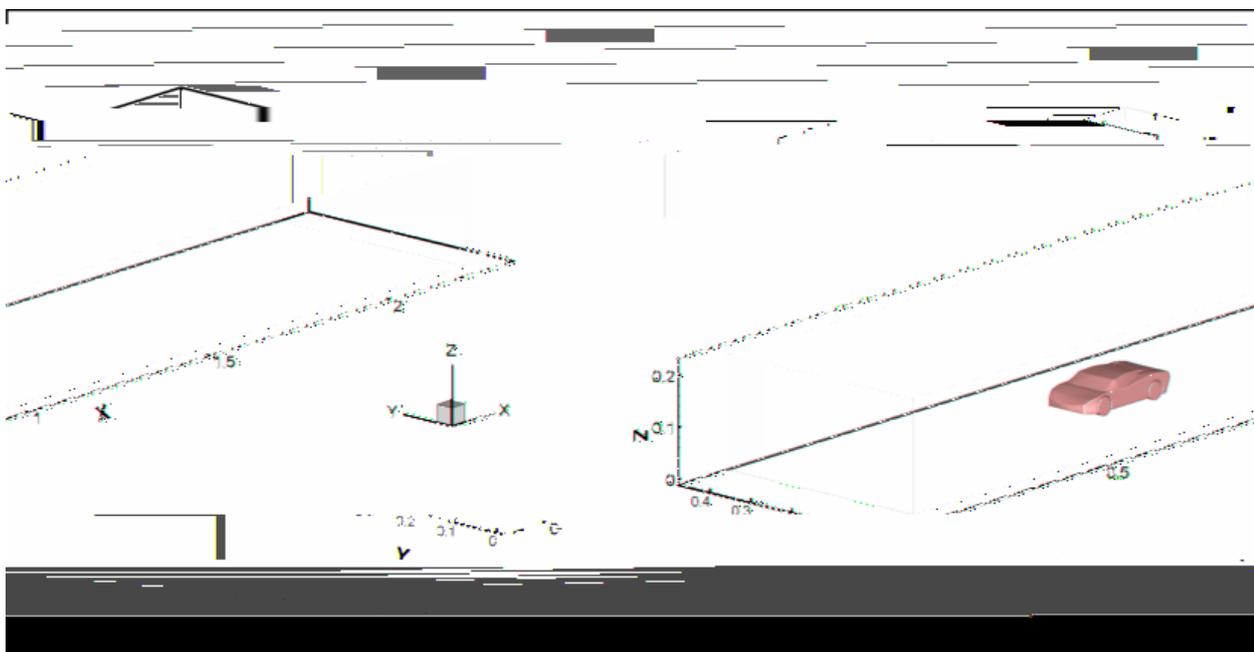


Figura 7.44 - Domínio de cálculo para simulação do escoamento sobre um protótipo de automóvel.

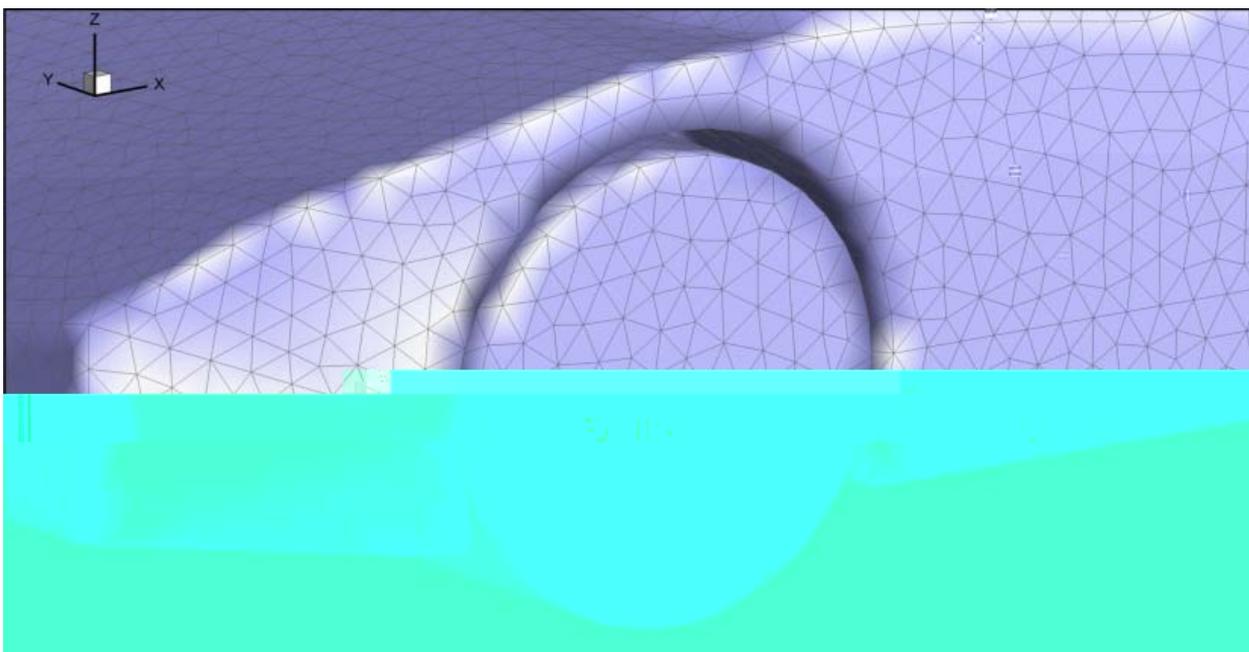
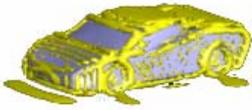
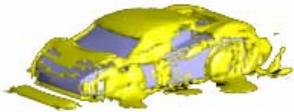


Figura 7.45 - Detalhe da roda dianteira do protótipo do automóvel.

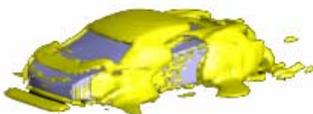
A Figura 7.46 (a)-(f) mostra a evolução de isosuperfícies de $Q = 5$ para o escoamento sobre o protótipo de automóvel. É possível notar estruturas coerentes como os grampos de cabelo a jusante e a montante, e estruturas do tipo ferradura de cavalo. É importante dizer que as estruturas tipo ferradura de cavalo, neste caso, apareceram em consequência da condição de contorno para $z=0$, a qual foi imposta como condição de parede com velocidade nula. Se a parede estivesse a uma velocidade U_∞ , estas estruturas turbilhonares não seriam notadas.



(a)



(b)



(c)

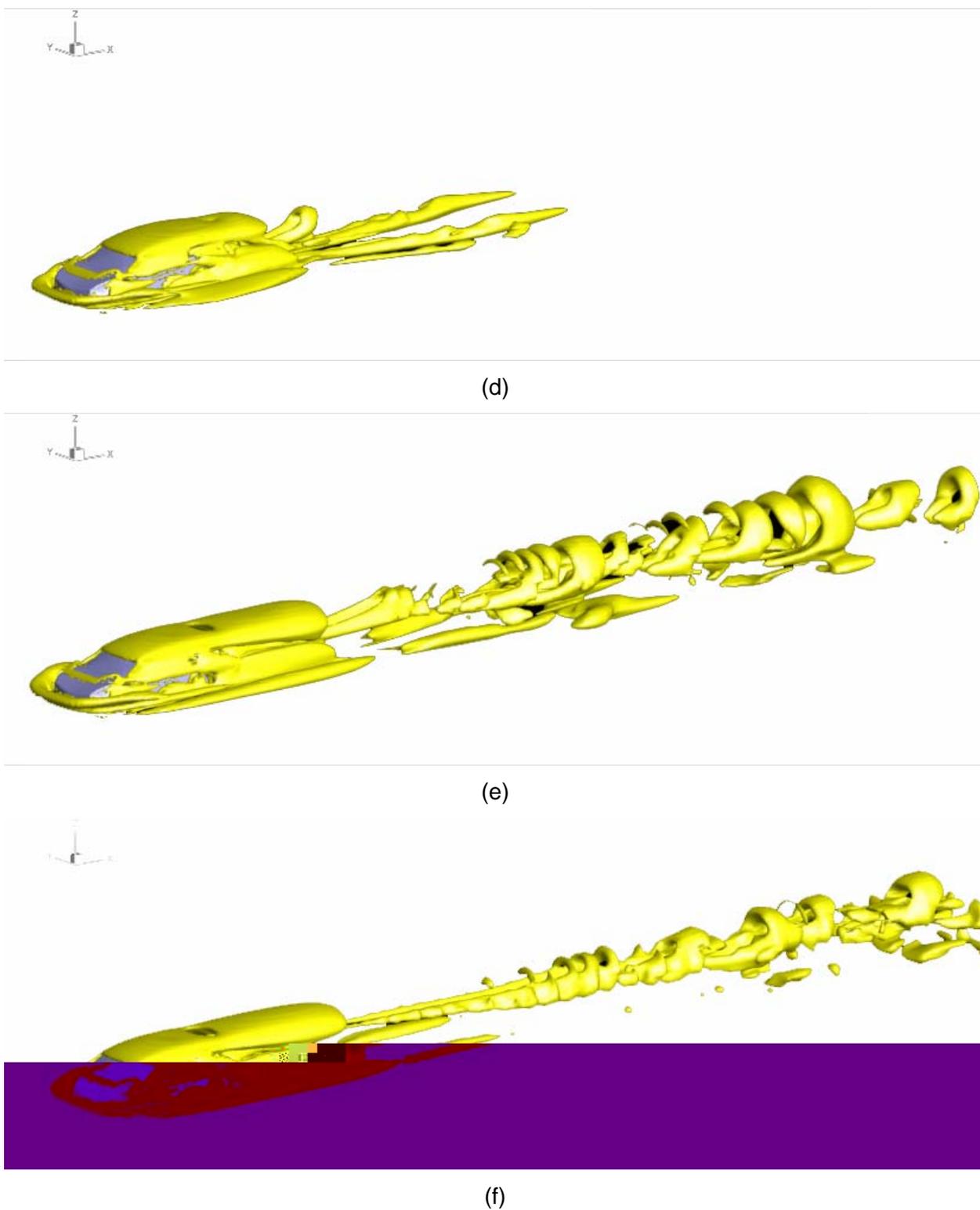


Figura 7.46 - Evolução temporal de isosuperfícies de $Q = 5$ para os seguintes tempos físicos: (a) $t = 0,005$ s, (b) $t = 0,0045$ s, (c) $t = 0,09$ s, (d) $t = 0,8$ s, (e) $t = 2,7$ s e (f) $t = 4,7$ s.

A Figura 7.47 mostra, em três vistas, detalhes das isosuperfícies de $Q = 10$ para $t = 0,6$ s. Nota-se claramente os vórtices laterais, a estrutura tipo ferradura de cavalo a montante, e uma grande estrutura tipo grampo de cabelo surgindo a jusante.

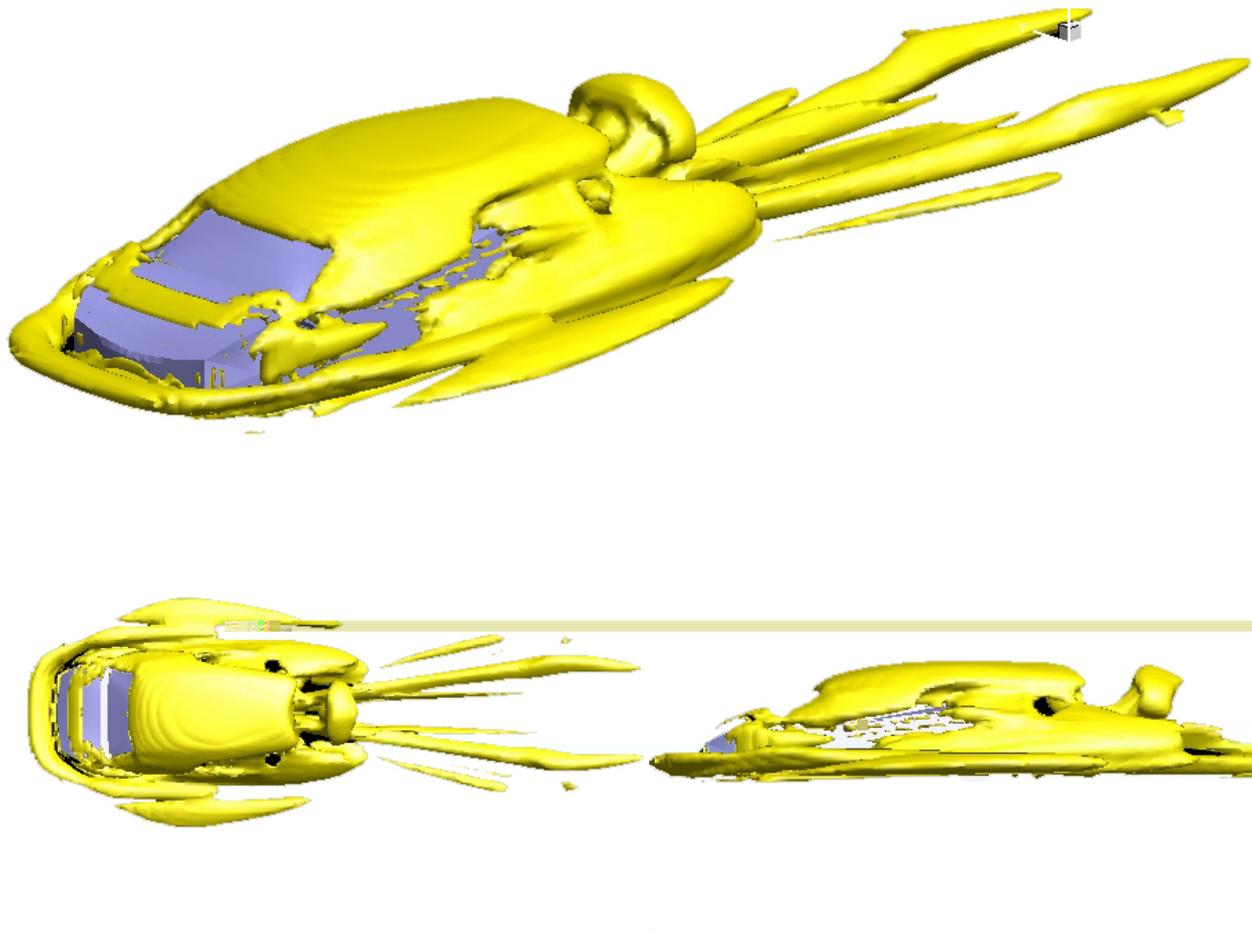


Figura 7.47 - Detalhes das isosuperfícies de $Q = 10$ para $t = 0,6$ s.

A $t = 2,7$ s, a Figura 7.46 (e) e (f), mostram uma grande recirculação logo a jusante da fronteira imersa. Com o intuito de estudar estas estruturas formadas, traçou-se linhas de corrente pelo escoamento. A Figura 7.48 mostra uma vista geral destas linhas de corrente. Nota-se novamente todas as estruturas turbilhonares já citadas, como as ferraduras de cavalo e grampos de cabelo.

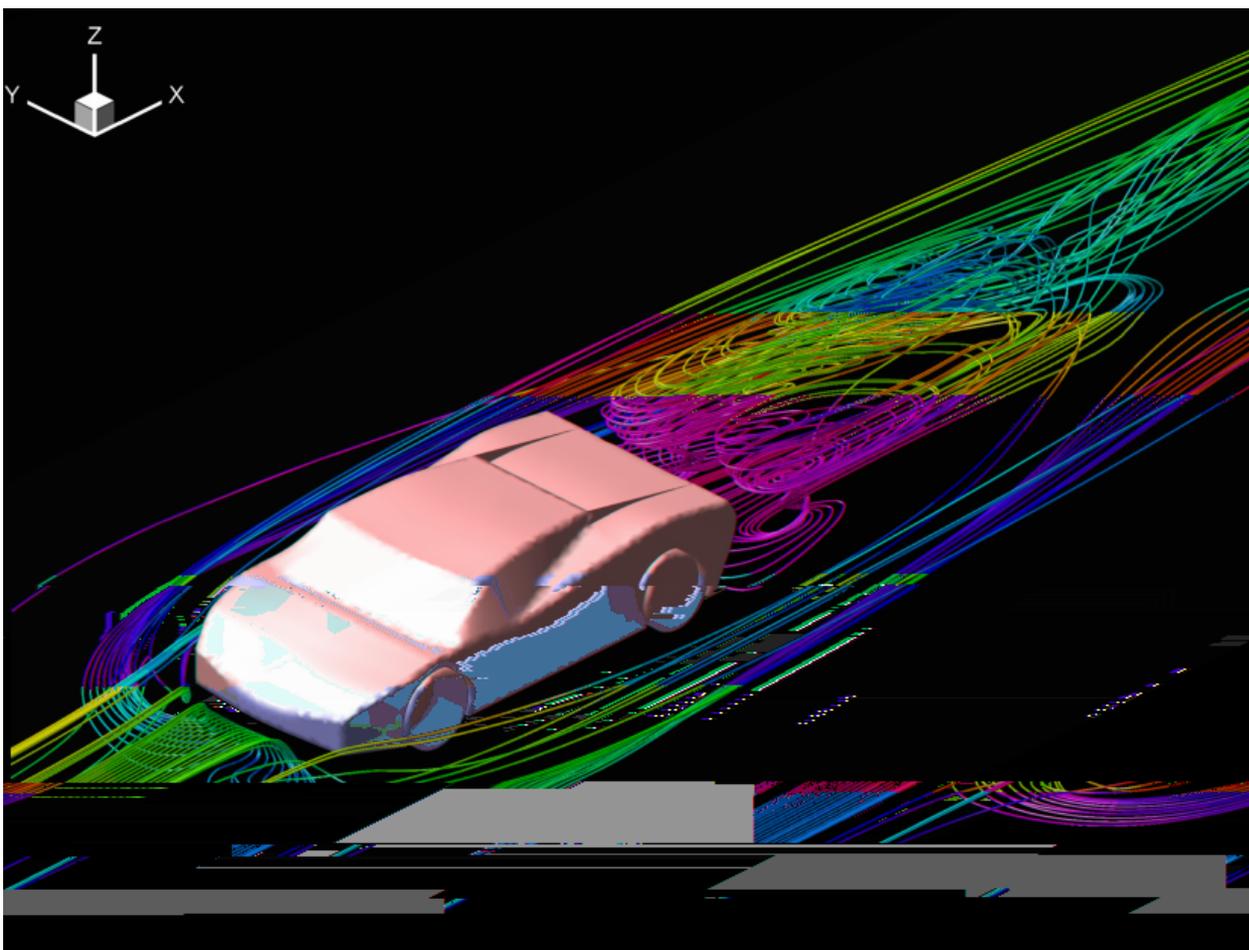


Figura 7.48 - Linhas de corrente do escoamento sobre um protótipo de automóvel, $t = 2,6$ s.

A Figura 7.49 mostra uma vista posterior do objeto imerso. Pode-se notar ainda melhor as recirculações geradas a jusante do protótipo nesta figura. O detalhe do surgimento destas recirculações, para um lado do protótipo pode ser vista na Figura 7.50

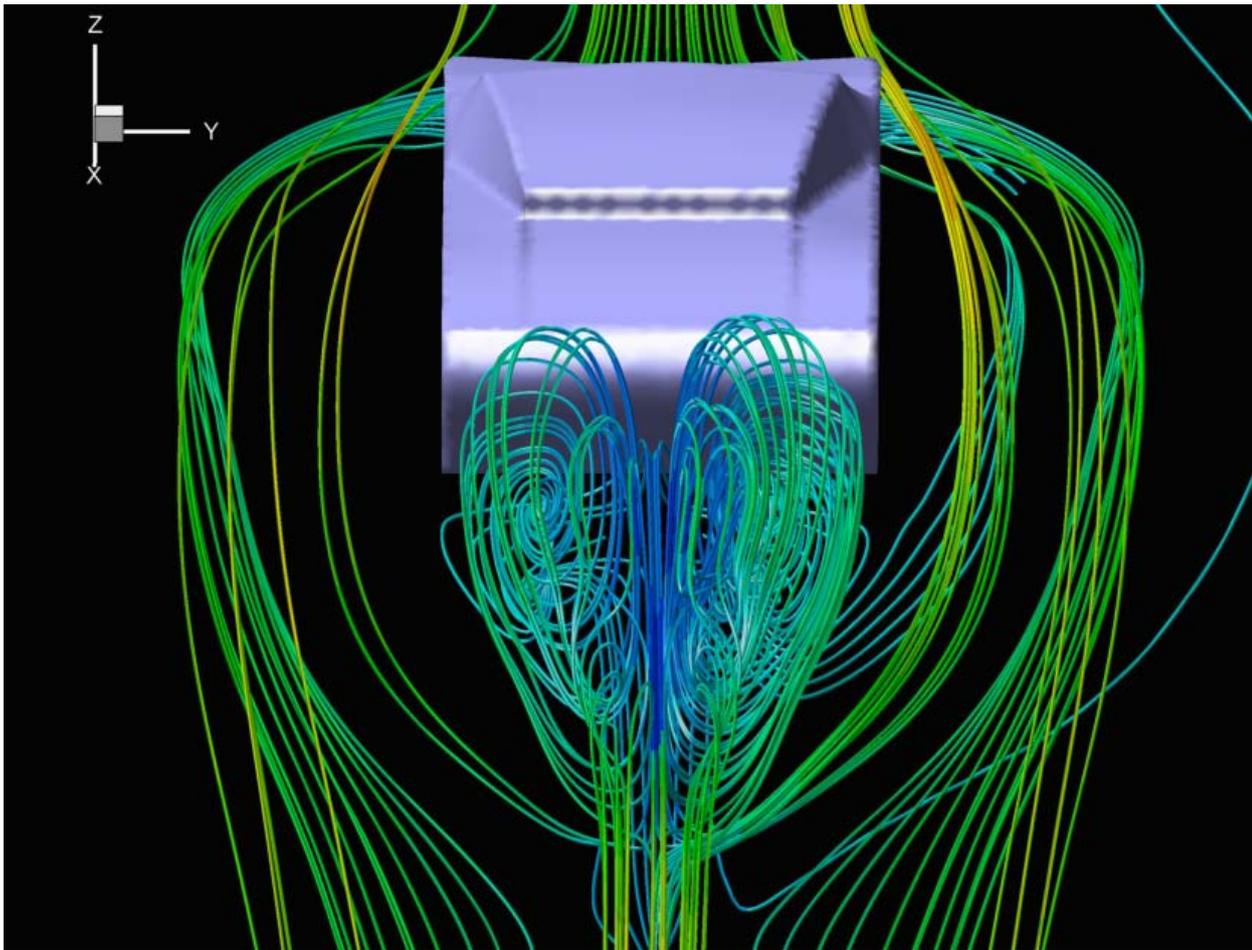


Figura 7.49 - Vista posterior das linhas de corrente a jusante do protótipo de automóvel, $t = 2,6$ s.

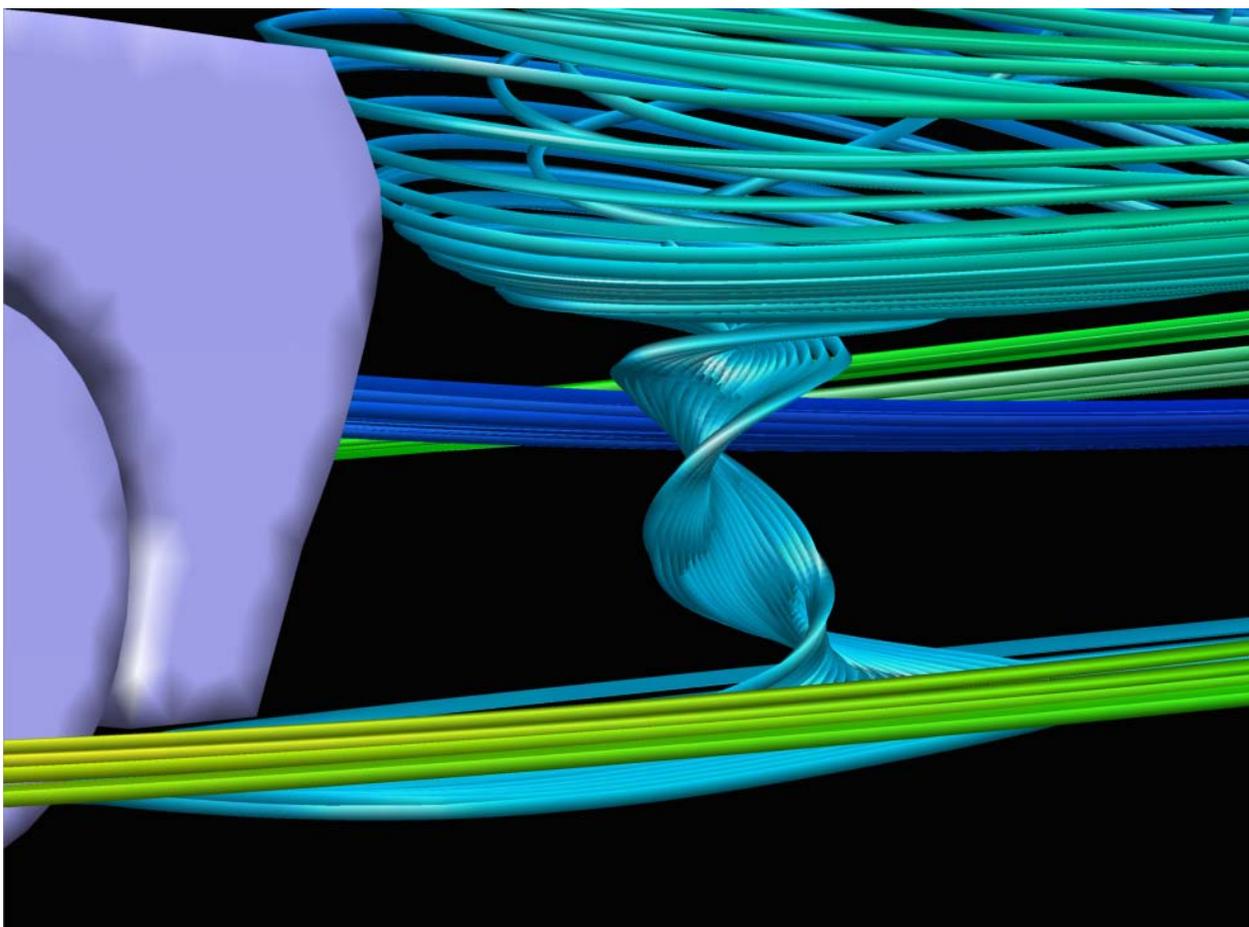
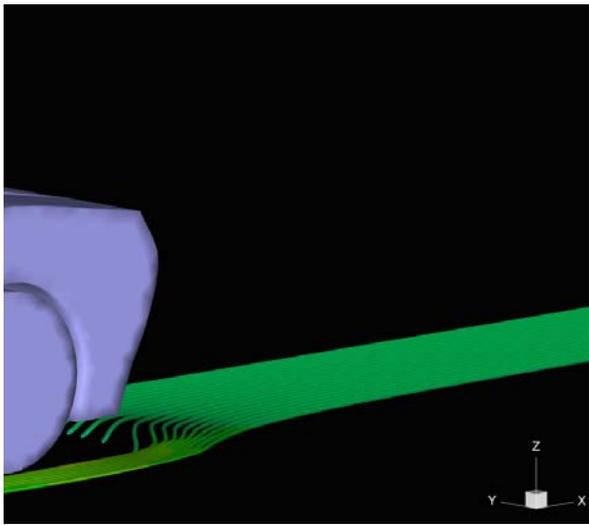
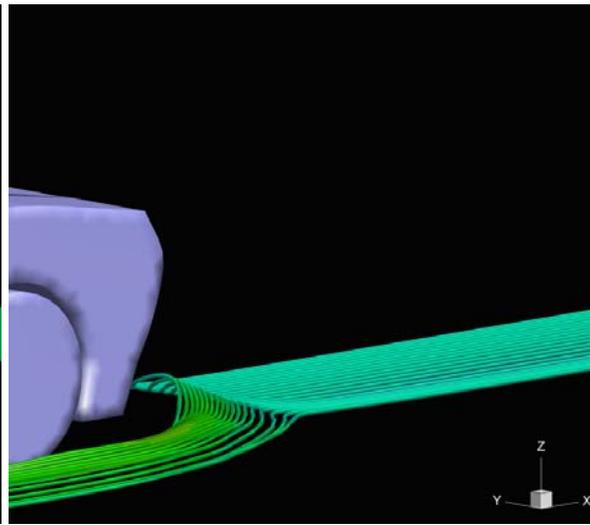


Figura 7.50 - Detalhe do início de uma estrutura helicoidal a jusante do protótipo de automóvel, $t = 2,6$ s.

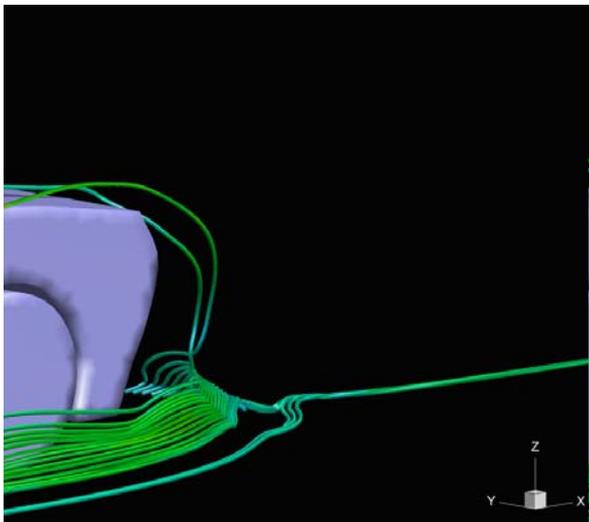
A Figura 7.51 mostra a evolução temporal do vórtice mostrado na Figura 7.50. É possível notar naquela figura, estágios como o surgimento, desenvolvimento e as estruturas turbilhonares sendo transportadas pelo escoamento.



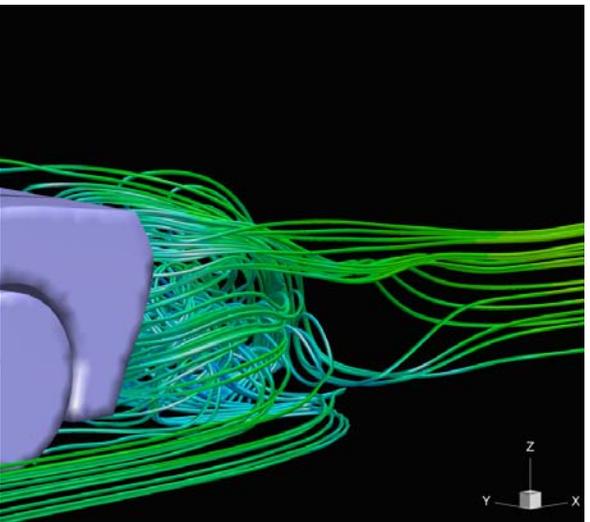
(a)



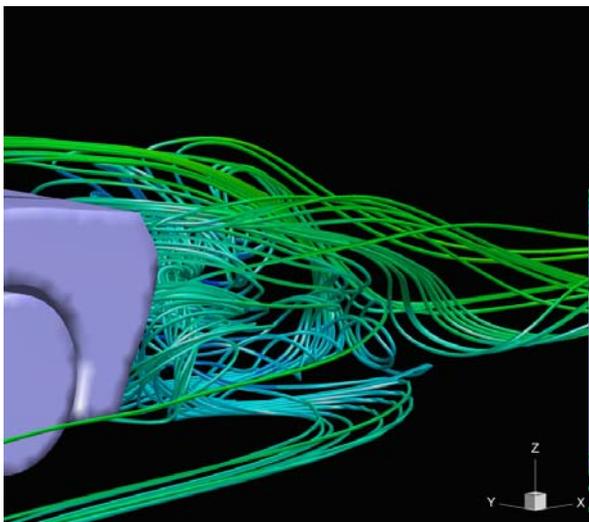
(b)



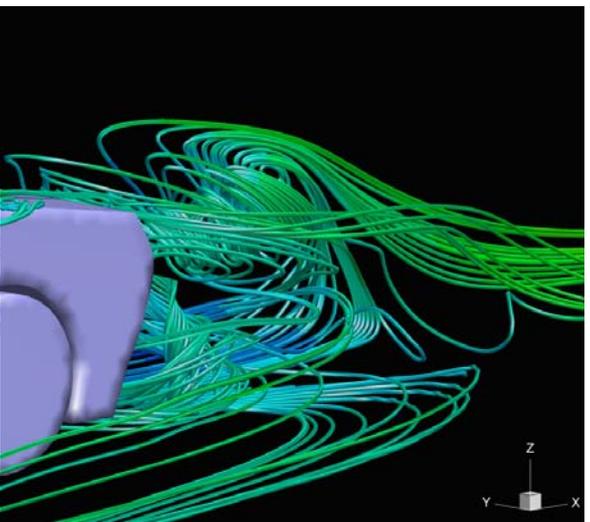
(c)



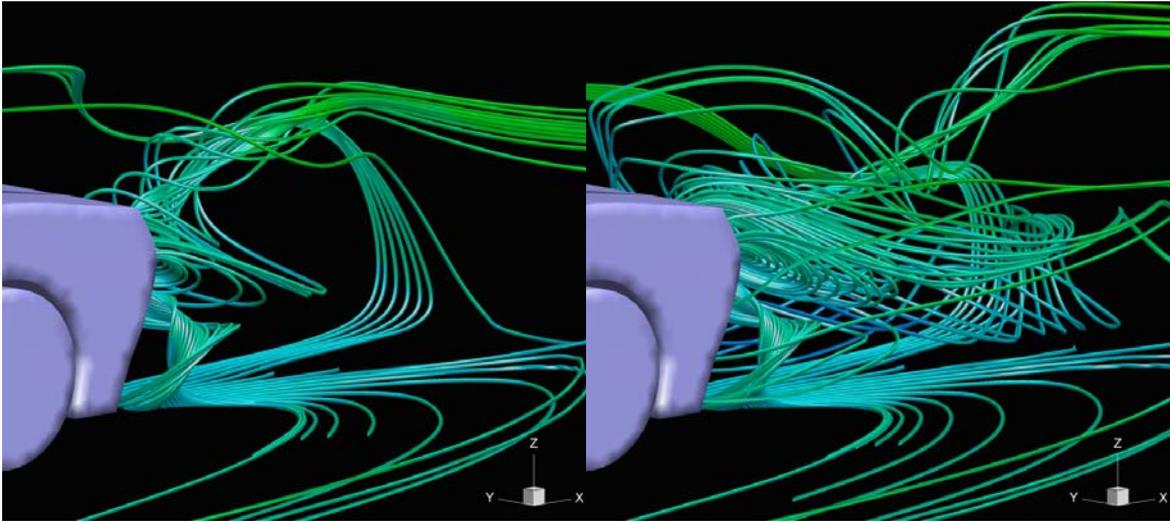
(d)



(e)

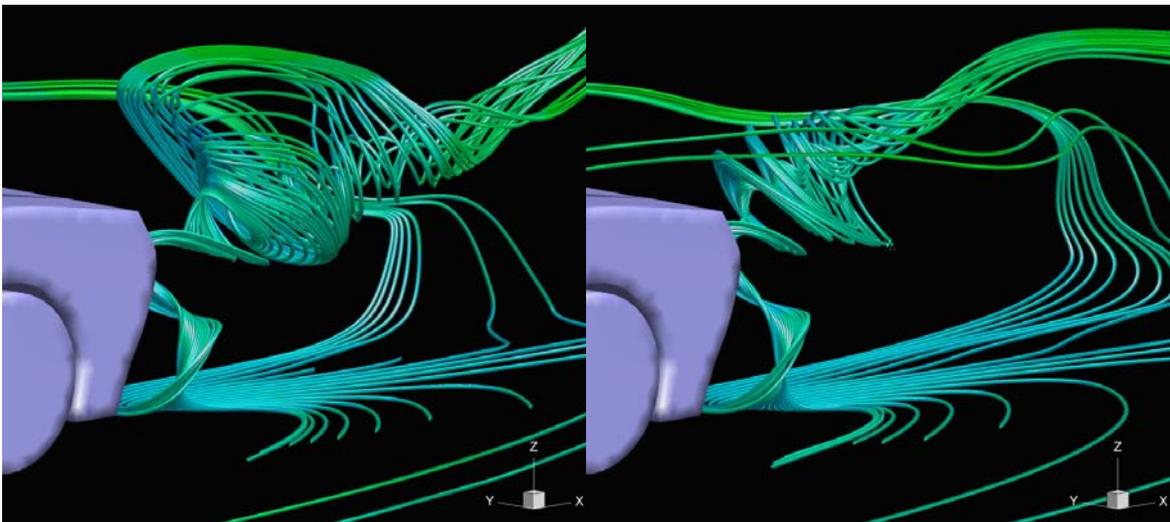


(f)



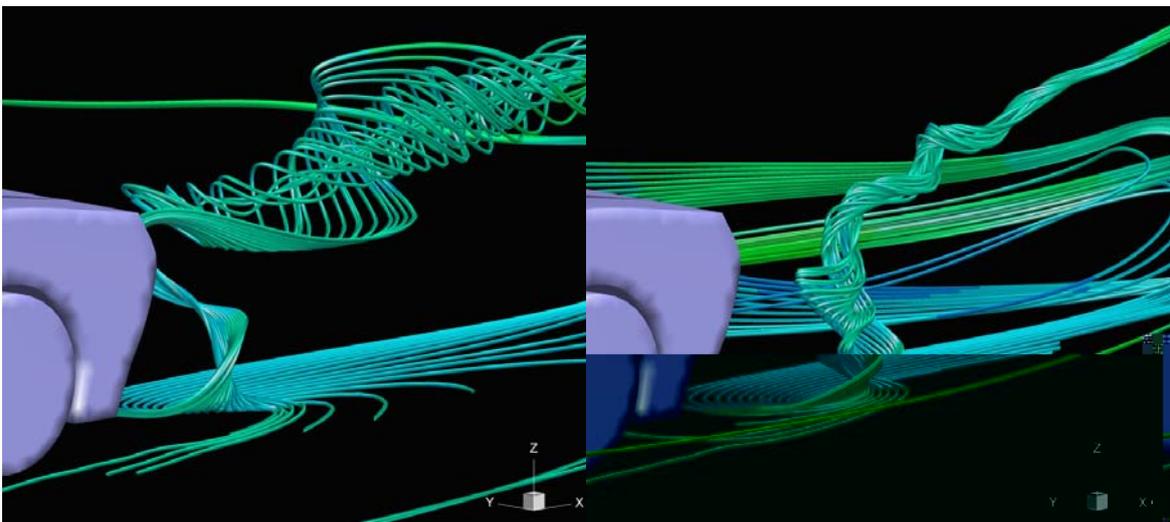
(g)

(h)



(i)

(j)



(k)

(l)

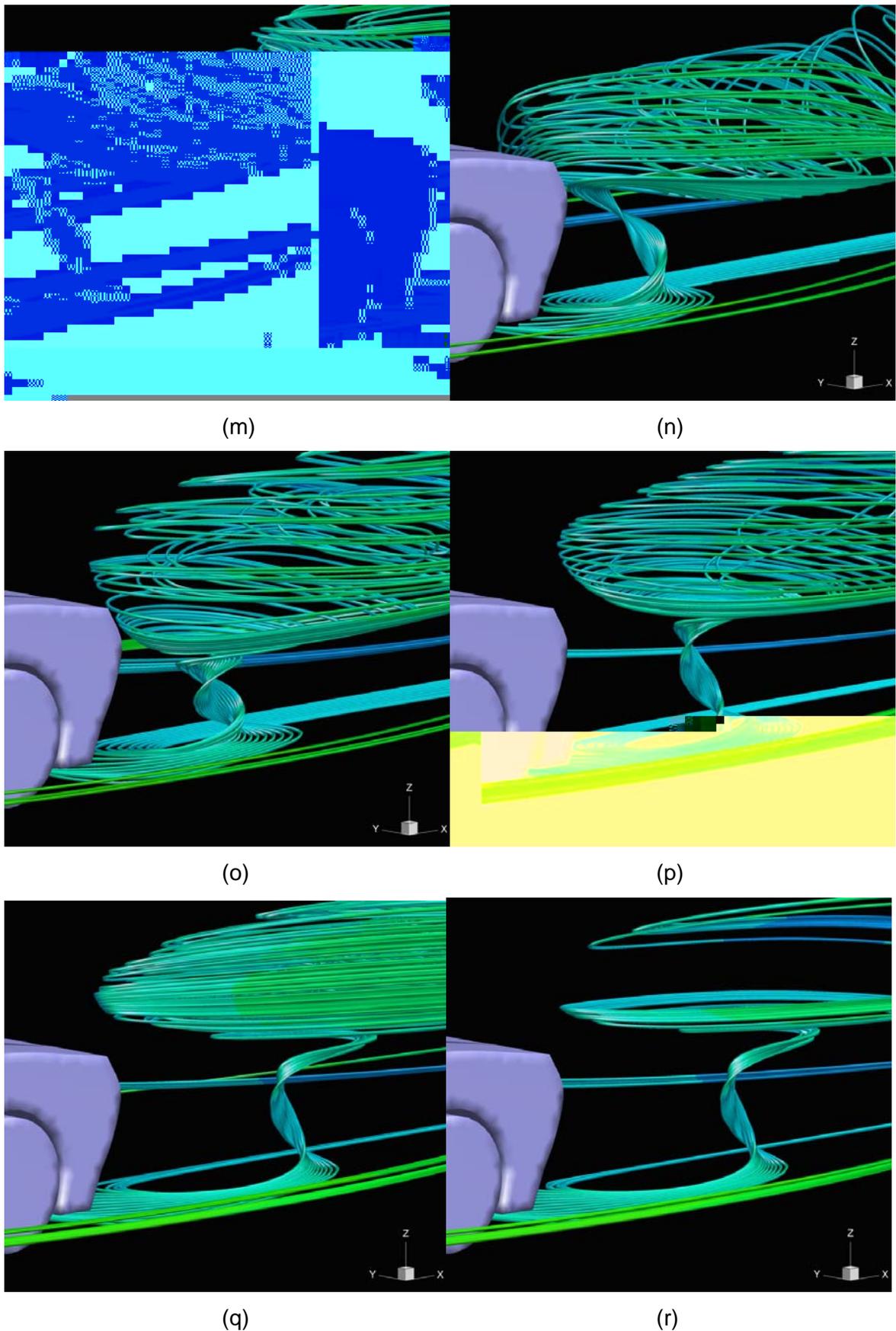


Figura 7.51 – Evolução de uma estrutura turbilhonar a jusante do protótipo de automóvel.

Os estágios de evolução das linhas de corrente da Figura 7.51, são para os seguintes tempos físicos (de cima para baixo): (a) $t=0,02s$, (b) $t=0,075s$, (c) $t=0,1s$, (d) $t=0,2s$, (e) $t=0,3s$, (f) $t=0,6s$, (g) $t=0,7s$, (h) $t=0,8s$, (i) $t=1,0s$, (j) $t=1,25s$, (k) $t=1,5s$, (l) $t=1,75s$, (m) $t=2,0s$, (n) $t=2,75s$, (o) $t=3,25s$, (p) $t=3,75s$, (q) $t=4,5s$, (r) $t=4,75s$.

7.2.5 Escoamento sobre um protótipo de aeronave

Nesta seção serão apresentados resultados qualitativos para o escoamento sobre um protótipo de aeronave. O protótipo teve suas características gerais baseadas no jato executivo Legacy, de fabricação da Embraer, assim como o protótipo da seção anterior. Mais detalhes sobre a geração do protótipo da aeronave pode ser encontrada em Oliveira (2007). A malha de elementos triangulares que representam o protótipo pode ser visualizada na Figura 7.52.

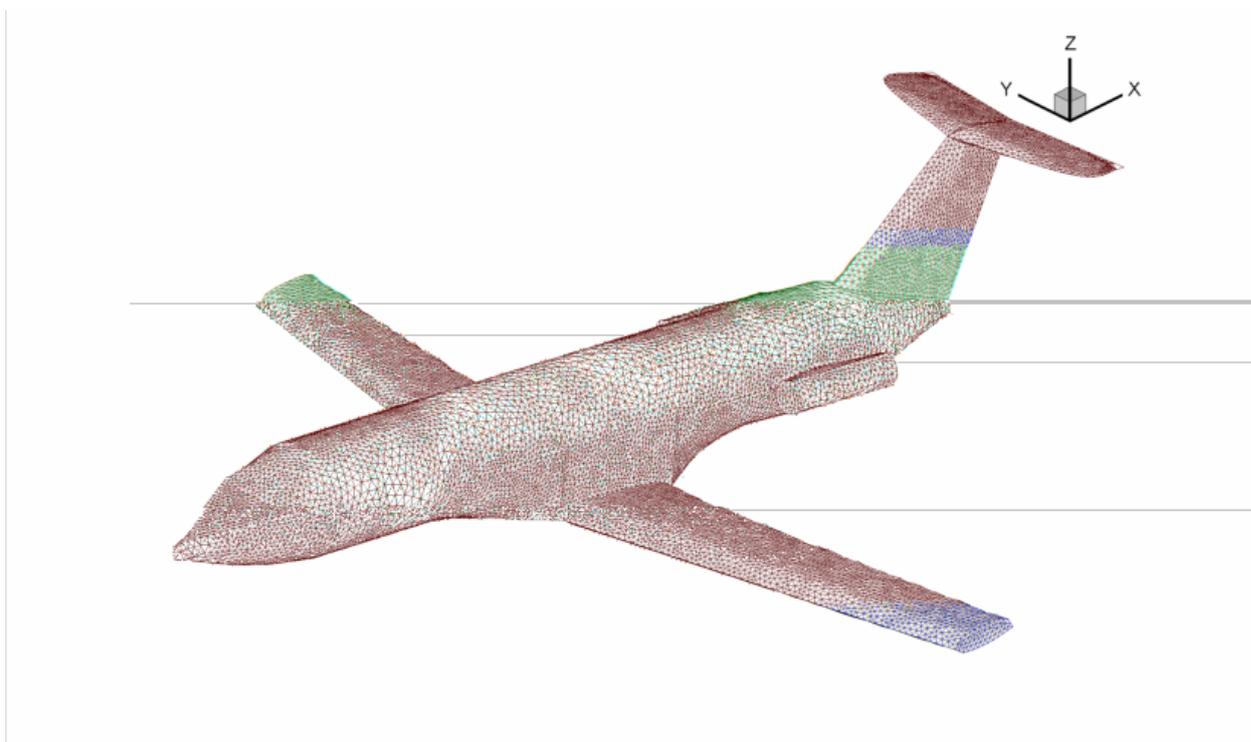


Figura 7.52 - Malha de elementos triangulares do protótipo de aeronave (OLIVEIRA, 2007).

A malha mostrada na Figura 7.52, possui 21.854 nós e 43.702 elementos. O domínio euleriano possui dimensões tais que na direção x, a montante tem-se 1,5 vezes o comprimento

da aeronave, e a jusante, 5 vezes o comprimento. Na direção y , em ambas as laterais, tem-se 1,5 vezes a largura do protótipo e na direção z , tem-se uma altura 2,5 vezes a altura do protótipo. A malha euleriana é composta por $227 \times 150 \times 74$ volumes, totalizando aproximadamente 2.420.000 volumes, subdivididos em quatro processadores. Como condições de contorno do domínio euleriano têm-se superfícies livres em $z=0$, $z=z_{\max}$, $y=0$ e $y=y_{\max}$. O perfil de velocidades na entrada é uniforme com $u=U_{\infty}$, $v=w=0$. O número de Reynolds foi de 500, baseado na envergadura (distância entre as pontas das asas) do objeto imerso.

Uma vez que um dos objetivos deste trabalho é verificar a capacidade de importação de geometrias complexas, não houve ainda uma preocupação de se simular o protótipo da aeronave à altíssimos números de Reynolds, ou realizar a simulação em domínios com razões de aspecto muito grandes. Desta forma os resultados mostrados nesta seção são apenas qualitativos. Como já comentado anteriormente, é necessário que do tamanho característico de cada malha lagrangiana seja aproximadamente igual ao tamanho da malha euleriana na região, isto porque se a disparidade entre os tamanhos característicos das malhas lagrangiana e euleriana forem muito altas, podem ocorrer problemas como perfuração de linhas de corrente (indicando que a densidade de força gerada pela fronteira imersa não foi o suficiente para “afastar” o fluido da região da fronteira). Pode haver um aumento no coeficiente de arrasto, por exemplo, devido ao fato de a fronteira imersa ter gerado uma força maior que a força fisicamente correta.

A Figura 7.53 mostra algumas linhas de corrente posicionadas sobre o protótipo da aeronave. Nota-se que apesar do domínio escolhido não ser totalmente adequado (o tamanho característico da malha euleriana na região da fronteira imersa é duas vezes maior que o tamanho característico da malha lagrangiana), o código convergiu e mostrou resultados fisicamente consistentes.

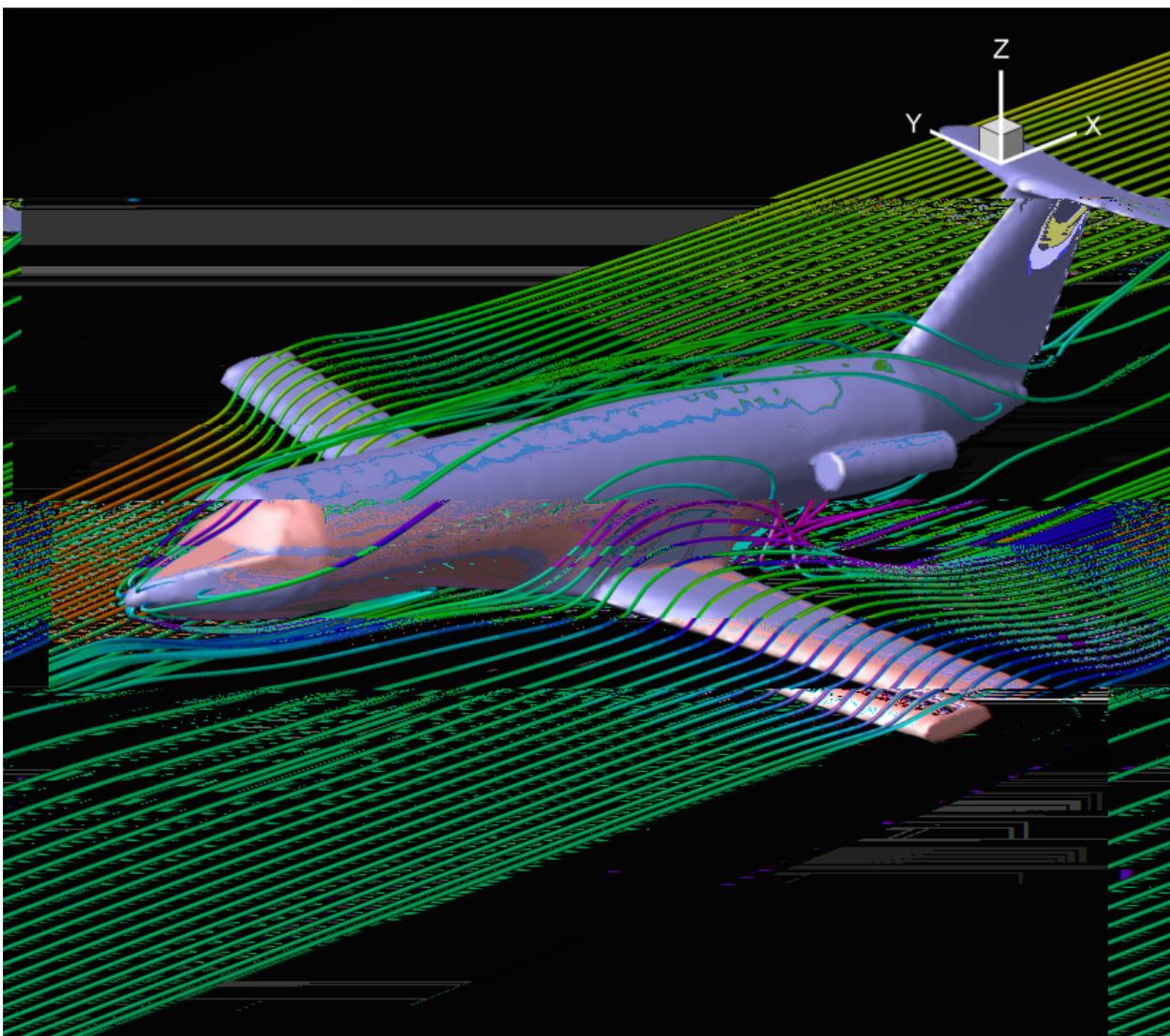


Figura 7.53 - Linhas de corrente indicando padrão de escoamento sobre o protótipo de aeronave, para $t = 0,6$ s.

A Figura 7.54 mostra uma visão lateral do escoamento mostrado na Figura 7.53.

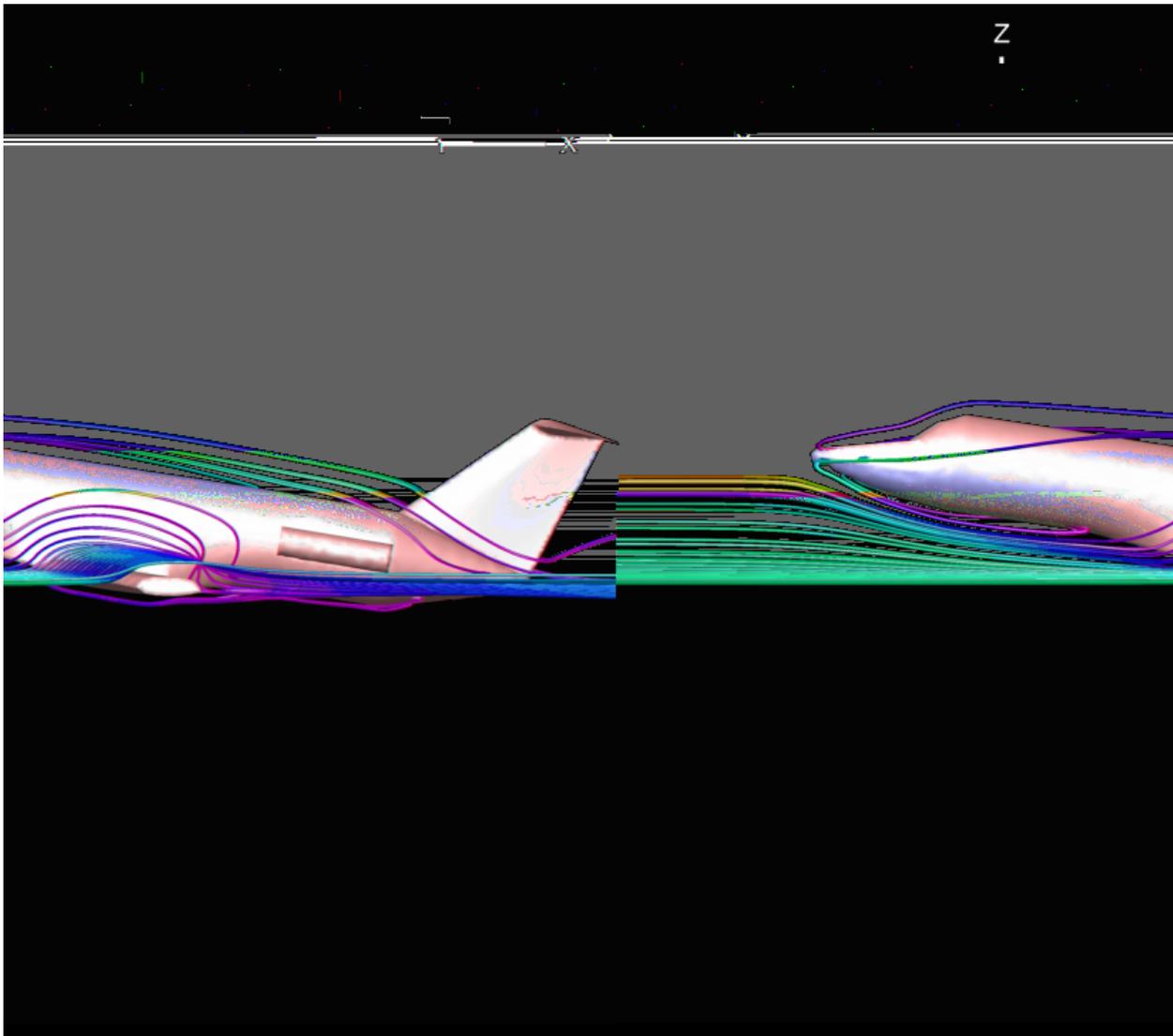


Figura 7.54 - Vista lateral da Figura 7.53.

CAPÍTULO VIII

8 Conclusões e Perspectivas

No presente trabalho, foram apresentadas as mudanças realizadas no código Fluids-3D, desenvolvido no LTCM, de forma a torná-lo uma ferramenta numérica capaz de, sem grandes dificuldades, simular o escoamento sobre qualquer tipo de geometria complexa, em números de Reynolds baixos a moderados. O método da fronteira imersa, baseado no Modelo físico virtual, possibilitaram resultados consistentes. Mostraram-se robustos e confiáveis o bastante nas diversas simulações, nos diversos tipos de geometrias simuladas. Além disto a facilidade da geração de malhas (tanto as malhas eulerianas quanto lagrangianas) foi significativa. O *set-up* de uma simulação, considerando uma geometria como a do protótipo de automóvel, ou do protótipo de aeronave, seria computacionalmente bem mais caro e complicado se fossem utilizadas malhas não-estruturadas. Mesmo a implementação do *solver* seria muito mais complicada.

Com o intuito de facilitar o *set-up* de novas simulações, várias mudanças foram feitas no código fonte do programa Fluids-3D. Estas mudanças foram feitas de tal forma que não fossem necessárias re-compilações para cada nova simulação. Isto é particularmente importante, pois desta maneira, o código está apto a receber uma interface gráfica de usuário, além de minimizar a probabilidade de inserção de erros no código – fonte do programa. Uma outra realização importante foi a implementação de uma sub-rotina de leitura e importação de arquivos **.stl*. A importação deste tipo de arquivo significa uma liberdade maior, por parte do usuário, de criação de novas geometrias a serem simuladas, isto porque, estão disponíveis na Internet diversos softwares livres para criação e modificação deste tipo de arquivo (Oliveira, 2007).

Nas várias simulações realizadas, detalhes físicos dos escoamentos puderam ser vistos, mostrando que o código Fluids-3D, pode ser considerado uma ferramenta confiável de simulação de escoamentos sobre geometrias e com um custo computacional competitivo se comparado com códigos comerciais.

Como perspectivas para novos trabalhos, pode-se citar:

- Continuar o trabalho de implementação da modelagem da turbulência aplicada a escoamentos ao redor de geometrias imersas, a Reynolds elevados. Três modelos de

turbulência já estão implementados no código Fluids-3D, mas carecem de testes de validação;

- Estender a metodologia de paralelização também ao domínio lagrangiano, de forma a se ganhar maior autonomia no uso do *cluster*;
- Avaliar a possibilidade de implementar malhas adaptativas e particionamento do domínio de forma a otimizar o balanceamento de carga no *cluster*;
- Avaliar a possibilidade e necessidade do uso da topologia tridimensional de paralelização em simulações de casos que exijam uma razão de aspecto alta;
- Promover mudanças no código, de forma a este possuir a capacidade de simulação de escoamentos com densidade variável - escoamentos compressíveis.

CAPÍTULO IX

9 Referências Bibliográficas

Arruda, J., 2004, “Modelagem Matemática de escoamentos Internos Forçados utilizando o Método da fronteira imersa e o Modelo físico virtual”, Tese de Doutorado, Universidade Federal de Uberlândia.

Arruda, J., Lima e Silva, A., Roma, A. e Silveira-Neto., A., 2004, “Simulação Numérica de escoamentos sobre Cavidades Abertas Rasas utilizando o Método da fronteira imersa”, In Proceedings of the 12th Brazilian Congress of Thermal Sciences and Engineering – ENCIT.

Campregher, R., 2005, “Modelagem Matemática Tridimensional Para Problemas de Iteração Fluido-Estrutura”. Tese de Doutorado. Universidade Federal de Uberlândia.

Craft, T.J.; Gerasimov, A.V.; Launder, B.E.; Robinson C.M.E., 2006; “A computational study of the near-field generation and decay of wingtip vortices”. International Journal of Heat and Fluid Flow.

Ferziger, J. e Peric, M, 2002, Computational Methods for Fluid Dynamics 3rd Ed. Springer Verlag, New York, USA.

Fortuna, A.O., 2000, “Técnicas computacionais para dinâmica dos fluidos”, Edusp.

Flynn, M.J., 1966, “Very high-speed computing systems”, Proc. IEEE 54 (12), 1901-1909.

Glowinski, R., Pan, T. e Périaux, J., 1998, « Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies”, Comput. Methods Appl. Mech. Engrg. 151, 181-194.

Griffith, B. e Peskin, C., 2005, “On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems.”, Journal of Computational Physics 208, 75-105.

Gropp, W., Lusk, E., Thakur, R., 1999, "Using MPI-2 - Advanced Features of the Message-Passing Interface", Massachusetts Institute of Technology.

Gustafson, J., 1988, "Reevaluating Amdahl's Law", Communications of ACM, May, 1988, Volume 31, No 5.

Hwang, J., Yang, K., 2004, "Numerical study of vertical structures around a wall-mounted cubic obstacle in channel flow", Physics of Fluids, Vol. 16, No. 7, 2382-2394.

Jeong, J. e Hussain, F., 1995, "On the identification of a vortex", Journal of Fluid Mechanics 285,69-94.

Kim, J., Kim, D. e Choi, H., 2001, "An immersed-boundary Finite-Volume method for simulations of flow in complex geometries", Journal of Computational Physics 171, 132-150.

Lai, M., 1998, "Simulations of the flow past an array of circular cylinders as a test of the Immersed Boundary Method", PhD Dissertation, New York University.

Lima e Silva, A. L. F., 2002, "Desenvolvimento e implementação de uma nova metodologia para modelagem de escoamentos sobre geometrias complexas: método da fronteira imersa como Modelo físico virtual", Tese de Doutorado, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.

Lima e Silva, A. L. F., Silveira-Neto, A. e Damasceno, J., 2003, "Numerical simulation of twodimensional flows over a circular cylinder using the immersed boundary method", Journal of Computational Physics 189, 351-370.

Lima e Silva, A. L. F., Silva, A. R. e Silveira-Neto, A., 2005, "Numerical simulation of twodimensional complex flows over bluff bodies using the immersed boundary method", Aceito para publicação no Journal of Fluid and Structures.

Mariano, F. P., 2007, "Simulação de escoamentos não periódicos utilizando a metodologia pseudo-espectral de Fourier acoplado com o método da fronteira imersa", Dissertação em redação, Universidade Federal de Uberlândia, Uberlândia.

Mariano, F. P., Moreira, L. Q. e Silveira Neto, A., 2006, "Resolução de escoamentos não periódicos utilizando métodos pseudo-espectrais e o método da fronteira imersa", 16° Posmec.

Marinho, W., Campregher, R., e Silveira-Neto, A., 2004, "Three-dimensional heat transfer simulation as a benchmark for an in-house Beowulf-class cluster", In Proceedings of the 10th Brazilian Congress of Thermal Sciences and Engineering – ENCIT.

Mohd-Yusof, J., 1997, "Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries", In CTR Annual Research Briefs, NASA Ames/Stanford University.

Muzaferija, S., Peric, M., 1997, "Computational of free-surface flows using the Finite-Volume method and moving grids", Numerical Heat Transfer, Part B 32, 369-384.

Padilla, E. L.M., 2007, "Aplicações de simulação numérica para análise de escoamentos transicionais em canais cilíndrico-anulares com excentricidade variável", Relatório técnico de convênio FEMEC – PETROBRAS. (Projeto No: 005000007895.04.3)

Patankar, S., 1980, "Numerical Heat Transfer and Fluid Flow", Hemisphere Publishing Corporation, New York, USA.

Peskin, C., 1972, "Flow patterns around heart valves: A numerical method", Journal of Computational Physics 10, 252-271

Peskin, C., 1977, "Numerical analysis of blood flow in the heart", Journal of Computational Physics 25, 220-252.

Pitanga, M, 2002, "Construindo supercomputadores com Linux", Brasport.

Oliveira, J. E. S., 2006, "Método de Fronteira Imersa Aplicado à Modelagem Matemática e Simulação Numérica de Escoamentos Turbulentos sobre Geometrias Móveis e Deformáveis", Tese de Doutorado, Universidade Federal de Uberlândia.

Oliveira, G. P., 2007, "Desenvolvimento de interface gráfica para o solver de dinâmica dos fluids computacional Fluids-3D", Dissertação em redação, Universidade Federal de Uberlândia, Uberlândia.

Oliveira, J., Lima e Silva, A., Guimarães, G. e Silveira-Neto, A., 2004, "Simulação numérica do escoamento a baixo Reynolds sobre o cilindro de diâmetro variável usando MFI/MFV", In Proceedings of the 10th Brazilian Congress of Thermal Sciences and Engineering.

Oliveira, J., Lima e Silva, A., Souza, F., Guimarães, G. e Silveira-Neto, A., 2004, "Comparative analysis between different methodologies of turbulence modeling: URANS, DES e LES", In 4º Escola de Primavera de Transição e Turbulência.

Rhie, C e Chow, W., 1983, "Numerical study of the turbulent flow past an airfoil with trailing edge separation", AIAA Journal, 21(11), 1525-1532.

Roma, A., Peskin, C. e Berger, M., 1999, "An adaptive version of the immersed boundary method", Journal of Computational Physics 153, 509-534.

Saha, A. K., 2004, "Three-dimensional numerical simulation of the transition of flow past a cube", Physics of Fluids, Vol. 16, No. 5, 1630-1646.

Silva, A. R., 2004, "Simulação Numérica de Escoamentos em Transição sobre Cilindros Imersos", Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG.

Van Doormal, J. e Raithby, G., 1984, "Enhancements of the simple method for predicting incompressible fluid flows", Numerical Heat Transfer 7, 147-163.

Vedovoto, J.M., Campregher, R, Silveira Neto, A., 2006, "Mathematical modeling and numerical simulation of a three-dimensional flow over complex geometries using the immersed boundary method", In proceedings of 11th Brazilian Congress of Thermal Sciences and Engineering – ENCIT.

Versteeg, H. e Malalasekera, W., 1995, "An introduction to computational fluid dynamics", Prentice Hall.

Vilaça, A., Oliveira, O., Lima e Silva, A. e Silveira-Neto, A., 2004, "Modelagem matemática e simulação numérica do escoamento sobre uma partícula em queda livre", Submetido ao XXXI ENEMP – Congresso Brasileiro de Sistemas Particulados.

Unverdi, S. e Tryggvason, G., 1992, "A front-tracking method for viscous, incompressible, multifluid flows", *Journal of Computational Physics* 100, 25-37.

Winkelmann, R., Hauser, J., Williams, R., 1999, "Strategies for parallel and numerical scalability of CFD codes", *Computational. Methods Applied. Mechanical. Engineering*, 174, 433 456.

Yu, Z., 2005, "A DLM/FD method for fluid/flexible-body interactions", *Journal of Computational Physics* 207, 1-27.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)