



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

FERNANDO MOLON ABREU

**SIMULAÇÕES NUMÉRICAS DE PROBLEMAS
DESCRITOS PELAS EQUAÇÕES DE
NAVIER-STOKES INCOMPRESSÍVEIS VIA
BIBLIOTECA FEniCS.**

VITÓRIA
2017

FERNANDO MOLON ABREU

**SIMULAÇÕES NUMÉRICAS DE PROBLEMAS
DESCRITOS PELAS EQUAÇÕES DE
NAVIER-STOKES INCOMPRESSÍVEIS VIA
BIBLIOTECA FEniCS.**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Mecânica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Mecânica.

Orientadora:

Prof^ª. Dr^a. LUCIA CATABRIGA

VITÓRIA

2017

FERNANDO MOLON ABREU

SIMULAÇÕES NUMÉRICAS DE PROBLEMAS DESCRITOS
PELAS EQUAÇÕES DE NAVIER-STOKES INCOMPRESSÍVEIS
VIA BIBLIOTECA FEniCS./ FERNANDO MOLON ABREU. –
VITÓRIA, 2017 -

92 p.

Orientadora: Prof^a. Dr^a. LUCIA CATABRIGA

Dissertação de Mestrado – UNIVERSIDADE FEDERAL DO ESPÍ-
RITO SANTO. CENTRO TECNOLÓGICO. PROGRAMA DE PÓS-
GRADUAÇÃO EM ENGENHARIA MECÂNICA. 2017.

1. Método dos Elementos Finitos. 2. Formulações Estabilizadas.
3. Métodos de Projeção. 4. Interação Fluido-Estrutura. 5. FEniCS.
I. CATABRIGA, L. II. LOEFFLER, C. F. III. SANTOS, I. P. IV.
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO. CENTRO
TECNOLÓGICO. V. SIMULAÇÕES NUMÉRICAS DE PROBLE-
MAS DESCRITOS PELAS EQUAÇÕES DE NAVIER-STOKES
INCOMPRESSÍVEIS VIA BIBLIOTECA FEniCS.

FERNANDO MOLON ABREU

**SIMULAÇÕES NUMÉRICAS DE PROBLEMAS
DESCRITOS PELAS EQUAÇÕES DE NAVIER-STOKES
INCOMPRESSÍVEIS VIA BIBLIOTECA FEniCS.**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Mecânica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Mecânica.

Trabalho aprovado. VITÓRIA, 19 de Dezembro de 2017:

Prof^a. Dr^a. Lucia Catabriga
Universidade Federal do Espírito Santo

**Prof. Dr. Carlos Friedrich Loeffler
Neto**
Universidade Federal do Espírito Santo

Prof. Dr. Isaac Pinheiro dos Santos
Universidade Federal do Espírito Santo

VITÓRIA
2017

Agradecimentos

Agradeço a Deus por ter me dado força para realizar este trabalho.

Aos meus pais, Alceio e Anésia, pelo apoio incondicional em todos os momentos, e aos meus irmãos, Eliana e Julio, pelo apoio e incentivo. A minha namorada, Mizia, que de forma especial me deu força e coragem, me apoiando em grandes dificuldades.

A minha orientadora, professora Lucia Catabriga, pela orientação e compreensão das dificuldades encontradas.

A secretária do PPGEM, Andréia, sempre disposta a auxiliar os alunos e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela bolsa de estudos e ao PPGEM pelos auxílios.

A todas as pessoas que direta ou indiretamente contribuíram para este trabalho.

*"Importante não é ver o que ninguém nunca viu, mas sim, pensar o que ninguém nunca
pensou sobre algo que todo mundo vê".
(Arthur Schopenhauer)*

Resumo

Este trabalho tem por objetivo estudar as equações de Navier-Stokes solucionadas pelo método dos elementos finitos utilizando a biblioteca de domínio público FEniCS. Um estudo comparativo de métodos de projeção e estabilizados é apresentado. Os métodos de projeções implementados foram: esquema de correção não incremental da pressão (Chorin) (*Non-Incremental Pressure Correction Scheme - Chorin*), esquema de correção incremental da pressão (IPCS) (*Incremental Pressure Correction Scheme - IPCS*) e esquema de divisão consistente (CSS) (*Consistent Splitting Scheme - CSS*). O esquema adaptativo de escolha do passo de tempo, PID, é considerado nos métodos de projeção e estabilizados. Também, são investigados problemas de interação fluido-estrutura através do método de impulso acoplado (*CMM-FSI - Coupled Momentum Method for FSI*). O método CMM-FSI considera a formulação convencional de elementos finitos para as equações de Navier-Stokes em um domínio rígido, acrescentando modificações de tal maneira que a deformação da membrana seja considerada. Para a formulação variacional das equações de Navier-Stokes é considerada a formulação estabilizada *Streamline-Upwind/Petrov-Galerkin* (SUPG) para a velocidade e a formulação estabilizada *Pressure-Stabilizing/Petrov-Galerkin* (PSPG) para a pressão enriquecida com a estabilização *Least-Squares on Incompressibility Constraint* (LSIC). A biblioteca FEniCS demonstrou eficiência para tratamento das equações de Navier-Stokes e na solução de problemas de interação fluido-estrutura pelo método CMM-FSI.

Palavras-chave: Método dos Elementos Finitos. Formulações Estabilizadas. Métodos de Projeção. Interação Fluido-Estrutura. FEniCS.

Abstract

This work aims to study the Navier-Stokes equations solved by the finite element method using the FEniCS public domain library. A comparative study of projection and stabilized methods is presented. The projection methods implemented were: *non-incremental pressure correction scheme* (Chorin), *incremental pressure correction scheme* (IPCS) and *consistent splitting scheme* (CSS). The adaptive choice of time step scheme, PID, is considered in the projection and stabilized methods. In addition, fluid-structure interaction problems are investigated through the coupled momentum method for FSI (CMM-FSI). The CMM-FSI method considers the conventional formulation of finite elements for the Navier-Stokes equations in a rigid domain, adding modifications in such a way that the membrane deformation is considered. For the variable formulation of the Navier-Stokes equations a stabilized *Streamline-Upwind/Petrov-Galerkin* (SUPG) formulation is considered for the velocity and stabilized *Pressure-Stabilizing/Petrov-Galerkin* (PSPG) formulation for the pressure enriched with *Least-Squares on Incompressibility Constraint* (LSIC) stabilization. The FEniCS library demonstrated efficiency in the treatment of the Navier-Stokes equations in general and in solution of fluid-structure interaction problems by the CMM-FSI method.

Keywords: Finite Element Method. Stabilized Formulations. Projection Methods. Fluid-Structure Interaction. FEniCS.

Lista de ilustrações

Figura 1 – Seleção do passo de tempo como um controlador	25
Figura 2 – Representação dos domínios, adaptado de (FIGUEROA et al., 2006)	26
Figura 3 – Esforço na membrana com carga transversal, retirado de (FIGUEROA et al., 2006)	31
Figura 4 – Alguns dos principais componentes e classes da DOLFIN, adaptado de (LOGG; MARDAL; WELLS, 2012)	34
Figura 5 – Linhas de corrente para o problema da cavidade bidimensional utilizando o método Chorin	54
Figura 6 – Resultados para o problema da cavidade bidimensional	54
Figura 7 – Velocidade e pressão para o problema do canal quadrado utilizando o método IPCS	55
Figura 8 – Resultados para o problema do canal retangular	55
Figura 9 – Vórtices ao passar pelo obstáculo no tempo $t=4$ segundos	57
Figura 10 – Resultados para o problema do canal com obstáculo	57
Figura 11 – Vetores velocidade e pressão para o problema do vórtice de Taylor-Green	58
Figura 12 – Resultados para o problema do vórtice de Taylor-Green	58
Figura 13 – Vetores velocidade para o problema do fluxo de Beltrami	59
Figura 14 – Resultados para o problema do do fluxo de Beltrami	60
Figura 15 – Velocidade para o problema do aneurisma idealizado	61
Figura 16 – Resultados para o problema do aneurisma idealizado	61
Figura 17 – Desempenho dos métodos	62
Figura 18 – Velocidade na direção x no centro da cavidade para número de Reynolds 100	64
Figura 19 – Velocidade na direção y no centro da cavidade para número de Reynolds 100	65
Figura 20 – Velocidade na direção x no centro da cavidade para número de Reynolds 400	66
Figura 21 – Velocidade na direção y no centro da cavidade para número de Reynolds 400	67
Figura 22 – Velocidade na direção x no centro da cavidade para número de Reynolds 1000	68
Figura 23 – Velocidade na direção y no centro da cavidade para número de Reynolds 1000	69
Figura 24 – Solução para o problema do escoamento em uma cavidade com tampa deslizante para número de Reynolds 100 utilizando o método IPCS	70

Figura 25 – Solução para o problema do escoamento em uma cavidade com tampa deslizante para para número de Reynolds 400 utilizando o método IPCS	71
Figura 26 – Solução para o problema do escoamento em uma cavidade com tampa deslizante para número de Reynolds 1000 utilizando o método IPCS . .	72
Figura 27 – Descrição do modelo e da malha para o problema do escoamento com alargamento de canal	72
Figura 28 – Linhas de corrente para o problema do escoamento com alargamento de canal utilizando o método IPCS	73
Figura 29 – Comprimentos de referências para o problema do escoamento com alargamento de canal	73
Figura 30 – Solução para o problema do escoamento com alargamento de canal para número de Reynolds 1 utilizando o método IPCS	74
Figura 31 – Solução para o problema do escoamento com alargamento de canal para número de Reynolds 100 utilizando o método IPCS	75
Figura 32 – Solução para o problema do escoamento com alargamento de canal para número de Reynolds 250 utilizando o método IPCS	76
Figura 33 – Malha para o problema de parede deformável	76
Figura 34 – Velocidade e deformação dos nós nas paredes (deslocamento zero) . . .	77
Figura 35 – Velocidade e deformação dos nós nas paredes (pressão hidrostática) . .	78
Figura 36 – Dimensões e malha para o problema da artéria idealizada	79
Figura 37 – Perfis de velocidade e pressão na artéria idealizada	79
Figura 38 – Vetores deslocamento para o problema da artéria idealizada	80
Figura 39 – Parâmetros geométricos e malha de elementos finito para artéria aneurismática	81
Figura 40 – Campo de velocidade e pressão para o problema da artéria aneurismática	82
Figura 41 – Vetores velocidade e deslocamento para o problema da artéria aneurismática	83

Lista de tabelas

Tabela 1 – Experimentos testes e parâmetros de interesse	52
Tabela 2 – Níveis de refinamento para a cavidade bidimensional	53
Tabela 3 – Níveis de refinamento para o canal retangular com obstáculo	56
Tabela 4 – Níveis de refinamento para o fluxo de Beltrami	59
Tabela 5 – Níveis de refinamento para o problema do aneurisma idealizado	60
Tabela 6 – Descrição da malha para o problema do escoamento em uma cavidade bidimensional com tampa deslizante	63
Tabela 7 – Tempo de CPU em segundos para número de Reynolds 100	64
Tabela 8 – Tamanho médio do passo de tempo para número de Reynolds 100	64
Tabela 9 – Tempo de CPU em segundos para número de Reynolds 400	65
Tabela 10 – Tamanho médio do passo de tempo para número de Reynolds 400	66
Tabela 11 – Tempo de CPU em segundos para número de Reynolds 1000	67
Tabela 12 – Tamanho médio do passo de tempo para número de Reynolds 1000	67
Tabela 13 – Comprimento característico para números de Reynolds 1, 100 e 250	70
Tabela 14 – Tempo de CPU em segundos para o escoamento com alargamento de canal	73
Tabela 15 – Tamanho médio do passo de tempo para o escoamento com alargamento de canal	74
Tabela 16 – Propriedades e parâmetros para parede deformável	75
Tabela 17 – Posição dos nós	76
Tabela 18 – Propriedades e parâmetros para o problema da artéria idealizada	79

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	13
1.2	Objetivos do Trabalho	16
1.3	Organização da Dissertação	17
2	EQUAÇÕES DE NAVIER-STOKES	18
2.1	Equações Governantes	18
2.2	Formulação Variacional	19
2.2.1	Formulação estabilizada	20
2.2.2	Métodos de projeções	21
2.3	Técnicas Adaptativas de Avanço no Tempo	25
3	INTERAÇÃO FLUIDO-ESTRUTURA	26
3.1	Equações Governantes	26
3.2	Formulação Variacional	27
3.3	Acoplamento fluido-sólido	29
4	FENICS	34
4.1	Implementação das Equações de Navier-Stokes	35
4.2	Implementação do Problema IFE	44
5	EXPERIMENTOS NUMÉRICOS	51
5.1	Experimentos das equações de Navier-Stokes	51
5.1.1	Comparação de Métodos	52
5.1.2	Escoamento em uma Cavidade Bidimensional com Tampa Deslizante	62
5.1.3	Escoamento com Alargamento de Canal	68
5.2	Experimentos de Interação Fluido-Estrutura	73
5.2.1	Parede Deformável	75
5.2.2	Artéria Idealizada	78
5.2.3	Artéria Aneurismática	81
6	CONCLUSÕES	84
	REFERÊNCIAS	86

1 INTRODUÇÃO

1.1 Motivação

As equações de Navier-Stokes (ENS), denominadas em homenagem aos estudos do engenheiro e físico Claude Louis Marie Henri Navier e do matemático e físico George Gabriel Stokes, descrevem o movimento de substâncias fluidas viscosas. As equações se baseiam na conservação da quantidade de momento linear, proveniente de forças viscosas e variações da pressão, dentre outras grandezas físicas, além da conservação da massa em um volume infinitesimal.

Tais equações são de suma importância em várias áreas do conhecimento. Podem ser utilizadas para modelar mudanças climáticas, correntes oceânicas, escoamento em dutos, projetos de aerofólios e aeronaves, estudos de fluxo sanguíneo, entre outras aplicações (ELIAS, 2007). Na grande maioria das aplicações, o fluido exerce interações com outro meio e vice-versa, por exemplo, projetos de aerofólios, aeronaves e fluxo sanguíneo, no qual o fluido interage com um sólido recaindo nos chamados problemas de interação fluido-estrutura. Quando a influência do sólido e a resposta do fluido tornam-se relevantes, os efeitos desta interação devem ser levados em consideração (BAZILEVS; TAKIZAWA; TEZDUYAR, 2013).

A interação fluido-estrutura (IFE) constitui o acoplamento entre diferentes leis da física, em especial as da fluidodinâmica e mecânica estrutural. Os fenômenos em estudo possuem dependência mútua entre as áreas dinâmica dos sólidos e mecânica dos fluidos. O comportamento do fluxo depende da estrutura e o movimento e a deformação do sólido dependem das forças que agem sobre a estrutura.

Os problemas IFE estão presentes em várias áreas das engenharias, tais como: civil, mecânica, aeronáutica (STEIN et al., 2000; WREN et al., 1997; BAZILEVS; TAKIZAWA; TEZDUYAR, 2013; GOMES, 2013), em ramos das ciências, por exemplo em simulações geológicas (KUMAGAI, 2002), e na área da medicina, como em problemas de biomecânica (KAMENSKY et al., 2015; VIGNON-CLEMENTEL et al., 2006; FIGUEROA et al., 2006; HELMIG, 2013; SELIM; LOGG, 2017), dentre outras áreas de aplicação.

Um problema de interação fluido-estrutura é caracterizado como um conjunto de equações diferenciais parciais e condições de contorno relacionadas a cada um dos domínios (fluido e sólido), em que formam um único conjunto acoplado de equações. A natureza inerentemente não linear e dependente do tempo em problemas IFE tornam difíceis a busca por soluções analíticas. Apesar de existir algumas soluções analíticas para

problemas de fluidos e/ou estruturais, há pouco progresso em solução para os problemas IFE (BAZILEVS; TAKIZAWA; TEZDUYAR, 2013). Todavia, grandes avanços na pesquisa de métodos computacionais vêm sendo desenvolvidas nas últimas décadas, especialmente aplicações em problemas de mecânica dos fluidos e de interação fluido-estrutura (MITTAL; TEZDUYAR, 1995; KUHL; HULSHOFF; BORST, 2003; FIGUEROA et al., 2006; BAZILEVS; TAKIZAWA; TEZDUYAR, 2013; COURT; FOURNIÉ; LOZINSKI, 2014; COURT; FOURNIÉ, 2015). O Método dos Elementos Finitos (MEF) já é um método consolidado, sendo utilizado há décadas na área de análise numérica e tendo um grande destaque na área de mecânica dos fluidos.

Modelos matemáticos têm sido cada vez mais aplicados, devido aos avanços computacionais, em campos da medicina e biologia, de modo especial para aplicações que envolvam artérias tridimensionais com configurações geometricamente complexas. Essas abordagens possibilitam estudar regiões propensas à aterosclerose e aneurismas (PERKTOLD; RESCH; PETER, 1991; TAYLOR; HUGHES; ZARINS, 1998; STEINMAN, 2002), válvulas reguladoras do coração (MAGOSSO; URSINO, 2002; BREBBIA, 2009), e vão além, possibilitando planejamentos cirúrgicos (LEVAL et al., 1996; MIGLIAVACCA et al., 1999; TAYLOR et al., 1999; LAGANÁ et al., 2002). O fluxo sanguíneo é normal em pessoas saudáveis, porém indivíduos com certas patologias estão propensos a fluxos anormais, e a formação e o desenvolvimento dos aneurismas são diretamente influenciados por estes campos de fluxos nas artérias (GLAGOV et al., 1988; MOORE, 1991). Dessa forma, a fisiologia do corpo humano pode ser melhor avaliada, sem risco à vida, e ainda, simulações podem possibilitar melhorias de diagnósticos e tratamentos (STERGIOPULOS; YOUNG; ROGGE, 1992).

Um expressivo número de aplicações de modelos arteriais não levam em conta a deformação do vaso, assumindo o mesmo como rígido (OSHIMA et al., 2001; CEBRAL et al., 2003; SALMON; MARC; GERBEAU, 2003; SHOJIMA et al., 2004), ou as configurações geométricas são altamente simplificadas (PERKTOLD; RAPPITSCH, 1995; VOSSE et al., 2003). Porém essas hipóteses não representam de forma realista as práticas destas aplicações, devido a deformação do vaso estar relacionada com as forças do fluido e vice-versa. Portanto, existe a necessidade de contabilizar a deformação do sólido acoplando as equações provenientes da mecânica dos fluidos e mecânica estrutural, resultando nos problemas de interação fluido-estrutura.

Uma das abordagens mais utilizadas para incluir a deformação da parede do vaso é a formulação Lagrangeana-Euleriana-Arbitrária (ALE) para resolver os problemas de interação fluido-estrutura (TEZDUYAR; BEHR; LIOU, 1992; MITTAL; TEZDUYAR, 1994; SANCHES; CODA, 2009). Entretanto, esta técnica encontra grandes desafios para modelagem de fluxo sanguíneo tridimensional, pois necessitam de atualizar, a cada passo de tempo, a geometria do fluido e da estrutura, resultando em uma dispendiosa abordagem.

Além disso, algumas aplicações fisiológicas necessitam que os modelos sejam resolvidos em um período de tempo clinicamente relevantes (FIGUEROA et al., 2006). Dessa forma, a busca por métodos mais simples para incorporar a deformação da parede são essenciais para essas aplicações.

O método de impulso acoplado para problemas de interação fluido-estrutura (CMM-FSI - *Coupled Momentum Method for FSI*) foi proposto por (FIGUEROA et al., 2006; FIGUEROA et al., 2004), evitando as dificuldades encontradas com a formulação ALE. Este método considera o sólido sendo de parede fina, utilizando um modelo de membrana. Os diâmetros das artérias são pequenos comparados com o comprimento de onda do pulso cardíaco, fazendo com que o sólido apresente um comportamento de membrana ao invés de flexão, não necessitando de graus adicionais de liberdade na parede do sólido.

O método CMM-FSI considera a formulação convencional de elementos finitos para as equações de Navier-Stokes em um domínio rígido, acrescentando modificações de tal maneira que a deformação da membrana seja considerada. Uma abordagem de cinemática linear é então considerada na equação do sólido permitindo a utilização do mesmo modelo Euleriano usado nas equações do fluido. Além do mais, uma membrana linear é reforçada com cisalhamento transversal (HUGHES, 1987), pois há instabilidade da membrana linear sob cargas transversais, não sendo necessário graus de liberdade adicionais na interface fluido-sólido além dos deslocamentos. Assim sendo, o modelo não necessita de remalhamento e são incorporados alguns termos adicionais na interface, tornando o método computacionalmente mais vantajoso do que a formulação ALE (FIGUEROA et al., 2006).

A formulação padrão de Galerkin para as equações de Navier-Stokes incompressíveis, utilizando polinômios interpoladores de mesma ordem, apresenta instabilidades. Dessa forma, é considerada a formulação estabilizada *Stremline-Upwind/Petrov-Galerkin* (SUPG) (BROOKS; HUGHES, 1982) para a velocidade, a formulação estabilizada *Pressure-Stabilizing/Petrov-Galerkin* (PSPG) (TEZDUYAR, 1991) para a pressão, e um termo de captura *Least-Squares on Incompressibility Constraint* (LSIC) (TEZDUYAR; OSAWA, 2000).

Em geral, as formulações estabilizadas tratam velocidade e pressão de forma acopladas. Outras abordagens possíveis são os métodos de projeções, utilizados para as equações de Navier-Stokes incompressíveis e dependente do tempo, apresentado por (CHORIN, 1968) e posteriormente por (TEMAM, 1969). Essa abordagem trata velocidade e pressão de forma desacopladas e, com isso, a cada passo de tempo é necessário resolver uma sequência de equações mais simples. Dessa forma, são considerados três métodos de projeção: esquema de correção não incremental da pressão (Chorin) (*non-incremental pressure correction scheme - Chorin*) (CHORIN, 1968), esquema de correção incremental da pressão (IPCS) (*incremental pressure correction scheme - IPCS*) (GODA, 1979), esquema de

divisão consistente (CSS) (*consistent splitting scheme* - CSS) (GUERMOND; SHEN, 2003; GUERMOND; MINEV; SHEN, 2006), com polinômios interpoladores quadráticos para a velocidade e lineares para a pressão, denominados de elementos Taylor-Hood (P2/P1).

O tamanho do passo de tempo é de suma importância em problemas que possuem dependência temporal, pois este influencia na solução a ser obtida, no tempo computacional, entre outros fatores. A fim de obter um passo de tempo adequado, foi utilizado o controlador Proporcional-Integral-Diferencial (PID) (VALLI; CAREY; COUTINHO, 2002) para as equações de Navier-Stokes.

FEniCS é uma biblioteca de domínio público para solução de equações diferenciais baseada no método de elementos finitos (LOGG; MARDAL; WELLS, 2012). O projeto FEniCS iniciou-se em 2003 com a finalidade de facilitar a solução automatizada de modelos matemáticos. É uma biblioteca abrangente de elementos finitos que inclui: controles de erros automatizados e adaptabilidade, álgebra linear de alto desempenho, computação em paralelo, dentre outras funcionalidades. A implementação é simples, pois a formulação variacional é detalhada em uma notação quase matemática, incluindo todo tipo de forma variacional, o que caracteriza a generalidade da biblioteca. Programas FEniCS podem ser implementados nas linguagens Python e C++.

1.2 Objetivos do Trabalho

Este trabalho tem por objetivo geral o estudo das equações de Navier-Stokes incompressíveis e problemas de interação fluido-estrutura via método dos elementos finitos considerando a biblioteca de domínio público FEniCS.

Para as equações de Navier-Stokes, é realizada a comparação de três métodos de projeção e as formulações estabilizadas, avaliando a acurácia e o tempo computacional para diferentes problemas e tamanhos de malhas, selecionando dois desses métodos. Os métodos escolhidos foram aplicados em dois problemas clássicos da literatura com a técnica adaptativa PID de avanço no tempo, a fim de avaliar a acurácia das soluções e o tempo computacional.

Os problemas de interação fluido-estrutura utilizam a abordagem CMM-FSI. Portanto, é avaliado a acurácia, vantagens e desvantagens do método CMM-FSI aplicado a problemas fisiológicos.

A biblioteca FEniCS escolhida para implementação dos experimentos numéricos gerencia as dificuldades dos problemas, liberando o pesquisador para se concentrar nos aspectos mais relevantes da simulação. Assim sendo, é estudado o desempenho da FEniCS aplicada as equações de Navier-Stokes e problemas de interação fluido-estrutura.

1.3 Organização da Dissertação

No Capítulo 2 são apresentadas as equações de Navier-Stokes para fluidos incompressíveis e sua formulação variacional. Além disso, métodos estabilizados e de projeção são descritos. Por fim, é apresentada a técnica PID de avanço no tempo.

O Capítulo 3 discute a estratégia adotada para o problema de interação fluido-estrutura. São apresentadas as equações do sólido, a formulação variacional e o acoplamento fluido-estrutura utilizando o método CMM-FSI.

O Capítulo 4 descreve a implementação dos problemas utilizando a biblioteca FEniCS. Primeiramente é descrito uma introdução à biblioteca e, em seguida, as implementações para as equações de Navier-Stokes e, por fim, as implementações para os problemas de interação fluido-estrutura.

No Capítulo 5 são apresentados os resultados da pesquisa. Primeiramente, é realizada uma seleção de dois métodos para serem aplicados em dois problemas distintos com a técnica adaptativa no tempo: cavidade bidimensional com tampa deslizante e o problema do escoamento com alargamento de canal. Posteriormente, são realizados três experimentos em problemas de interação fluido-estrutura: parede deformável, artéria idealizada e artéria aneurismática.

Por fim, no Capítulo 7 são feitas as conclusões e sugestões para futuros trabalhos.

2 Equações de Navier-Stokes

Neste capítulo apresentamos as equações governantes do problema da mecânica dos fluidos, governado pelas equações de Navier-Stokes (ENS), bem como as condições de contorno pertinentes ao problema. Além disso, é apresentada a formulação padrão semi-discreta de Galerkin, que é imprescindível para o método dos elementos finitos. Entretanto, para escoamento com convecção dominante, a forma padrão de Galerkin apresenta instabilidades numéricas, utilizando polinômios interpoladores de mesma ordem para pressão e velocidade. Portanto, serão utilizadas formulações estabilizadas, conforme descrito em (ELIAS, 2007), sendo que pressão e velocidade são resolvidas simultaneamente, ou seja, estão acopladas.

Uma forma alternativa de solução são os chamados métodos de projeção e, neste trabalho, foram utilizados polinômios com grau dois para a velocidade e grau um para a pressão, denominados de elementos Taylor-Hood (P2/P1), conforme descrito em (LOGG; MARDAL; WELLS, 2012). Além disso, técnicas adaptativas de avanço no tempo são consideradas com o objetivo de melhorar a acurácia da solução e/ou acelerar a convergência.

2.1 Equações Governantes

O fluido é considerado como newtoniano, viscoso e incompressível, ocupando um domínio $\Omega \in \mathcal{R}^{dim}$ com contorno Γ , onde dim é a dimensão no espaço. As equações de Navier-Stokes (ENS) são provenientes da lei da conservação da massa e do momento, que com condições de contorno pertinentes formam um sistema fechado de equações diferenciais parciais. Seja $\Omega \in \mathcal{R}^{dim}$ e Γ o contorno, tal que $\Gamma = \Gamma_g \cup \Gamma_h$ e $\Gamma_g \cap \Gamma_h = \emptyset$, as equações de Navier-Stokes, descritas em (ELIAS, 2007), são dadas por:

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} - \nabla \cdot \sigma(\vec{v}, p) = \vec{f}, \text{ em } \Omega \times (0, T), \quad (2.1)$$

$$\nabla \cdot \vec{v} = 0, \text{ em } \Omega \times (0, T), \quad (2.2)$$

$$\vec{v} = \vec{g}, \text{ em } \Gamma_g \times (0, T), \quad (2.3)$$

$$\vec{t}_{\vec{n}} = \sigma(\vec{v}, p) \vec{n} = \vec{h}, \text{ em } \Gamma_h \times (0, T), \quad (2.4)$$

$$\vec{v}(\vec{x}, 0) = \vec{v}^0(\vec{x}), \text{ em } \Omega^0, \quad (2.5)$$

onde \vec{v} é o vetor velocidade do fluido, \vec{f} é a força de corpo, T é o tempo final, \vec{g} é a condição de contorno do tipo Dirichlet, \vec{x} é o vetor de coordenadas espaciais, \vec{n} é o vetor normal e $\vec{v}^0(\vec{x})$ é a condição inicial do problema. A Eq. (2.4) é a condição de contorno do tipo Neumann.

Para fluidos newtonianos, o tensor das tensões cisalhantes relaciona-se por uma constante de proporcionalidade com a taxa de deformação do fluido, denominada de viscosidade cinemática. Em fluidos não-newtonianos esta relação não é constante e não será tratada neste trabalho. Portanto, o tensor das tensões de Cauchy $\sigma(\vec{v}, p)$ é dado por

$$\sigma(\vec{v}, p) = 2\nu\epsilon(\vec{v}) - pI, \quad (2.6)$$

onde p é a pressão do fluido, ν é a viscosidade cinemática do fluido, I é o tensor identidade e $\epsilon(\vec{v})$ é o tensor deformação do fluido,

$$\epsilon(\vec{v}) = \frac{1}{2} \left(\nabla \vec{v} + (\nabla \vec{v})^T \right). \quad (2.7)$$

As Eqs. (2.1 - 2.7) estão na forma adimensional. Portanto, a viscosidade cinemática é dada como o inverso do número de Reynolds, onde $Re = \frac{vL}{\nu}$ e L é o comprimento efetivo do escoamento.

2.2 Formulação Variacional

Vamos considerar a discretização do domínio Ω , subdivido em vários domínios elementares Ω_e . Sejam os seguintes espaços variacionais (FIGUEROA et al., 2006):

$$\mathcal{V}_h^k = \{ \vec{v} | \vec{v}(\cdot, t) \in H^k(\Omega)^{dim}, t \in [0, T], \vec{v}|_{\vec{x} \in \Omega_e} \in P_k(\Omega_e)^{dim}, \vec{v}(\cdot, t) = \vec{g} \text{ em } \Gamma_g \}, \quad (2.8)$$

$$\mathcal{W}_h^k = \{ \vec{w} | \vec{w}(\cdot, t) \in H^k(\Omega)^{dim}, t \in [0, T], \vec{w}|_{\vec{x} \in \Omega_e} \in P_k(\Omega_e)^{dim}, \vec{w}(\cdot, t) = \vec{0} \text{ em } \Gamma_g \}, \quad (2.9)$$

$$\mathcal{P}_h^m = \{ p | p(\cdot, t) \in H^m(\Omega), t \in [0, T], p|_{\vec{x} \in \Omega_e} \in P_m(\Omega_e) \}. \quad (2.10)$$

O campo da velocidades é vetorial e o da pressão é escalar, porém, por simplicidade de notação, neste trabalho não vamos utilizar a notação vetorial para o campo de velocidades. A derivada temporal será aproximada pelo método das diferenças finitas atrasada $\left(\frac{\partial v}{\partial t} = \frac{v - v^n}{\Delta t} \right)$, onde v^n é a velocidade do passo de tempo anterior e Δt é o tamanho do passo de tempo. Considerando os espaços definidos nas Eqs. (2.8 - 2.10) e as Eqs. (2.1 - 2.5), a formulação semi-discreta de Galerkin resulta em:

Encontrar $v \in \mathcal{V}_h^k$ e $p \in \mathcal{P}_h^m$, de tal modo que

$$F(v, p; w, q) = \int_{\Omega} \left\{ w \cdot \left(\frac{v - v^n}{\Delta t} + v \cdot \nabla v - f \right) + \nabla w : \sigma(v, p) - q \nabla \cdot v \right\} d\Omega - \int_{\Gamma_h} w \cdot h d\Gamma = 0 \quad (2.11)$$

$\forall w \in \mathcal{V}_h^k$ e $q \in \mathcal{P}_h^m$.

Para uma classe representativa de escoamentos regidos pelas equações de Navier-Stokes a formulação padrão de Galerkin considerando polinômios interpoladores de mesma ordem é instável, uma vez que podem não satisfazer a condição inf-sup discreta, definida por Ladyzhenskaya-Babuska-Brezzi (LBB) (BREZZI, 1974). A fim de contornar as possíveis instabilidades numéricas, escolher interpolações de ordem $k + 1$ para a velocidade e k para a pressão ou acrescentar termos estabilizadores tem sido considerado com sucesso nas últimas décadas. Neste trabalho iremos adotar métodos de projeções, utilizando polinômios de ordem dois para a velocidade e um para a pressão, que são os chamados elementos Taylor-Hood (P2/P1), e também, formulações estabilizadas com polinômios lineares para velocidade e pressão: Formulação SUPG para a velocidade, Formulação PSPG para a pressão enriquecida com a estabilização LSIC.

2.2.1 Formulação estabilizada

Para tornar a formulação de Galerkin estável, termos estabilizadores são adicionados à Eq. (2.11). As formulações SUPG e PSPG podem ser simplificadas, segundo descrito em (DONEA; HUERTA, 2003), tendo por objetivo: acrescentar

$$\sum_e \int_{\Omega^e} \mathcal{P}(w) \tau \mathcal{R}(v, p) d\Omega, \quad (2.12)$$

na forma semi-discreta de Galerkin, onde $\mathcal{P}(w)$ é um determinado operador aplicado no espaço de funções teste, τ é o parâmetro de estabilização e $\mathcal{R}(v, p)$ é o resíduo da equação do momento, dado por

$$\mathcal{R}(v, p) = \frac{v - v^n}{\Delta t} + v \cdot \nabla v - \nabla \cdot \sigma(v, p) - f. \quad (2.13)$$

O termo PSPG insere uma quantidade de compressibilidade ao problema, estabilizando a pressão. Já para a velocidade, é utilizado o termo SUPG que aumenta a difusão quando necessário. Por fim, é considerada uma difusão artificial atuando quando o número de Reynolds é elevado (TEZDUYAR; OSAWA, 2000), denominada LSIC. Portanto a Eq. (2.11) estabilizada é definida por:

Encontrar $v \in \mathcal{V}_h^k$ e $p \in \mathcal{P}_h^m$, de tal modo que

$$\begin{aligned} F(v, p; w, q) &= \int_{\Omega} \left\{ w \cdot \left(\frac{v - v^n}{\Delta t} + v \cdot \nabla v - f \right) + \nabla w : \sigma(v, p) - q \nabla \cdot v \right\} d\Omega \\ &+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} (\tau_{SUPG} v \cdot \nabla w + \tau_{PSPG} \nabla q) \cdot R(v, p) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{LSIC} \nabla \cdot w \nabla \cdot v d\Omega \\ &- \int_{\Gamma_h} w \cdot h d\Gamma = 0, \end{aligned} \quad (2.14)$$

$\forall w \in \mathcal{V}_h^k$ e $q \in \mathcal{P}_h^m$, onde τ_{SUPG} , τ_{PSPG} e τ_{LSIC} são, respectivamente, os parâmetros de estabilização SUPG, PSPG e LSIC. Os parâmetros SUPG e PSPG foram definidos por (SHAKIB; HUGHES; JOHAN, 1991), e são dados por:

$$\tau_{SUPG} = \tau_{PSPG} = \left(\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\|v\|}{h} \right)^2 + 9 \left(\frac{4\nu}{h^2} \right)^2 \right)^{-\frac{1}{2}}, \quad (2.15)$$

e o parâmetro LSIC é descrito em (TEZDUYAR; OSAWA, 2000), dado por:

$$\tau_{LSIC} = \frac{\|v\|h}{2} \quad (2.16)$$

2.2.2 Métodos de projeções

Uma das grandes dificuldades na solução das equações de Navier-Stokes incompressíveis esta no acoplamento da velocidade e pressão. O método de projeção foi introduzido por (CHORIN, 1968) e posteriormente aperfeiçoado (TEMAM, 1969), a fim de solucionar as equações de Navier-Stokes para escoamentos incompressíveis de forma eficiente. O método de projeção baseou-se no pressuposto de que, para escoamentos incompressíveis, a pressão está presente apenas como um multiplicador de Lagrange para a restrição de incompressibilidade, não tendo significado termodinâmico (CHORIN, 1968). Devido a esta observação, motivou-se um esquema de divisão que desacopla velocidade e pressão, característica chave dos métodos de projeções. Esses métodos são bem compreendidos para número de Reynolds moderado e altamente influenciados pelas condições de contorno, podendo introduzir oscilações numéricas nos campos de pressão e velocidade, ocasionando imprecisões nos resultados (WEIMAN; LIU, 1995). Para os métodos de projeções listados a seguir são considerados polinômios interpoladores de ordem quadrática para a velocidade e lineares para a pressão.

Chorin

O método, muitas vezes denotado por *esquema de correção não incremental da pressão*, foi proposto por (CHORIN, 1968) e (TEMAM, 1969). Este esquema usa um tratamento de passo fracionário. É utilizado uma aproximação da equação do momento, Eq. (2.1), para estimar uma velocidade provisória e, posteriormente, é solucionada uma

equação elíptica, na qual impõe a restrição da divergência, proveniente da equação da continuidade Eq. (2.2), que determina a pressão (BROWN; CORTEZ; MINION, 2001).

Portanto, a ideia do método é determinar uma estimativa para a velocidade, em que a pressão é desconsiderada na equação do momento e em seguida projetar a velocidade nos campos vetoriais livres de divergência. A primeira etapa representa os efeitos viscosos, enquanto a segunda trata da incompressibilidade (GUERMOND; MINEV; SHEN, 2006).

Este método consiste em:

Passo 1 - Estimativa da velocidade:

$$\int_{\Omega} \left\{ \left(\frac{v^* - v^n}{\Delta t} + v^n \cdot \nabla v^n - f \right) \cdot w + \nu \nabla v^* \cdot \nabla w \right\} d\Omega = 0 \quad (2.17)$$

Passo 2 - Determinação da pressão:

$$\int_{\Omega} \nabla p \cdot \nabla q d\Omega = -\frac{1}{\Delta t} \int_{\Omega} \nabla \cdot v^* q d\Omega \quad (2.18)$$

Passo 3 - Correção da velocidade:

$$\int_{\Omega} v \cdot w d\Omega = \int_{\Omega} v^* \cdot w d\Omega - \Delta t \int_{\Omega} \nabla p \cdot w d\Omega \quad (2.19)$$

onde v^* é a estimativa para a velocidade.

Esquema de Correção Incremental da Pressão (IPCS)

O *esquema de correção incremental da pressão*, proposto por (GODA, 1979), é derivado do método Chorin, sendo que para determinar a estimativa da velocidade é utilizada a pressão do passo de tempo anterior. O esquema de desacoplamento considera a pressão da Eq. (2.1) do passo de tempo anterior (p^n) ao invés da atual (p). O termo convectivo é linearizado de uma forma semi-implícita, $(v^n \cdot \nabla) U$, e ainda, os termos viscoso e convectivo são de segunda ordem, onde $U = v^{n-1/2} = \frac{1}{2}(v + v^n)$, conforme (BROWN; CORTEZ; MINION, 2001). Este primeiro passo irá determinar uma estimativa para a velocidade mantendo a pressão desconhecida. Adiante a pressão é calculada, com base na velocidade aproximada encontrada e no último passo a velocidade é corrigida. Para mais detalhes consultar (GODA, 1979).

O esquema de correção incremental da pressão resulta em:

Passo 1 - Estimativa da velocidade em função da pressão do passo de tempo anterior:

$$\begin{aligned} \int_{\Omega} \left\{ w \cdot \left(\frac{v^* - v^n}{\Delta t} + v^n \cdot \nabla U - f \right) + \epsilon(w) : \sigma(U, p^n) \right\} d\Omega \\ - \int_{\Gamma} \sigma(U, p^n) n \cdot w d\Gamma = 0 \end{aligned} \quad (2.20)$$

Passo 2 -Determinação da pressão:

$$\int_{\Omega} \nabla q \cdot \nabla p d\Omega - \int_{\Omega} \nabla q \cdot \nabla p^n d\Omega + \int_{\Omega} \frac{1}{\Delta t} q \nabla \cdot v^* d\Omega = 0 \quad (2.21)$$

Passo 3 - Correção da velocidade:

$$\int_{\Omega} w \cdot v d\Omega - \int_{\Omega} w \cdot v^* d\Omega + \Delta t \int_{\Omega} w \cdot (\nabla p - \nabla p^n) d\Omega = 0 \quad (2.22)$$

onde n é o vetor normal.

Esquema de divisão consistente (CSS)

Os métodos de projeções referenciados acima se baseiam em uma aproximação inicial para a velocidade, sem a restrição de incompressibilidade, e nas equações seguintes essa restrição é imposta. O método CSS, descrito por (GUERMOND; SHEN, 2003) e (GUERMOND; MINEV; SHEN, 2006), é diferente dos outros esquemas já apresentados. Este esquema é fundamentado em encontrar uma equação para a pressão p , multiplicando a equação do momento das equações de Navier-Stokes (Eq. (2.1)) pelo gradiente da função de teste da pressão q . A combinação dessa equação com a restrição de incompressibilidade, gera uma equação para a pressão. A velocidade é solucionada com base na equação do momento das equações de Navier-Stokes por uma extrapolação da pressão.

O método consiste em multiplicar a equação do momento das ENS, Eq. (2.1), por ∇q integrando-o sobre o domínio, resultando em $\left(\frac{\partial v}{\partial t} + v \cdot \nabla v - \nu \Delta v + \nabla p, \nabla q\right) = (f, \nabla q)$. A parte temporal pode ser reescrita como $\left(\frac{\partial v}{\partial t}, \nabla q\right) = (\nabla \cdot v, -q) + \left(\frac{\partial v}{\partial t}, qn\right)_{\Gamma} = 0$, onde n é o vetor normal. Impondo a restrição de incompressibilidade ($\nabla \cdot v = 0$), e assumindo que $\frac{\partial v}{\partial t} = 0$ em Γ , a seguinte equação é obtida:

$$(\nabla p, \nabla q) = (f - v \cdot \nabla v + \nu \Delta v, \nabla q) \quad (2.23)$$

Utilizando a identidade

$$\Delta A \equiv \nabla \nabla \cdot A - \nabla \times \nabla \times A \quad (2.24)$$

combinada com a restrição de incompressibilidade, a Eq. (2.23), escrita em forma rotativa, resulta em:

$$(\nabla p, \nabla q) = (f - v \cdot \nabla v - \nu \nabla \times \nabla \times v, \nabla q). \quad (2.25)$$

A Eq. (2.25) é a base para o método. Portanto, o CSS tende a solucionar os seguintes

problemas (LOGG; MARDAL; WELLS, 2012):

$$\left(\frac{v - v^n}{\Delta t}, w\right) + (v^n \cdot \nabla v^n, w) + (\nu \nabla v, \nabla w) - (p^*, \nabla \cdot w) = (f, w), \quad (2.26)$$

$$(\nabla p, \nabla q) = (f - v^n \cdot \nabla v^n - \nu \nabla \times \nabla \times v, \nabla q), \quad (2.27)$$

onde p^* é a extrapolação para a pressão.

O caso mais simples é utilizando aproximação linear para a pressão $p^* = p^n$, embora seja possível utilizar ordens superiores, tais como de segunda ordem $p^* = 2p^{n-1} - p^{n-2}$. Ambos os casos serão utilizados neste trabalho.

É desejável eliminar os termos de rotação presentes na Eq. (2.27). Para isso, a Eq. (2.26) é reescrita e multiplicada pelo gradiente de q , resultando em:

$$\left(\frac{v - v^n}{\Delta t} + v^n \cdot \nabla v^n - \nu \Delta v + \nabla p^*, \nabla q\right) = (f, \nabla q) \quad (2.28)$$

Diante disto, a Eq. (2.28) é subtraída da Eq. (2.27) e utilizando novamente a identidade Eq. (2.24), a Eq. (2.27) resulta em:

$$(\nabla p - \nabla p^*, \nabla q) = \left(\frac{v - v^n}{\Delta t} - \nu \nabla \nabla \cdot v, \nabla q\right), \quad (2.29)$$

ou

$$\{\nabla(p - p^* + \nu \nabla \cdot v), \nabla q\} = \left(\frac{v - v^n}{\Delta t}, \nabla q\right). \quad (2.30)$$

Por fim, é definido um parâmetro auxiliar ($\psi^n = p - p^* + \nu \nabla \cdot v$) para escrever a Eq. (2.30) como:

$$(\nabla \psi^n, \nabla q) = \left(\frac{v - v^n}{\Delta t}, \nabla q\right) \quad (2.31)$$

Portanto, os passos do método CSS são:

Passo 1 - Selecione uma aproximação para a pressão p^* :

$$p^* = \begin{cases} p^n, & \text{para CSS1} \\ 2p^n - p^{n-1}, & \text{para CSS2} \end{cases} \quad (2.32)$$

Passo 2 - Cálculo da velocidade em função da pressão aproximada:

$$\begin{aligned} \int_{\Omega} \left\{ \left(\frac{v - v^n}{\Delta t} + v^n \cdot \nabla v^n - f \right) \cdot w + \sigma(v, p^*) : \epsilon(w) \right\} d\Omega \\ \int_{\Gamma} (\bar{p}n - \nu n \cdot (\nabla v)^T) \cdot w d\Gamma = 0, \end{aligned} \quad (2.33)$$

onde \bar{p} é uma condição de contorno imposta para a pressão.

Passo 3 - Resolva a correção para a pressão ψ :

$$\int_{\Omega} \nabla \psi \cdot \nabla q d\Omega = \int_{\Omega} \frac{v - v^n}{\Delta t} \cdot w d\Omega - \int_{\Gamma} \frac{v - v^n}{\Delta t} \cdot w d\Gamma \quad (2.34)$$

Passo 4 - Cálculo da pressão p :

$$\int_{\Omega} p q d\Omega = \int_{\Omega} (p^* + \psi - \nu \nabla \cdot v) q d\Omega. \quad (2.35)$$

2.3 Técnicas Adaptativas de Avanço no Tempo

A determinação do passo de tempo é essencial para obter uma boa solução e/ou redução do tempo computacional. Para isso foi utilizado o controlador Proporcional-Integral-Diferencial (PID), descrito em (VALLI; CAREY; COUTINHO, 2002). A Fig. 1 fornece uma ideia básica deste tipo de controlador em que estipulada uma tolerância, o controle mantém o erro dentro de uma tolerância pré-fixada.

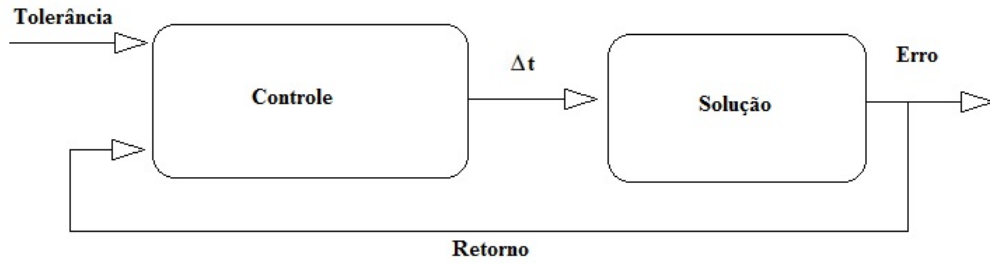


Figura 1 – Seleção do passo de tempo como um controlador

O tamanho do passo de tempo é definido em (VALLI; CAREY; COUTINHO, 2002) por:

$$\Delta t_{n+1} = PID \Delta t_n, \quad (2.36)$$

onde Δt_n é o tamanho do passo de tempo atual, Δt_{n+1} é o novo passo de tempo e PID é um fator multiplicador dado por:

$$PID = \left(\frac{e_{n-1}}{e_n} \right)^{k_P} \left(\frac{tol}{e_n} \right)^{k_I} \left(\frac{e_{n-1}^2}{e_n e_{n-2}} \right)^{k_D} \quad (2.37)$$

onde $k_P = 0.075$, $k_I = 0.175$ e $k_D = 0.01$ são parâmetros do controlador, tol é uma tolerância pré-fixada, e_n , e_{n-1} e e_{n-2} são normas euclidianas relativa da velocidade, dadas como

$$e_n = \frac{\|v - v^n\|}{\|v\|}, \quad (2.38)$$

$$e_{n-1} = \frac{\|v^n - v^{n-1}\|}{\|v^n\|}, \quad (2.39)$$

$$e_{n-2} = \frac{\|v^{n-1} - v^{n-2}\|}{\|v^{n-1}\|}. \quad (2.40)$$

3 Interação Fluido-Estrutura

Nesta seção é apresentado o método de impulso acoplado para problemas de interação fluido-estrutura (CMM-FSI - Coupled Momentum Method for FSI). Neste método o sólido é definido como uma membrana, sendo consideradas formulações variacionais para as equações de Navier-Stokes em um domínio rígido, modificando-as de tal forma que a deformidade da membrana é levada em consideração. Em seguida, é apresentado o problema da mecânica dos sólidos, governado pelas clássicas equações elastodinâmicas, suas condições de contorno e a formulação variacional para o problema do sólido. Além disso, é feito o acoplamento das equações do sólido na forma variacional do fluido como uma condição de contorno para os problemas de interação fluido-estrutura. Para isso, é utilizada a formulação estabilizada das ENS e a condição de não deslizamento é substituída por uma condição de tração na parede (condição de Neumann). A Fig. 2 descreve o domínio do problema, sendo que a estrutura é considerada como uma parede fina. Ressaltamos que o método descrito neste trabalho tem por finalidade simular problemas fisiológicos.

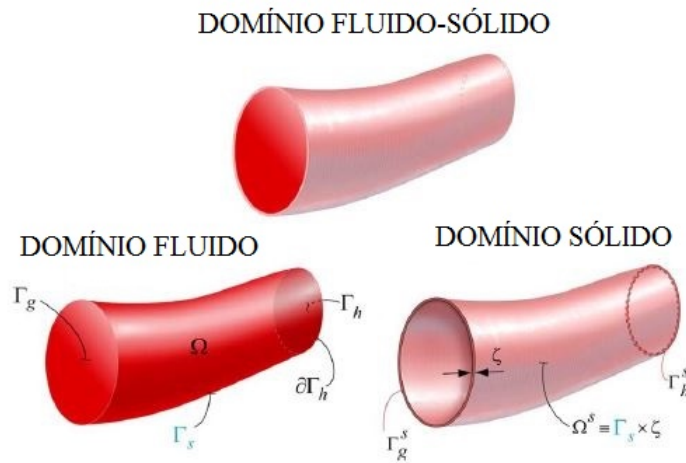


Figura 2 – Representação dos domínios, adaptado de (FIGUEROA et al., 2006)

3.1 Equações Governantes

O sólido é descrito pelas clássicas equações elastodinâmicas dependentes do tempo. Considere o domínio Ω^s e Γ^s o contorno, tal que $\Gamma^s = \Gamma_g^s \cup \Gamma_h^s$ e $\Gamma_g^s \cap \Gamma_h^s = \emptyset$, as equações, descritas em (LARSON M. B., 2013), são dadas por:

$$\rho^s \frac{\partial^2 \vec{u}}{\partial t^2} = \nabla \cdot \sigma^s + \vec{b}^s, \text{ em } \Omega^s \times (0, T), \quad (3.1)$$

$$\vec{u} = \vec{g}^s, \text{ em } \Gamma_h^s \times (0, T), \quad (3.2)$$

$$\vec{t}_{\vec{n}} = \sigma^s \vec{n} = \vec{h}^s, \text{ em } \Gamma_g^s \times (0, T), \quad (3.3)$$

$$\vec{u}(\vec{x}, 0) = \vec{u}^0(\vec{x}) \text{ em } \Omega_0^s, \quad (3.4)$$

$$\frac{\partial \vec{u}(\vec{x}, 0)}{\partial t} = \frac{\partial \vec{u}^0(\vec{x})}{\partial t} \text{ em } \Omega_0^s, \quad (3.5)$$

onde \vec{u} é o vetor deslocamento, \vec{b}^s força de corpo do sólido por unidade de volume, σ^s é o tensor das tensões da membrana, \vec{h}^s é a força de tração na interface, e as Eqs. (3.4) e (3.5) representam as condições iniciais.

3.2 Formulação Variacional

Considerando estruturas de paredes finas, existe assim uma dependência do domínio do sólido (Ω^s) com a interface do fluido (Γ_s). A formulação variacional do sólido se relaciona com a força de tração do fluido na interface fluido-sólido. Além disso, é considerado um modelo de membrana para a estrutura, que define um significativo acoplamento dos graus de liberdade do domínio do fluido e do sólido na interface. Neste contexto (FIGUEROA et al., 2006) sugere a inserção do termo

$$\int_{\Gamma_s} -\vec{w} \cdot \vec{t}^f d\Gamma \quad (3.6)$$

nas equações de Navier-Stokes, representando a tração na parede, desconsiderando a condição de não escorregamento.

Dessa forma a formulação variacional para o problema de Navier-Stokes resulta em:

Encontrar $v \in \mathcal{V}_h^k$ e $p \in \mathcal{P}_h^m$, de tal modo que

$$\begin{aligned} F(v, p; w, q) = \int_{\Omega} \left\{ w \cdot \left(\rho \frac{v - v^n}{\Delta t} + \rho v \cdot \nabla v - f \right) + \nabla w : \sigma(v, p) - q \nabla \cdot v \right\} d\Omega \\ - \int_{\Gamma_h} w \cdot h d\Gamma - \int_{\Gamma_s} w \cdot t^f d\Gamma = 0 \end{aligned} \quad (3.7)$$

O sólido é considerado sendo de parede fina e, além disso, queremos definir uma relação das equações do sólido com a Eq. (3.6), ou seja, relacionar as equações de domínio do sólido com a interface do fluido. Então faz-se necessário definir uma relação do domínio Ω^s da membrana com a interface Γ_s do fluido. Assim, podemos utilizar as seguintes relações:

$$\int_{\Omega^s} (\cdot) d\Omega = \zeta \int_{\Gamma_s} (\cdot) d\Gamma, \quad (3.8)$$

$$\int_{\Gamma_h^s} (\cdot) d\Gamma = \zeta \int_{\partial\Gamma_h} (\cdot) dl, \quad (3.9)$$

onde ζ é a espessura da parede.

A força que o fluido faz na parede do sólido é igual e oposta a que o sólido faz na interface do fluido

$$\vec{t}^f = -\vec{t}^s. \quad (3.10)$$

Em uma estrutura fina o suficiente, a tração de superfície interna \vec{t}^s é distribuída de forma uniforme sobre a espessura da parede ζ . Portanto, segue-se que

$$\vec{t}^s = \frac{\vec{b}^s}{\zeta}, \quad (3.11)$$

sendo possível relacionar o termo integral desconhecido, Eq. (3.6), com a forma fraca para o problema do sólido. Essas definições, para acoplar as equações do sólido ao nível variacional do fluido, são análogas à derivação de (WOMERSLEY, 1955) de uma solução analítica para o fluxo sanguíneo pulsátil axisimétrico em um vaso elástico cilíndrico. Porém para o presente trabalho o tensor de tensão do sólido é modificado como descrito por (FIGUEROA et al., 2006), a ser definido mais adiante.

Portanto, existe um forte acoplamento dos graus de liberdade entre os domínios do fluido e do sólido. Dessa forma, os campos de deslocamento, velocidade e aceleração na interface fluido-sólido são idênticos. Além disso, as equações que descrevem o comportamento da parede possuem as mesmas condições de diferenciação daquelas descritas pela forma variacional do fluido. Com isso, os mesmos espaços de funções podem ser utilizados para a formulação do sólido, ou seja:

$$\mathcal{V}_h^{sk} = \{\vec{u} | \vec{u}(\cdot, t) \in H^k(\Omega^s)^3, t \in [0, T], \vec{u}|_{\vec{x} \in \Omega_e^s} \in P_k(\Omega_e^s)^3, \vec{u}(\cdot, t) = \vec{g}^s \text{ em } \Gamma_g^s\}, \quad (3.12)$$

$$\mathcal{W}_h^{sk} = \{\vec{w} | \vec{w}(\cdot, t) \in H^k(\Omega^s)^3, t \in [0, T], \vec{w}|_{\vec{x} \in \Omega_e^s} \in P_k(\Omega_e^s)^3, \vec{w}(\cdot, t) = \vec{0} \text{ em } \Gamma_g^s\}. \quad (3.13)$$

Portanto, a formulação semi-discreta de Galerkin para o sólido resulta em:

Encontrar $\vec{u} \in \mathcal{V}_h^{sk}$, de tal modo que:

$$\int_{\Omega^s} \rho^s \vec{w} \cdot \frac{\partial^2 \vec{u}}{\partial t^2} d\Omega + \int_{\Omega^s} \nabla \vec{w} : \sigma^s d\Omega = \int_{\Omega^s} \vec{w} \cdot \vec{b}^s d\Omega + \int_{\Gamma_h^s} \vec{w} \cdot \vec{h}^s d\Gamma \quad (3.14)$$

$$\forall \vec{w} \in \mathcal{W}_h^{sk}.$$

3.3 Acoplamento fluido-sólido

O método CMM-FSI tem por objetivo considerar a formulação variacional do sólido em função da velocidade e aceleração do fluido na interface. Como a derivada segunda do deslocamento na interface está relacionada com a aceleração do fluido, tem-se:

$$\vec{v} = \frac{\partial \vec{u}}{\partial t}, \quad (3.15)$$

e utilizando as Eqs. (3.8 - 3.11) podemos reescrever, resultando em:

$$\zeta \int_{\Gamma_s} \rho^s \vec{w} \cdot \frac{\partial \vec{v}}{\partial t} d\Gamma + \zeta \int_{\Gamma_s} \nabla \vec{w} : \sigma^s d\Gamma = \int_{\Gamma_s} \vec{w} \cdot \vec{t}^f d\Gamma + \zeta \int_{\partial\Gamma_h} \vec{w} \cdot \vec{h}^s dl, \quad (3.16)$$

ou

$$- \int_{\Gamma_s} \vec{w} \cdot \vec{t}^f d\Gamma = \zeta \int_{\Gamma_s} \rho^s \vec{w} \cdot \frac{\partial \vec{v}}{\partial t} d\Gamma + \zeta \int_{\Gamma_s} \nabla \vec{w} : \sigma^s d\Gamma - \zeta \int_{\partial\Gamma_h} \vec{w} \cdot \vec{h}^s dl. \quad (3.17)$$

A Eq. (3.17) relaciona as equações do sólido com a interface do fluido por uma condição de contorno. Logo, as equações elastodinâmicas provenientes do problema do sólido podem ser acopladas com a forma fraca do fluido (Eq. (3.7)), como uma condição de contorno, resultando em:

$$\begin{aligned} F(\vec{v}, p; \vec{w}, q) &= \int_{\Omega} \left\{ \vec{w} \cdot \left(\rho \frac{\partial \vec{v}}{\partial t} + \rho \vec{v} \cdot \nabla \vec{v} - \vec{f} \right) + \nabla \vec{w} : \sigma(\vec{v}, p) - q \nabla \cdot \vec{v} \right\} d\Omega \\ &- \int_{\Gamma_h} \vec{w} \cdot \vec{h} d\Gamma + \zeta \int_{\Gamma_s} \rho^s \vec{w} \cdot \frac{\partial \vec{v}}{\partial t} d\Gamma + \zeta \int_{\Gamma_s} \nabla \vec{w} : \sigma^s d\Gamma - \zeta \int_{\partial\Gamma_h} \vec{w} \cdot \vec{h}^s dl = 0, \end{aligned} \quad (3.18)$$

Todos os termos da Eq. (3.18) já foram descritos, com exceção do tensor das tensões do sólido que será detalhado mais adiante. As integrais adicionadas a formulação variacional fazem com que a deformidade seja considerada. Além disso, a formulação estabilizada não é alterada, pois os termos estabilizadores alteram apenas o elementos internos do fluido. Portanto, o método CMM-FSI na formulação estabilizada é definido por:

Encontrar $v \in \mathcal{V}_h^1$ e $p \in \mathcal{P}_h^1$, de tal modo que

$$\begin{aligned} F(v, p; w, q) &= \int_{\Omega} \left\{ w \cdot \left(\rho \frac{\partial v}{\partial t} + \rho v \cdot \nabla v - f \right) + \nabla w : \sigma(v, p) - q \nabla \cdot v \right\} d\Omega \\ &+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{1}{\rho} (\tau_{SUPG} \rho v \cdot \nabla w + \tau_{PSPG} \nabla q) \cdot R(v, p) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{LSIC} \rho \nabla \cdot w \nabla \cdot v d\Omega \\ &- \int_{\Gamma_h} w \cdot h d\Gamma + \zeta \int_{\Gamma_s} \rho^s w \cdot \frac{\partial v}{\partial t} d\Gamma + \zeta \int_{\Gamma_s} \nabla w : \sigma^s(u) d\Gamma - \zeta \int_{\partial\Gamma_h} w \cdot h^s dl = 0 \end{aligned} \quad (3.19)$$

$\forall w \in \mathcal{V}_h^1$ e $q \in \mathcal{P}_h^1$.

Discretização da parede do vaso

Neste trabalho a discretização do tensor σ^s , que representa a tensão da parede, será considerada como em (FIGUEROA et al., 2006). Evidências experimentais demonstram que o comportamento da parede do vaso pode ser considerado linear dentro dos limites de pressões fisiológicas (ZHOU; FUNG, 1997). Para garantir um incremento mínimo de esforço computacional comparando com problemas de parede rígida, é essencial a combinação de um tratamento de cinemática linear e malha fixa. Portanto, um elemento de membrana modificado é utilizado para descrever o forte acoplamento dos graus de liberdade da parede do vaso e da interface do fluido.

Dessa forma, faz-se necessário expressar um tensor adequado apenas para a tensão do sólido $\sigma^s(\vec{u})$, uma vez que o termo de massa da membrana

$$\zeta \int_{\Gamma_s} \vec{w} \cdot \rho^s \frac{\partial \vec{v}}{\partial t} d\Gamma$$

não necessita de um tratamento especial devido o seu forte acoplamento com os graus de liberdade da velocidade na interface.

A malha do sólido é definida por triângulos lineares enquanto que a malha do fluido por elementos tetraédricos. Um triângulo linear é um triângulo plano cujo campo varia linearmente com as coordenadas x e y . Na análise do sólido, o deslocamento linear produz uma tensão constante. Dessa forma, o triângulo linear pode ser chamado de triângulos de tensões constantes (*Constant Strain Triangles* - CST). Este tipo de elemento, aplicado em geometrias tridimensionais, gera um estresse de cisalhamento espúrio, ou seja, exibe uma deformação por cisalhamento transversal, que não deveria existir. Essa tensão indesejável absorve energia, de maneira que se for prescrito um deslocamento, o esforço será maior do que o necessário para produzi-la, aumentando a rigidez e deixando o problema instável (COOK et al., 2001). Dessa forma, será considerada uma rigidez adicional ao elemento do vaso com cargas transversais – detalhes na Fig. 3 – contornando a falta de instabilidade sem adicionar graus de liberdade ao problema. Além disso, a adição de rigidez, considerada a nível local, deve mapear o tensor a nível global.

O tensor da tensão para um material linear elástico é dado por:

$$\sigma^s = C : \epsilon(u), \quad (3.20)$$

onde C representa um tensor de quarta ordem com as propriedades do material e $\epsilon(u)$ é o tensor deslocamento para o sólido, dado por:

$$\epsilon = \frac{1}{2} (\nabla u + (\nabla u)^T). \quad (3.21)$$

A parede da membrana é fina e como a tensão é distribuída de forma uniforme sobre a espessura da parede, a tensão normal na direção da espessura é negligenciada

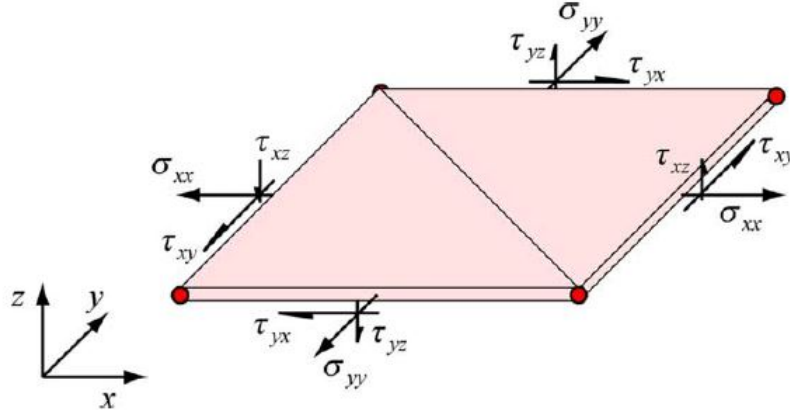


Figura 3 – Esforço na membrana com carga transversal, retirado de (FIGUEROA et al., 2006)

($\sigma_z^l = 0$). A simetria de C , σ^s e ϵ nos permite trabalhar com ordens reduzidas e um tensor de segunda ordem para as propriedades do material. Portanto, os tensores da tensão e deslocamento modificados, na forma reduzida, são dados por:

$$\{\sigma^{s,l}\} = \begin{Bmatrix} \sigma_x^l \\ \sigma_y^l \\ 0 \\ \tau_{xy}^l \\ \tau_{xz}^l \\ \tau_{yz}^l \end{Bmatrix}, \quad \{\epsilon^l\} = \begin{Bmatrix} \frac{\partial u^l}{\partial x^l} \\ \frac{\partial v^l}{\partial y^l} \\ 0 \\ \frac{\partial u^l}{\partial y^l} + \frac{\partial v^l}{\partial x^l} \\ \frac{\partial w^l}{\partial x^l} \\ \frac{\partial w^l}{\partial y^l} \end{Bmatrix}, \quad (3.22)$$

onde u^l , v^l e w^l são as componentes do deslocamento local e, x^l e y^l são as coordenadas locais da membrana.

O tensor, na forma reduzida, das propriedades de um material isotrópico, incompressível utilizando a simetria, é definido por:

$$D = \frac{E}{(1 - (\nu^s)^2)} \begin{bmatrix} 1 & \nu^s & 0 & 0 & 0 & 0 \\ \nu^s & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5(1 - \nu^s) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5k(1 - \nu^s) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5k(1 - \nu^s) \end{bmatrix}, \quad (3.23)$$

onde E e $\nu^s = 0.5$ são o módulo de Young e o coeficiente de Poisson. O parâmetro k foi inserido a fim de descrever uma variação parabólica do esforço de cisalhamento transversal. (HUGHES, 1987) sugere um valor de $k = 5/6$, para uma membrana sólida e homogênea.

Devido a simetria do tensor das tensões, a expressão

$$\zeta \int_{\Gamma_s} \nabla w : \sigma^s(u) d\Gamma$$

pode ser reescrita como

$$\zeta \int_{\Gamma_s} \epsilon(w) : \sigma^s(u) d\Gamma = \zeta \int_{\Gamma_s} \epsilon(w) : C : \epsilon(u) d\Gamma. \quad (3.24)$$

Por fim, é necessário fazer o mapeamento das variáveis locais para as variáveis globais na Eq. (3.22). Para isso é definido um vetor normal local n^l do triângulo planar e outros dois vetores tangente t_1^l e t_2^l , que abrangem todo o plano local. Portanto, o tensor do deslocamento local pode ser mapeado na referência global, conforme em (HELMIG, 2013), por:

$$\{\epsilon(u)\} = \begin{pmatrix} (t_1^l)^T \nabla u t_1^l \\ (t_2^l)^T \nabla u t_2^l \\ 0 \\ (t_1^l)^T \nabla u t_2^l + (t_2^l)^T \nabla u t_1^l \\ (n^l)^T \nabla u t_1^l \\ (n^l)^T \nabla u t_2^l \end{pmatrix} \quad (3.25)$$

Formas de inicialização do deslocamento

A derivada temporal da velocidade será aproximada por diferenças finitas atrasada, Eq. (3.26), enquanto que o deslocamento será determinado de forma explícita – fórmula de Newmark – (Eq. (3.27)), relacionando a velocidade e aceleração do fluido na interface:

$$\frac{\partial v}{\partial t} = \frac{v - v^n}{\Delta t}, \quad (3.26)$$

$$u_{n+1} = u_n + \Delta t v_n + \frac{\Delta t^2}{2} \left((1 - 2\beta) v_{,t_n} + 2\beta v_{,t_{n+1}} \right), \quad (3.27)$$

considerando $\beta = 1.0$.

A inicialização do método CMM-FSI, dar-se-á resolvendo o problema de parede rígida, obtendo a velocidade que será a condição inicial para o problema deformável. Já a inicialização do campo de deslocamento é considerada de duas formas: deslocamento zero e pressão hidrostática.

Os principais passos do **deslocamento zero** são:

- Resolva o problema de parede rígida, dado pela Eq. (3.7) acrescido dos termos estabilizados presentes na Eq. (3.19);
- Inicialize o deslocamento e velocidade na interface como zero;

- Resolva o problema deformável, dado pela Eq. (3.19).

Os principais passos da **pressão hidrostática** estão descritos a seguir.

A ideia básica é requerer que a tensão do fluido na parede se equilibre com a do sólido na interface, começando a partir de um ponto de equilíbrio (FIGUEROA et al., 2006). Dessa forma, a parede deformável é carregada com a pressão média na interface, proveniente do problema de parede rígida. Caracterizando-se, portanto, como um caso hidrostático, uma vez que a pressão na parede inicialmente é estritamente devido a pressão estática ($v = \partial v / \partial t = \tau = 0$). Dessa forma, a Eq. (3.18) pode ser reescrita por:

$$\zeta \int_{\Gamma_s} \nabla w : \sigma^s d\Gamma = \int_{\Gamma_h} w \mathcal{P} \cdot n d\Gamma \quad (3.28)$$

onde \mathcal{P} é a pressão média na interface Γ_s . Com isso a inicialização do algoritmo pode ser resumida por:

- Resolva o problema de parede rígida, dado pela Eq. (3.7) acrescido dos termos estabilizados presentes na Eq. (3.19);
- Inicialize a velocidade na interface como zero, solucione Eq. (3.28) e inicialize o deslocamento;
- Resolva o problema deformável, dado pela Eq. (3.19).

4 FEniCS

Neste capítulo é introduzida a biblioteca FEniCS. Os códigos para as equações de Navier-Stokes são apresentados e detalhados para os métodos de projeção e formulações estabilizadas. Além disso, são apresentadas as implementações para os problemas de interação fluido-estrutura utilizando o método CMM-FSI, sendo que todos os códigos apresentados utilizam a linguagem Python.

A FEniCS é uma biblioteca que contém vários componentes de softwares livres e códigos abertos com a ideia de possibilitar a solução automatizada de equações diferenciais com base no método dos elementos finitos. Os códigos FEniCS utilizam como principal componente a interface DOLFIN. A DOLFIN oferece um ambiente de interface entre o usuário e a solução para as equações diferenciais, e implementa as funcionalidades da FEniCS como: montagem das matrizes de elementos finitos, estrutura de dados e algoritmos para malhas computacionais. A fim de possibilitar uma simples implementação, a FEniCS envolve comunicações entre componentes de mais baixo nível. Os códigos podem ser implementados em C++ ou Python, chamando no cabeçalho do programa a biblioteca DOLFIN, contendo classes de malhas, funções, espaços variacionais, matrizes, vetores, etc., que modelam todas as características necessárias para o método de elementos finitos. Além disso, dentre suas funcionalidades destacamos sua comunicação com bibliotecas externas como por exemplo: álgebra linear (PETSc, Trilinos); particionamento de malhas (SCOTCH, ParMETIS); processamento em paralelo (MPI, OpenMP) (LOGG; MARDAL; WELLS, 2012). A Fig. 4 mostra algumas das principais funções presentes na biblioteca.

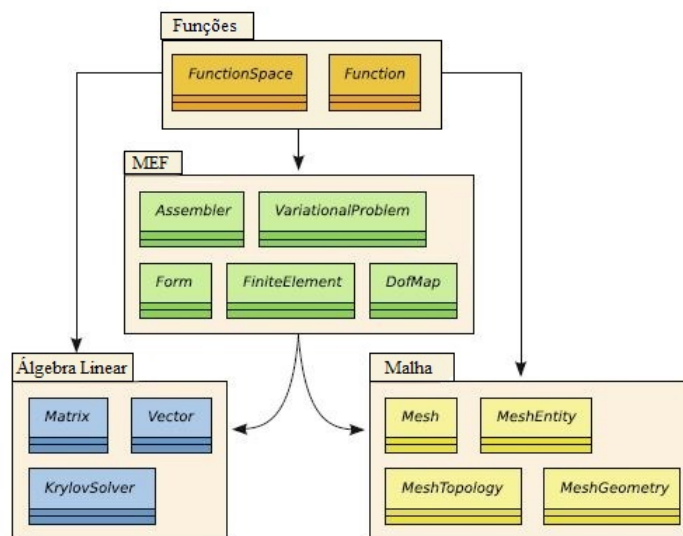


Figura 4 – Alguns dos principais componentes e classes da DOLFIN, adaptado de (LOGG; MARDAL; WELLS, 2012)

A fim de exemplificar a simplicidade dos códigos FEniCS, são apresentadas as principais partes das implementações para as equações de Navier-Stokes e para os problemas de interação fluido-estrutura.

4.1 Implementação das Equações de Navier-Stokes

Inicialmente é explicitado uma classe padrão para todos os problemas implementados, no qual é importado o componente DOLFIN, e é realizado o cálculo do parâmetro de tempo. Em seguida, o problema do escoamento em uma cavidade bidimensional é descrito. Adiante, é demonstrado a geração de uma malha tridimensional, a importação de malha, conversão de tipos de malhas e, também, condições de contorno com expressões. Na sequência é implementado os tensores para o fluido, implementação do método IPCS, formulação estabilizada e a implementação com a técnica adaptativa no tempo.

A classe padrão para todos os experimentos é apresentada no Cód. 4.1. Na linha 1 é importada a biblioteca DOLFIN e na linha 12 é determinado o tamanho do passo de tempo, com base na velocidade máxima do problema (linha 9) e o parâmetro característico de malha (linha 10), e na sequência, o número de passos de tempo é calculado. Além disso, é gerado um vetor contendo os tempos na linha 16. Esta classe retorna o tamanho do passo de tempo e o vetor contendo os tempos, descrito na linha 17.

Código 4.1 – Classe padrão para todos os experimentos

```

1 from dolfin import *
2 import numpy as np
3 class ProblemBase:
4     "Classe padrao para todos os problemas"
5     def timestep(self, problem):
6         "parametros para tempo"
7         #Dados para calcular o passo de tempo
8         T = problem.T #Tempo final
9         Vmax = problem.Vmax #velocidade maxima
10        h = self.mesh.hmin()
11        #Computando o passo de tempo
12        dt = 0.2*(h/Vmax)
13        n = int(T/dt + 1.0)
14        dt = T/n
15        #Vetor contendo os tempos (comecando em t>0)
16        t_range = np.linspace(0,T,n+1)[1:]
17        return dt, t_range

```

Após importar a biblioteca, as informações relacionadas ao problema devem ser informadas. Para exemplificar, descrevemos a implementação do problema de escoamento em uma cavidade bidimensional com tampa deslizante. No Cód. 4.2 é gerada a malha

unitária, estruturada com N divisões na direção longitudinal e transversal. A força de corpo é um vetor com valor nulo, conforme é descrito no próximo comando e, logo abaixo, são definidos o número de Reynolds, velocidade máxima do problema e o tempo final. Nas linhas 15 e 16 são definidas as condições iniciais para a velocidade e pressão, respectivamente.

Adiante, são impostas as condições de contorno. Na linha 21 é definido o valor para a velocidade na tampa deslizante, com valor unitário na direção longitudinal (x) e zero na transversal (y), aplicado no campo $b1$ representando uma marcação booleana para $y = 1$ (linha 25). De forma similar, para as paredes laterais da cavidade e a parte inferior é imposta a condição de velocidade zero ($g2$) (linhas 22 e 26). Na origem do sistema de coordenadas é definido um valor zero para a pressão ($g3$), conforme consta na linha 23. Assim sendo, as condições de contorno do tipo Dirichlet são inseridas, conforme as linhas 28, 29 e 30. Para isso, deve ser informado qual o espaço variacional, o valor e a parte do contorno que será imposta a condição do tipo Dirichlet. O espaço variacional $W.sub(0)$ e $W.sub(1)$ representam, respectivamente, o espaço variacional para a velocidade e para a pressão.

No final do código é realizada a marcação booleana do contorno, onde "Tampa" é localizada em $y = 1$ e "Walls" são as paredes laterais e a parte inferior da cavidade, além da marcação "zero" localizada na origem do sistema de coordenadas.

Código 4.2 – Escoamento em uma cavidade bidimensional com tampa deslizante

```

1  #Gerando a malha
2  N = options['N']
3  self.mesh = UnitSquareMesh(N, N)
4  #Força de corpo
5  self.f = Constant( (0.0, 0.0) )
6  #Viscosidade
7  Re = 1000.0
8  self.nu = 1.0/Re
9  #Maxima velocidade
10 self.Vmax = 1.0
11 #Tempo final
12 self.T = 2.5
13
14 def condicao_inicial(self, W):
15     v0 = Constant( (0.0, 0.0) )
16     p0 = Constant(0.0)
17     return v0, p0
18
19 def condicao_contorno(self, W, t):
20     #Valores
21     self.g1 = Constant( (1.0, 0.0) )
22     self.g2 = Constant( (0.0, 0.0) )
23     self.g3 = Constant(0.0)

```

```

24     #Regiao
25     self.b1 = Tampa()
26     self.b2 = Walls()
27     #Condicao
28     bc1 = DirichletBC(W.sub(0), self.g1, self.b1)
29     bc2 = DirichletBC(W.sub(0), self.g2, self.b2)
30     bc3 = DirichletBC(W.sub(1), self.g3, zero, 'pointwise')
31
32     bcs = [bc1, bc2, bc3]
33     return bcs
34
35 "Marcando o contorno"
36 class Tampa(SubDomain):
37     def inside(self, x, on_boundary):
38         return (on_boundary and near(x[1], 1.0))
39
40 class Walls(SubDomain):
41     def inside(self, x, on_boundary):
42         return on_boundary and (near(x[0], 0.0) or near(x[0], 1.0) or near(x
43         [1], 0.0))
44
45 def zero(x):
46     return (near(x[0], 0.0) and near(x[1], 0.0))

```

O Cód. 4.3 mostra um conjunto de comandos relacionados a malha. Na segunda linha é gerada uma malha tridimensional, estruturada, com N divisões nas três direções (x , y e z). Além de gerar as malhas, a FEniCS possibilita importar malhas no formato .xml, conforme consta na quarta linha ou, ainda, converter outros tipos de formatos de malha em formato .xml, como por exemplo, o comando dado na linha 6 converte uma malha .msh em .xml. Além disso, condições de contorno podem estar relacionada com as coordenadas da malha, tempo, funções senoidais, exponenciais, entre outras. O valor g_1 na linha 24 representa uma expressão vetorial sendo dependente da coordenada $y = x[1]$ e também de uma função exponencial do tempo.

Código 4.3 – Exemplo de malha tridimensional e outros tipos de condições de contorno

```

1     #Gerando a malha tridimensional
2     self.mesh2 = UnitCubeMesh(N, N, N)
3     #Importando a malha
4     self.mesh1 = Mesh("malha.xml")
5     #Convertendo a malha
6     dolfin-convert geometry.msh geometry.xml
7
8     def condicao_contorno(self, W, t):
9         #Valores
10        self.g1=Expression('(x[1]*(ymax-x[1]))*exp(-t)',ymax=0.41,t=t,degree=2)

```

O Cód. 4.4 representa o tensor deformação (Eq. (2.7)) e o tensor das tensões de Cauchy (Eq. (2.6)), onde $Identity(len(v))$ é o tensor identidade com dimensões $n \times n$, sendo n o tamanho da dimensão espacial.

Código 4.4 – Tensor das tensões e tensor deformação

```

1 def epsilon(v):
2     "Tensor das deformacoes"
3     return sym(nabla_grad(v))
4
5 def sigma(v, p, nu):
6     "Tensor das tensoes de Cauchy"
7     return 2*nu*epsilon(v) - p*Identity(len(v))

```

Após gerar a malha, definir os parâmetros utilizados nos problemas e as condições de contorno é definido o método. O método IPCS, Chorin e CSS possuem implementações semelhantes, cada um com suas equações específicas. Portanto, é apresentado apenas o método IPCS no Cód. 4.5. Inicialmente é definido o espaço variacional para a velocidade e para a pressão nas linhas 2 e 3, definindo, respectivamente, elementos Lagrangeanos com polinômios quadráticos para a velocidade e linear para a pressão. Na sequência, são explicitadas as funções triais, testes e as funções incógnitas para o problema. Note que primeiro é definido a variável v como uma *TrialFunction*, utilizada para representar a formulação variacional e, posteriormente, definimos a variável $v1$ (incógnitas) como uma *Function* que representa a solução para a velocidade, ou seja, a função calculada pela interpolação do método dos elementos finitos.

Adiante, a formulação variacional é implementada. As linhas de 18 a 20 representam a estimativa da velocidade, dada pela Eq. (2.20). Já $F2$ representa a determinação da pressão (Eq. (2.21)) e a linha 27 é a correção da velocidade, dada pela Eq. (2.22). Em seguida é tomada a forma bilinear $a = a(v, w)$ e a forma linear $L = L(w)$, fazendo a montagem das matrizes nas linhas 34-36.

Na sequência é iniciado o loop temporal (linha 39). Caso as condições de contorno dependam do tempo, faz-se necessário atualiza-las no início do laço temporal, conforme descrito na linha 41. A linha 44 faz a montagem do vetor dos termos independentes para a estimativa da velocidade e, posteriormente, é aplicado as condições de contorno. Em seguida, a linha 46 soluciona o problema variacional para a estimativa da velocidade, com o método iterativo GMRES e preconditionador ILU(0). A mesma ideia é usada nas linhas 49 até 56. Solucionado o problema, velocidade e pressão do passo de tempo anterior são atualizadas nas linhas 60 e 61.

Código 4.5 – Implementação do método IPCS

```

1 #Espacos variacionais
2 V = VectorFunctionSpace(mesh, "P", 2)

```

```

3 P = FunctionSpace(mesh, "P", 1)
4 #Definindo as funcoes triais e testes
5 v = TrialFunction(V)
6 p = TrialFunction(P)
7 w = TestFunction(V)
8 q = TestFunction(P)
9 #Definindo as funcoes
10 v1 = Function(V)
11 p1 = Function(P)
12 vn = interpolate(v0, V)
13 pn = interpolate(p0, P)
14 #Velocidade  $u^{(n-1/2)}$ 
15 U = 0.5*(vn + v)
16
17 #Problema variacional para a tentativa de velocidade
18 F1 = (1.0/dt)*inner(v - vn,w)*dx + inner(dot(vn,nabla_grad(U)),w)*dx\
19     +inner(sigma(U, pn, nu), epsilon(w))*dx - inner(f, w)*dx \
20     +inner(pn*n, w)*ds - nu*inner(w, grad(U).T*n)*ds
21
22 #Problema variacional para o problema da pressao
23 F2 = inner(nabla_grad(p - pn), nabla_grad(q))*dx \
24     + (1.0/dt)*nabla_div(v1)*q*dx
25
26 #Problema variacional para a atualizacao da velocidade
27 F3 = inner(v - v1, w)*dx + dt*inner(nabla_grad(p1 - pn), w)*dx
28
29 #Formas linear e bilinear
30 a1, L1 = system(F1)
31 a2, L2 = system(F2)
32 a3, L3 = system(F3)
33 #Montando as matrizes
34 A1 = assemble(a1)
35 A2 = assemble(a2)
36 A3 = assemble(a3)
37
38 #Laco para o tempo
39 for t in t_range:
40     #Condicoes de contorno para o problema atualizar no tempo
41     bcv, bcp = problem.condicao_contorno(V, P, t)
42
43     #Estimativa da velocidade
44     b = assemble(L1)
45     [bc.apply(A1, b) for bc in bcv]
46     solve(A1, v1.vector(), b, 'gmres', 'ilu')
47
48     #Determinando a pressao
49     b = assemble(L2)

```

```

50     [bc.apply(A2, b) for bc in bcp]
51     solve(A2, p1.vector(), b, 'gmres', 'hybre_amg')
52
53     #Corrigindo a velocidade
54     b = assemble(L3)
55     [bc.apply(A3, b) for bc in bcv]
56     solve(A3, v1.vector(), b, 'gmres', 'ilu')
57
58     #Atualizando o problema
59     self.update(problem, t, v1, p1)
60     vn.assign(v1)
61     pn.assign(p1)
62
63     return v1, p1

```

O Cód. 4.6 fornece a implementação para a formulação estabilizada SUPG/PSPG + LSIC. Inicialmente é definido o espaço variacional para a velocidade e pressão nas linhas 2 e 3, respectivamente, definindo elementos Lagrangeanos com polinômios lineares. Diferente dos métodos acima mencionados, a formulação estabilizada trata pressão e velocidade de forma acopladas, resolvendo essas variáveis simultaneamente sendo, portanto, necessário definir um espaço misto, conforme descrito na quarta linha. Adiante, são definidas as funções triais, testes e as funções incógnitas, onde vp e $vp1$ são os vetores contendo informações referentes a velocidade e pressão. O comando **split**, presente na linha 7, separa velocidade e pressão.

Em seguida, a formulação variacional de Galerkin, dada pela Eq. (2.11), é definida nas linhas 15-17. O termo não-linear é linearizado e a variável atrasada é $v1$. Nas linhas a seguir, é implementada a formulação estabilizada. Para isso é definido o parâmetro de malha e a norma da velocidade nas linhas 20 e 21, respectivamente. O resíduo da equação do momento das equações de Navier-Stokes, dado pela Eq. (2.13), é definido pela linha 23. Nas linhas 26, 29 e 32 são implementadas as formulações estabilizadas, presentes na Eq. (2.14) e nas linhas 25, 28 e 31 os parâmetros de estabilização, dados pela Eq. (2.15). Diante disso, as formulações estabilizadas são acrescentadas na formulação padrão de Galerkin na linha 34. Na sequência, é computado o jacobiano, definido o método não-linear, linear, preconditionador e os parâmetros utilizados, como: tolerância absoluta e relativa, número máximo de iterações, dentre outros.

Seguidamente, é iniciado o loop temporal na linha 57. A linha 59 faz a atualização das condições de contorno, caso necessário. Em seguida, o problema é solucionado, onde é informado a formulação variacional (F1), a variável de solução contendo velocidade e pressão acopladas (vp1), as condições de contorno (bcs), o jacobiano (J) e os parâmetros para os métodos iterativos (par). Na linha 63 pressão e velocidade são separadas e, por fim, a velocidade do passo de tempo anterior é atualizada.

Código 4.6 – Implementação da formulação estabilizada

```

1  #Funcoes de espaco
2  V = VectorFunctionSpace(mesh, "P", 1)
3  P = FunctionSpace(mesh, "P", 1)
4  W = MixedFunctionSpace([V, P])
5  #Definindo as funcoes triais e testes
6  vp = TrialFunction(W)
7  v, p = ufl.split(vp)
8  w, q = TestFunctions(W)
9  #Definindo as funcoes
10 vp1 = Function(W)
11 v1, p1 = vp1.split()
12 vn = interpolate(v0, V)
13
14 #Formulacao padrao de Galerkin
15 F=(1.0/dt)*inner(v - vn, w)*dx + inner(dot(v1, nabla_grad(v)), w)*dx\
16     + inner(sigma(v,p,nu), epsilon(w))*dx-q*nabla_div(v)*dx-inner(f,w)*dx\
17     + inner(p*n, w)*ds - nu*inner(w, grad(v).T*n)*ds
18
19 #Parametros para a estabilizacao
20 h = CellSize(mesh)
21 vnorm = sqrt(dot(v1, v1))
22 #Residuo do momento
23 R = (1.0/dt)*(v-v_n)+dot(v1, nabla_grad(v))-div(sigma(v,p,nu))-f
24 #PSPG
25 tau_pspg=((2.0/dt)**2+(2.0*vnorm/h)**2+9.0*(4.0*nu/(h**2))**2)**(-0.5)
26 F_pspg = tau_pspg*inner(R, nabla_grad(q))*dx
27 #SUPG
28 tau_supg=((2.0/dt)**2+(2.0*vnorm/h)**2+9.0*(4.0*nu/(h**2))**2)**(-0.5)
29 F_supg = tau_supg*inner(R, dot(v1, nabla_grad(w)))*dx
30 #LSIC
31 tau_lsic = (vnorm*h)/2.0
32 F_lsic = tau_lsic*inner(nabla_div(v), nabla_div(w))*dx
33 #Adicionando os termos estabilizadores
34 F += -F_pspg + F_supg + F_lsic
35
36 #Jacobiano
37 F1 = action(F, vp1)
38 J = derivative(F1, vp1, vp)
39
40 #Parametros para solucao
41 tol_r = "relative_tolerance"
42 tol_a = "absolute_tolerance"
43 ln = "linear_solver"
44 ns = "newton_solver"
45 prec = "preconditioner"
46 ks = "krylov_solver"

```

```

47     mi    = "maximum_iterations"
48     enon  = "error_on_nonconvergence"
49
50     linear = {tolr: 1.0E-8, tola: 1.0E-8, mi: 10000, enon:False,
51               prec:{ "ilu":{ "fill_level":0}}}
52     nonlinear = {tolr: 1.0E-4, tola: 1.0E-4, mi: 20,
53                  ln:"gmres",prec:"ilu",enon:False,ks:linear}
54     par = {ns:nonlinear}
55
56     #Laco para o tempo
57     for t in t_range:
58         #Condicoes de contorno para o problema atualizar no tempo
59         bcs = problema.condicao_contorno(W, t)
60         #Resolvendo o problema
61         solve(F1 == 0, vp1, bcs=bcs, J=J, solver_parameters = par)
62
63         v1, p1 = vp1.split(True)
64
65         #Atualizando o problema
66         self.update(problema, t, v1, p1)
67         vn.assign(v1)
68
69     return v1, p1

```

As implementações dos métodos com a técnica adaptativa no tempo são semelhantes, sendo apresentada apenas a formulação estabilizada com o controlador PID. As definições feitas antes de iniciar o laço temporal no Cód. 4.6 são análogas as presentes no Cód. 4.7. Porém, para a implementação com o controlador é necessário definir duas funções a mais: $v2n$ e $v3n$, que são, respectivamente, a velocidade em dois e três passos de tempo anteriores. As linhas de 2-6 são alguns parâmetros utilizados na implementação com a técnica adaptativa PID, onde $tolpid$ é a tolerância presente na Eq. (2.37), $dt1$ é o novo passo de tempo, k é o passo de tempo atual e PID é o fator multiplicador dado na Eq.(2.37).

A parte inicial do laço temporal (linhas 11-15) são idênticas as implementadas no Cód. 4.6. Nas linhas 17-19 é determinado a diferença entre as velocidades e na sequência, linhas 21-13, é calculada a norma absoluta dessas velocidades. Posteriormente, linhas 25-27, é determinada a norma das velocidades. A norma relativa da velocidade é obtida pela razão entre a norma absoluta e a norma da velocidade. Nos dois primeiros passos de tempo, as normas relativas $erv2$ e $erv3$ valem zero, portanto o controlador começa a atuar a partir do terceiro laço no tempo. Assim sendo, quando o processo estiver em $iter \geq 2$ é definido os parâmetros do controlador (linhas 39-41) e determinado o fator multiplicador PID pela Eq. (2.37). Na sequência, o problema é atualizado, linhas 47-50, o novo passo de tempo é calculado pela linha 52, em seguida, é atualizado o tamanho do passo de tempo presente na formulação variacional (linha 53) e o tempo corrente na linha 54.

Código 4.7 – Implementação da formulação estabilizada com controlador PID

```

1 #Parametros para o controlador PID
2 tolpid = 1.0E-2
3 dt1 = k
4 t = k
5 PID = 1.0
6 iter = 0
7
8 #Laco para o tempo
9 while t <= T:
10     #Condicoes de contorno para o problema atualizar no tempo
11     bcs = problema.condicao_contorno(W, t)
12     #Resolvendo o problema
13     solve(F1 == 0, vp1, bcs=bcs, J=J, solver_parameters = par)
14
15     v1, p1 = vp1.split(True)
16     #Norma absoluta
17     diffv1 = v1.vector().array() - vn.vector().array()
18     diffv2 = vn.vector().array() - v2n.vector().array()
19     diffv3 = v2n.vector().array() - v3n.vector().array()
20
21     epsv1 = np.linalg.norm(diffv1, ord=2)
22     epsv2 = np.linalg.norm(diffv2, ord=2)
23     epsv3 = np.linalg.norm(diffv3, ord=2)
24     #Norma euclidiana
25     env1 = norm(v1, 'l2')
26     env2 = norm(vn, 'l2')
27     env3 = norm(v2n, 'l2')
28     #Norma relativa
29     erv1 = epsv1/env1
30     #Controlador PID
31     if (iter < 2):
32         PID = 1.0
33
34     else:
35
36         erv2 = epsv2/env2
37         erv3 = epsv3/env3
38         #Parametros para o controlador
39         KP = 0.075
40         KI = 0.175
41         KD = 0.01
42         #Calculando o fator PID
43         PID=((erv2/erv1)**KP)*((tol_pid/erv1)**KI)*((erv2*erv2/(erv1*erv3))**KD
44         )
45     #Atualizando o problema

```

```

46 self.update(problema, t, v1, p1)
47 v3n.assign(v2n)
48 v2n.assign(vn)
49 vn.assign(v1)
50 pn.assign(p1)
51
52 dt1 = PID*k #Novo passo de tempo
53 dt.k = dt1 #Atualizando o passo de tempo
54 t += dt1 #Atualizando o tempo corrente
55 iter += 1

```

4.2 Implementação do Problema IFE

Os problemas de interação fluido-estrutura aqui apresentados utilizam o método CMM-FSI. Nesta abordagem o problema de parede rígida é solucionado inicialmente, descrito pela Eq. (3.7) sem o termo dado pela Eq. (3.6), em seguida, o problema deformável (Eq. (3.19)) é inicializado com a solução proveniente do problema de parede rígida. Portanto, é apresentada apenas a implementação do problema deformável.

O problema de interação fluido-estrutura apresentado representa uma artéria com condição de entrada parabólica. Primeiramente, no Cód. 4.8 é realizada a geração ou importação da malha e as marcações do contorno. Adiante, no Cód. 4.9 são definidos os parâmetros para o fluido e sólido, espaços variacionais, condições de contorno e funções utilizadas no problema. Posteriormente, são explicitados os tensores do fluido e os tensores modificados do sólido, no Cód. 4.10. O Cód. 4.11 trata do cálculo do deslocamento e as formas de inicialização do deslocamento. Na sequência o problema variacional é definido e, posteriormente, é solucionado o problema acoplado.

A malha representa um cilindro com raio 0.3 e comprimento 4.2, podendo ser gerada ou importada, conforme descrito nas linhas 7-11 do Cód. 4.8 e, posteriormente, é realizada a marcação booleana do contorno. Nas linhas 32-38 são feitas marcações nas partes do contorno.

Código 4.8 – Gerando a malha e marcação do contorno

```

1 #Gerando ou importando a malha
2 P0 = Point(0.0, 0.0, 0.0)
3 P1 = Point(4.2, 0.0, 0.0)
4 ri = 0.3
5 rf = 0.3
6 res = 80
7 #Importando a malha
8 try:
9     mesh = Mesh("Arteria/malhas/arteria_%d.xml" % res)

```

```

10 #Gerando a malha
11 except:
12     domain = Cylinder(P0, P1, ri, rf, 100)
13     mesh = generate_mesh(domain, res)
14 #Marcando o contorno
15 class Inlet(SubDomain):
16     def inside(self, x, on_boundary):
17         return on_boundary and near(x[0], 0.0)
18
19 class Outlet(SubDomain):
20     def inside(self, x, on_boundary):
21         return on_boundary and near(x[0], 4.2)
22
23 class Interface(SubDomain):
24     def inside(self, x, on_boundary):
25         dy = x[1]
26         dz = x[2]
27         dr = sqrt(dy*dy + dz*dz)
28         bmarg = DOLFIN_EPS + 1.0E-3
29         return on_boundary and (dr > ri - bmarg) and between(x[0], (0.2, 4.0))
30
31 #Definindo valores para as marcacoes
32 parts = FacetFunction('size_t', mesh)
33 parts.set_all(0)
34 DomainBoundary().mark(parts, 3)
35 Inlet().mark(parts, 1)
36 Outlet().mark(parts, 2)
37 Interface().mark(parts, 4)
38 ds = Measure("ds", domain=mesh, subdomain_data=parts)

```

No Cód. 4.9 são fornecidos os parâmetros do fluido e do sólido, definido os espaços variacionais, e as funções utilizadas. Esta parte é semelhante ao problema rígido. Porém, a condição de Dirichlet na interface é retirada (linha 26) e, também, são definidas as funções para o deslocamento, descritas na linhas 35, 39 e 43.

Código 4.9 – Propriedades; espaços variacionais e funções utilizadas

```

1 #Parametros para o fluido
2 mu_f = 0.04
3 rho_f = 1.06
4 nu_f = Constant(mu_f/rho_f)
5 f = Constant((0.0, 0.0, 0.0))
6 #Parametros para o solido
7 nu_s = 0.5
8 rho_s = 1.09
9 E = 4.07E6
10 qsi = 0.03

```

```

11 ks      = 5.0/6.0
12 beta    = 1.0
13
14 #Espacios variacionais
15 V = VectorFunctionSpace(mesh, 'P', 1)
16 P = FunctionSpace(mesh, 'P', 1)
17 W = MixedFunctionSpace([V, P])
18 #Condicoes de contorno
19 vx = 'Vm*((r*r-x[1]*x[1]-x[2]*x[2])/(r*r))'
20 vin  = Expression((vx, '0.0', '0.0'), r=ri, Vm=12.5, degree=3)
21 vwalls = Constant((0.0, 0.0, 0.0))
22 pout   = Constant(1330.0)
23
24 bcvin      = DirichletBC(W.sub(0), vin, parts, 1)
25 bcvwalls   = DirichletBC(W.sub(0), vwalls, parts, 3)
26 bcvinterface = DirichletBC(W.sub(0), vwalls, parts, 4)
27 bcpout     = DirichletBC(W.sub(1), pout, parts, 2)
28
29 bcs = [bcvin, bcvwalls, bcpout]
30
31 #Funcoes para o problema
32 w, q  = TestFunctions(W)
33 vp    = TrialFunction(W)
34 v, p  = ufl.split(vp)
35 u     = TrialFunction(V)
36
37 vp_    = Function(W)
38 v_, p_ = vp_.split()
39 u_     = Function(V)
40
41 v_n    = Function(V)
42 v_2n   = Function(V)
43 u_n    = Function(V)

```

Os tensores das tensões e deformação do fluido são os mesmos descritos anteriormente. O modelo de membrana utilizado para o sólido é modificado, necessitando de modificações nos tensores da tensão e deslocamento. Portanto, é necessário realizar o mapeamento a nível global desses tensores. O Cód. 4.10 representa as modificações realizadas nos tensores e o seu respectivo mapeamento no nível global. A linha 136 é a implementação da Eq. (3.25) e a linha 155 é a implementação do tensor da tensão presente na Eq. (3.22).

Código 4.10 – Tensores para o fluido e sólido

```

1 def epsilon_f(v):
2     return sym(nabla_grad(v))
3

```

```

4 def sigma_f(v, p, mu_f):
5     return 2*mu_f*epsilon_f(v) - p*Identity(len(v))
6
7 #vetores tangentes e normal modificados
8 n = FacetNormal(mesh)
9 norm_n = (n[0]*n[0] + n[1]*n[1] + n[2]*n[2])** (1/2.0)
10 n1 = 1/norm_n*n
11
12 #Primeiro vetor tangente
13 t1 = as_vector((n1[1] - n1[2], n1[2] - n1[0], n1[0] - n1[1]))
14 norm_t1 = (t1[0]*t1[0] + t1[1]*t1[1] + t1[2]*t1[2])** (1/2.0)
15 t1 = 1/norm_t1*t1
16
17 #segundo vetor tangente
18 t2x = n1[1]*t1[2] - n1[2]*t1[1]
19 t2y = n1[2]*t1[0] - n1[0]*t1[2]
20 t2z = n1[0]*t1[1] - n1[1]*t1[0]
21
22 t2 = as_vector((t2x, t2y, t2z))
23 norm_t2 = (t2[0]*t2[0] + t2[1]*t2[1] + t2[2]*t2[2])** (1/2.0)
24 t2 = 1/norm_t2*t2
25
26 #Tensor do deslocamento e da tensao para o solido
27 def epsilon_s(u):
28     e=grad(u)
29
30     exx = dot(dot(t1, e), t1)
31     eyy = dot(dot(t2, e), t2)
32     ezz = 0.0*dot(dot(n1, e), n1)
33     txy = dot(dot(t1, e), t2) + dot(dot(t2, e), t1)
34     txz = dot(dot(n1, e), t1)
35     tyz = dot(dot(n1, e), t2)
36
37     ep = as_vector([ exx, \
38                     eyy, \
39                     ezz, \
40                     txy, \
41                     txz, \
42                     tyz ])
43
44     return ep
45
46 def sigma_s(u):
47     C_coef = E/(1.0 - nu_s*nu_s)
48
49     ttzz = 0.5*ks*(1.0-nu_s)
50     ttxy = 0.5*(1.0-nu_s)

```

```

51
52 C = as_matrix([[1.0 ,nu_s, 0.0, 0.0 , 0.0 , 0.0 ],
53                [nu_s,1.0 , 0.0, 0.0 , 0.0 , 0.0 ],
54                [ 0.0,0.0 , 0.0, 0.0 , 0.0 , 0.0 ],
55                [ 0.0,0.0 , 0.0, ttxy , 0.0 , 0.0 ],
56                [ 0.0,0.0 , 0.0, 0.0 , ttzz , 0.0 ],
57                [ 0.0,0.0 , 0.0, 0.0 , 0.0 , ttzz ]])
58
59 C = C*C_coef
60
61 stress = dot(C,epsilon_s(u))
62
63 return stress

```

O deslocamento é determinado de forma explícita pela fórmula de Newmark (Eq. (3.27)), sendo implementada na segunda linha do Cód. 4.11. Nas linhas 10 e 11 o usuário escolhe as formas de inicialização do deslocamento: pressão hidrostática ou deslocamento zero. Na sequência, linhas 19-23, é importada a solução proveniente do problema de parede rígida. Em ambos os casos de inicialização a velocidade é inicializada com a solução do problema rígido. Caso a forma de inicialização do deslocamento seja pressão hidrostática, a Eq. (3.28) deve ser resolvida. Porém, não é possível resolver esta equação na FEniCS, sendo necessário modifica-la, conforme descrito nas linhas 48 e 49. Assim sendo, o deslocamento é inicializado.

Código 4.11 – Deslocamento e formas de inicialização

```

1 #Deslocamento explícito
2 def disp(u_n, v, v_n, v_2n, dt):
3
4     dd1 = dt*v_n+(dt**2/2.0)
5     dd2 = ((1.0-2.0*beta)*((v_n-v_2n)/dt)+2.0*beta*((v-v_n)/dt))
6
7     return u_n + dd1*dd2
8
9 #Formas de inicializacao do deslocamento
10 INIT = {"zero_displacement": False,
11         "hydrostatic_pressure": True}
12
13 init = INIT.copy()
14
15 #Inicializando o deslocamento
16 vr, pr = Function(V), Function(P)
17
18 #Importanto a solucao
19 input_file=HDF5File(mesh.mpi_comm(), 'velocity.h5', 'r')
20 input_file.read(vr, 'velocity')

```



```

21
22 input_file = HDF5File(mesh.mpi_comm(), 'pressure.h5', 'r')
23 input_file.read(pr, 'pressure')
24
25 #Deslocamento zero
26 if init["zero_displacement"] and not init["hydrostatic_pressure"]:
27
28     v_n.assign(vr)
29     v_2n.assign(vr)
30     p_n.assign(pr)
31
32     name = "zero_displacement"
33
34 #Pressao hidrostática
35 if init["hydrostatic_pressure"] and not init["zero_displacement"]:
36
37     bc1 = DirichletBC(V, vwalls, parts, 1)
38     bc2 = DirichletBC(V, vwalls, parts, 2)
39     bc3 = DirichletBC(V, vwalls, parts, 3)
40     bcu=[bc1, bc2, bc3]
41
42 #Pressao media na interface
43 pavg = (assemble(pr*ds(4)))/(assemble(Constant(1.0)*ds(4)))
44 w1 = TestFunction(V)
45 u1 = TrialFunction(V)
46 us = Function(V)
47
48 Fhp=1E-5*inner(nabla_grad(u1),nabla_grad(w1))*dx-inner(pavg*n, w1)*ds(4)\
49     + qsi*inner(sigma_s(u1), epsilon_s(w1))*ds(4)
50
51 ahp, Lhp = system(Fhp)
52 Ahp, bhp = assemble(ahp), assemble(Lhp)
53 [bc.apply(Ahp, bhp) for bc in bcu]
54 solve(Ahp, us.vector(), bhp, 'gmres', 'ilu')
55
56 v_n.assign(vr)
57 v_2n.assign(vr)
58 p_n.assign(pr)
59 u_n.assign(us)
60
61 name = "hydrostatic_pressure"

```

As formulações variacionais para esses tipos de problemas são análogas aos casos rígidos, porém a condição de Dirichlet na interface é retirada e são inseridas as equações do sólido na formulação variacional do fluido como uma condição de contorno. Portanto, as linhas 2 e 3 do Cód. 4.12 acrescentam as equações do sólido na formulação variacional

do fluido, dado pela Eq. (3.19).

Código 4.12 – Forma variacional acoplada

```

1 #Acoplando o problema
2 F += qsi*(rho_s/dt)*inner(v - v_n, w)*ds(4) \
3   +qsi*inner(sigma_s(dispatch(u_n, v, v_n, v_2n, dt)), epsilon_s(w))*ds(4)

```

As equações acrescentadas na formulação variacional do fluido não alteram as formulações estabilizadas, sendo idênticas ao caso rígido. Definida a formulação e os demais parâmetros acima mencionados, o laço no tempo é iniciado na primeira linha do Cód. (4.13). Caso as condições de contorno sejam em função do tempo, é necessário atualizá-las antes da quarta linha. Solucionado o problema e separando pressão e velocidade, é determinado o deslocamento de forma explícita, conforme a linha 10. Como o deslocamento é calculado por uma expressão, é indispensável projetá-lo sobre o espaço variacional V , conforme descrito na linha 11.

Código 4.13 – Solução do deslocamento

```

1 for t in t_range:
2
3     #solve
4     solve(F_ == 0, vp_, bcs=bcs, J=J_, solver_parameters = par)
5
6     #split
7     v_, p_ = vp_.split(True)
8
9     #Resolvendo a deformacao
10    u_ = disp(u_n, v_, v_n, v_2n, dt)
11    u_ = project(u_, V)

```

Após solucionar o problema variacional, dado pela Eq. (3.19), é necessário atualizar as variáveis do passo de tempo anterior, conforme descrito nas linhas 2 até 5 do Cód. 4.14

Código 4.14 – Atualização das variáveis

```

1 #Atualizando o problema
2 v_2n.assign(v_n)
3 v_n.assign(v_)
4 p_n.assign(p_)
5 u_n.assign(u_)

```

5 Experimentos Numéricos

Nesta capítulo são apresentados os principais experimentos numéricos realizados. Primeiramente, apresentamos uma comparação dos métodos de projeção e estabilizados para as equações de Navier-Stokes incompressíveis, sendo avaliados erros de aproximação e tempos computacionais. São selecionados dois métodos para aplicação, com a técnica adaptativa de avanço no tempo PID, em dois problemas clássicos da literatura: escoamento em uma cavidade bidimensional com tampa deslizante e escoamento com alargamento de canal, comparando-os com resultados obtidos na literatura.

Para os problemas de interação fluido-estrutura foram considerados três experimentos. Inicialmente, é considerado um problema teste com paredes laterais deformáveis. Posteriormente, uma artéria idealizada com um fluxo constante na entrada e uma artéria aneurismática com um fluxo senoidal de entrada, são considerados.

Todos os experimentos foram executados pela biblioteca FEniCS em uma máquina com 7.3 GB de memória RAM e processador com velocidade de clock de 3 GHz.

5.1 Experimentos das equações de Navier-Stokes

Nesta seção é realizada a comparação dos quatro métodos de projeção (Chorin, IPCS, CSS1 e CSS2) e método estabilizado (SUPG/PSPG + LSIC) descritos nos capítulos anteriores, implementados para seis problemas distintos: cavidade bidimensional, canal quadrado, canal retangular com obstáculo, vórtice de Taylor-Green, fluxo de Beltrami e aneurisma idealizado. Foram avaliadas a precisão e eficiência dos métodos em vários níveis de refinamento das malhas.

Em seguida é selecionado um método de projeção e o método estabilizado SUPG/PSPG + LSIC para aplicação no problema do escoamento em uma cavidade bidimensional com tampa deslizante e do escoamento com alargamento de canal, juntamente com a técnica adaptativa de avanço no tempo.

Para solução dos sistemas lineares, foi utilizado o método iterativo GMRES com preconditionador ILU(0), com tolerância absoluta de 10^{-15} e tolerância relativa de 10^{-12} para todos os métodos no primeiro conjunto de testes. Para os testes com PID foi considerado 10^{-10} para ambas as tolerâncias. A não linearidade da formulação SUPG/PSPG + LSIC foi aproximada pelo método de Newton com tolerância de 10^{-6} .

5.1.1 Comparação de Métodos

A fim de avaliar a eficiência e precisão dos métodos, seis problemas testes são considerados para diferentes níveis de refinamento de malha. Para cada problema é calculado um parâmetro de interesse e comparado com valores de referência. Os problemas testes e o parâmetro de interesse são apresentados na Tab. 1. Os níveis de refinamento das malhas, valores de referência e como determinar o parâmetro de interesse são detalhados nas seções de cada problema.

A acurácia dos métodos é realizada gerando o gráfico de erro pelo número de graus de liberdade, e a eficiência plotando o tempo de CPU pelo número de graus de liberdade. O erro é determinado pelo módulo da diferença entre o parâmetro de interesse e o valor de referência. O número de graus de liberdade é a quantidade de incógnitas à serem resolvidas. Os métodos de projeção considerados neste trabalho utilizam polinômios quadráticos para a velocidade e lineares para a pressão, enquanto que a formulação SUPG/PSPG + LSIC utiliza polinômios lineares para pressão e velocidade. Portanto, o número de graus de liberdade será maior para os métodos de projeção em comparação ao método SUPG/PSPG + LSIC, considerando o mesmo nível de refinamento. Dessa forma, ao comparar os gráficos de erro e tempo de CPU conclusões errôneas podem ser tomadas. Assim sendo, os gráficos foram gerados em escalas logarítmicas. Em todos os gráficos o método estabilizado SUPG/PSPG + LSIC é denominado por SUPG para simplificar as legendas.

Tabela 1 – Experimentos testes e parâmetros de interesse

Problema teste	Parâmetro de interesse
Cavidade Bidimensional	Valor mínimo da função de fluxo para $t=2.5s$
Canal Quadrado	Velocidade na direção de x , $v_x(1, 0.5)$, para $t = 0.5s$
Canal Retangular com Obstáculo	Diferença de pressão no cilindro para $t = 8.0s$
Vórtice de Taylor-Green	Energia cinética em $t=0.5s$
Fluxo de Beltrami	Erro relativo (L^2) da velocidade para $t = 0.5s$
Aneurisma Idealizado	Velocidade $v_x(0.025, -0.006, 0)$ para $t = 0.05s$

Cavidade Bidimensional

A cavidade bidimensional com tampa deslizante é um problema clássico da literatura (PANDIT; KALITA; DALAL, 2007); (LOGG; MARDAL; WELLS, 2012). Este experimento é definido por uma cavidade quadrada de lados unitários. São inseridas condições de contorno de não deslizamento nas paredes laterais e na parte inferior, ou seja, $v(x, 0) = v(0, y) = v(1, y) = (0, 0)$ e na parte superior é imposta velocidade unitária na direção de x , $v(x, 1.0) = (1, 0)$. A condição inicial para a velocidade é considerada zero e

um número de Reynolds igual a $R_e = 1000$ para o problema. A Tab. 2 mostra os níveis de refinamento para o problema.

Tabela 2 – Níveis de refinamento para a cavidade bidimensional

Nível de refinamento	Quantidade de vértices	Quantidade de elementos
1	81	128
2	144	242
3	289	512
4	576	1058
5	1089	2048
6	2116	4050
7	4225	8192
8	8464	16562
9	8464	32768

O parâmetro de interesse para a cavidade bidimensional é o mínimo da função de fluxo (Eq. (5.1)) no tempo final $t = 2.5$ segundos. O valor de referência é adotado como $\psi_{ref} = -0.061076605$, dado por (LOGG; MARDAL; WELLS, 2012).

A função de fluxo, descrita em (WHITE, 1991), é dada por:

$$v = \nabla \times \psi = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right), \quad (5.1)$$

podendo ser determinada solucionando a equação de Poisson

$$-\nabla^2 \psi = \omega, \quad (5.2)$$

onde ω é a vorticidade, definida por:

$$\omega = \nabla \times v = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}. \quad (5.3)$$

As linhas de corrente para o tempo $t = 2.5$ segundos são apresentadas na Fig. 5. À medida que o problema avança no tempo, o vórtice no canto superior direito tende a deslocar-se para o centro da cavidade. Além disso, nos cantos inferiores inicia-se o processo de formação de outros dois vórtices.

A Fig. 6 fornece os resultados para o problema da cavidade bidimensional. O método de projeção Chorin apresenta o menor erro. Para o último nível de refinamento os métodos IPCS, CSS2 e SUPG apresentam erros próximos. O método SUPG/PSPG + LSIC apresenta os maiores tempos computacionais, e nota-se a grande diferença entre os erros dos métodos CSS1 e CSS2.

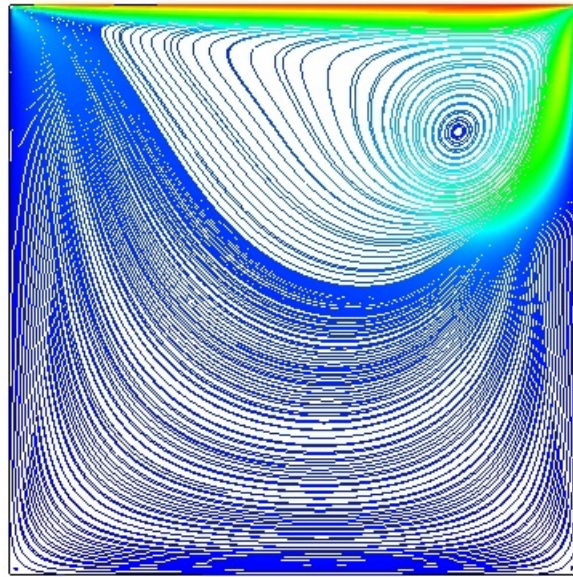


Figura 5 – Linhas de corrente para o problema da cavidade bidimensional utilizando o método Chorin

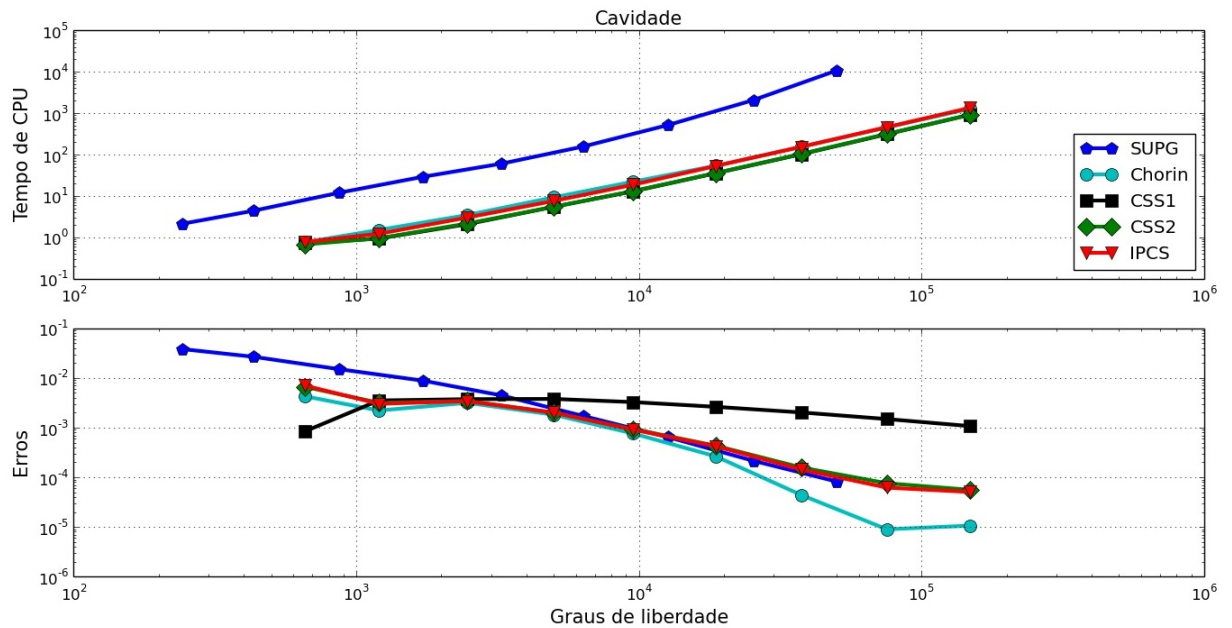


Figura 6 – Resultados para o problema da cavidade bidimensional

Canal Quadrado

Neste teste o escoamento é devido a diferença de pressão impostas na entrada e saída. O problema é definido por um quadrado de lado unitário e número de Reynolds igual a $Re = 8$. Os níveis de refinamento são os mesmos dado pela Tab. 2. Na parede superior e inferior é imposta a condição de não deslizamento. São aplicadas condições de contorno do tipo Dirichlet, em relação a pressão, na entrada e saída, respectivamente, $p(0, y) = 1$ e $p(1, y) = 0$. A condição inicial para a velocidade é imposta sendo nula. O parâmetro de interesse é a componente x da velocidade no ponto $(x, y) = (1.0, 0.5)$ no

tempo final $t = 0.5$ segundos. O valor de referência é proveniente da expansão da série de Fourier no ponto, dado por (LOGG; MARDAL; WELLS, 2012):

$$v_x(1.0, 0.5, t) = 1 - \sum_{n=1,3,\dots}^{\infty} \frac{32}{\pi^3 n^3} e^{-\frac{\pi^2 n^2 t}{8}} (-1)^{(n-1)/2}$$

Portanto, o valor de referência no tempo final é $v_x \approx 0.44321183655681595$.

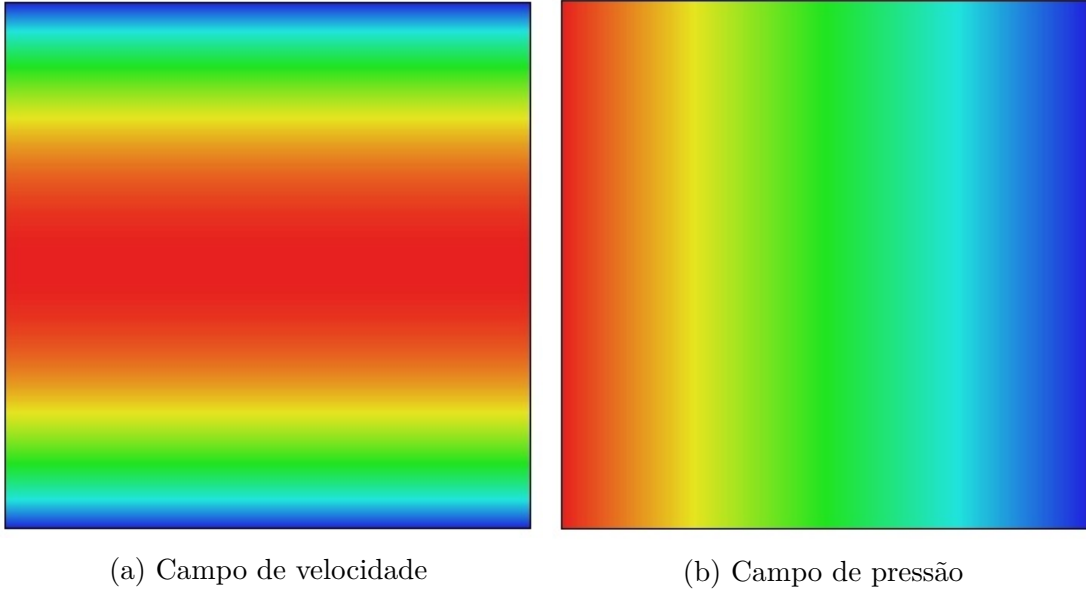


Figura 7 – Velocidade e pressão para o problema do canal quadrado utilizando o método IPCS

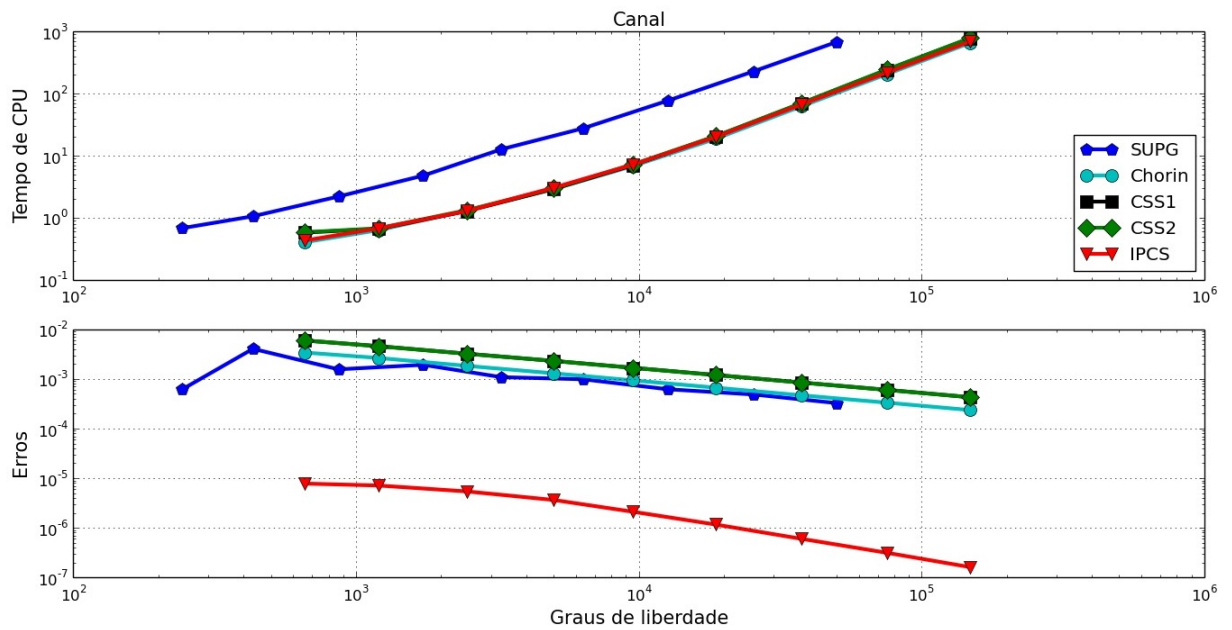


Figura 8 – Resultados para o problema do canal retangular

O escoamento do problema do canal quadrado se dá devido a diferença de pressão imposta, conforme podemos observar na Fig. 7.

A Fig. 8 mostra os resultados para o problema teste do canal quadrado. O menor erro é obtido pelo método IPCS. Os demais apresentam erros próximos e o tempo computacional do SUPG/PSPG + LSIC foi novamente maior.

Canal Retangular com Obstáculo

O próximo problema teste é um canal retangular com obstáculo circular inserido dentro do canal. A geometria do canal tem dimensões $[0, 2.2] \times [0, 0.41]$ e o obstáculo tem raio $r = 0.05$ e centro em $(x_c, y_c) = (0.2, 0.2)$. Os níveis de refinamentos para o experimento são dados na Tab. 3, sendo consideradas malhas triangulares não estruturadas. Condições de contorno de não deslizamento são aplicadas na parte inferior e superior do canal e, também, ao redor do cilindro. Na saída é a imposta condição de Dirichlet, para a pressão, com valor nulo. Para a entrada é prescrita uma condição de velocidade com perfil parabólico e dependente do tempo na direção x , definida por

$$v_x = \left(4V_{max} (y_{max} - y) \sin(\pi t/8) / y_{max}^4, 0 \right), \quad t \leq 8,$$

onde $V_{max} = 1.5$ e $y_{max} = 0.41$. O número de Reynolds para o problema é $Re = 1000$. O parâmetro de interesse para o problema é a diferença de pressão entre a frente e atrás do obstáculo, no tempo final $t = 8$ segundos, definido por: $\Delta p = p(0.15, 0.2, 8) - p(0.25, 0.2, 8)$.

O valor de referência para este problema vale $\Delta p = -0.111444953719$, obtido por (LOGG; MARDAL; WELLS, 2012), no qual foi considerado o método IPCS para uma malha bem refinada com passo de tempo bem pequeno.

Tabela 3 – Níveis de refinamento para o canal retangular com obstáculo

Nível de refinamento	Quantidade de vértices	Quantidade de elementos
1	575	1060
2	806	1501
3	1289	2413
4	2592	4963
5	3836	7364

A Fig. 9 mostra a velocidade do problema no tempo $t = 4$ segundos. O escoamento, após passar pelo obstáculo circular, torna-se irregular. Podemos observar na Fig. 10 que o menor erro obtido é com o método IPCS e próximo estão os métodos CSS2 e Chorin. Mais uma vez o tempo computacional do método SUPG/PSPG + LSIC foi o mais elevado.

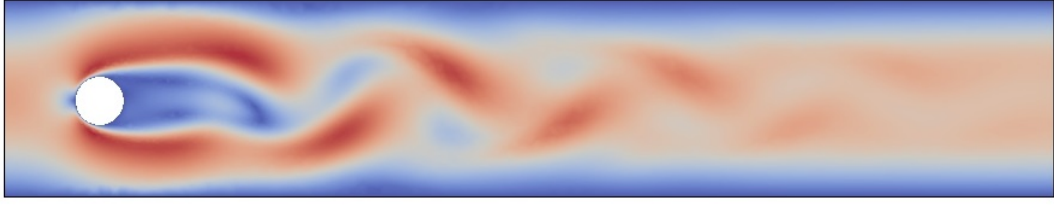
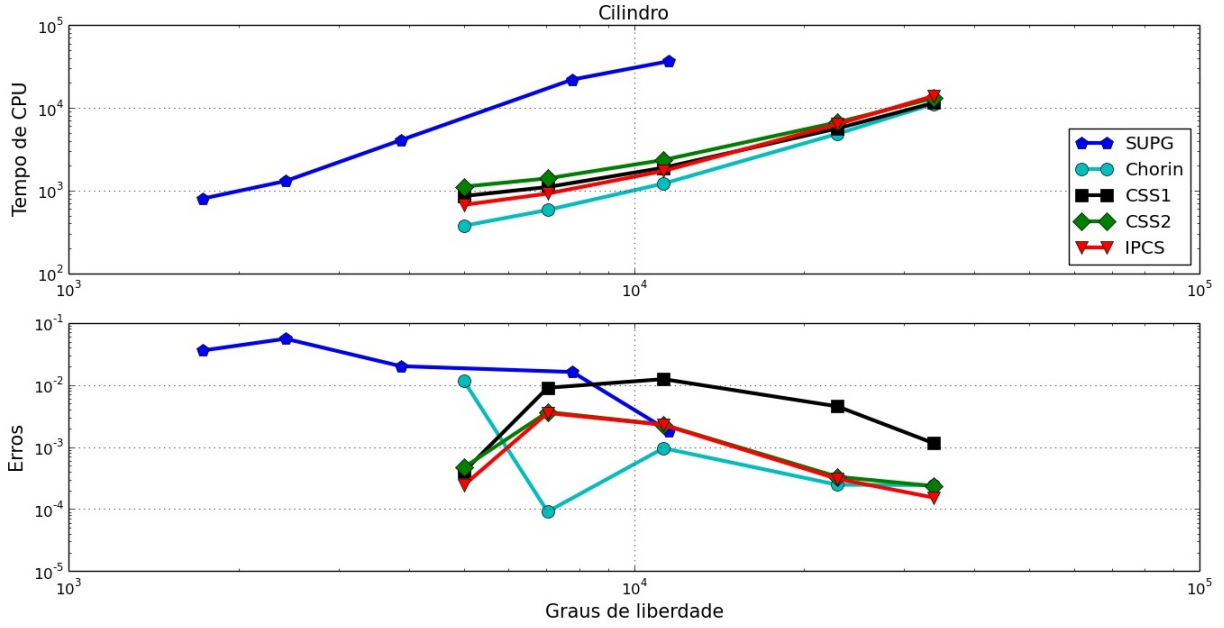
Figura 9 – Vórtices ao passar pelo obstáculo no tempo $t=4$ segundos

Figura 10 – Resultados para o problema do canal com obstáculo

Vórtice de Taylor-Green

Este problema, com solução exata em um domínio quadrado $[-1, 1] \times [-1, 1]$, caracteriza-se por um escoamento com fluxo periódico descrito por (CANUTO et al., 2007). Os níveis de refinamento para o problema estão presentes na Tab. 2 e o número de Reynolds para o problema vale $Re = 100$. Condições de contorno e condição inicial são definidas pela solução exata, definida por:

$$v(x, y, t) = \left(-\cos(\pi x) \sin(\pi y) e^{-2\nu\pi^2 t}, \cos(\pi y) \sin(\pi x) e^{-2\nu\pi^2 t} \right),$$

$$p(x, y, t) = -0.25 (\cos(2\pi x) + \cos(2\pi y)) e^{-4\nu\pi^2 t}.$$

Foi considerado a energia cinética $K = \frac{1}{2} \|v\|_{L^2}^2$ no tempo final $t = 0.5$ como parâmetro de interesse e o valor de referência é obtido pela solução exata.

A Fig. 11a mostra os vetores de velocidade formando vários vórtices, e a pressão no tempo final obtido utilizando o método IPCS. Observamos na Fig. 12 que o menor erro é obtido pelo método IPCS. Além disso, os tempos computacionais dos métodos de projeção são próximos.

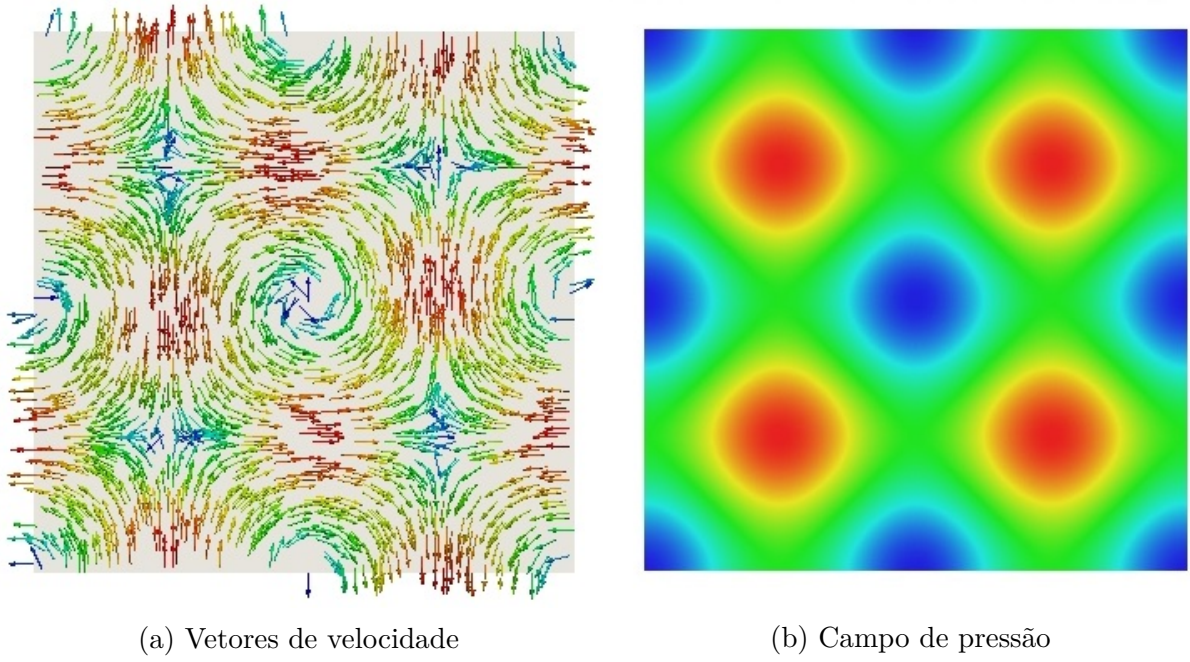


Figura 11 – Vetores velocidade e pressão para o problema do vórtice de Taylor-Green

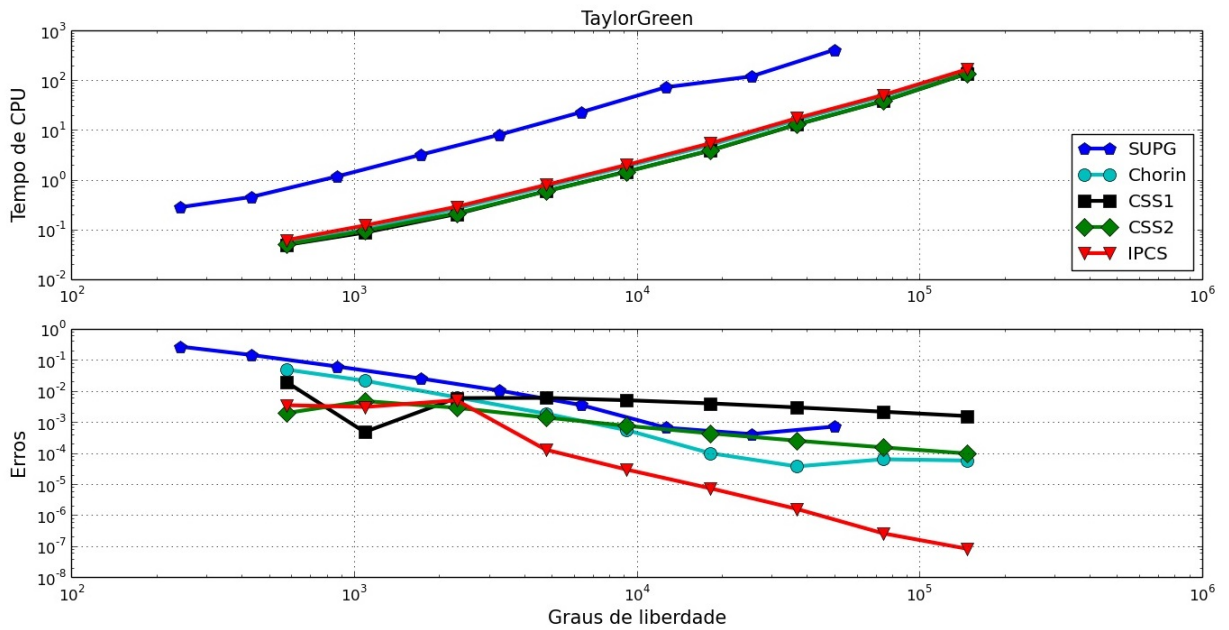


Figura 12 – Resultados para o problema do vórtice de Taylor-Green

Fluxo de Beltrami

O próximo problema a ser considerado é uma solução exata tridimensional, descrito em (ETHIER; STEINMAN, 1994). O domínio tem dimensões $[-1, 1] \times [-1, 1] \times [-1, 1]$, e os níveis de refinamento são dados na Tab. 4 e o número de Reynolds para o experimento é $R_e = 1.0$.

As condições de contorno e condição inicial são descritas pelas soluções exatas,

Tabela 4 – Níveis de refinamento para o fluxo de Beltrami

Nível de refinamento	Quantidade de vértices	Quantidade de elementos
1	216	750
2	729	3072
3	1728	7986
4	4913	24576
5	13824	73002

definidas por

$$\begin{aligned}
 v_x(x, y, t) &= -c_1 e^{-c_2^2 t} (e^{c_1 x} \sin(c_1 y + c_2 z) + e^{c_1 z} \cos(c_1 x + c_2 y)), \\
 v_y(x, y, t) &= -c_1 e^{-c_2^2 t} (e^{c_1 y} \sin(c_1 z + c_2 x) + e^{c_1 x} \cos(c_1 y + c_2 z)), \\
 v_z(x, y, t) &= -c_1 e^{-c_2^2 t} (e^{c_1 z} \sin(c_1 x + c_2 y) + e^{c_1 y} \cos(c_1 z + c_2 x)),
 \end{aligned}$$

para a velocidade, e

$$\begin{aligned}
 p(x, y, z, t) &= -0.5 c_1^2 e^{-2c_2^2 t} (e^{2c_1 x} + e^{2c_1 y} + e^{2c_1 z} + 2e^{c_1(y+z)} \sin(c_1 x + c_2 y) \cos(c_1 z + c_2 x) + \\
 &\quad 2e^{c_1(x+z)} \sin(c_1 y + c_2 z) \cos(c_1 x + c_2 y) + 2e^{c_1(x+y)} \sin(c_1 z + c_2 x) \cos(c_1 y + c_2 z)). \quad (5.4)
 \end{aligned}$$

para a pressão, onde $c_1 = \pi/4$ e $c_2 = \pi/2$. O parâmetro de interesse foi determinado pela norma euclidiana da velocidade no tempo final $t = 0.5$ segundos.

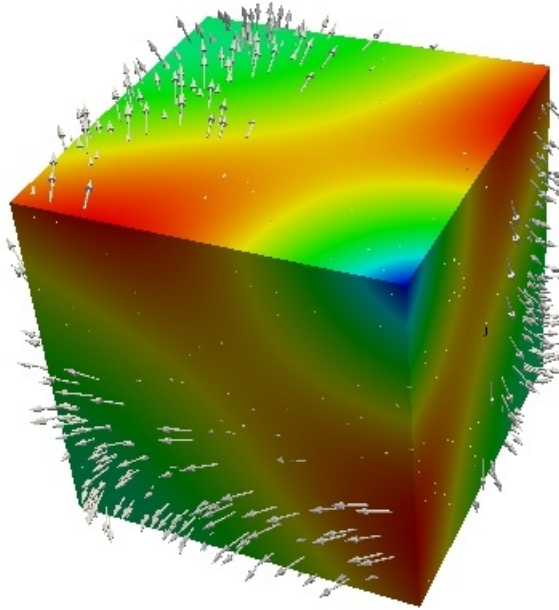


Figura 13 – Vetores velocidade para o problema do fluxo de Beltrami

A Fig. 14 representa os resultados dos erros e tempos para o fluxo de Beltrami e a Fig. 13 os vetores velocidade. Os menores erros são alcançados utilizando os métodos

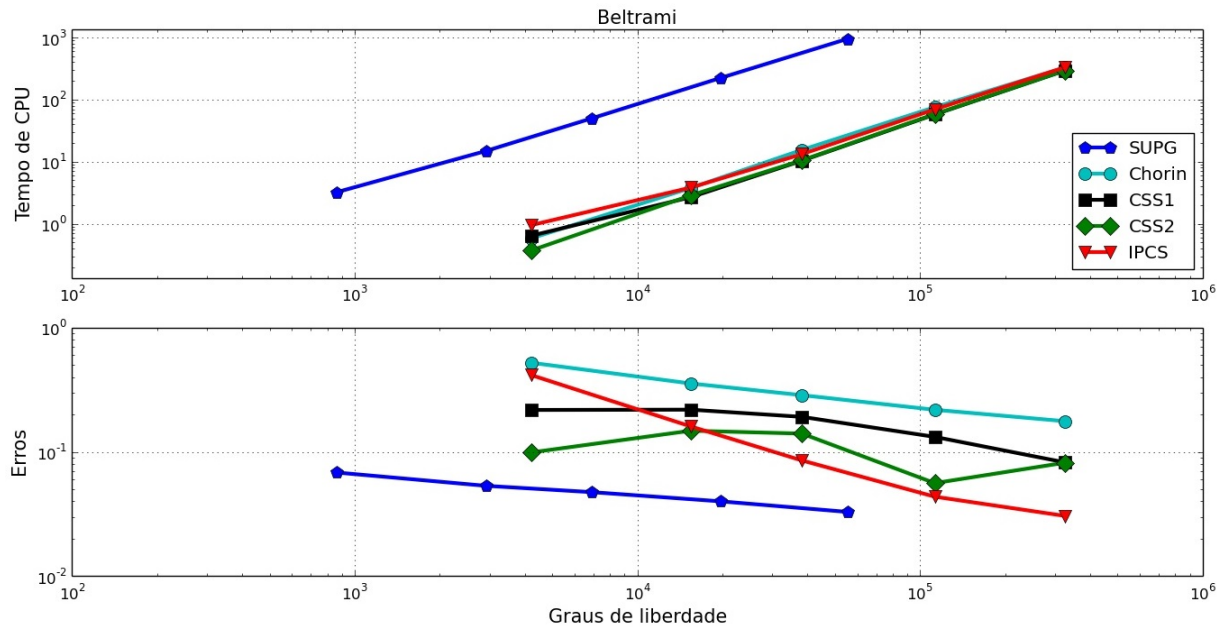


Figura 14 – Resultados para o problema do do fluxo de Beltrami

SUPG e IPCS, enquanto que os maiores erros são obtidos utilizando o método Chorin. Embora o SUPG tenha uma boa acurácia para este problema, o mesmo apresenta elevado tempo computacional.

Aneurisma Idealizado

O último problema teste representa uma artéria aneurismática idealizada. A geometria da artéria possui raio de 2 mm e 50 mm de comprimento. No ponto médio da artéria existe uma esfera com 3.5 mm de raio representando o aneurisma, conforme podemos observar na Fig. 15. Os níveis de refinamentos para o experimento são dados na Tab. 5. A velocidade na entrada é dada por: $v_x = \sin(30t)(r^2 - y^2 - z^2)/r^2$, onde r é o raio da artéria. Já para a saída foi prescrita uma condição de pressão nula. A condição inicial para a velocidade é zero e o número de Reynolds para o problema é de aproximadamente $R_e \approx 1.025 \times 10^6$.

Tabela 5 – Níveis de refinamento para o problema do aneurisma idealizado

Nível de refinamento	Quantidade de vértices	Quantidade de elementos
1	786	2077
2	2063	6909
3	5730	25402
4	10771	50051
5	16140	74738

O parâmetro de interesse para este problema teste é a componente x da velocidade,

no tempo final $t = 0.05$ ms localizada no ponto $(x, y, z) = (0.025, -0.006, 0)(mm)$. O valor de referência, obtido por (LOGG; MARDAL; WELLS, 2012), vale $v_{ref} = -0.0355$.

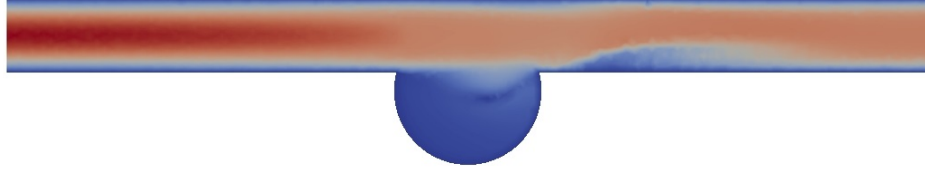


Figura 15 – Velocidade para o problema do aneurisma idealizado

A Fig. 16 representa os resultados para o problema do aneurisma idealizado. Observa-se que todos os métodos de projeção apresentam erros parecidos e menor do que o método SUPG. Além disso, o tempo computacional do método SUPG é mais elevado do que os demais.

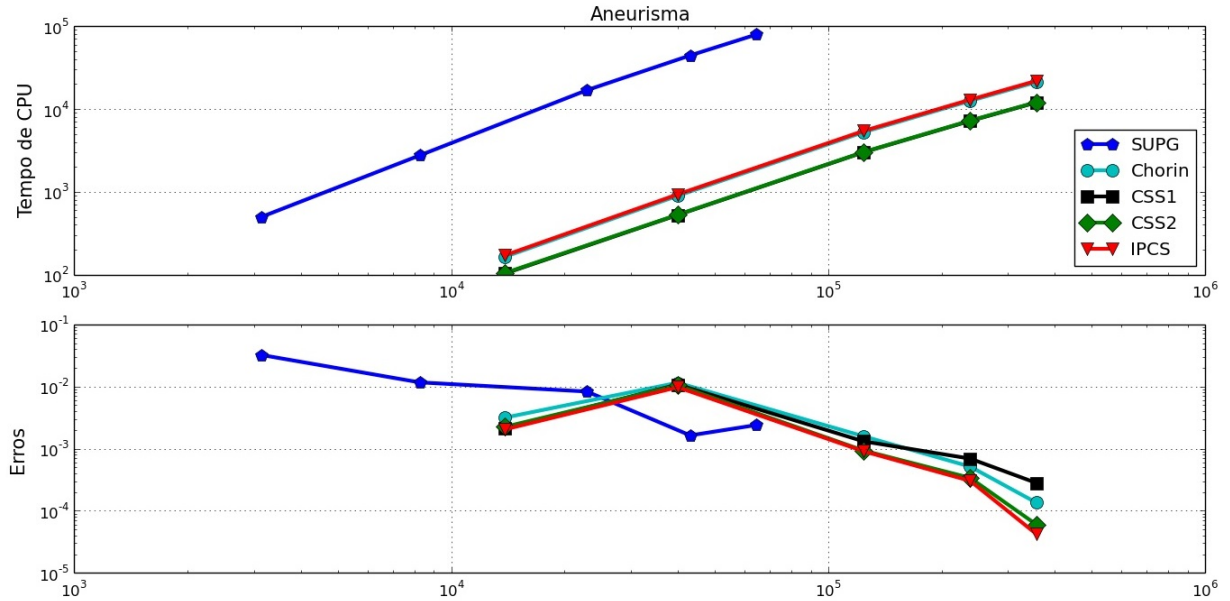


Figura 16 – Resultados para o problema do aneurisma idealizado

Seleção dos métodos

A fim de obter um método com bom desempenho computacional e boa acurácia, o gráfico de dispersão dos tempos e erros foi gerado. O gráfico contém todos os problemas testes para todos os métodos contemplando todos os níveis de refinamento, onde cada ponto do gráfico representa o resultado de um método para certo problema com determinado nível de refinamento. É determinado o tempo médio de CPU pela média de todos os solucionadores em um determinado nível de refinamento, e o tempo de CPU é fracionado pelo tempo médio. Os erros são fracionados de forma análoga. Além disso, os tempos de CPU e erros foram fracionados pelo número de graus de liberdade. A Fig. 17 apresenta um resumo do comportamento de todos os métodos para todos os problemas. O melhor

método é o que se encontra mais próximo do canto inferior esquerdo, ou seja, rápido e preciso.

Podemos observar na Fig. 17 que os métodos CSS1, CSS2 e Chorin possuem desempenho semelhante, pois estão mais agrupados. O método IPCS é o que se encontra menos agrupado, localizado mais no canto inferior esquerdo. Já o método SUPG está localizado no canto superior direito. Dessa forma, o método IPCS é o que apresenta o melhor desempenho no geral, ou seja, uma boa precisão com um tempo computacional relativamente menor. Como a tolerância considerada para a solução do sistema linear resultante foi muito grande (10^{-15} e 10^{-12}), o método SUPG necessitou de um número de iterações maior, uma vez que reconhecidamente formulações estabilizadas tendem a piorar o condicionamento dos sistemas lineares resultantes. Entretanto, boas soluções podem ser obtidas considerando uma tolerância maior. Embora o método estabilizado necessite de mais tempo computacional que os demais, ele será utilizado para aproximar as equações de Navier-Stokes nos problemas de interação fluido-estrutura, uma vez que o método CMM-FSI prevê aproximações de mesma ordem para velocidade e pressão.

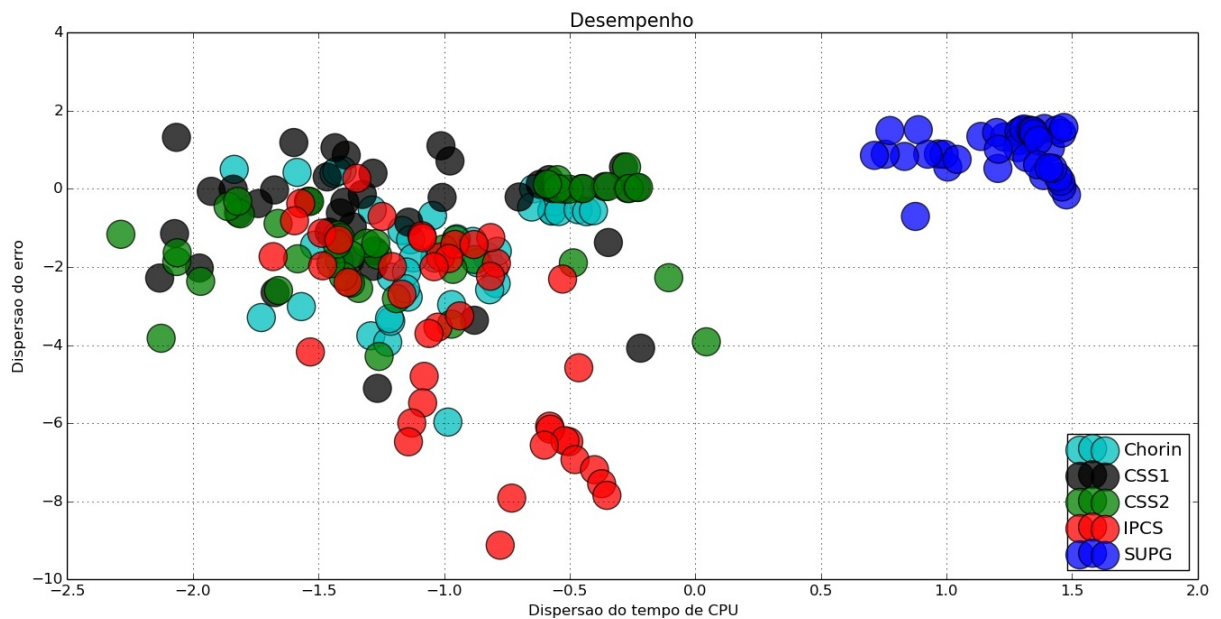


Figura 17 – Desempenho dos métodos

5.1.2 Escoamento em uma Cavidade Bidimensional com Tampa Deslizante

O problema do escoamento em uma cavidade bidimensional com tampa deslizante tem sido estudado por vários pesquisadores. Soluções de referência são encontradas em (GHIA; GHIA; SHIN, 1982). O problema é definido por uma cavidade bidimensional quadrada de lado unitário. Neste experimento são consideradas três tipos malhas, descritas na Tab. 6, onde nas paredes laterais e inferior são impostas condições de Dirichlet de

não deslizamento, velocidade nula, e na parte superior é inserida a velocidade unitária na direção de x .

Tabela 6 – Descrição da malha para o problema do escoamento em uma cavidade bidimensional com tampa deslizante

Malhas	Quantidade de vértices	Quantidade de elementos
Malha 20x20	441	800
Malha 40x40	1681	3200
Malha 60x60	3721	7200

Foi utilizado a técnica adaptativa de avanço no tempo PID com tolerância de 10^{-2} (tol na Eq. (2.37)). A norma relativa da velocidade é utilizada como critério de parada para o estado estacionário, quando atingir uma tolerância pré-estabelecida de 10^{-5} . Nas legendas das Figs. 18 - 23, IPCS + PID é a solução do problema utilizando o método de projeção com a técnica adaptativa de controle do tempo, o mesmo é válido para o método estabilizado SUPG e GHIA são os valores de referência.

Observa-se nas Figs. 18 e 19 que todos os métodos utilizados apresentam o mesmo resultado. Além disso, o controle de tempo com PID não alterou a solução. À medida que refinamos a malha do problema, os resultados obtidos vão se aproximando dos resultados da referência, embora para uma malha menos refinada a solução já se encontra próxima dos valores da literatura. Vale ressaltar que os dados obtidos por (GHIA; GHIA; SHIN, 1982) são adquiridos utilizando uma malha 129x129.

O tempo computacional é reduzido acima de 30% para o método IPCS e mais que 45% para o SUPG, utilizando o controle adaptativo de avanço no tempo, como podemos observar na Tab. 7. Além do mais, o tempo computacional do método SUPG é maior do que o IPCS, mesmo utilizando o controle do tempo. Os tamanhos dos passos de tempos, descritos na Tab. 8, utilizando a técnica PID, são os valores médios em toda a análise. A medida que refinamos a malha, é necessário diminuir o passo de tempo. Podemos observar ainda que, o método IPCS requer um tamanho de passo de tempo ligeiramente menor do que aqueles do método SUPG. Embora o tamanho do passo de tempo médio do método SUPG com o controle PID seja maior, o tempo computacional ainda é maior pois, em geral os sistemas resultantes dos métodos estabilizados são mal-condicionados, necessitando de um número maior de iterações lineares. Além disso, na formulação estabilizada um loop não linear é considerado, enquanto que nos métodos de projeção soluções atrasadas são consideradas para tratar a não linearidade.

À medida que o número de Reynolds aumenta, é necessário refinar a malha para obter melhores resultados, como podemos observar nas Figs. 20 e 21 comparadas com os resultados anteriores. Nota-se pequenas diferenças nas soluções entre os métodos SUPG e

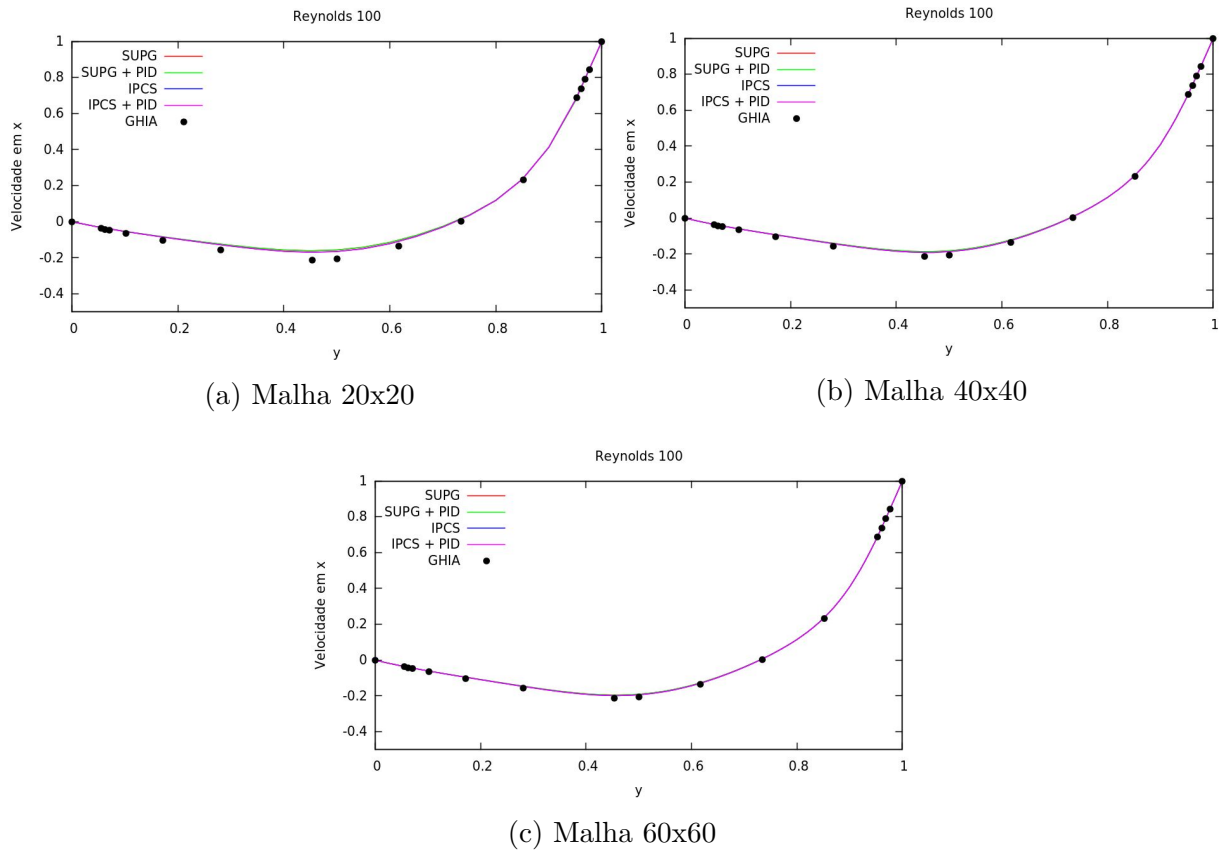
Figura 18 – Velocidade na direção x no centro da cavidade para número de Reynolds 100

Tabela 7 – Tempo de CPU em segundos para número de Reynolds 100

	Malha 20x20	Malha 40x40	Malha 60x60
IPCS	82.21	451.78	1484.35
IPCS + PID	58.02	254.64	921.13
SUPG	294.5	1537.98	5021.68
SUPG + PID	160.36	969.69	2844.61

Tabela 8 – Tamanho médio do passo de tempo para número de Reynolds 100

	Malha 20x20	Malha 40x40	Malha 60x60
IPCS e SUPG	0.014142	0.007071	0.004714
IPCS + PID	0.02280	0.01130	0.007512
SUPG + PID	0.02386	0.01183	0.007858

IPCS nas Figs. 20) e 21. Porém, conforme a malha é refinada ambos os métodos apresentam, praticamente, o mesmo resultado. A solução obtida encontra-se próxima da referência e podemos obter melhores resultados utilizando malhas refinadas. Além disso, a técnica adaptativa de avanço no tempo não alterou a solução.

Mais uma vez, o tempo de CPU do método SUPG é superior ao IPCS, mesmo

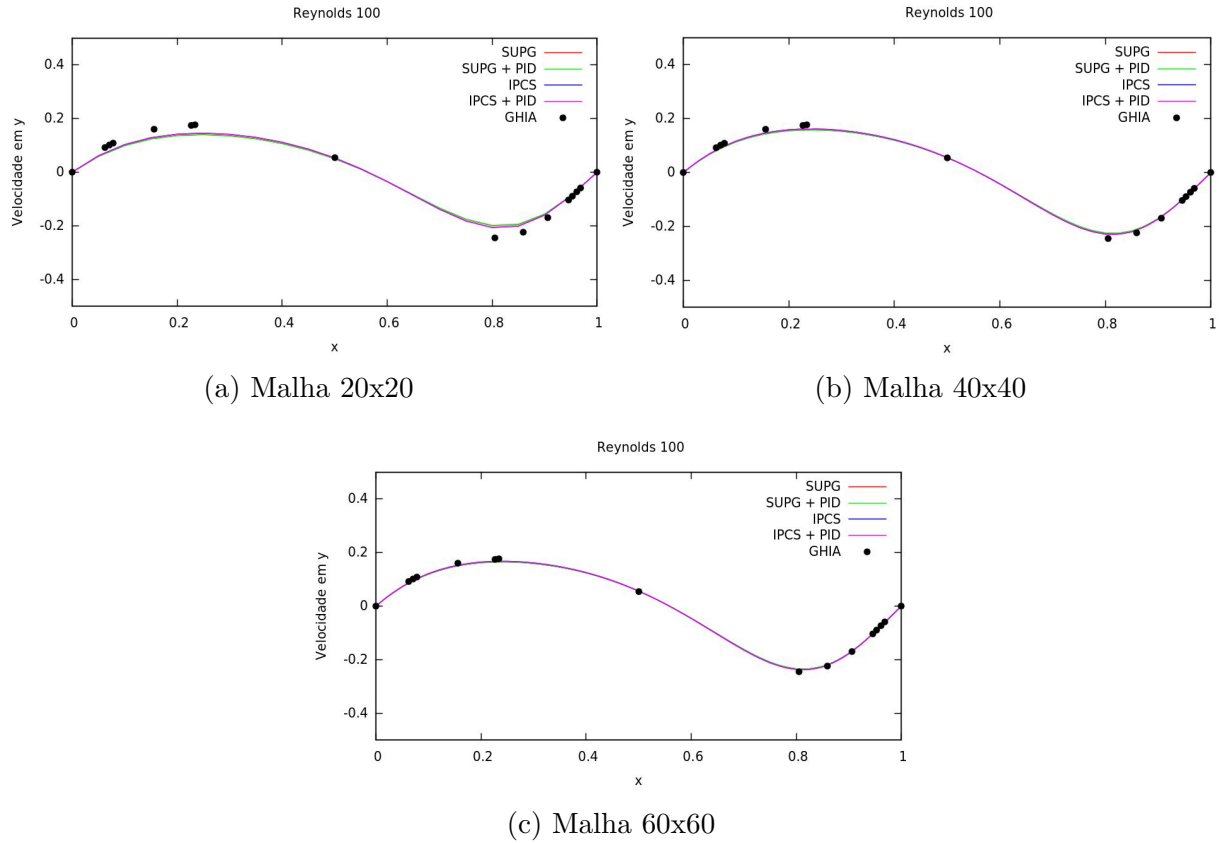


Figura 19 – Velocidade na direção y no centro da cavidade para número de Reynolds 100

utilizando a técnica adaptativa PID, conforme pode ser observado na Tab. 9. Reduções significativas são obtidas ajustando o passo de tempo, girando em torno de 40%, exceto para o método IPCS com a malha 20x20, que obteve uma redução de aproximadamente 17%. Além do mais, para um maior número de Reynolds, o tempo computacional aumenta consideravelmente. Novamente, os passos de tempo médios para o método SUPG são ligeiramente maiores, comparados ao do método IPCS, conforme observa-se na Tab. 10. O problema requer tamanho de passo de tempo menor à medida que o número de Reynolds aumenta.

Tabela 9 – Tempo de CPU em segundos para número de Reynolds 400

	Malha 20x20	Malha 40x40	Malha 60x60
IPCS	234.64	1195.72	3736.21
IPCS + PID	193.21	662.15	2187.23
SUPG	817.27	3925.07	12142.94
SUPG + PID	441.53	2521.41	6999.53

Podemos observar nas Figs. 22 e 23 que é imprescindível uma malha com maior número de elementos para obter melhores resultados. As soluções dos métodos IPCS e SUPG são próximas, mesmo com a técnica adaptativa de avanço no tempo, diferindo

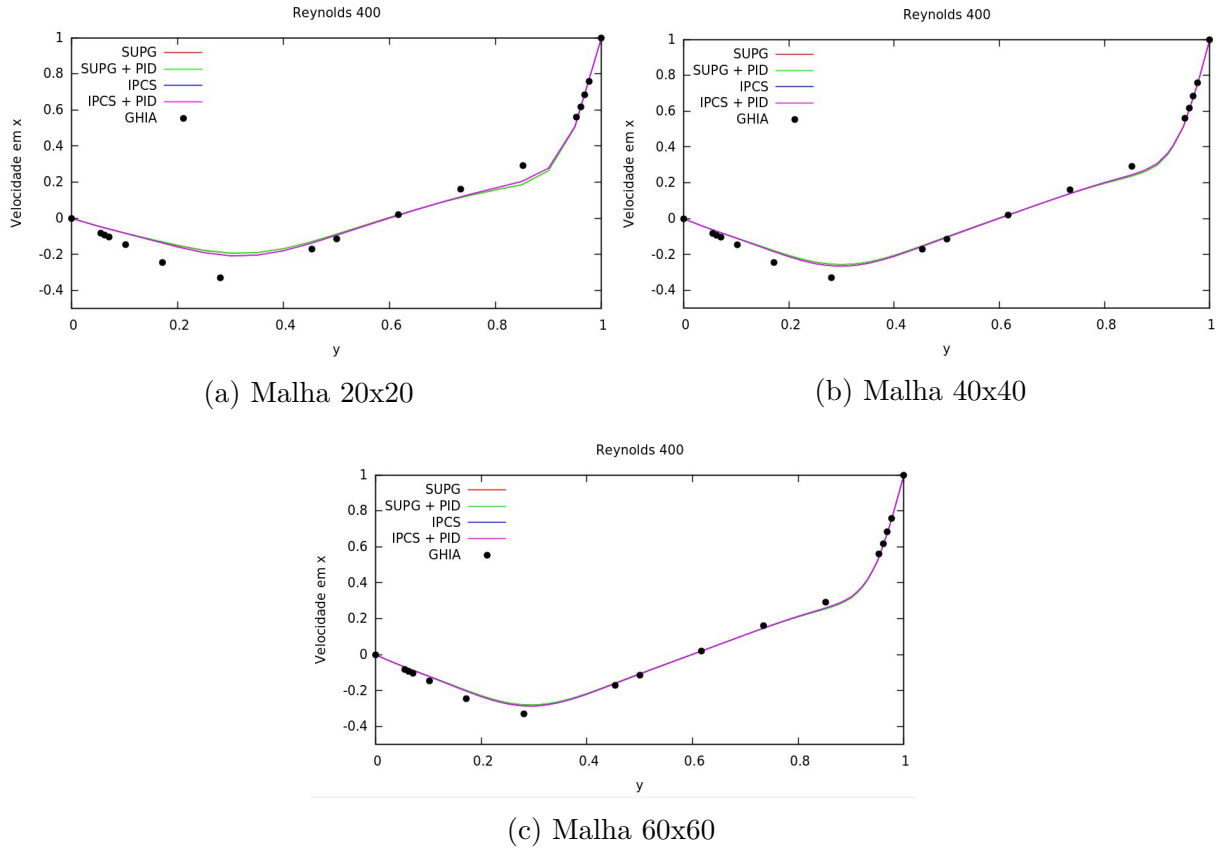
Figura 20 – Velocidade na direção x no centro da cavidade para número de Reynolds 400

Tabela 10 – Tamanho médio do passo de tempo para número de Reynolds 400

	Malha 20x20	Malha 40x40	Malha 60x60
IPCS e SUPG	0.01414	0.007071	0.004714
IPCS + PID	0.02261	0.01114	0.007367
SUPG + PID	0.02424	0.01173	0.007738

apenas em algumas regiões. Ao refinar a malha as soluções se aproximam da solução de referência. Nos intervalos entre 0 e 0.3 e de 0.8 até 1.0 encontramos as maiores diferenças de resultados, comparados com a de referência, tanto para a velocidade na direção x quanto para a velocidade na direção y . Este fato pode ser minimizado refinando apenas essas regiões, melhorando a solução e não comprometendo tanto o tempo computacional.

A Tab. 11 apresenta os tempos computacionais para os diferentes tipos de malhas e métodos utilizados. Conforme a malha é refinada o tempo de CPU aumenta consideravelmente e, novamente, o método IPCS é mais rápido que o SUPG. Além disso, aplicando a técnica adaptativa de controle do passo de tempo, o tempo computacional é reduzido em cerca de 40%, exceto para a malha 20×20 utilizando o método IPCS e a malha 40×40 com o método SUPG. Mais uma vez podemos observar na Tab. 12 que os passos de tempo médios para o método SUPG são maiores do que para o método IPCS. Além do mais,

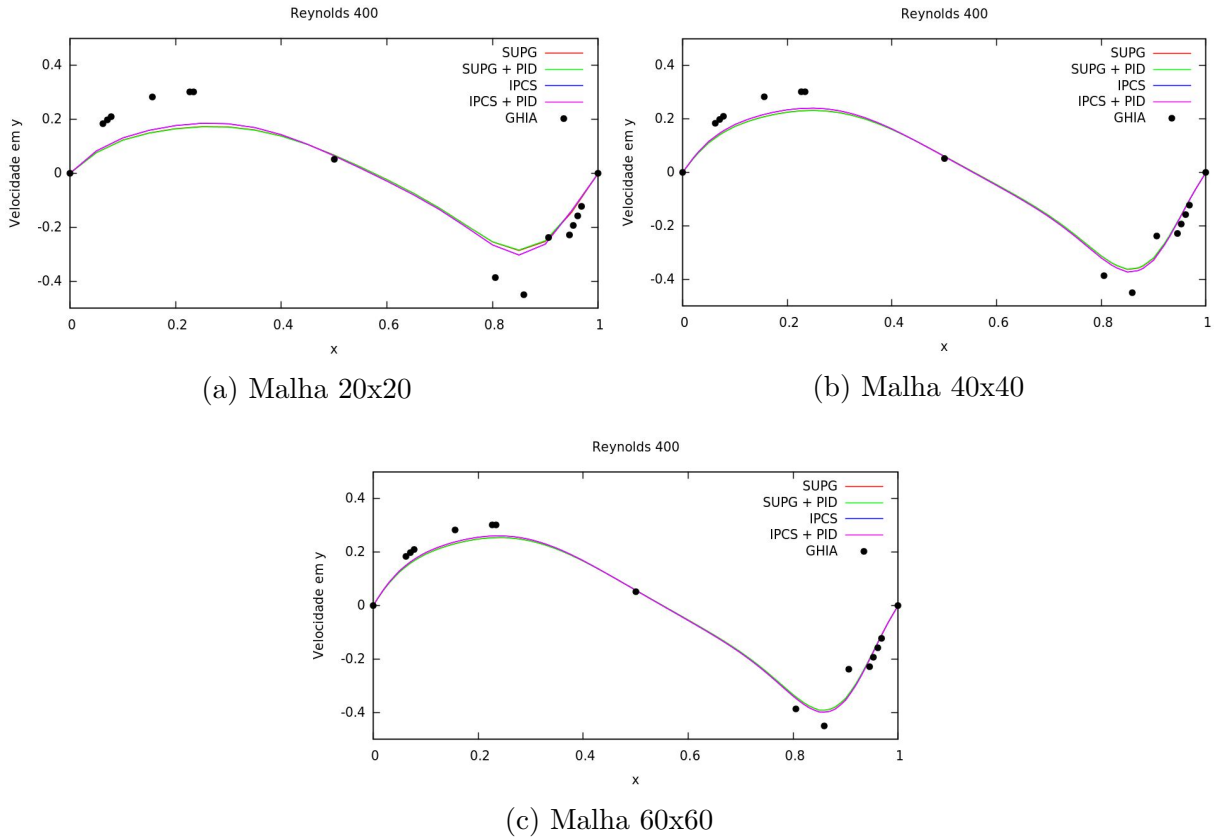


Figura 21 – Velocidade na direção y no centro da cavidade para número de Reynolds 400

aumentando o número de Reynolds é requerido um passo de tempo menor.

Tabela 11 – Tempo de CPU em segundos para número de Reynolds 1000

	Malha 20x20	Malha 40x40	Malha 60x60
IPCS	408.68	2078.46	5928.26
IPCS + PID	312.46	1191.14	3690.54
SUPG	1231.37	5986.23	18309.40
SUPG + PID	773.90	4054.08	10864.38

Tabela 12 – Tamanho médio do passo de tempo para número de Reynolds 1000

	Malha 20x20	Malha 40x40	Malha 60x60
IPCS e SUPG	0.01414	0.007071	0.004714
IPCS + PID	0.02053	0.01038	0.006384
SUPG + PID	0.02087	0.01099	0.006874

As Fis. 24 - 26 representam as soluções do problema pelo método IPCS utilizando a malha 60x60.

Podemos observar nas Figs. 24c, 25c e 26c que o canto superior esquerdo possui

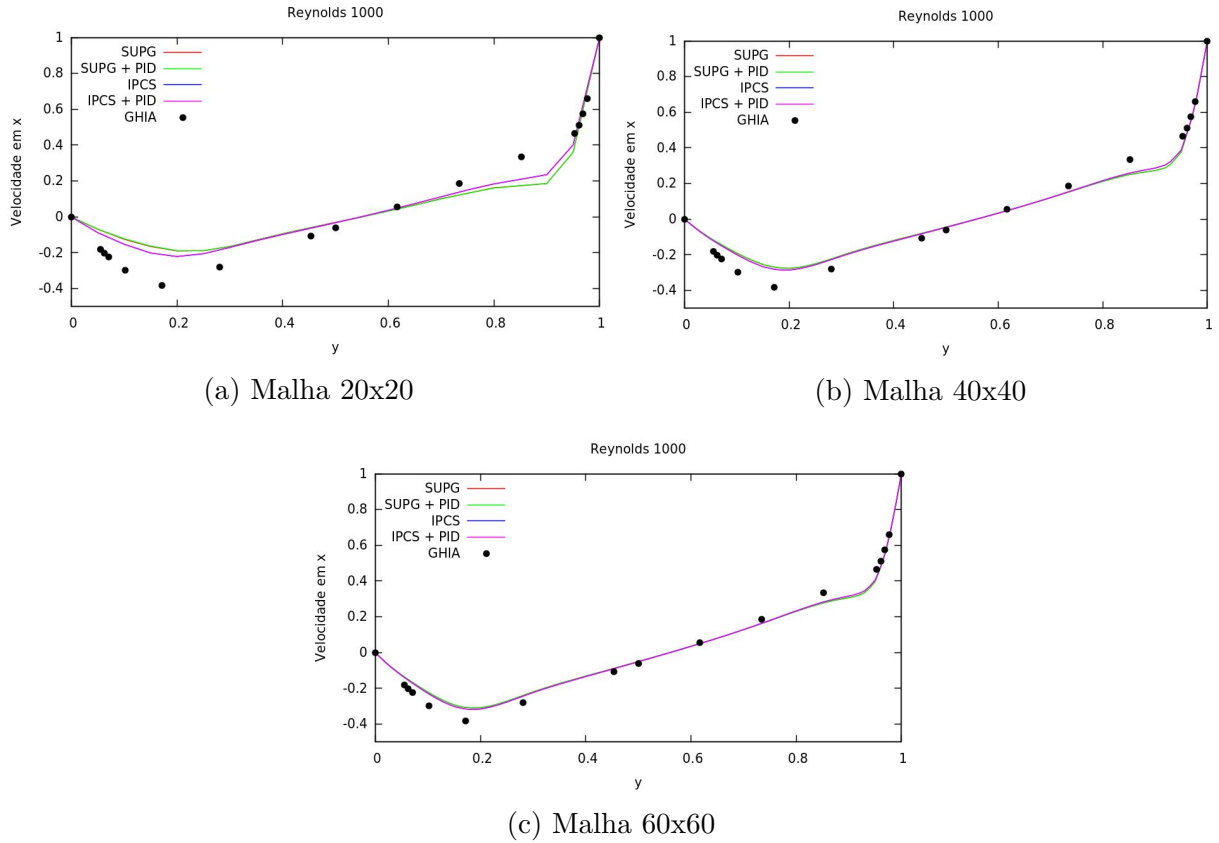


Figura 22 – Velocidade na direção x no centro da cavidade para número de Reynolds 1000

pressão negativa e o canto superior direito pressão positiva e, à medida que o número de Reynolds aumenta, a diferença entre as pressões nesses extremos diminuem. Nota-se ainda que, conforme o número de Reynolds aumenta o vórtice central vai se deslocando para o centro da cavidade. Além disso, aparecem vórtices secundários no canto inferior esquerdo com Reynolds igual a 100 e Reynolds 1000 iniciar-se outro vórtice no canto superior esquerdo da cavidade.

5.1.3 Escoamento com Alargamento de Canal

O segundo experimento considerado, foi o escoamento com alargamento de canal (*Backward Facing Step*), outro problema clássico da literatura. Resultados numéricos podem ser obtidos em (GRIEBEL; DORNSHEIFER; NEUNHOEFFER, 1998). A geometria bem como as condições de contorno são dados na Fig. 27a, adaptada de (ELIAS, 2003). A malha do experimento foi refinada para um valor de $x < 18$, contendo 1877 vértices e 3356 elementos, conforme é apresentada na Fig. 27b.

Os fenômenos de interesse para o problema concentram-se no intervalo de $7.5 < x < 17$. Portanto, o domínio do problema foi seccionado, resultando na região $7 < x < 21$, a fim de visualização das soluções. As Figs. 28a - 28c mostram as linhas de corrente

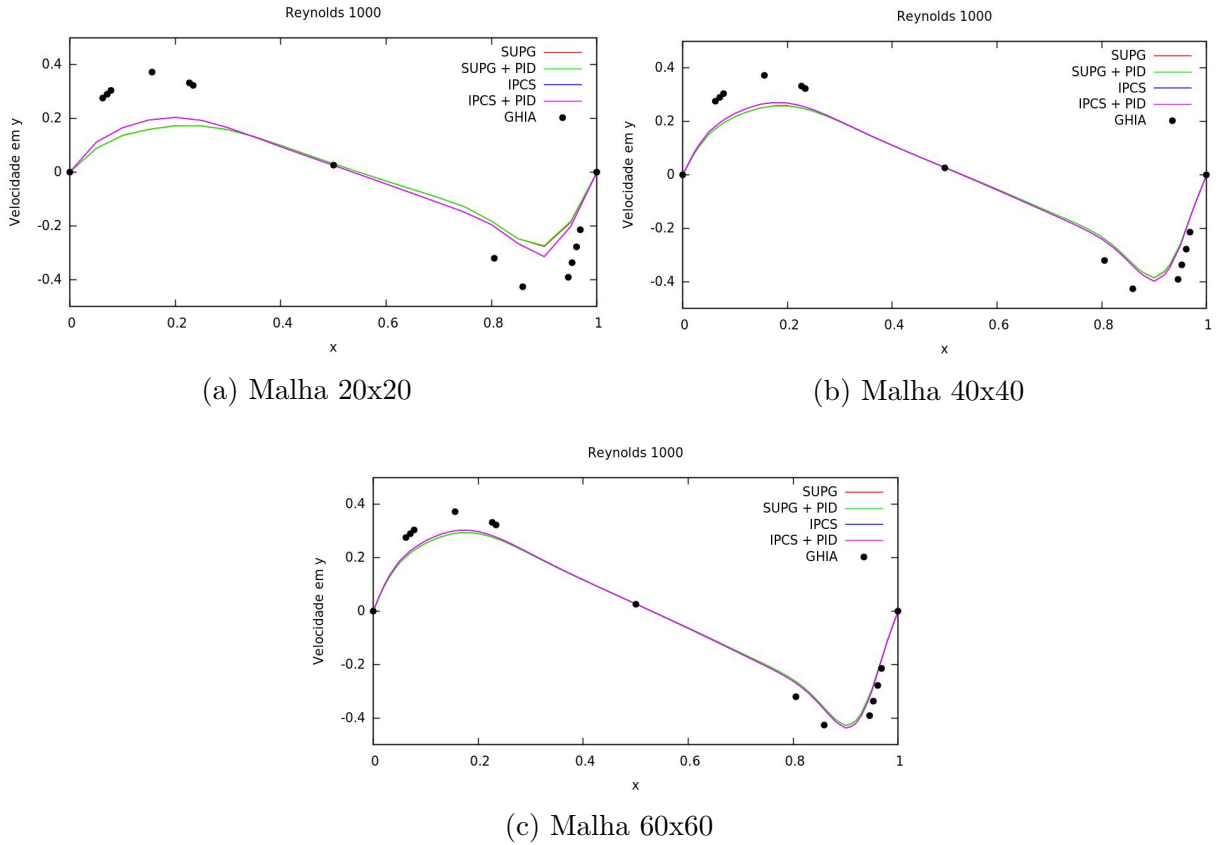


Figura 23 – Velocidade na direção y no centro da cavidade para número de Reynolds 1000

para o problema do escoamento com alargamento de canal para os números de Reynolds 1, 100 e 250, respectivamente. Para um número de Reynolds pequeno o escoamento é regular, expandindo rapidamente após o ressalto e, à medida que o número de Reynolds aumenta, o escoamento expande-se mais a frente, gerando vórtices no canto inferior esquerdo, deslocando o fluxo principal para baixo. Podemos observar ainda que, para Reynolds acima de 100, o fluxo principal é deslocado mais ainda para baixo, aumentando o primeiro vórtice e iniciando um segundo na parte superior do canal.

Os tamanhos dos vórtices são utilizados para qualificar implementações numéricas aplicadas a esse tipo de problema. Todos os comprimentos obtidos são divididos pela altura do ressalto, que no experimento foi utilizado $s = 0.75$. Os comprimentos característicos são descritos na Fig. 29.

A Tab. 13 representa os resultados obtidos na simulação numérica em comparação aos resultados de referência para os comprimentos característicos, denominado por GRIEBEL, encontrados em (GRIEBEL; DORNSHEIFER; NEUNHOEFFER, 1998). Os métodos SUPG e IPCS apresentam os mesmos resultados, utilizando ou não a técnica adaptativa de avanço no tempo PID. Dessa forma, a Tab. 13 apresenta apenas os resultados utilizando o método IPCS. Pode-se avaliar, que para número de Reynolds menores os métodos conseguiram reproduzir de forma satisfatória os resultados. Já para número

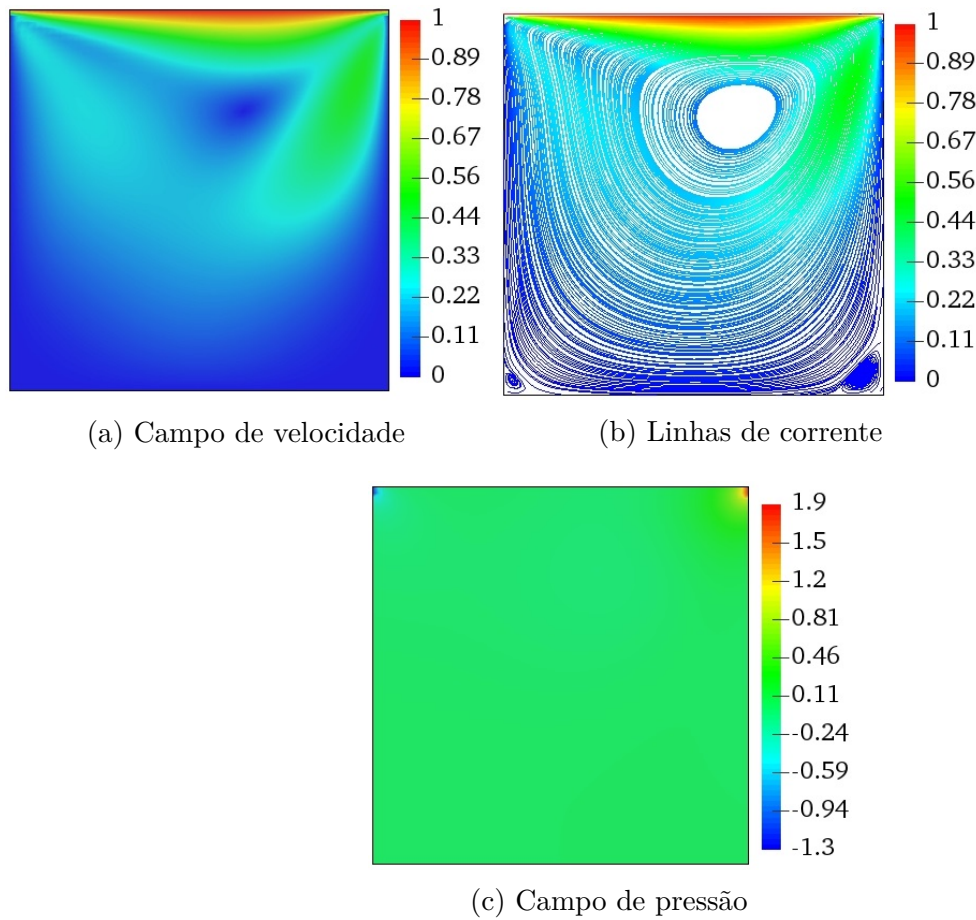


Figura 24 – Solução para o problema do escoamento em uma cavidade com tampa deslizando para número de Reynolds 100 utilizando o método IPCS

de Reynolds igual a 250, os resultados obtidos diferem um pouco da referência para o comprimento característico do vórtice na parte superior. Vale ressaltar que, (GRIEBEL; DORNSHEIFER; NEUNHOEFFER, 1998) utiliza o método das diferenças finitas para solução do problema com uma malha bastante refinada.

Tabela 13 – Comprimento característico para números de Reynolds 1, 100 e 250

R_e	GRIEBEL			IPCS		
	x_1/s	x_2/s	x_3/s	x_1/s	x_2/s	x_3/s
1	0.30	—	—	0.293	—	—
100	3.80	—	—	3.573	—	—
250	5.80	5.40	4.40	5.6	3.6	5.47

A Tab. 14 apresenta os tempos computacionais para os diferentes números de Reynolds e métodos utilizados. O tempo computacional dos métodos SUPG e IPCS são próximos, para Reynolds igual a 1 e utilizando a técnica de controle adaptativo PID o tempo de CPU diminuí drasticamente. Isso se deve ao fato de que o tamanho do passo de tempo utilizado é muito pequeno, não necessitando desse tamanho para solucionar o

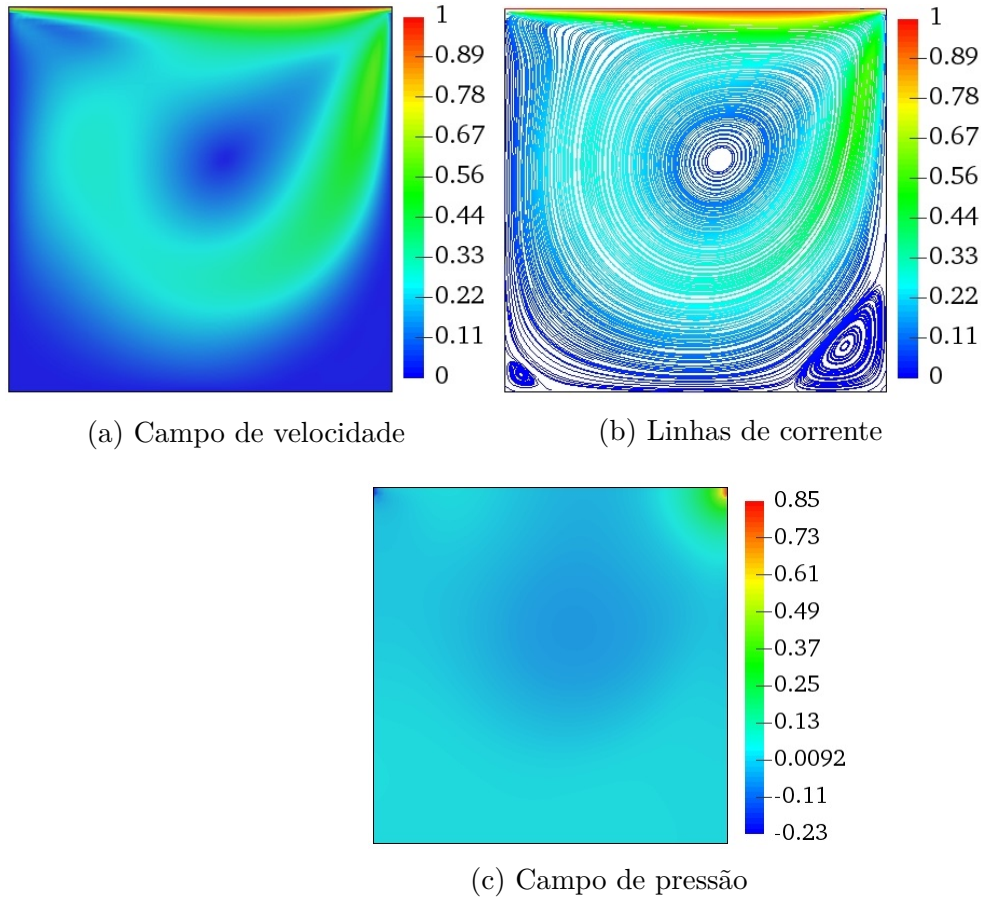


Figura 25 – Solução para o problema do escoamento em uma cavidade com tampa deslizante para número de Reynolds 400 utilizando o método IPCS

problema. A medida que o número de Reynolds é elevado, o tempo computacional aumenta consideravelmente. Além disso, o tempo computacional, utilizando o método SUPG, é maior que do método IPCS, mesmo utilizando a técnica de controle PID, que reduz o tempo de CPU em mais de 35%.

Os tamanhos dos passos de tempo, descritos na Tab. 15, utilizando a técnica PID, são os valores médios. Podemos observar que, os passos de tempo médios para o método SUPG são ligeiramente maiores do que aqueles do método IPCS. Ressaltamos que com o aumento do número de Reynolds, é requerido um passo de tempo menor para a solução do problema. Apesar do passo de tempo médio para o método SUPG com o controle adaptativo PID ser maior, o tempo de CPU ainda é maior.

As Figs. 30 - 32 apresentam as soluções do problema do escoamento com alargamento de canal utilizando o método IPCS. Podemos observar que, conforme o número de Reynolds aumenta, o escoamento se torna mais irregular, gerando um vórtice no canto inferior esquerdo e, à medida que este se eleva mais, o escoamento principal é deslocado para baixo. Devido a este fato, inicia-se outro vórtice na parte superior do canal, para o número de Reynolds igual a 250.

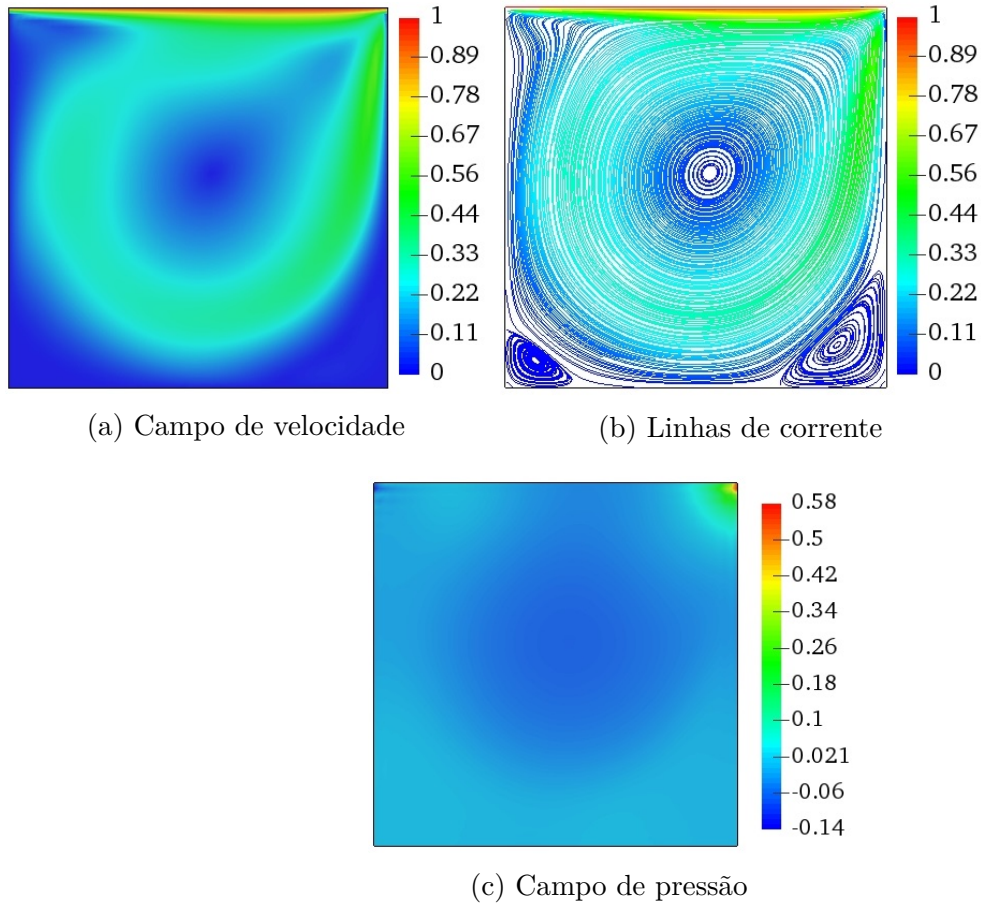


Figura 26 – Solução para o problema do escoamento em uma cavidade com tampa deslizante para número de Reynolds 1000 utilizando o método IPCS

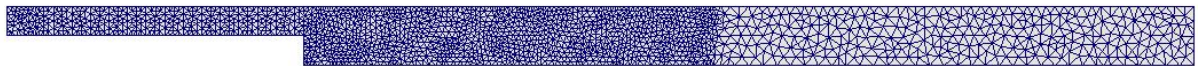
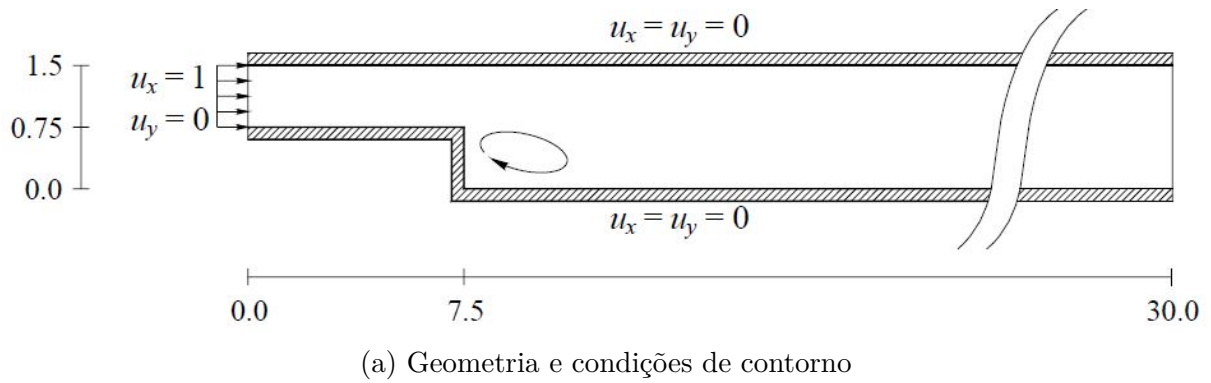


Figura 27 – Descrição do modelo e da malha para o problema do escoamento com alargamento de canal

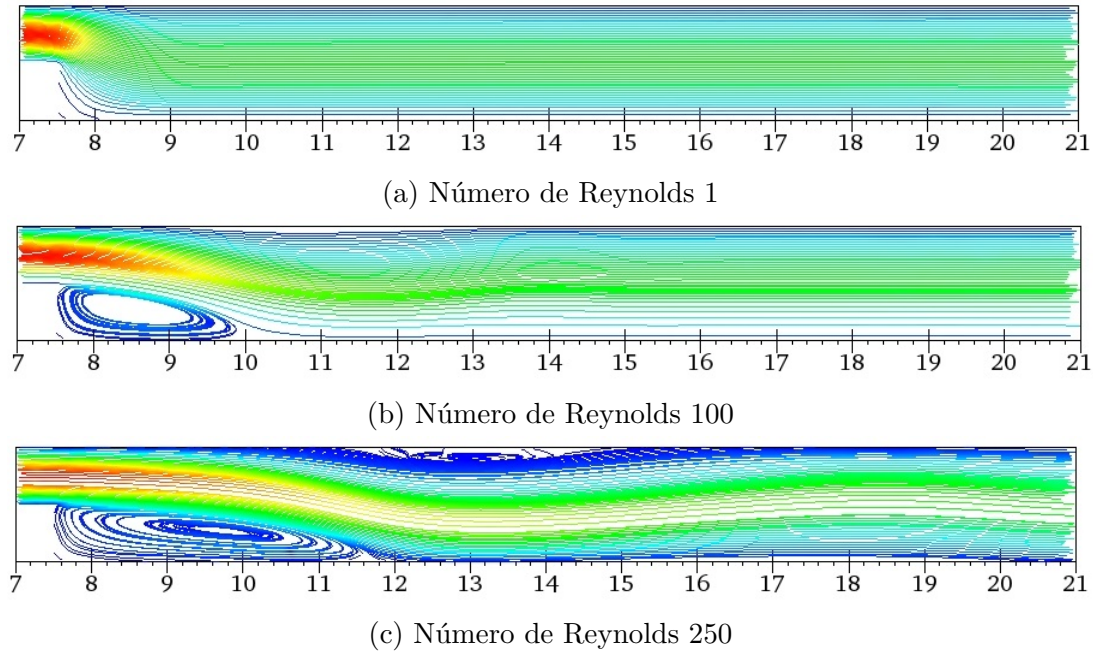


Figura 28 – Linhas de corrente para o problema do escoamento com alargamento de canal utilizando o método IPCS

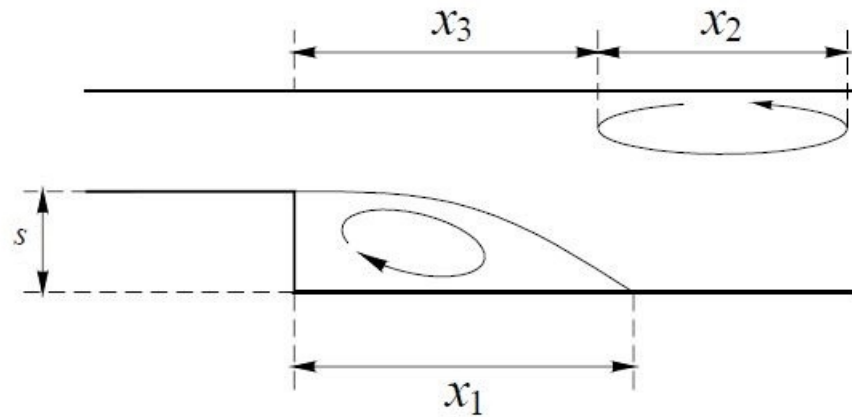


Figura 29 – Comprimentos de referências para o problema do escoamento com alargamento de canal

Tabela 14 – Tempo de CPU em segundos para o escoamento com alargamento de canal

	Reynolds 1	Reynolds 100	Reynolds 250
IPCS	145.19	1231.55	7433.19
IPCS + PID	38.67	815.22	4103.19
SUPG	161.32	4006.30	18101.65
SUPG + PID	42.96	1635.06	9631.14

5.2 Experimentos de Interação Fluido-Estrutura

Nesta seção são apresentados os experimentos numéricos dos problemas de interação fluido-estrutura utilizando o método CMM-FSI. Primeiramente, consideramos um problema

Tabela 15 – Tamanho médio do passo de tempo para o escoamento com alargamento de canal

	Reynolds 1	Reynolds 100	Reynolds 250
IPCS e SUPG	0.02254	0.01672	0.008361
IPCS + PID	0.02280	0.02026	0.01720
SUPG + PID	0.02283	0.02155	0.01764

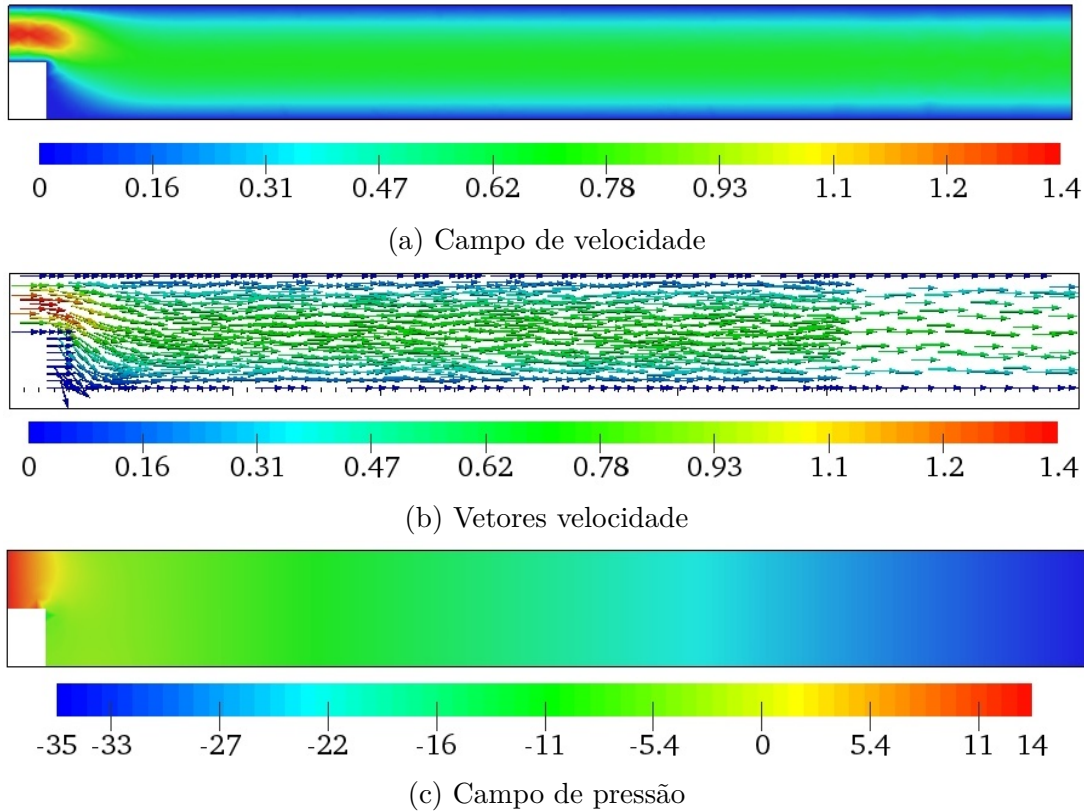


Figura 30 – Solução para o problema do escoamento com alargamento de canal para número de Reynolds 1 utilizando o método IPCS

teste com paredes laterais deformáveis e, em seguida, dois experimentos: uma artéria idealizada com um fluxo constante na entrada e uma artéria aneurismática com um fluxo senoidal de entrada. Embora o método IPCS tenha destacado nos experimentos anteriores, para os problemas IFE aproximados pelo método CMM-FSI vamos considerar o método estabilizado SUPG/PSPG + LSIC, uma vez que para o método CMM-FSI é primordial ter velocidade e pressão aproximados por funções de interpolação de mesma ordem. Para solução dos sistemas lineares resultantes foi utilizado o método iterativo GMRES com preconditionador ILU(2), com tolerância de 10^{-8} . A não linearidade do problema foi aproximada pelo método de Newton com tolerância de 10^{-5} .

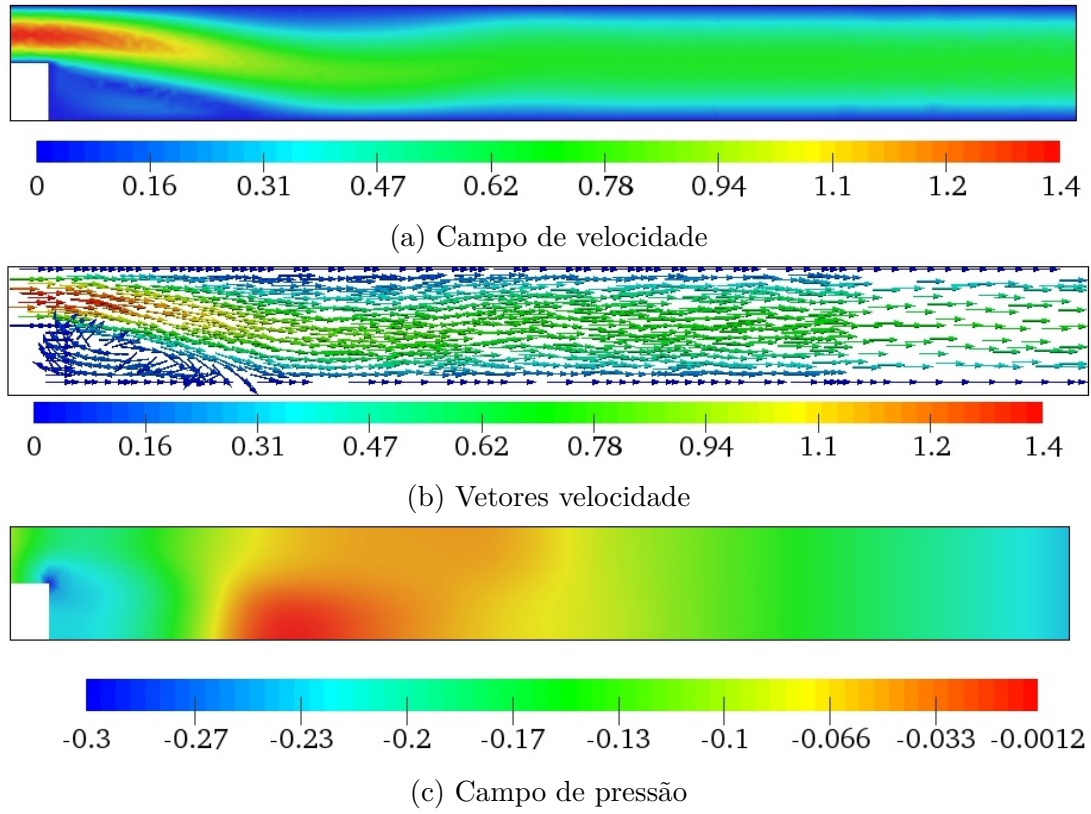


Figura 31 – Solução para o problema do escoamento com alargamento de canal para número de Reynolds 100 utilizando o método IPCS

5.2.1 Parede Deformável

O problema é definido por um canal retangular com paredes laterais deformáveis e fluxo constante. O domínio é dado por $\Omega = [-1, 1] \times [-0.5, 0.5] \times [-1, 1]$, com velocidade prescrita na entrada ($v_z = V_{max}(1 - x^2)$) e pressão constante na saída (p_{out}). Os valores das propriedades do fluido e do sólido, bem como outros parâmetros são apresentados na Tab. 16. Destacamos que, com exceção do módulo de Young, todos os parâmetros do experimento são fisiologicamente reais. Neste experimento é considerada uma malha com 6000 tetraedros, conforme mostra a Fig. 33, onde as paredes deformáveis são aquelas normais à direção x . O passo de tempo utilizado para este experimento foi de $\Delta t = 0.005$ segundos.

Tabela 16 – Propriedades e parâmetros para parede deformável

ρ	μ	E	ρ^s	ν^s	k	ζ	V_{max}	p_{out}
g/cm ³	dyn.s/cm ²	dyn/cm ²	g/cm ³	-	-	cm	cm/s	dyn/cm ²
1.06	0.04	5320	1.09	0.5	5/6	0.1	12.5	2660

Os resultados são apresentados em quatro nós, posicionados de acordo com a Tab.17. Os nós A e B podem ser visualizados na Fig. 33 e os nós C e D estão na parede paralela. Esses nós são livres para se moverem na direção do fluxo, devido a tensão de

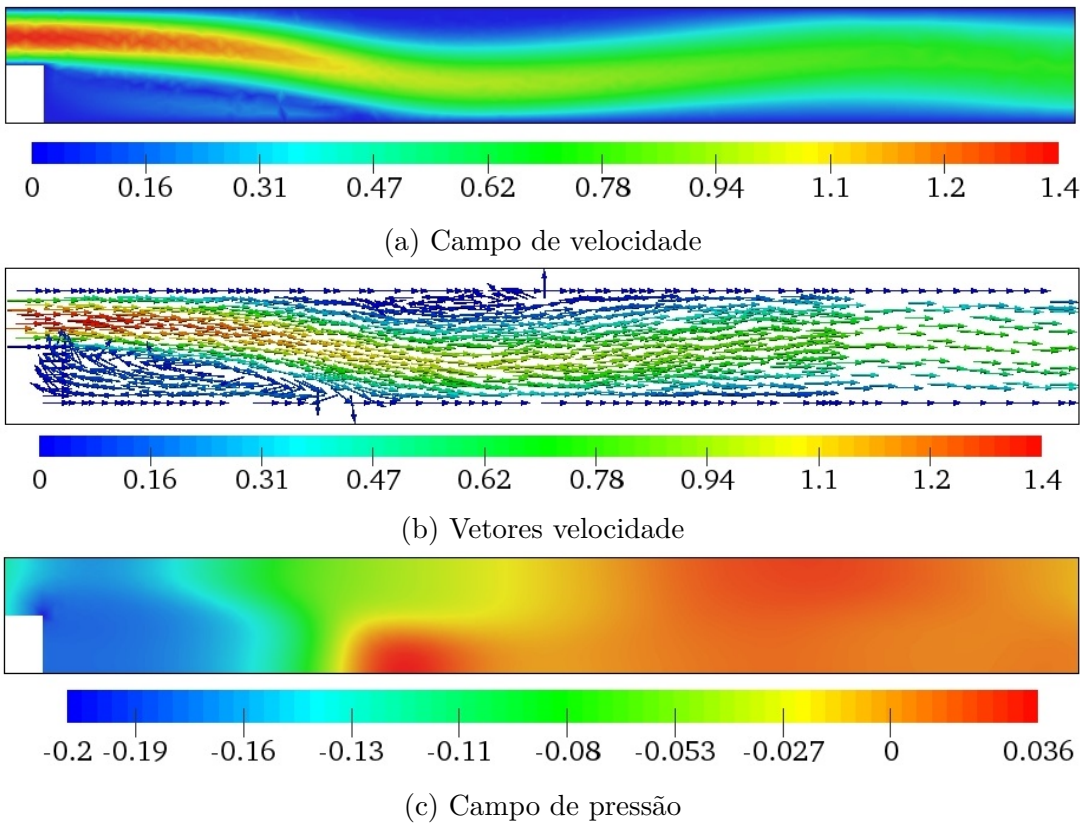


Figura 32 – Solução para o problema do escoamento com alargamento de canal para número de Reynolds 250 utilizando o método IPCS

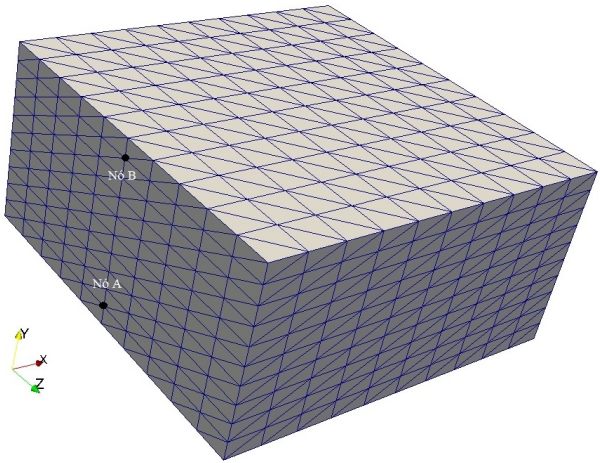


Figura 33 – Malha para o problema de parede deformável

Tabela 17 – Posição dos nós

-	x(cm)	y(cm)	z(cm)
Nó A	-1.0	-0.4	0.0
Nó B	-1.0	+0.4	0.0
Nó C	+1.0	+0.4	0.0
Nó D	+1.0	-0.4	0.0

cisalhamento, e na direção perpendicular a parede em razão da tensão normal. Portanto, o deslocamento e a velocidade nestes nós são observadas nas direções z e x . Os resultados dos deslocamentos e velocidades (Figs. 34 e 35), incluem as duas formas de inicialização: deslocamento zero e pressão hidrostática.

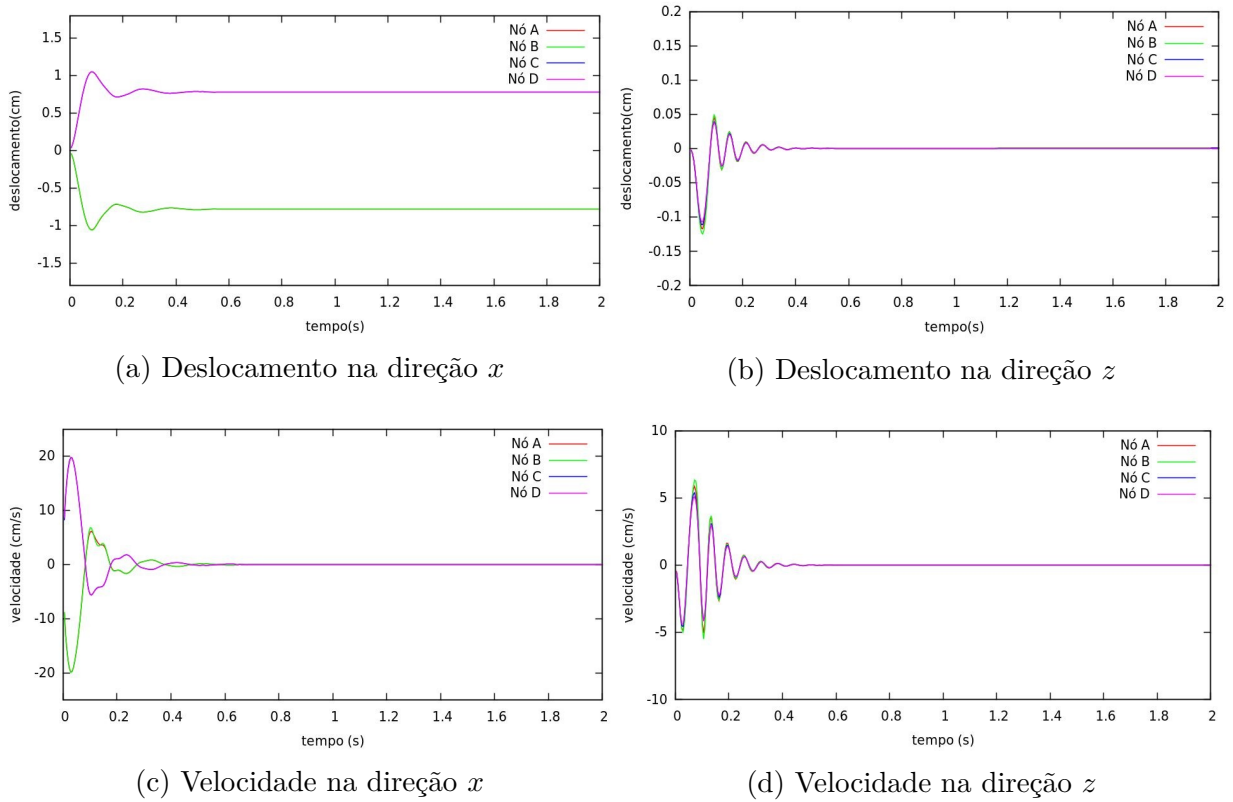


Figura 34 – Velocidade e deformação dos nós nas paredes (deslocamento zero)

Podemos observar na Fig. 34 que a variação inicial da tensão, devido a interação do fluido com a estrutura, causa uma oscilação do deslocamento nas paredes laterais. O sistema atinge o equilíbrio, como esperado, onde a velocidade na parede é zero e o deslocamento é um valor fixo. O deslocamento na direção x ocorre devido a pressão do fluido nas paredes laterais da estrutura e na direção longitudinal (z) ser nula, pois a velocidade na interface é zero. Vale ressaltar que inicialmente temos uma tensão maior devido à carga súbita imposta na parede.

O problema deformável é inicializado com a velocidade proveniente do problema de parede rígida. Devido a este fato, existe naturalmente uma deformação inicial, que é o deslocamento sofrido pelas forças de pressão do fluido na interface, o que do ponto de vista mecânico é muito desejável, uma vez que existe um equilíbrio entre as forças internas do fluido e o sólido na interface.

A Fig. 35 representa os resultados com a inicialização do algoritmo com pressão hidrostática, ou seja, deslocamento inicial diferente de zero. Podemos observar que neste

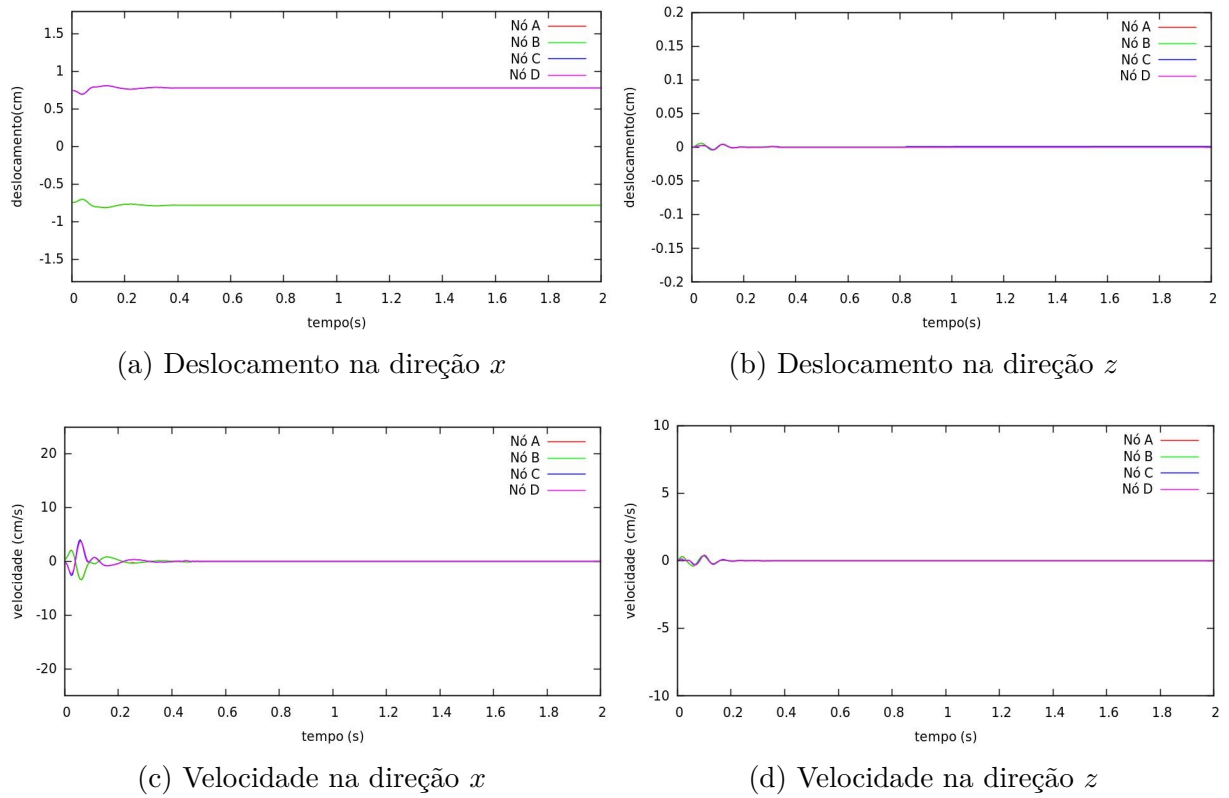


Figura 35 – Velocidade e deformação dos nós nas paredes (pressão hidrostática)

caso o regime estacionário é atingido mais rapidamente. As oscilações iniciais, tanto nos campos de velocidade como nos de deslocamento, são bem menores. Analisando mais detalhadamente a parte da velocidade em x , para as duas formas de inicialização do deslocamento (Figs. 34c e 35c), observamos que iniciando com deslocamento zero demora cerca de 0.6 segundos para atingir o estado estacionário e já no caso pré-carregado (pressão hidrostática) o tempo para atingir o estado estacionário é de cerca de 0.3 segundos. Portanto, é vantajoso iniciar o problema deformável com uma pré-carga. Além disso, do ponto de vista fisiológico, as paredes dos vasos são bem populares por serem pré-carregadas na direção radial e longitudinal (FIGUEROA et al., 2004).

Utilizamos o mesmo passo de tempo para as diferentes inicializações, entretanto o tempo de CPU foi cerca de 1470 segundos para o caso de deslocamento zero e 844 segundos para o caso hidrostático. Isso demonstra que o caso com inicialização hidrostática necessita de menos iterações lineares e não lineares para atingir as tolerâncias pré-fixadas, resultando em um procedimento numérico mais vantajoso.

5.2.2 Artéria Idealizada

Neste experimento uma artéria idealizada com raio de 0.3 cm e comprimento de 4.2 cm é considerada para descrever o fluxo sanguíneo. Na entrada é prescrito fluxo

constante (v_{in}) com um perfil parabólico e na saída uma pressão constante (p_{out}). A malha utilizada para o problema é formada por 35875 elementos e 6961 nós, sendo considerado um passo de tempo igual a 1.38×10^{-4} segundos. A Fig. 36 mostra a malha bem como as dimensões da artéria. Os resultados são obtidos utilizando o caso pré-carregado para inicializar a deformação, com uma pressão média de 1510 dyn/cm² na interface. As propriedades e parâmetros necessários para o problema se encontram na Tab. 18.

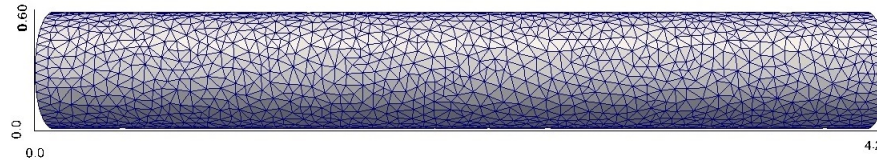
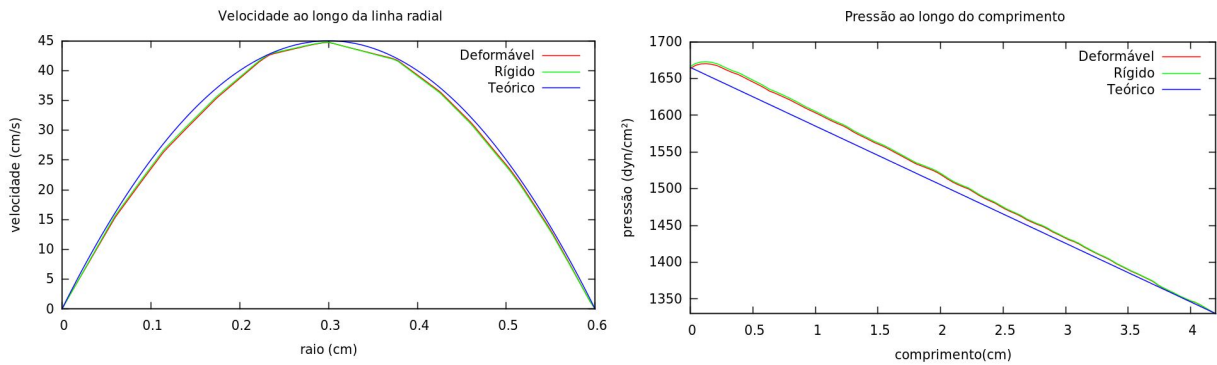


Figura 36 – Dimensões e malha para o problema da artéria idealizada

Tabela 18 – Propriedades e parâmetros para o problema da artéria idealizada

ρ	μ	E	ρ^s	ν^s	k	ζ	V_{max}	p_{out}
g/cm ³	dyn.s/cm ²	dyn/cm ²	g/cm ³	-	-	cm	cm/s	dyn/cm ²
1.06	0.04	4.07×10^6	1.09	0.5	5/6	0.03	45	1330



(a) Velocidade em uma linha radial no centro da artéria (b) Velocidade em uma linha ao longo do comprimento da artéria

Figura 37 – Perfis de velocidade e pressão na artéria idealizada

Em condições reais, o sangue fluindo através das artérias não é exatamente um escoamento laminar, mas para as condições impostas o escoamento é considerado como laminar. Após a norma relativa da velocidade atingir uma tolerância pré-estabelecida de 10^{-5} , o escoamento atinge o regime permanente. Devido a isso, o movimento do fluido é considerado como um escoamento de Poiseuille (PRITCHARD, 2010), em que a queda de pressão varia linearmente com o comprimento da artéria e o perfil de velocidade ao longo da linha de corrente permanece inalterado, variando apenas com na direção radial. Assim sendo, as soluções obtidas numericamente, após atingir o estado de equilíbrio, podem ser comparadas com equações presentes nas bibliografias (PRITCHARD, 2010). O perfil de

velocidade e pressão teórico são dados por:

$$p(x) = p_0 + cx,$$

$$v(r) = V_{max} \left(1 - (r/R)^2\right),$$

onde $p_0 = 1665 \text{ dyn/cm}^2$, $c = -79.762 \text{ dyn/cm}^3$ é o gradiente de pressão axial, R é o raio da artéria, r é a distância radial e V_{max} se encontra na Tab. 18.

A Fig. 37 apresenta o perfil de velocidade na linha radial ($x = 2.1 \text{ cm}$) e a pressão ao longo do comprimento para os problemas rígido e deformável até que uma tolerância pré-estabelecida de 10^{-5} seja atingida para a norma relativa da velocidade. Podemos observar que após atingir a tolerância, tanto o caso rígido como o deformável possuem a configuração parabólica próximas do teórico. Além disso, a velocidade na interface é zero e a pressão decai linearmente até o valor imposto na saída. Nota-se que, próximo a entrada a pressão apresenta um perfil pouco diferente do teórico. Isso pode ser devido a imprecisões numéricas. Esses resultados são esperados, pois devido a imposição de fluxo constante, o escoamento é consequência do gradiente constante de pressão fazendo com que o problema atinja o estado de equilíbrio, obtendo um deslocamento fixo.

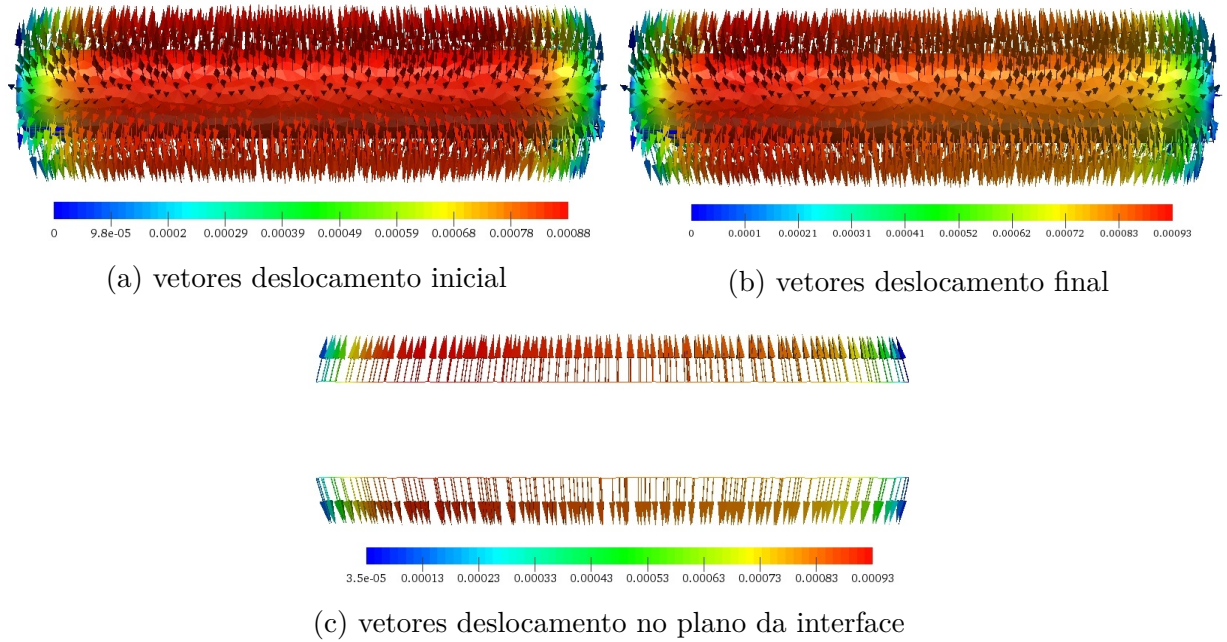


Figura 38 – Vetores deslocamento para o problema da artéria idealizada

A Fig. 38 mostra os vetores do deslocamento. (FIGUEROA et al., 2006) sugerem que à uma determinada distância da entrada e da saída, condições de parede rígida devem ser impostas para evitar que o deslocamento seja afetado pelas condições de limite. Dessa forma, a artéria foi truncada em $0.2 < x < 4.0$, para fins de visualização dos deslocamentos. O deslocamento inicial representa a inicialização do problema deformável com a pré-carga proveniente da pressão do caso rígido. Já o deslocamento final, representa o deslocamento

após o problema atingir a tolerância pré-fixada. Foi gerado um plano longitudinal no centro da artéria para visualização dos vetores deslocamento neste plano. Observa-se que o deslocamento inicial e a final são próximas, pois a inicialização do problema começou com uma carga de pressão inicial, proveniente do caso hidrostático, e as paredes dos problemas fisiológicos possuem uma carga inicial na direção radial e longitudinal. Os vetores de deslocamento apontam radialmente para fora da artéria, devido a força de pressão que o fluido exerce na interface. O deslocamento máximo obtido no problema foi de 0.00093 cm e o percentual de deformação ficou em torno de 3.1%, não ultrapassando o limite de pequenas deformações que é de 5% (FIGUEROA et al., 2006).

5.2.3 Artéria Aneurismática

Neste experimento vamos considerar uma velocidade pulsátil mapeada por um perfil parabólico na entrada, a fim de obter o comportamento do método para uma situação mais realista, uma vez que fluxos constantes não representam boas condições de contorno para modelos de artéria. A velocidade na entrada é dada por: $v_{in} = V_m(r^2 - y^2 - z^2)/r^2(1 + \sin(2\pi t/T))$, onde $V_m = 12.5$ cm/s, $r = 0.3$ cm, t é o tempo atual e $T = 0.2$ segundos é o período. Na saída foi considerada uma pressão constante $p_{out} = 1330$ dyn.s/cm². A geometria da artéria tem raio de 0.3 cm e comprimento de 4.2 cm, conforme Fig. 39. No ponto médio da artéria existe uma esfera com 0.45 cm de raio representando o aneurisma. A malha da artéria aneurismática é formada por 17241 elementos e 3548 nós, sendo considerado um passo de tempo igual a 5.076×10^{-4} segundos. As propriedades e demais parâmetros considerados encontram-se na Tab. 18.

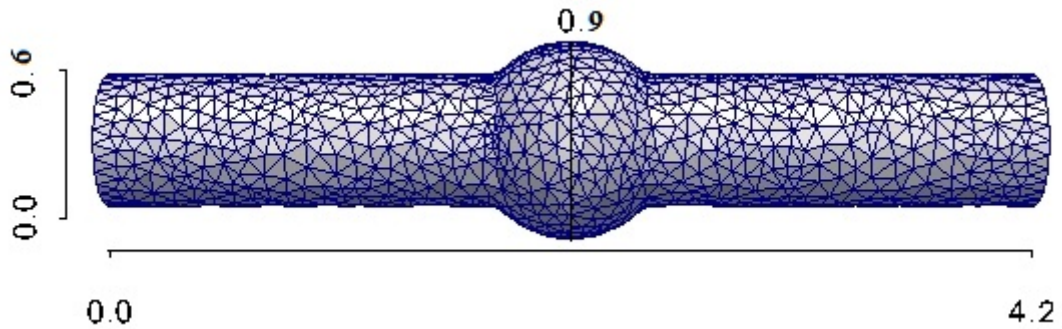


Figura 39 – Parâmetros geométricos e malha de elementos finito para artéria aneurismática

A Fig. 40 mostra os campos de pressão e velocidade em 0.05 segundos para os problemas rígido e deformável e a Fig. 41 apresentam os vetores deslocamento e vetores velocidade na interface e no fluido para o mesmo tempo de 0.05 segundos. Podemos observar na Fig. 40 que o comportamento da velocidade no caso rígido e deformável são diferentes, principalmente na região aneurismática. Já para o campo de pressão, temos uma pequena diferença nessa região, sendo que o caso deformável apresenta uma maior

pressão. Esta situação não ocorreu no caso de ser considerado fluxo constante e ausência de patologias nas artérias, conforme observa-se na Fig.37.

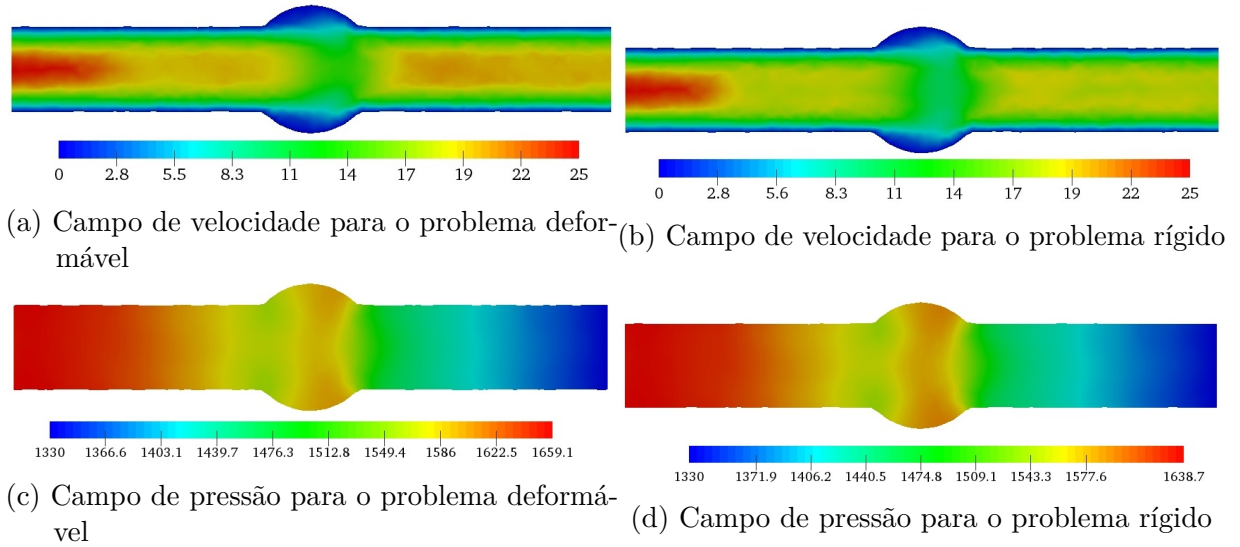


Figura 40 – Campo de velocidade e pressão para o problema da artéria aneurismática

Próximo a região aneurismática, observamos um aumento da velocidade na interface, caracterizando a irregularidade do escoamento na presença do aneurisma (Fig. 41d). Devido a esse fato, são geradas concentrações de tensões em determinadas regiões ocasionando um possível rompimento da parede. Na Fig. 41a observamos que os maiores deslocamentos estão no início e no fim do aneurisma, ocorrendo um aumento da tensão na parede da artéria nessa região. O máximo deslocamento no experimento foi de 0.0014 cm obtendo um percentual máximo de 4.67% de deformação, não ultrapassando o limite esperado para pequenas deformações. Além disso, os percentuais de deformação são superiores aos obtidos no experimento anterior, confirmando que as artérias estão propensas a maiores esforços na presença de patologias.

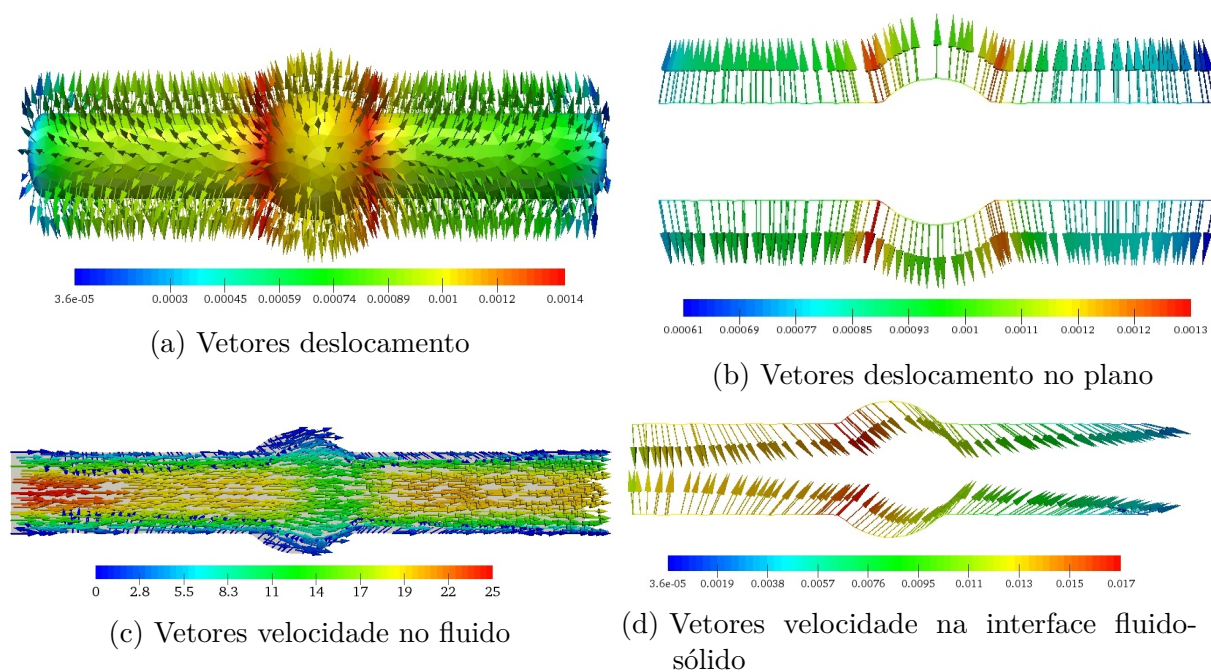


Figura 41 – Vetores velocidade e deslocamento para o problema da artéria aneurismática

6 Conclusões

Este trabalho apresentou um estudo de implementações, utilizando a biblioteca de domínio público FEniCS, para os métodos de projeção Chorin, IPCS, CSS e para os métodos estabilizados SUPG/PSPG e LSIC aplicados as equações de Navier-Stokes incompressíveis. Além disso, foram estudadas implementações via biblioteca FEniCS de formulações estabilizadas para o método de impulso acoplado em problemas de interação fluido-estrutura.

Para o estudo comparativo dos métodos de projeção e método SUPG/PSPG + LSIC, foram consideradas vários tipos de refinamento de malhas e problemas distintos da literatura, a fim de avaliar a acurácia e precisão dos métodos. O método de projeção IPCS foi o que obteve o melhor desempenho sendo o mais rápido e o mais preciso. A formulação estabilizada SUPG/PSPG e LSIC mostrou o pior desempenho em termos do tempo de processamento. Porém a tolerância considerada para a solução do sistema linear foi muito grande, e o método demandou um elevado número de iterações, visto que formulações estabilizadas tendem a piorar o condicionamento dos sistemas resultantes. Apesar disso, considerando uma tolerância maior, boas soluções podem ser obtidas, em tempo computacional menor.

A técnica adaptativa de avanço no tempo foi aplicada em dois problemas clássicos, utilizando o método de projeção IPCS e o método SUPG/PSPG + LSIC. Ambos os métodos apresentaram resultados similares. Isso porque a escala de comparação nessas aplicações são maiores do que aquelas do estudo comparativo de métodos, não conseguindo distinguir a acurácia entre os métodos. O método SUPG apresentou tempo computacional superior ao método IPCS, mesmo utilizando PID. Além disso, a técnica de controle do tempo obteve um bom desempenho, reduzindo consideravelmente o tempo computacional em que, na maioria das vezes, a redução foi acima de 30%.

O método CMM-FSI mostrou ser um método eficiente e simples para a simulação de problemas deformáveis tridimensionais. O método faz o acoplamento das equações do sólido ao problema variacional do fluido como uma condição de contorno. Sendo assim, é acrescido pouco esforço computacional em relação ao problema rígido. Além disso, o problema deformável depende fortemente da forma de inicialização da deformação, sendo desejável optar por uma pré carga, eliminando as oscilações iniciais. Embora o método seja válido apenas para sólidos de parede fina e trabalhe com pequenas deformações, o esforço computacional é muito inferior a abordagem ALE.

O uso da biblioteca FEniCS possibilitou implementações relativamente simples, gerenciando as dificuldades com as equações de Navier-Stokes, problemas IFE, condições

de contorno, dentre demais aspectos, possibilitando ao pesquisador para se aprofundar nos aspectos mais relevantes das aplicações em estudos.

Sugestões para trabalhos futuros

Nos experimentos apresentados para as equações de Navier-Stokes, não foi realizado um estudo aprofundado de importantes parâmetros tais como: tolerâncias para o método GMRES, tolerância para o método de Newton. Os valores utilizados resultaram em conclusões desfavoráveis para o método estabilizado SUPG/PSPG + LSIC. Assim, uma análise mais criteriosa considerando malhas menos refinadas e diferentes valores de tolerância devem ser realizadas.

As condições de contorno impostas e as geometrias consideradas para os problemas de interação fluido-estrutura foram simplificadas. Assim sendo, inserir condições de contorno realistas e geometrias complexas é um caminho natural para simular modelos mais realistas.

Referências

- BAZILEVS, Y.; TAKIZAWA, K.; TEZDUYAR, T. *Computational Fluid-Structure Interaction: Methods and Applications*. Wiley, 2013. (Wiley Series in Computational Mechanics). ISBN 9781118483572. Disponível em: <<https://books.google.com.br/books?id=R-3-5ZNSQnMC>>. Citado 2 vezes nas páginas 13 e 14.
- BREBBIA, C. *Modelling in Medicine and Biology VIII*. Wessex Institute of Technology, 2009. (WIT transactions on biomedicine and health). ISBN 9781845641832. Disponível em: <<https://books.google.mv/books?id=cNeWylqevOIC>>. Citado na página 14.
- BREZZI, F. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, Dunod, v. 8, n. R2, p. 129–151, 1974. Disponível em: <<http://eudml.org/doc/193255>>. Citado na página 20.
- BROOKS, A. N.; HUGHES, T. J. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, v. 32, n. 1, p. 199 – 259, 1982. ISSN 0045-7825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0045782582900718>>. Citado na página 15.
- BROWN, D. L.; CORTEZ, R.; MINION, M. L. Accurate projection methods for the incompressible navier-stokes equations. *Journal of Computational Physics*, v. 168, n. 2, p. 464 – 499, 2001. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0021999101967154>>. Citado na página 22.
- CANUTO, C. et al. *Spectral Methods: Fundamentals in Single Domains*. Springer Berlin Heidelberg, 2007. (Scientific Computation). ISBN 9783540307266. Disponível em: <<https://books.google.com.br/books?id=DFJB0kiq0CQC>>. Citado na página 57.
- CEBRAL, J. R. et al. Blood-flow models of the circle of willis from magnetic resonance data. v. 47, p. 369–386, 12 2003. Citado na página 14.
- CHORIN, A. Numerical solution of the navier-stokes equations. v. 22, p. 745–762, 10 1968. Citado 2 vezes nas páginas 15 e 21.
- COOK, R. D. et al. *Concepts and Applications of Finite Element Analysis, 4th Edition*. 4. ed. [S.l.]: Wiley, 2001. ISBN 0471356050. Citado na página 30.
- COURT, S.; FOURNIÉ, M. A fictitious domain finite element method for simulations of fluid-structure interactions: The navier-stokes equations coupled with a moving solid. *Journal of Fluids and Structures*, v. 55, p. 398–408, 2015. Citado na página 14.
- COURT, S.; FOURNIÉ, M.; LOZINSKI, A. A fictitious domain approach for the stokes problem based on the extended finite element method. *International Journal for Numerical Methods in Fluids*, Wiley Online Library, v. 74, n. 2, p. 73–99, 2014. Citado na página 14.

DONEA, J.; HUERTA, A. *Finite Element Methods for Flow Problems*. John Wiley & Sons, 2003. (Finite Element Methods for Flow Problems). ISBN 9780471496663. Disponível em: <<https://books.google.com.br/books?id=S4URqrTtSXoC>>. Citado na página 20.

ELIAS, R. N. *Método tipo-Newton inexatos para a solução de problemas não-lineares resultantes da formulação SUPG/PSPG das equações de Navier-Stokes incompressíveis em regime permanente*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 5 2003. Tese de Mestrado em Engenharia Civil. Disponível em: <<http://www.nacad.ufrj.br/~rnelias/msc.html>>. Citado na página 68.

ELIAS, R. N. *Estrutura de dados por arestas para a simulação paralela de escoamentos incompressíveis pelo método estabilizado de elementos finitos*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 12 2007. Tese de Doutorado em Engenharia Civil. Disponível em: <http://www.dominionpublico.gov.br/pesquisa/DetalheObraForm.do?select=_{}action=&co=_{}obra=8>. Citado 2 vezes nas páginas 13 e 18.

ETHIER, C. R.; STEINMAN, D. A. Exact fully 3d navier–stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, John Wiley & Sons, Ltd, v. 19, n. 5, p. 369–375, 1994. ISSN 1097-0363. Disponível em: <<http://dx.doi.org/10.1002/fld.1650190502>>. Citado na página 58.

FIGUEROA, C. et al. A coupled momentum method to model blood flow in deformable arteries. 01 2004. Citado 2 vezes nas páginas 15 e 78.

FIGUEROA, C. A. et al. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer Methods in Applied Mechanics and Engineering*, v. 195, n. 41, p. 5685 – 5706, 2006. ISSN 0045-7825. John H. Argyris Memorial Issue. Part II. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S004578250500513X>>. Citado 13 vezes nas páginas 9, 13, 14, 15, 19, 26, 27, 28, 30, 31, 33, 80 e 81.

GHIA, U.; GHIA, K.; SHIN, C. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, v. 48, n. 3, p. 387 – 411, 1982. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999182900584>>. Citado 2 vezes nas páginas 62 e 63.

GLAGOV, S. et al. Hemodynamics and atherosclerosis. insights and perspectives gained from studies of human arteries. v. 112, p. 1018–31, 11 1988. Citado na página 14.

GODA, K. A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *Journal of Computational Physics*, v. 30, n. 1, p. 76 – 95, 1979. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999179900883>>. Citado 2 vezes nas páginas 15 e 22.

GOMES, H. C. *Método dos elementos finitos com fronteiras imersas aplicado a problemas de dinâmica dos fluidos e interação fluido-estrutura*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 4 2013. Tese de Doutorado em Engenharia de Estruturas. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3144/tde-26122013-150059/>>. Citado na página 13.

GRIEBEL, M.; DORNSHEIFER, T.; NEUNHOEFFER, T. *Numerical Simulation in Fluid Dynamics: A Practical Introduction*. Society for Industrial and Applied Mathematics, 1998. (Monographs on Mathematical Modeling and Computation). ISBN 9780898713985. Disponível em: <<https://books.google.com.br/books?id=ehuzHQT5GcC>>. Citado 3 vezes nas páginas 68, 69 e 70.

GUERMOND, J.; MINEV, P.; SHEN, J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, v. 195, n. 44, p. 6011 – 6045, 2006. ISSN 0045-7825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0045782505004640>>. Citado 3 vezes nas páginas 16, 22 e 23.

GUERMOND, J.; SHEN, J. A new class of truly consistent splitting schemes for incompressible flows. *Journal of Computational Physics*, v. 192, n. 1, p. 262 – 276, 2003. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0021999103003784>>. Citado 2 vezes nas páginas 16 e 23.

HELMIG, J. *A Numerical Investigation of the Stability of Different Schemes for a Simplified Fluid-Structure Interaction Scheme with Application to Physiological Problems*. 2013. Bachelor Thesis. Citado 2 vezes nas páginas 13 e 32.

HUGHES, T. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987. ISBN 9780133170252. Disponível em: <<https://books.google.com.br/books?id=pF-IQgAACAAJ>>. Citado 2 vezes nas páginas 15 e 31.

KAMENSKY, D. et al. An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, v. 284, p. 1005–1053, 2015. ISSN 0045-7825. Isogeometric Analysis Special Issue. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0045782514004101>>. Citado na página 13.

KUHL, E.; HULSHOFF, S.; BORST, R. de. An arbitrary lagrangian eulerian finite-element approach for fluid-structure interaction phenomena. *International Journal for Numerical Methods in Engineering*, John Wiley & Sons, Ltd., v. 57, n. 1, p. 117–142, 2003. ISSN 1097-0207. Disponível em: <<http://dx.doi.org/10.1002/nme.749>>. Citado na página 14.

KUMAGAI, I. On the anatomy of mantle plumes: effect of the viscosity ratio on entrainment and stirring. *Earth and Planetary Science Letters*, v. 198, n. 1, p. 211 – 224, 2002. ISSN 0012-821X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0012821X02005137>>. Citado na página 13.

LAGANÁ, K. et al. Multiscale modeling as a tool to prescribe realistic boundary conditions for the study of surgical procedures. v. 39, p. 359–64, 02 2002. Citado na página 14.

LARSON M. B., B. F. *The Finite Element Method: Theory, Implementation and Applications*. Springer-Verlag Berlin Heidelberg, 2013. (Texts in Computational Science and Engineering). ISBN 978-3-642-33287-6. Disponível em: <<http://www.springer.com/us/book/9783642332869>>. Citado na página 26.

LEVAL, M. de et al. Use of computational fluid dynamics in the design of surgical procedures: Application to the study of competitive flows in

- cavopulmonary connections. *The Journal of Thoracic and Cardiovascular Surgery*, v. 111, n. 3, p. 502 – 513, 1996. ISSN 0022-5223. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022522396703021>. Citado na página 14.
- LOGG, A.; MARDAL, K.; WELLS, G. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer Berlin Heidelberg, 2012. (Lecture Notes in Computational Science and Engineering). ISBN 9783642230998. Disponível em: https://books.google.com.br/books?id=ASWN_{VRr1N. Citado 10 vezes nas páginas 9, 16, 18, 24, 34, 52, 53, 55, 56 e 61.
- MAGOSSO, E.; URSINO, M. Cardiovascular response to dynamic aerobic exercise: A mathematical model. *Medical and Biological Engineering and Computing*, v. 40, n. 6, p. 660–674, Nov 2002. ISSN 1741-0444. Disponível em: <https://doi.org/10.1007/BF02345305>. Citado na página 14.
- MIGLIAVACCA, F. et al. Computational fluid dynamic and magnetic resonance analyses of flow distribution between the lungs after total cavopulmonary connection. *IEEE transactions on bio-medical engineering*, v. 46 4, p. 393–9, 1999. Citado na página 14.
- MITTAL, S.; TEZDUYAR, T. Massively parallel finite element computation of incompressible flows involving fluid-body interactions. *Computer Methods in Applied Mechanics and Engineering*, v. 112, n. 1, p. 253 – 282, 1994. ISSN 0045-7825. Disponível em: <http://www.sciencedirect.com/science/article/pii/0045782594900299>. Citado na página 14.
- MITTAL, S.; TEZDUYAR, T. E. Parallel finite element simulation of 3d incompressible flows: Fluid-structure interactions. *International Journal for Numerical Methods in Fluids*, Wiley Online Library, v. 21, n. 10, p. 933–953, 1995. Citado na página 14.
- MOORE, W. *Vascular surgery: a comprehensive review*. Saunders, 1991. (Vascular Surgery: A Comprehensive Review, v. 331). ISBN 9780721635125. Disponível em: <https://books.google.com.br/books?id=IZBsAAAAMAAJ>. Citado na página 14.
- OSHIMA, M. et al. Finite element simulation of blood flow in the cerebral artery. v. 191, p. 661–671, 12 2001. Citado na página 14.
- PANDIT, S. K.; KALITA, J. C.; DALAL, D. A transient higher order compact scheme for incompressible viscous flows on geometries beyond rectangular. *Journal of Computational Physics*, v. 225, n. 1, p. 1100 – 1124, 2007. ISSN 0021-9991. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0021999107000319>. Citado na página 52.
- PERKTOLD, K.; RAPPITSCH, G. Computer simulation of local blood flow and vessel mechanics in a compliant carotid artery bifurcation model. v. 28, p. 845–56, 08 1995. Citado na página 14.
- PERKTOLD, K.; RESCH, M.; PETER, R. O. Three-dimensional numerical analysis of pulsatile flow and shear stress in the carotid artery bifurcation. v. 24, p. 409–20, 02 1991. Citado na página 14.
- PRITCHARD, P. *Fox and McDonald's Introduction to Fluid Mechanics*, 8th

- Edition*. John Wiley & Sons, 2010. ISBN 9781118139455. Disponível em: <https://books.google.com.br/books?id=RdIbAAAAQBAJ>. Citado na página 79.
- SALMON, S.; MARC, T.; GERBEAU, J.-F. Medical image - based computational model of pulsatile flow in saccular aneurisms. v. 37, 07 2003. Citado na página 14.
- SANCHES, R.; CODA, H. Análise numérica bidimensional de interação fluido-estrutura. *International journal for numerical methods in engineering*, Cadernos de engenharia de estruturas, São Carlos, v. 10, n. 43, p. 33–54, 2009. Citado na página 14.
- SELIM, K.; LOGG, A. Simulating heart valve dynamics in fenics. 08 2017. Citado na página 13.
- SHAKIB, F.; HUGHES, T. J.; JOHAN, Z. A new finite element formulation for computational fluid dynamics: X. the compressible euler and navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, v. 89, n. 1, p. 141 – 219, 1991. ISSN 0045-7825. Second World Congress on Computational Mechanics. Disponível em: <http://www.sciencedirect.com/science/article/pii/0045782591900414>. Citado na página 21.
- SHOJIMA, M. et al. Magnitude and role of wall shear stress on cerebral aneurysm: Computational fluid dynamic study of 20 middle cerebral artery aneurysms. v. 35, p. 2500–5, 11 2004. Citado na página 14.
- STEIN, K. et al. Parachute fluid–structure interactions: 3-d computation. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 190, n. 3, p. 373–386, 2000. Citado na página 13.
- STEINMAN, D. A. Image-based computational fluid dynamics modeling in realistic arterial geometries. *Annals of Biomedical Engineering*, v. 30, n. 4, p. 483–497, Apr 2002. ISSN 1573-9686. Disponível em: <https://doi.org/10.1114/1.1467679>. Citado na página 14.
- STERGIOPULOS, N.; YOUNG, D.; ROGGE, T. Computer simulation of arterial flow with applications to arterial and aortic stenoses. *Journal of Biomechanics*, v. 25, n. 12, p. 1477 – 1488, 1992. ISSN 0021-9290. Disponível em: <http://www.sciencedirect.com/science/article/pii/002192909290060E>. Citado na página 14.
- TAYLOR, C. A. et al. Predictive medicine: computational techniques in therapeutic decision-making. *Computer aided surgery*, Taylor & Francis, v. 4, n. 5, p. 231–247, 1999. Citado na página 14.
- TAYLOR, C. A.; HUGHES, T. J. R.; ZARINS, C. K. Finite element modeling of three-dimensional pulsatile flow in the abdominal aorta: Relevance to atherosclerosis. *Annals of Biomedical Engineering*, v. 26, n. 6, p. 975–987, Nov 1998. ISSN 1573-9686. Disponível em: <https://doi.org/10.1114/1.140>. Citado na página 14.
- TEMAM, T. Sur l’approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). v. 33, p. 377–385, 1 1969. Citado 2 vezes nas páginas 15 e 21.

- TEZDUYAR, T. Stabilized finite element formulations for incompressible flow computations. In: HUTCHINSON, J. W.; WU, T. Y. (Ed.). Elsevier, 1991, (Advances in Applied Mechanics, Supplement C). p. 1 – 44. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0065215608701534>. Citado na página 15.
- TEZDUYAR, T.; BEHR, M.; LIOU, J. A new strategy for finite element computations involving moving boundaries and interfaces-the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, v. 94, n. 3, p. 339 – 351, 1992. ISSN 0045-7825. Disponível em: <http://www.sciencedirect.com/science/article/pii/004578259290059S>. Citado na página 14.
- TEZDUYAR, T. E.; OSAWA, Y. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, v. 190, n. 3, p. 411 – 430, 2000. ISSN 0045-7825. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0045782500002115>. Citado 3 vezes nas páginas 15, 20 e 21.
- VALLI, A. M. P.; CAREY, G. F.; COUTINHO, A. L. G. A. Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Communications in Numerical Methods in Engineering*, v. 18, n. 2, p. 131 – 139, 2002. ISSN 1099 - 0887. Disponível em: <http://dx.doi.org/10.1002/cnm.475>. Citado 2 vezes nas páginas 16 e 25.
- VIGNON-CLEMENTEL, I. E. et al. Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. *Computer Methods in Applied Mechanics and Engineering*, v. 195, n. 29, p. 3776 – 3796, 2006. ISSN 0045-7825. Absorbing Boundary Conditions. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0045782505002586>. Citado na página 13.
- VOSSE, F. van de et al. Finite-element-based computational methods for cardiovascular fluid-structure interaction. *Journal of Engineering Mathematics*, v. 47, n. 3, p. 335–368, Dec 2003. ISSN 1573-2703. Disponível em: <https://doi.org/10.1023/B:ENGI.0000007985.17625.43>. Citado na página 14.
- WEIMAN, E.; LIU, J.-G. Projection method i: Convergence and numerical boundary layers. *Society for Industrial and Applied Mathematics*, v. 32, n. 4, p. 1017 – 1057, 1995. ISSN 1095-7170. Disponível em: <http://epubs.siam.org/doi/abs/10.1137/0732047>. Citado na página 21.
- WHITE, F. *Viscous Fluid Flow*. McGraw-Hill, 1991. (McGraw-Hill series in mechanical engineering). ISBN 9780070697126. Disponível em: <https://books.google.com.br/books?id=G6IeAQAIAAJ>. Citado na página 53.
- WOMERSLEY, J. Xxiv. oscillatory motion of a viscous liquid in a thin-walled elastic tube—i: The linear approximation for long waves. v. 46, p. 199–221, 02 1955. Citado na página 28.
- WREN, G. P. et al. Simulation of flow problems with moving mechanical components, fluid–structure interactions and two-fluid interfaces. *International journal for numerical*

methods in fluids, Wiley Online Library, v. 24, n. 12, p. 1433–1448, 1997. Citado na página [13](#).

ZHOU, J.; FUNG, Y. The degree of nonlinearity and anisotropy of blood vessel elasticity. v. 94, p. 14255–60, 12 1997. Citado na página [30](#).