

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

VITOR SPADETO VENTURIN

**SIMULAÇÃO COMPUTACIONAL DE BIO-ESCOAMENTOS
UTILIZANDO O OPENFOAM**

**VITÓRIA
2014**

VITOR SPADETO VENTURIN

**SIMULAÇÃO COMPUTACIONAL DE BIO-ESCOAMENTOS
UTILIZANDO O OPENFOAM**

Projeto de Graduação apresentado ao Departamento de Engenharia Mecânica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Mecânico.

Orientador: Prof. Dr. Juan Sergio Romero Saenz

**VITÓRIA
2014**

VITOR SPADETO VENTURIN

**SIMULAÇÃO COMPUTACIONAL DE BIO-ESCOAMENTOS
UTILIZANDO O OPENFOAM**

Projeto de Graduação apresentado ao Departamento de Engenharia Mecânica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Mecânico.

Aprovado em 22 de dezembro de 2014.

COMISSÃO EXAMINADORA:

Prof. Dr. Juan Sergio Romero Saenz
Universidade Federal do Espírito Santo
Orientador

Eng. Manuel E. E. Salazar Schiaffino
Universidade Federal do Espírito Santo
Examinador

Eng. Jaderson Nunes Pretti
Universidade Federal do Espírito Santo
Examinador

AGRADECIMENTOS

Agradeço primeiramente ao Senhor Jesus Cristo, que sempre existiu, que é a expressão completa de Deus na Terra (*“O qual é a imagem do Deus invisível, o primogênito de toda a criação - Cl 1:15*). Ele que tem me dado todo o sustento diante de todas as necessidades.

Agradeço também aos meus pais, que me acompanharam durante todo o curso, além de me darem muita força e incentivo.

Ao Prof. Dr. Juan Sergio Romero Saenz pelas orientações, conselhos e toda a atenção dedicada a este trabalho.

Aos amigos que conheci durante o projeto, sendo pessoas de várias nações, que direta ou indiretamente me instruíram. E a todos que, de uma forma ou de outra, contribuíram para a conclusão dessa etapa.

LISTA DE FIGURAS

Figura 4-1: Lista padrão de diretórios de um caso no OpenFOAM. Fonte: Guia de Usuário do OpenFOAM	17
Figura 4-2: Arquivo DecomposePar com divisão de malha em 4 partes	23
Figura 5-1: Exemplo de Malha Estruturada.....	26
Figura 5-2: Exemplo de Malha Não-Estruturada.....	27
Figura 5-3: Ortogonalidade	28
Figura 5-4: Razão de Aspecto.....	28
Figura 5-5: Assimetria.....	29
Figura 5-6: Interface Gráfica do Salome.....	30
Figura 6-1: Geometria do CASO I.....	31
Figura 6-2: Código da malha estruturada utilizada no CASO I	33
Figura 6-3: Pressões para o CASO I.....	34
Figura 6-4: Condições de contorno para velocidade de entrada máxima igual a 0,0025 m/s	35
Figura 6-5: Condições de contorno para velocidade de entrada máxima igual a 0,8 m/s.....	35
Figura 6-6: <i>fvSchemes</i> para o CASO I.....	37
Figura 6-7: <i>fvSolution</i> para o CASO I.....	39
Figura 6-8: Comparação do perfil de velocidade na saída utilizando o modelo Carreau-Yasuda e velocidade de entrada de 0,8 m/s.....	41
Figura 6-9: Comparação do perfil de velocidade na saída utilizando o modelo Carreau-Yasuda e velocidade de entrada de 0,025 m/s	42
Figura 6-10: Comparação do perfil de velocidade na saída utilizando o modelo Newtoniano e velocidade de entrada de 0,8 m/s.....	42
Figura 6-11: Comparação do perfil de velocidade na saída utilizando o modelo Newtoniano e velocidade de entrada de 0,025 m/s	43
Figura 6-12: Comparação do perfil de velocidade na saída utilizando o modelo Power-Law e velocidade de entrada de 0,8 m/s.....	43
Figura 6-13: Comparação do perfil de velocidade na saída utilizando o modelo Power-Law e velocidade de entrada de 0,025 m/s	44
Figura 6-14: Comparativo entre linhas de corrente obtidas na simulação e obtidas por Fodor para velocidade máxima de 0,0025 m/s	45
Figura 6-15: Comparativo entre linhas de corrente obtidas na simulação e obtidas por Fodor para velocidade máxima de 0,025 m/s	46
Figura 6-16: Comparativo entre linhas de corrente obtidas na simulação e obtidas por Fodor para velocidade máxima de 0,8 m/s.....	47
Figura 6-17: Anastomose vascular distal. O diâmetro D da artéria anfitriã é o mesmo da artéria secundária. A distância de A até o centro da estenose é d. (a) Perspectiva; (b) Vista do plano x-z de simetria	48
Figura 6-18: Parte da malha para distancia d igual a 3 geradas no SALOME.....	50
Figura 6-19: Condições de contorno para pressão na malha do CASO II	53
Figura 6-20: Condições de contorno para velocidade na malha do CASO II	54
Figura 6-21: Trecho do código BirdCarreau.C modificado	55
Figura 6-22: <i>transportProperties</i> do Caso II.....	56
Figura 6-23: <i>fvSchemes</i> para o Caso II.....	57
Figura 6-24: <i>fvSolution</i> para o Caso II	59

Figura 6-25: Localização dos pontos Pa, Pb e Pc para o CASO II	61
Figura 6-26: Comparação entre os perfis de velocidade D1.5-Q12 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	62
Figura 6-27: Comparação entre os perfis de velocidade D1.5-Q34 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	63
Figura 6-28: Comparação entre os perfis de velocidade D1.5-Q78 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	64
Figura 6-29: Comparação entre os perfis de velocidade D2-Q12 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	65
Figura 6-30: Comparação entre os perfis de velocidade D2-Q34 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	66
Figura 6-31: Comparação entre os perfis de velocidade D2-Q78 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	67
Figura 6-32: Comparação entre os perfis de velocidade D3-Q12 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	68
Figura 6-33: Comparação entre os perfis de velocidade D3-Q34 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	69
Figura 6-34: Comparação entre os perfis de velocidade D3-Q78 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	70
Figura 6-35: Comparação entre os perfis de velocidade D1.5-Q12 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	71
Figura 6-36: Comparação entre os perfis de velocidade D1.5-Q34 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	72
Figura 6-37: Comparação entre os perfis de velocidade D1.5-Q78 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	73
Figura 6-38: Comparação entre os perfis de velocidade D2-Q12 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	74
Figura 6-39: Comparação entre os perfis de velocidade D2-Q34 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	75
Figura 6-40: Comparação entre os perfis de velocidade D2-Q78 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	76

Figura 6-41: Comparação entre os perfis de velocidade D3-Q12 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	77
Figura 6-42: Comparação entre os perfis de velocidade D3-Q34 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	78
Figura 6-43: Comparação entre os perfis obtidos para D3-Q78 na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho.....	79
Figura 6-44: Perfil de velocidades instantâneas para as duas entradas. A curva superior é o fluxo instantâneo na artéria secundária. A curva inferior corresponde ao fluxo instantâneo na artéria anfitriã. Fonte: Bertolotti e Deplano(2000)	81
Figura 6-45: Condições de contorno para velocidade.....	82
Figura 6-46: Perfis de velocidade média instantânea nas seções de entrada.....	83
Figura 6-47: Condições de contorno para pressões no CASO III	84
Figura 6-48: <i>transportProperties</i> para o CASO III.....	85
Figura 6-49: <i>fvSchemes</i> para o CASO III	86
Figura 6-50: Localização dos pontos Pa, Pb e Pc para o CASO III	88
Figura 6-51: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,24 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	89
Figura 6-52: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,48 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	90
Figura 6-53: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,64 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	91
Figura 6-54: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,88 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	92
Figura 6-55: Comparação entre perfis de velocidade nas linhas horizontais passando por Pa, Pb e Pc para o tempo de 0,24 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	93
Figura 6-56: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,48 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	94
Figura 6-57: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,64 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	95
Figura 6-58: Comparação entre perfis de velocidade nas linhas horizontais passando por Pa, Pb e Pc para o tempo de 0,88 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM	96

LISTA DE TABELAS

Tabela 4.1: Controle de tempo.....	20
Tabela 4.2: Controle de dados.....	20
Tabela 4.3: Passo de Tempo Ajustável.....	21
Tabela 4.4: Leitura de dados.....	22
Tabela 4.5: Funcionalidade Runtime carregável.....	22
Tabela 6.1: Caracterização das situações simuladas do CASO I.....	32
Tabela 6.2: Saída do <i>blockMesh</i>	41
Tabela 6.3: Descrição das vazões adotadas para as simulações.....	49
Tabela 6.4: Parametros de qualidade para a malha do CASO II – malha D1.5.....	51
Tabela 6.5: Parametros de qualidade para a malha do CASO II – malha D2.....	51
Tabela 6.6: Parametros de qualidade para a malha do CASO II – malha D3.....	51

SUMÁRIO

1	INTRODUÇÃO	11
2	OBJETIVOS	13
3	MODELOS ADOTADOS	13
3.1	O SANGUE	13
3.1.1	DENSIDADE	13
3.1.2	VISCOSIDADE	14
3.1.2.1	Modelos de viscosidade para fluidos Não-Newtonianos	15
3.1.3	LAMINAR	16
4	OPENFOAM	16
4.1	DIRETÓRIO <i>CONSTANT</i>	18
4.1.1	SUBDIRETÓRIO POLYMESH	18
4.2	DIRETÓRIO <i>SYSTEM</i>	19
4.2.1	<i>CONTROL</i> DICT	19
4.2.2	<i>FV</i> SCHEMES	22
4.2.3	<i>FV</i> SOLUTION	22
4.2.4	<i>DECOMPOSE</i> PARDICT	22
4.3	DIRETÓRIO <i>TIME DIRECTORIES</i>	24
5	GERAÇÃO DE MALHAS	25
5.1	PROCESSO DE CRIAÇÃO DA GEOMETRIA	25
5.1.1	PROCESSO DE GERAÇÃO DE MALHA	25
5.1.1.1	Malhas estruturadas	26
5.1.1.2	Malhas não-estruturadas	26
5.1.1.3	Malhas híbridas	27
5.2	PARÂMETROS DE QUALIDADE	27
5.2.1	ORTOGONALIDADE	27
5.2.2	RAZÃO DE ASPECTO	28
5.2.3	ASSIMETRIA	29
5.3	SALOME	29
6	METODOLOGIA E RESULTADOS	30
6.1	CASO I	31
6.1.1	DESCRIÇÃO DO CASO	31

6.1.2	SIMULAÇÃO	32
6.1.2.1	Geração de malha	32
6.1.2.2	Condições iniciais e de contorno	34
6.1.2.3	Propriedades físicas	36
6.1.2.4	Esquemas numéricos	37
6.1.2.5	Controle da solução	39
6.2	CASO II	48
6.2.1	DESCRIÇÃO DO CASO	48
6.2.2	SIMULAÇÃO	49
6.2.2.1	Geração de malha	50
6.2.2.2	Condições iniciais e de contorno	52
6.2.2.3	Propriedades físicas	54
6.2.2.4	Esquemas numéricos	56
6.2.2.5	Controle da solução	59
6.2.2.6	Resultados e comentários adicionais	60
6.3	CASO III	81
6.3.1	DESCRIÇÃO DO CASO	81
6.3.2	SIMULAÇÃO	81
6.3.2.1	Geração de malha	81
6.3.2.2	Condições iniciais de de contorno	82
6.3.2.3	Propriedades físicas	85
6.3.2.4	Esquemas numéricos	85
6.3.2.5	Controle da Solução	86
6.3.2.6	Resultados e comentários adicionais	87
7	CONSIDERAÇÕES FINAIS E SUGESTÕES	97
8	REFERENCIAS BIBLIOGRÁFICAS	99

RESUMO

O presente trabalho de conclusão de curso tem como tema a análise de viabilidade do OpenFOAM direcionado à simulação de bio-escoamentos, levando-se em conta a natureza do fluido, bem como as condições reais de escoamento. Foram feitas simulações computacionais considerando o sangue como Newtoniano e Não-Newtoniano. Foi considerado a característica laminar do sangue em todos os casos abordados (quase todo sistema circulatório humano tem comportamento laminar, o que é eficiente). Para geração de malhas, analisou-se geradores padrões do OpenFOAM, bem como geradores externos, como o programa livre SALOME. Manteve-se cautela na configuração de cada caso a fim de reduzir o custo computacional, embora se tenha buscado melhor exatidão nas soluções. Foram analisados casos em regime permanente e transiente.

Palavras-chave:

OpenFOAM, Simulação Computacional, Estenose, Sangue, Carreau-Yasuda, Power-Law.

ABSTRACT

This course conclusion work is on the topic OpenFOAM feasibility analysis directed to the simulation of bio-flows taking into account the nature of the fluid and the actual flow conditions. There was computer simulations considering the blood as Newtonian and non-Newtonian. It was considered laminar blood characteristic (almost all human circulatory system has laminar flow, which is effective). For mesh generation, it was analyzed OpenFOAM standards generators as well as external generators, such as opensource software SALOME. Caution was maintained in each simulation setting in order to reduce the computational cost, although it has sought better accuracy in the solutions. Cases were analyzed in steady-state and transient.

Keywords:

OpenFOAM, blood system, Computer Simulation, Finite volumes, stenosis, blood, Carreau-Yasuda, Power-Law.

1 INTRODUÇÃO

Em décadas passadas, o estudo da Dinâmica dos Fluidos era feito principalmente por métodos experimentais, pois geralmente a solução analítica era complexa e inviável. Os métodos experimentais muitas vezes requeriam muito tempo e custo financeiro, como por exemplo, a utilização de túneis de vento para o desenvolvimento de aviões ou estudo de aerodinâmica em veículos. Além disso, alguns fenômenos são de difícil reprodução experimental, gerando às vezes informações pouco precisas.

Isso foi revolucionado com o desenvolvimento da Dinâmica dos Fluidos Computacional, ou CFD (do inglês *Computational Fluids Dynamics*), que utiliza métodos numéricos e algoritmos para resolver problemas complexos que não apresentavam soluções analíticas. Com as simulações em computador pôde-se acelerar o desenvolvimento de um projeto, por exemplo, reduzindo o número de modelos testados em túneis de vento, economizando também no custo do projeto.

Então, os métodos computacionais surgiram como poderosas ferramentas para a modelagem de fluxo de sangue, de pressão, e de deformação de vasos do sistema cardiovascular. Aplicações na pesquisa de doenças, *design* de dispositivos médicos, e no planejamento cirúrgico têm exigido métodos numéricos cada vez mais sofisticados que vão desde modelos unidimensionais até às árvores estruturadas de sistemas arteriais para simulação tridimensional de escoamentos. Alguns métodos são tão avançados que incluem métodos de iteração fluido-estrutura, isto é, considerando a deformação da malha em função da pressão momentânea do sangue na artéria. Destes métodos, aglomerados sistemas de equações (como as equações de Navier-Stokes) são resolvidos. O fluxo de sangue e ondas de pressão no sistema cardiovascular podem ser modelados a fim de se resolver as equações que governam um fluido incompressível ou compressível, em um domínio Newtoniano ou não-Newtoniano.

A modelagem do fluxo sanguíneo e a propagação da onda de pressão no sistema cardiovascular é baseada na resolução numérica das equações que governam o fluxo sanguíneo. Em estados normais e de doença em crianças e adultos, a quantificação tridimensional de fenômenos de fluxo de campos de pressão é importante para se entender a resposta do sistema cardiovascular a forças biomecânicas. Além disso, a

capacidade de se simular adequadamente o fluxo e a pressão é importante para a modelagem de desempenho de dispositivos, tais como válvulas cardíacas, LVADs (dispositivos de assistência ventricular esquerda), filtros, *stents* e endopróteses.

As simulações computacionais tem tido caminho aberto para tais aplicações, visto que os métodos atuais são invasivos. Não obstante, as simulações computacionais são não-invasivas. Dados de fluxo e pressão tridimensionais também podem fornecer informações importantes para determinar a significância de uma obstrução ou prevenção de efeitos maiores no paciente.

As simulações aplicadas para bio-escoamentos estão ainda em estágio de avaliação e desenvolvimento. Muitos obstáculos ainda estão presentes. Um deles é a obtenção de valores reais de medições para serem inseridas como condições de contorno (especialmente medições que variam com o tempo são geralmente impraticáveis, para o fluxo, bem como para a pressão). Além disso, mesmo que se cada fluxo ou pressão forem adquiridas simultaneamente, é muito difícil sincronizar essas formas de onda de uma maneira consistente.

Estas medições se tornam mais difíceis caso se tenha o efeito da elasticidade das artérias. Na verdade, capacidades de simulação atuais incluem tanto o pressuposto da parede rígida ou a modelagem de deformação de parede com informações limitadas sobre suas propriedades. Estas propriedades geralmente são o módulo de elasticidade, espessura e variações espaciais das artérias. E mesmo que estas sejam bem conhecidas, ainda se tem o fato de que os órgãos e demais partes do corpo humano interferem diretamente na dilatação das artérias, uma vez que estes agentes externos às artérias atuam com pressão sobre a mesma.

Embora se tenham muitos desafios a serem superados, a melhor estratégia é usar métodos unidimensionais para modelar os vasos a jusante e montante e gerar assim as condições de contorno para os métodos mais computacionalmente intensivos, os tridimensionais (usados para as principais artérias).

Sem dúvida alguma, atualmente os pacotes CFD mais desenvolvidos são os comerciais, como o Ansys CFX, FLUENT, CD-Adapco, PHOENICS e SU2. Todavia, um pacote de algum desses softwares pode custar em média US\$20.000 a US\$25.000 dólares para uma única licença. A licença tem validade de um ano,

apenas. Embora seja desmotivante por conta dos altos preços, estes mesmos preços tem se tornado motivação para o desenvolvimento de softwares livres, como o OpenFOAM.

2 OBJETIVOS

O objetivo deste trabalho é estudar o funcionamento do software livre OpenFOAM com respeito às suas capacidades no que tange a simulação de bio-escoamentos. O software será analisado com base nos resultados obtidos de várias simulações utilizando modelos Newtonianos e Não-Newtonianos, ambos laminares e incompressíveis e de parede rígida. Objetivos secundários são: otimizar a geração de malha através do software livre SALOME, além de buscar entendimento para se fazer uma configuração mais detalhada dos esquemas numéricos de discretização implantados no OpenFOAM.

3 MODELOS ADOTADOS

A seguir, são apresentados os principais modelos adotados neste trabalho.

3.1 O SANGUE

O sangue é uma suspensão de células (eritrócitos, leucócitos e trombócitos) em plasma. As células que existem em maior quantidade são os eritrócitos, ou glóbulos vermelhos, sendo determinantes para definir as propriedades reológicas do sangue. As considerações sobre o sangue são apresentados nos itens 3.1.1, 3.1.2 e 3.1.3.

3.1.1 Densidade

No que se refere à densidade, o sangue é conhecido por não ter compressibilidade significativa com relação a análises de escoamento. Nas condições reais de

circulação, o sangue flui numa temperatura de aproximadamente 37° C. Nesta temperatura seu valor de densidade corresponde a 1050 kg/m³. Alguns autores já consideram uma densidade um pouco maior, 1410 kg/m³, como é o caso de Chen(2006).

Numa situação real de escoamento. Uma artéria ou veia se dilata muito antes do fluido sanguíneo. Isso se deve à resistência a uma deformação volumétrica no sangue, caracterizada pelo fato de o módulo de elasticidade volumétrico ser de aproximadamente 10⁹ N/m². A artéria, todavia, apresenta resistência à variação do volume da ordem de 10⁵ N/m². Frente à deformação da artéria, o sangue então não apresenta deformação volumétrica significativa para efeitos de cálculo.

Portanto, quando se analisa sua variação de densidade em função da pressão, considera-se o sangue como fluido incompressível, em relação à deformação da artéria.

3.1.2 Viscosidade

Quanto a viscosidade, a comunidade científica classifica um fluido como sendo Newtoniano ou não-Newtoniano. Estes são os dois grandes grupos classificatórios. Um fluido newtoniano é um fluido em que as tensões viscosas que decorrem do seu fluxo, em cada ponto do escoamento, são linearmente proporcionais à taxa de deformação local (a taxa de variação da sua deformação ao longo do tempo).

O grupo dos fluidos não-Newtonianos são líquidos cujas propriedades de fluxo diferem dos fluidos newtonianos. Mais vulgarmente, as tensões viscosas já não são linearmente proporcionais à taxa de deformação local. Numericamente falando, no escoamento não-Newtoniano a viscosidade não é constante.

Quanto às equações, têm-se:

Para fluido Newtoniano:

$$\tau = \mu \dot{\gamma} = \mu \frac{\partial u}{\partial y} \quad (1)$$

sendo τ a tensão de cisalhamento, μ a viscosidade do fluido e $\dot{\gamma}$ a taxa de deformação.

Para fluidos não-Newtonianos:

$$\mu_{ap} = \frac{\tau}{\dot{\gamma}} \quad (2)$$

Onde μ_{ap} é a viscosidade aparente.

3.1.2.1 Modelos de viscosidade para fluidos Não-Newtonianos

Neste trabalho, serão utilizados alguns modelos de viscosidade para fluidos não-newtonianos.

Existem vários modelos desenvolvidos pelos estudiosos, cada um se adequando melhor para determinadas situações e composições do sangue. Dois dos conhecidos são: o Modelo Power-Law, e o Modelo Carreau-Yasuda.

O modelo de Power-Law tem a seguinte definição:

$$\eta = \eta_0 \dot{\gamma}^{n-1} \quad (3)$$

onde η_0 é a medida da viscosidade média do fluido (também chamado de índice de consistência) e n é uma medida do desvio do fluido newtoniano (também chamado de índice de Power-Law). Se n for igual a 1, o fluido será newtoniano, isto é, a função viscosidade será igual ao índice de consistência, que é a própria viscosidade média do fluido.

Já o modelo de Carreau-Yasuda, a equação de definição é:

$$\eta = \eta_{\infty} + (\eta_0 - \eta_{\infty}) [1 + (\lambda \dot{\gamma})^a]^{\frac{n-2}{a}} \quad (4)$$

onde η_0 é a viscosidade para taxa de cisalhamento nula, η_{∞} é a viscosidade para taxa de cisalhamento infinita, λ é o tempo de relaxamento e n é o índice de potência.

3.1.3 Laminar

Dentre as classificações quanto ao perfil de escoamento, o fluxo pode ser laminar ou turbulento. Pode-se passar de um regime a outro, simplesmente variando a velocidade de escoamento. Abaixo dessa velocidade, o fluxo é laminar. Ultrapassando-a, o mesmo se torna turbulento. Para a maioria dos trechos sanguíneos do sistema circulatório humano, o escoamento geralmente é laminar [2].

A velocidade limite, para transição laminar-turbulento é chamada de velocidade crítica (V_c), que se relaciona com as seguintes variáveis:

$$V_c = R_e \frac{\eta}{\rho \cdot D} \quad (5)$$

onde ρ é a densidade do fluido, R_e é o número de Reynolds, e η a viscosidade do fluido e D o diâmetro, no caso, das artérias.

4 OPENFOAM

O desenvolvimento original do OpenFOAM começou no final de 1980 no Imperial College, de Londres, para desenvolver uma plataforma de simulação geral, mais poderosa e flexível do que o padrão de fato na época, Fortran. Na época, o programa foi escrito em C++ como linguagem de programação, devido à sua modularidade e características orientadas a objetos. O predecessor, FOAM, foi vendido pela empresa britânica Nabla Ltd. antes de ser lançado como open source em 2004. Em 15 de agosto de 2011, OpenCFD anunciou sua aquisição pela Silicon Graphics International (SGI). Em 12 setembro de 2012, a ESI Group anunciou a aquisição da OpenFOAM Ltd da SGI.

No programa, os usuários podem criar objetos personalizados, tais como condições de contorno ou modelos de turbulência, que irão trabalhar com solucionadores existentes sem ter que modificar ou recompilar o código fonte existente. O OpenFOAM ainda faz isso através da combinação de construtores virtuais com o uso de classes de base simplificados como interfaces. Como resultado, isto dá ao OpenFOAM boas qualidades de extensibilidade.

Os solvers do OpenFOAM para escoamentos utilizam (dentre muitas outras) equações como a da Continuidade e as de Conservação de quantidade de movimento linear:

$$\frac{\partial}{\partial t}(\rho) + \nabla \cdot (\rho \vec{V}) = 0 \quad (6)$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho uu)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho wu)}{\partial z} = \frac{\partial T_{xx}}{\partial x} + \frac{\partial T_{yx}}{\partial y} + \frac{\partial T_{zx}}{\partial z} + \rho \cdot g_x \quad (7)$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho vv)}{\partial y} + \frac{\partial(\rho wv)}{\partial z} = \frac{\partial T_{xy}}{\partial x} + \frac{\partial T_{yy}}{\partial y} + \frac{\partial T_{zy}}{\partial z} + \rho \cdot g_y \quad (8)$$

$$\frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho uw)}{\partial x} + \frac{\partial(\rho vw)}{\partial y} + \frac{\partial(\rho ww)}{\partial z} = \frac{\partial T_{xz}}{\partial x} + \frac{\partial T_{yz}}{\partial y} + \frac{\partial T_{zz}}{\partial z} + \rho \cdot g_z \quad (9)$$

2.1 ESTRUTURA DE ARQUIVOS

O diretório para cada caso simulado no Openfoam é composto por um conjunto mínimo de arquivos, como é mostrado na Figura 4.1:

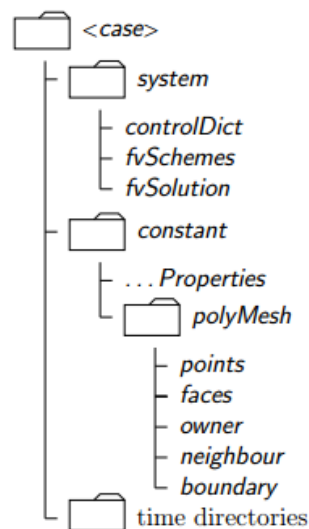


Figura 4-1: Lista padrão de diretórios de um caso no OpenFOAM. Fonte: Guia de Usuário do OpenFOAM

Estes diretórios armazenam os arquivos de texto que armazenam os códigos que definem a simulação por completo. Estes códigos contém palavras-chave reconhecidas pelo OpenFoam.

4.1 DIRETÓRIO *CONSTANT*

O diretório *constant* contém arquivos especificando as propriedades físicas – *Properties*, descrevendo as propriedades físicas, tal como a viscosidade necessária.

Além deste, arquivos de parâmetros do modelo de turbulência estão localizados aqui, caso se adote escoamento turbulentos.

4.1.1 Subdiretório PolyMesh

Há ainda o subdiretório *polyMesh*, que recebe o conjunto de arquivos descritivos da malha. O mesmo contém os seguintes arquivos de representação de malha:

points - o ficheiro de pontos é uma lista ordenada de vetores que descrevem a localização tridimensional de cada ponto da malha em ordem sequencial. Cada linha contém as coordenadas x, y, e z globais do ponto em dimensões de metros.

faces - o arquivo *faces* contém as faces definidas pelos vértices do arquivo de pontos. Cada linha da lista corresponde a uma face de modo sequencial e é constituída por um contador que indica o número de vértices que compõem a respectiva face, seguindo-se um vector com os rótulos dos vértices correspondentes.

owner - cada face no OpenFOAM está associada a uma célula proprietária e uma célula vizinha. O arquivo proprietário é uma lista sequencial de números de células que corresponde diretamente à lista sequencial no arquivo *faces*.

neighbour - é uma lista sequencial do número de células semelhantes, para o arquivo *owner*, exceto os números de células listados correspondem às células vizinhas para

cada face. Dentro deste arquivo, nas faces de contorno são aplicados índices de -1 como cada face fronteira pertence a apenas uma célula proprietário.

boundary - este arquivo contém uma lista de *patches* (definições geométricas dos limites) para a simulação.

O diretório polymesh também contém o arquivo *blockMeshDict* que é usado para entrada ao gerador de malha padrão do OpenFOAM, o *blockMesh*. Para um usuário típico é relativamente difícil de gerar malhas complexas usando *blockMesh* porque a ferramenta é baseada totalmente na entrada arquivo de texto e não tem interface gráfica para visualizar a malha projetada.

4.2 DIRETÓRIO SYSTEM

Acomoda os parâmetros que rege o processo de solução das equações envolvidas. É composto ao menos por três arquivos: *controlDict*, *fvSchemes* e *fvSolution*.

4.2.1 *controlDict*

O arquivo *controlDict* armazena os parâmetros de controle de execução que são definidos, tais como o tempo de início e fim, passo de tempo e parâmetros para a saída de dados.

As entradas de palavras-chave em *controlDict* estão listadas nas a seguir. Apenas o controle de tempo e entradas *writeInterval* são verdadeiramente obrigatórias.

Tabela 4.1: Controle de tempo

startFrom	Controla o tempo de início da simulação
- firstTime	Primeiro passo de tempo a partir do conjunto de diretórios de tempo
- startTime	Tempo especificado pela entrada de palavra-chave startTime
- latestTime	Passo de tempo mais recente do conjunto de diretórios de tempo
startTime	Hora de início para a simulação com startFrom startTime
stopAt	Controla o tempo final da simulação
- endTime	Tempo especificado pelo endTime entrada palavra-chave
- writeNow	Para a simulação na conclusão do passo de tempo atual e grava dados
- noWriteNow	Para simulação na conclusão do passo de tempo atual e não grava dados
- nextWrite	Para a simulação na conclusão do próximo tempo de gravação programado, especificado por writeControl
endTime	Tempo de fim para a simulação quando stopat e endTime forem especificados
deltaT	Passo de tempo da simulação

Fonte: Guia do Usuário do OpenFOAM

Tabela 4.2: Controle de dados

writeControl	Controla o tempo de saída de gravação para o arquivo
- timeStep†	Grava dados a cada passo de tempo writeInterval
- runTime	Grava dados a cada segundo de tempo simulado writeInterval
- adjustableRun Time	Grava dados a cada segundo de tempo simulado writeInterval, ajustando os passos de tempo para coincidir com o writeInterval se necessário - utilizados em casos de ajuste de passo de tempo automático
- cpuTime	Grava dados a cada segundo de tempo de CPU writeInterval
- clockTime	Grava dados a cada segundos de tempo real writeInterval
writeInterval	Escalar usado em conjunto com writeControl descrito acima
purgeWrite	

	Inteiro representando um limite para o número de diretórios de tempo que estão armazenados, substituindo diretórios de tempo em uma base cíclica
writeFormat	Especifica o formato dos arquivos de dados
- ascii†	Formato ASCII, escrito para writePrecision algarismos significativos
- binary	Formato binário
writePrecision	Inteiro usado em conjunto com writeFormat descrito acima, 6 † é o valor padrão
writeCompression	Especifica a compressão dos ficheiros de dados
- uncompressed	Sem compressão †
- compressed	Compressão no formato .gzip
timeFormat	A escolha do formato da nomeação dos diretórios de tempo
- fixed	+/-m.ddddddd em que o número d é definido por timePrecision
- scientific	+/- m.B123ddddddd +/- xx em que o número d é definido por timePrecision
- general†	Especifica o formato científico se o expoente é inferior a -4 ou maior ou igual à especificado por timePrecision
timePrecision	Inteiro usado em conjunto com timeFormat descrito acima, 6 † como padrão
graphFormat	Formato para dados do gráfico escrito por um aplicativo
- raw†	Formato ASCII Raw em colunas
- gnuplot	Dados em <i>gnuplot</i> formato
- xmgr	Dados em <i>Graça / xmgr</i> formato
- jplot	Dados em <i>jPlot</i> formato

† denota a entrada padrão se palavra-chave associada é omitido.

Continuação. Fonte: Guia do Usuário do OpenFOAM

Tabela 4.3: Passo de Tempo Ajustável

adjustTimeStep	sim† / não. Interruptor para OpenFOAM para ajustar o intervalo de tempo durante a simulação, geralmente de acordo com Maxco
maxCo	Número máximo Courant, <i>por exemplo</i> , 0,5

Fonte: Guia do Usuário do OpenFOAM

Tabela 4.4: Leitura de dados

runTimeModifiable	sim† / não. interruptor para se dicionários, <i>por exemplo controlDict</i> , que são re-lidos por OpenFOAM no início de cada intervalo de tempo
-------------------	--

Fonte: Guia do Usuário do OpenFOAM

Tabela 4.5: Funcionalidade Runtime carregável

libs	Lista de bibliotecas adicionais (em $\$ LD_LIBRARY_path$) a serem carregados em tempo de execução, <i>por exemplo</i> ("libUser1.so" "libUser2.so")
functions	Lista de funções, <i>por exemplo</i> , as sondas a serem carregados em tempo de execução; ver exemplos no $\$ FOAM_TUTORIALS$

Fonte: Guia do Usuário do OpenFOAM

4.2.2 *fvSchemes*

Em *fvSchemes* os esquemas de discretização utilizados na solução podem ser selecionados em tempo de execução.

4.2.3 *fvSolution*

fvSolution é o arquivo que define os solucionadores de equações, tolerâncias e outros controles de algoritmos para a simulação.

4.2.4 *decomposeParDict*

Este arquivo define a decomposição dos domínios para uso da função *decomposePar*, que habilita o usuário efetuar a simulação paralela. Não é um arquivo obrigatório para as simulações, exceto quando se deseja executar simulação em paralelo.

A decomposição dos domínios é feita selecionando o método de divisão da malha. O método mais comum para usuários do OpenFOAM é o *simple*, no qual o usuário diretamente define a divisão da malha para cada direção no sistema cartesiano de coordenadas.

Por exemplo, se o usuário deseja dividir a malha em 2 partes na direção x, 4 partes na direção y e 1 parte na direção z, sua malha será decomposta em 8 domínios. Na Figura abaixo, pode-se observar que selecionando o método *simple*, o usuário divide a malha em 4 partes:

```

18 numberOfSubdomains 4;
19
20 method          simple;
21
22 simpleCoeffs
23 {
24     n              ( 2 2 1 );
25     delta          0.001;
26 }
27
28 hierarchicalCoeffs
29 {
30     n              ( 1 1 1 );
31     delta          0.001;
32     order          xyz;
33 }
34
35 manualCoeffs
36 {
37     dataFile        "";
38 }
39
40 distributed      no;
41
42 roots            ( );
43
44
45 // ***** //

```

Figura 4-2: Arquivo DecomposePar com divisão de malha em 4 partes

Quando todos os diretórios da simulação forem criados, incluindo o arquivo *DecomposePar*, o usuário apenas insere no terminal do LINUX o comando:

DecomposePar

Tal comando é responsável pela divisão dos domínios requeridos para a malha. Neste momento, são gerados arquivos para cada domínio. O comando seguinte dá a partida no OpenFOAM:

```
mpirun --hostfile machines -np <número de núcleos> <solver utilizado para a
simulação> m -parallel > log &
```

Por exemplo, para uma simulação num Processador Intel® Core™ i7, com 4 núcleos (capaz de simular virtualmente um total de 8 núcleos), onde a malha é dividida em 4 partes, utilizando-se o solver *icofoam* do OpenFOAM, aplicaria-se o seguinte comando no terminal do LINUX:

```
mpirun --hostfile machines -np 4 icoFoam -parallel > log &
```

Nota-se que o número de núcleos especificado não foi maior que 8, tornando possível a simulação paralela. Este valor não deve exceder o número máximo de processadores, caso contrário uma das partes divididas ficaria sem solução, deixando o sistema de equações do OpenFOAM em aberto. O número de núcleos é o correspondente do processador do computador utilizado. Este valor também não pode ser menor do que o número de divisões da malha, visto que cada núcleo fica responsável para cada domínio gerado na divisão da malha.

Após todas as divisões de malha serem simuladas, é utilizado o comando *reconstructPar*, responsável pela reconstrução da malha e dos resultados finais.

4.3 DIRETÓRIO *TIME DIRECTORIES*

Em *time directories* são armazenados valores para condição inicial e condições de contorno, além dos arquivos gerados após a simulação, que são os dados de saída.

Para cada simulação, as condições iniciais são definidas dentro do diretório *0*. Para cada simulação inicializada é necessário ter um arquivo correspondente no diretório *0*; por exemplo, o campo de velocidade e de pressão são definidas pelos arquivos de *U* e *p*, respectivamente. Cada arquivo segue um esquema de formatação específica.

Dimensões de cada campo são especificados por um vetor que corresponde a uma lista de unidades SI de medida.

Para consultas, este vetor que representa a lista de unidades é descrito no Guia de Usuário do OpenFOAM.

5 GERAÇÃO DE MALHAS

5.1 PROCESSO DE CRIAÇÃO DA GEOMETRIA

Antes da geração da malha, ainda há o estágio de criação de geometria. Para geometrias complexas, é imprescindível a utilização de softwares com interface gráfica. Estes são softwares CAD que geram um modelo tridimensional desejado. Uma boa geometria, isto é, eficiente, é favorável na obtenção de uma boa malha (considerando uma malha gerada restringindo-se os parâmetros qualidade). Uma geometria ineficiente diminui a qualidade das malhas.

Basicamente a geometria ruim é caracterizada principalmente pela divisão das faces superficiais. As arestas da superfície irão definir as linhas de malha, isto é, a própria linha de divisão das faces da geometria se tornarão a divisão que unirá duas células. Portanto, O ideal no processo de criação de superfície é minimizar ao máximo a divisão de faces em uma superfície. A justificativa para tal minimização se dá pelo fato de as arestas limitarem o software de geração de malha. Isto causará diminuição dos parâmetros de qualidade de uma malha.

5.1.1 Processo de geração de malha

As malhas tem significativo impacto em uma simulação. Elas afetam diretamente a taxa de convergência (ou até mesmo a falta de convergência), a exatidão da solução, além do tempo requerido de processamento. Genericamente elas são classificadas em três tipos: malhas estruturadas, malhas não-estruturadas e malhas híbridas.

5.1.1.1 Malhas estruturadas

Uma malha estruturada é caracterizada por conectividade regular que pode ser expresso como uma matriz de duas ou três dimensões. Isso restringe as escolhas de elementos para quadriláteros em ou hexaedros em 3D. A malha abaixo é uma malha estruturada:

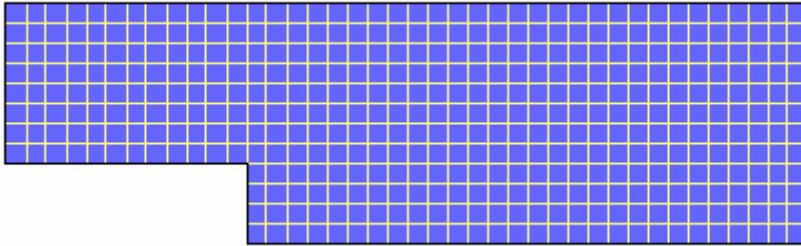


Figura 5-1: Exemplo de Malha Estruturada

5.1.1.2 Malhas não-estruturadas

Uma malha não-estruturada é caracterizada por conectividade irregular. Esta não é facilmente expressa como uma matriz bidimensional ou tridimensional na memória do computador. Em comparação com malhas estruturadas, os requisitos de armazenamento para uma malha não estruturada pode ser substancialmente maior. A seguir é mostrada uma malha não-estruturada:

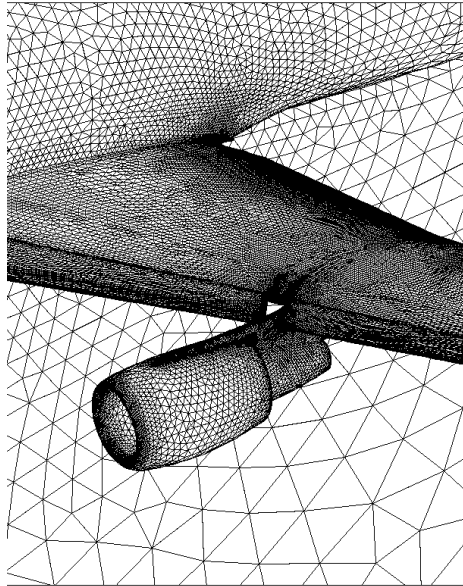


Figura 5-2: Exemplo de Malha Não-Estruturada

5.1.1.3 Malhas híbridas

A malha híbrida é aquela que contém porções estruturadas e não-estruturadas. Esta definição requer conhecimento do modo como a malha é armazenada (e usada).

5.2 PARÂMETROS DE QUALIDADE

5.2.1 Ortogonalidade

O conceito de ortogonalidade da malha está relacionado com o quão os ângulos entre faces de elementos ou entre arestas de elementos adjacentes estão próximos de um ângulo ótimo (90° para elementos com faces quadrilaterais e 60° para elementos com faces triangulares). Nas malhas não-ortogonais, os centros dos volumes de controle adjacentes estão distribuídos de modo que a linha imaginária que une os seus centros não seja ortogonal à face comum. Tal conceito é mostrado na figura a seguir:

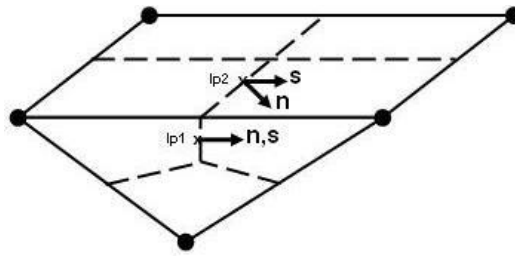


Figura 5-3: Ortogonalidade

5.2.2 Razão de aspecto

O conceito de razão de aspecto da malha é relacionado ao grau com que os elementos da malha se deformam. A razão de aspecto envolve a razão entre as áreas máxima e mínima das superfícies de todos os volumes de uma malha.

O esquema a seguir substantifica o conceito:

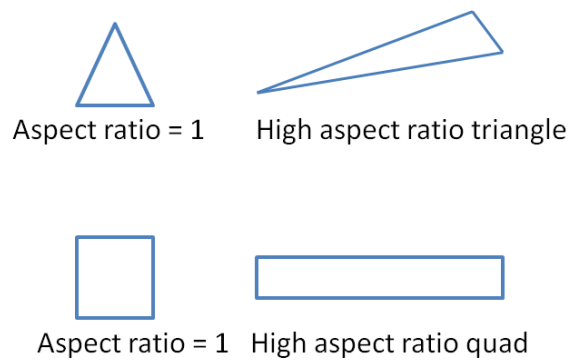


Figura 5-4: Razão de Aspecto

5.2.3 Assimetria

A assimetria é outra medida utilizada para quantificar a qualidade de uma malha. Assimetria determina quão próximo do ideal uma malha é, isto é, a propriedade de ser equilátera. Abaixo pode ser observado tal conceito:

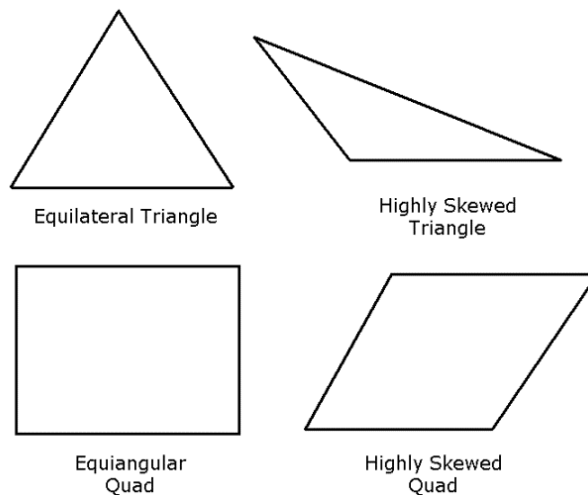


Figura 5-5: Assimetria

5.3 SALOME

O OpenFOAM já tem suporte para geração de malhas. São os blockMesh e SnappyMesh. Ambos são práticos, quando voltados para geração de malhas simples. Entretanto sua praticidade não é tão grande quando comparado com softwares externos voltados para geração de malhas, uma vez que os geradores de malha do OpenFoam não têm interface gráfica. Todas as malhas são geradas via arquivos de texto. Diante disso, convém assumir que uma escolha de software com interface gráfica torna-se extremamente vantajosa.

Na comunidade de simulação opensource, há algumas opções. Um dos softwares geralmente adotados é o Salome. Suas capacidades são deste o pré-processamento até o pós processamento de simulações numéricas. É distribuído como opensource nos termos da licença GNU LGPL.

O SALOME tem vários ambientes. A seguir pode-se observar o ambiente gráfico de geração e manuseio de malhas:

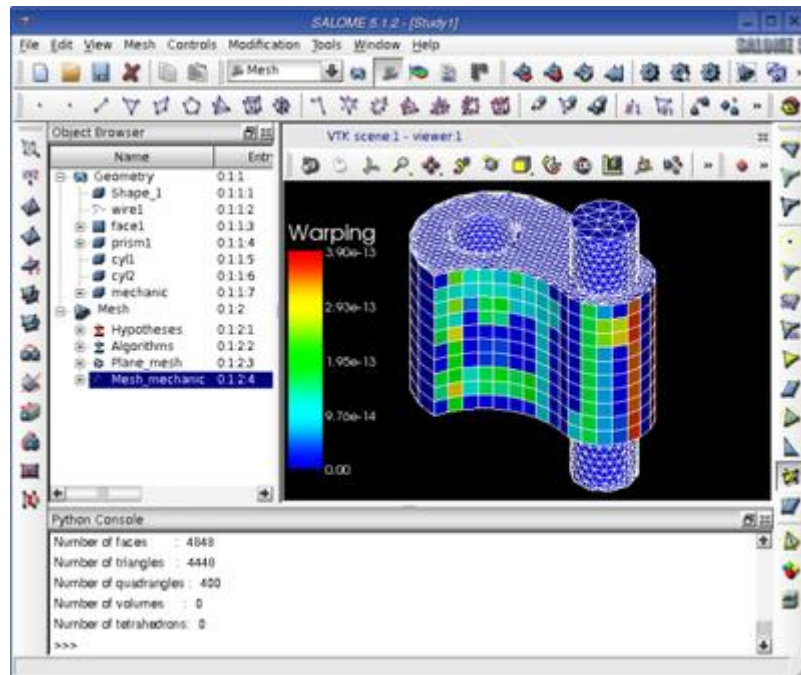


Figura 5-6: Interface Grafica do Salome

6 METODOLOGIA E RESULTADOS

Neste trabalho foram feitos estudos de caso para três artigos científicos, ambos tratando da simulação numérica para o sangue. O primeiro deles trata de uma simulação em regime permanente e bidimensional. O segundo é uma variante do primeiro, alterando apenas a dimensão, sendo este agora tridimensional e com uma geometria mais complexa. O terceiro e último, se difere dos anteriores pelo fato de a simulação ser em regime transiente e tridimensional. Todas elas foram realizadas com o intuito de mostrar que o OpenFOAM tem capacidades variadas, além de ser versátil.

6.1 CASO I

O primeiro caso teve como foco a comparação dos resultados obtidos por Fodor (2009). Neste artigo, o autor utilizou o software comercial *COMSOL Multiphysics*, que utiliza o método de elementos finitos.

6.1.1 Descrição do caso

No presente caso, o fluxo de fluido ao longo de uma geometria bidimensional é analisado a fim de se gerar dados para diferentes modelos de viscosidade do sangue. Dois modelos não newtonianos, bem como o modelo newtoniano são estudados. Os modelos não newtonianos utilizados neste caso são o modelo de Carreau, e o modelo de Power-Law.

A geometria adotada pelo autor é mostrada na Figura a seguir:

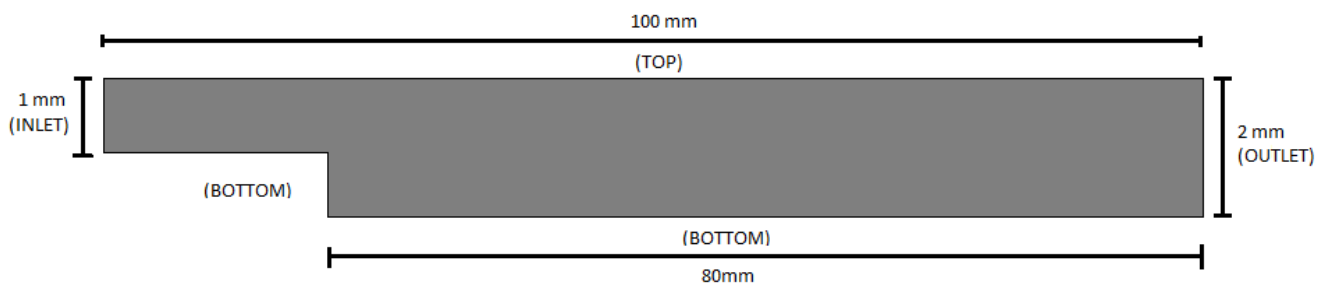


Figura 6-1: Geometria do CASO I

O modelo newtoniano e dois modelos não-newtonianos são utilizados neste caso: o modelo de viscosidade Carreau-Yasuda e o modelo de viscosidade Power-Law. Para o modelo newtoniano, adota-se viscosidade igual a 0,0035 Pa.s. Para o modelo Power-Law, os parâmetros para a função viscosidade são: η_0 igual a 0,035 Pa.s e n igual a 0,6. E no modelo Carreau-Yasuda, os parâmetros são: η_0 igual a 0,0035 Pa.s, η_∞ tendo o valor de 0,056 Pa.s, λ valendo 3,313 s e n igual a 0,3568.

Foram executadas, no total, seis simulações. Conforme Fodor (2009) propôs, para a geometria bidimensional, assume-se perfil parabólico de velocidade na entrada (INLET) com velocidades máximas de perfil igual a 0,0025 m/s e 0,8 m/s. A condição

de contorno para a saída (OUTLET) é de pressão relativa nula. Para as paredes (TOP E BOTTOM), o autor adotou velocidade nula na interface de contato fluido-parede. Assim, no total foram realizadas seis simulações. Para o perfil com velocidade máxima de 0,0025 m/s, foram realizadas uma simulação para cada modelo (são três modelos). E para o perfil com velocidade máxima de 0,8 m/s, outras três. Pela tabela a seguir, podem-se apresentar tais casos:

Tabela 6.1: Caracterização das situações simuladas do CASO I

Simulação	Velocidade máx. na entrada [m/s]	Modelo de viscosidade adotado
A	0,0025	Newtoniano
B	0,0025	Power-Law
C	0,0025	Carreau
D	0,8	Newtoniano
E	0,8	Power-Law
F	0,8	Carreau

Para todas as simulações, Fodor (2009) admite o fluido como sendo incompressível e laminar. A densidade para o sangue foi igual a 1050 kg/m³.

6.1.2 Simulação

Neste item, estão contidos os códigos utilizados no OpenFOAM para recriar as simulações feitas por Fodor(2009).

6.1.2.1 Geração de malha

A malha poderia ser criada utilizando-se o software SALOME. Não obstante, optou-se por utilizar o gerador de malha padrão do Openfoam, o blockMesh. Expõe-se então o código da malha:

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
// *****

convertToMeters 0.001;

vertices
(
    (0 0 0)
    (20 0 0)
    (20 0.1 0)
    (0 0.1 0)
    (0 0 1)
    (20 0 1)
    (20 0.1 1)
    (0 0.1 1)

    (100 0 0)
    (100 0.1 0)
    (20 0 2)
    (100 0 2)
    (100 0.1 2)
    (20 0.1 2)
    (100 0 1)
    (100 0.1 1)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (400 1 20) simpleGrading (1 1 1)
    hex (1 8 9 2 5 14 15 6) (480 1 20) simpleGrading (1 1 1)
    hex (5 14 15 6 10 11 12 13) (480 1 20) simpleGrading (1 1 1)
);

edges
(
);

boundary
(
    outlet
    {
        type patch;
        faces
        (
            (8 9 15 14)
            (14 15 12 11)
        );
    }

    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }

    wall
    {
        type wall;
        faces
        (
            (0 3 2 1)
            (1 2 9 8)
            (4 5 6 7)
            (5 10 13 6)
            (10 11 12 13)
        );
    }

    frontAndBack
    {
        type empty;
        faces
        (
            (0 1 5 4)
            (3 7 6 2)
            (1 8 14 5)
            (10 5 14 11)
            (6 15 9 2)
            (13 12 15 6)
        );
    }
);

mergePatchPairs
(
);

// *****

```

Figura 6-2: Código da malha estruturada utilizada no CASO I

6.1.2.2 Condições iniciais e de contorno

Com relação às condições de contorno, criou-se o arquivo “p”, localizado na pasta “0” do diretório de casos de uma simulação. O conteúdo do arquivo é apresentado na figura abaixo:

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  object       p;
}
// *****

dimensions     [0 2 -2 0 0 0];
internalField   uniform 0;
boundaryField
{
  inlet
  {
    type        zeroGradient;
  }

  outlet
  {
    type        fixedValue; // The outlet is taken as exposed to atmosphere, so you need to keep this as 0
    value       uniform 0;
  }

  wall
  {
    type        zeroGradient; // you had used the outlet conditions here, which is wrong.
  }
  frontAndBack
  {
    type        empty;
  }
}
// *****
```

Figura 6-3: Pressões para o CASO I

Para as condições de contorno assumidas por Fodor (2009), é imprescindível que se adote condições condizentes com a realidade. Para tal efeito, assume-se gradiente nulo na entrada (INLET) e nas paredes (TOP e BOTTOM). Estas condições de contorno são aplicadas para todas as simulações deste caso (CASO I).

Para o campo de velocidades, criou-se o arquivo “U”, também localizado no diretório “0”. Os arquivo contendo as condições de contorno para velocidade são:

Para as simulações com velocidade máxima do perfil de entrada igual a 0,0025 m/s:

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    inlet
    {
        type      groovyBC;
        variables  "zp=pos().z;velocity=(-0.025*(1-(pow(0.0005-zp,2)/0.00000025)))*normal()";
        valueExpression "velocity";
    }
    outlet
    {
        type      zeroGradient; // At the outlet the velocity is unknown so we take the gradient of velocity as zero here
    }
    wall
    {
        type      fixedValue;
        value      uniform (0 0 0); // we use the no slip condition at the walls so u = v = 0
    }
    frontAndBack
    {
        type      empty;
    }
}
// *****

```

Figura 6-4: Condições de contorno para velocidade de entrada máxima igual a 0,0025 m/s

E para as simulações com velocidade máxima do perfil de entrada igual a 0,8 m/s:

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    inlet
    {
        type      groovyBC;
        variables  "zp=pos().z;velocity=(-0.8*(1-(pow(0.0005-zp,2)/0.00000025)))*normal()";
        valueExpression "velocity";
    }
    outlet
    {
        type      zeroGradient; // At the outlet the velocity is unknown so we take the gradient of velocity as zero here
    }
    wall
    {
        type      fixedValue;
        value      uniform (0 0 0); // we use the no slip condition at the walls so u = v = 0
    }
    frontAndBack
    {
        type      empty;
    }
}

```

Figura 6-5: Condições de contorno para velocidade de entrada máxima igual a 0,8 m/s

É importante expor que para se inserir a condição de contorno de um perfil parabólico, foi necessário instalar uma ferramenta adicional chamada *swak4foam*. Esta biblioteca contém muitos utilitários. Um deles é o *groovyBC*, que fornece facilidades na criação de condições de contorno diferenciadas. O *groovyBC* substitui a programação de baixo nível por uma linguagem acessível que fornece palavras-chave (*keywords*) para simplificar a criação dos códigos. Esta biblioteca, em palavras simples, é direcionada para aqueles que não entendem a linguagem baixo nível de programação do OpenFOAM. No caso dos perfis de velocidades, foi inserida uma equação que fornece o perfil parabólico completamente desenvolvido. A variável dependente desta equação é a posição na direção z , denotada por “ z_p ”, na Figura 6.5.

6.1.2.3 Propriedades físicas

Cada um dos modelos de viscosidade foi implementado: Newtoniano, Power-Law e Carreau, conforme Fodor (2009).

6.1.2.4 Esquemas numéricos

Para todas as seis simulações, adotaram-se os mostrados na Figura abaixo, no arquivo *fvSchemes*:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSchemes;
}
// *****
ddtSchemes
{
    default      steadyState;
}
gradSchemes
{
    default      Gauss linear;
}
divSchemes
{
    default      none;
    div(phi,U)   bounded Gauss upwind;
    div(phi,k)   bounded Gauss upwind;
    div(phi,epsilon) bounded Gauss upwind;
    div(phi,R)   bounded Gauss upwind;
    div(R)       Gauss linear;
    div(phi,nuTilda) bounded Gauss upwind;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default      Gauss linear corrected;
}
interpolationSchemes
{
    default      linear;
}
snGradSchemes
{
    default      corrected;
}
fluxRequired
{
    default      no;
    p            ;
}
// *****
```

Figura 6-6: *fvSchemes* para o CASO I

Observando-se detalhadamente a Figura 6.6, nota-se que há vários esquemas de discretização selecionados. Os esquemas de tempo (ddtschemes) não são necessários neste caso, que não é transiente. Daí, utiliza-se o termo *steady-state*.

Para os esquemas de discretização de gradientes, selecionou-se o esquema de Gauss, *Gauss linear*. O termo complementar *linear* indica que o esquema de discretização de Gauss é resolvido utilizando a interpolação linear. Para os esquemas de divergência, *divSchemes*, é também adotado o esquema de Gauss, com a diferença de ser interpolado por interpolação *upwind*. Os termos laplacianos, *laplacianSchemes*, são unicamente resolvidos pelo esquema de Gauss (é a única opção para este tipo de esquema) e requer um esquema de interpolação (*linear*) e um esquema para tratar com gradientes normais de superfície (*corrected*). Para interpolações de valores de centro de volumes finitos até aos centros de face dos mesmos, em *interpolationSchemes*, utiliza-se a interpolação linear. Para tratar das não-ortogonalidades de superfícies, empregou-se o termo *corrected* (esta palavra-chave não é importante nestas simulações, visto que as malhas são estruturadas e são totalmente ortogonais). Por fim, *fluxRequired* é para a pressão, p . Isto é, a seleção da pressão significa que o fluxo será gerado depois da resolução de uma equação de pressão.

6.1.2.5 Controle da solução

O arquivo *fvSolution* deve ser criado com intuito de controlar as tolerâncias e os algoritmos gerais. O *fvSolution* utilizado em todas as simulações foi:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location      "system";
    object        fvSolution;
}
// *****
solvers
{
    p
    {
        solver          GAMG;
        tolerance        1e-08;
        relTol           0.1;
        smoother         GaussSeidel;
        nPreSweeps        0;
        nPostSweeps       2;
        cacheAgglomeration on;
        agglomerator       faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels        1;
    }
    "(U|nuTilda)"
    {
        solver          smoothSolver;
        smoother         symGaussSeidel;
        tolerance        1e-07;
        relTol           0.1;
    }
}
SIMPLE
{
    nNonOrthogonalCorrectors 0;

    residualControl
    {
        p               1e-4;
        U               1e-5;
    }
}
relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        U               0.7;
        nuTilda         0.7;
    }
}
// *****
```

Figura 6-7: *fvSolution* para o CASO I

Os *solvers* de matrizes esparsas são iterativos, isto é, eles se baseiam na redução da equação residual sobre uma sucessão de soluções. O resíduo é aparentemente uma medida do erro na solução de modo que quanto menor for, mais exata será a solução. Mais precisamente, o resíduo é avaliado através da substituição da solução atual na equação e tendo a magnitude da diferença entre os lados esquerdo e direito. Este resíduo também é normalizado para torná-lo independente da dimensão do problema que está sendo analisado. Isto intercambia várias simulações, podendo-se então comparar as curvas de convergência.

Antes de resolver uma equação para um campo específico, o resíduo inicial é avaliado com base nos valores atuais do campo. Após cada iteração o resíduo é reavaliado. O solver pára se qualquer uma das seguintes condições são atingidas:

- as quedas residuais abaixo da tolerância, *tolerance*;
- a o valor de resíduos iniciais cair abaixo da tolerância relativa, *relTol*;
- o número de iterações exceder um número máximo de iterações, *maxiter* (esta função é opcional);

Os valores de *tolerance* e *relTol* podem ser visualizados na Figura 6.7 .

O método SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) permite acoplar as equações de Navier-Stokes e resolvê-las com um processo iterativo.

No campo *relaxationfactors*, estão presentes fatores para as variáveis da simulação. Estes fatores são utilizados para melhorar a estabilidade da simulação. Geralmente os fatores adotados neste caso são adotados por quase todos os usuários do OpenFOAM. Dificilmente, ou em casos muito específicos, estes valores são alterados.

Para análise de malha, o openFOAM disponibiliza a função *checkMesh*, responsável por calcular parâmetros de qualidade para malhas. Para a malha em questão, dentre os vários parâmetros que o *checkMesh* calcula, os principais para caracterizar uma boa malha são apresentados a seguir, com seus respectivos quantitativos:

Tabela 6.2: Saída do *blockMesh* para o CASO I

Parâmetro	Valor	Unidade SI
Máxima razão de aspecto	3,33	-
Máxima não-ortogonalidade	0	°
Máxima assimetria	0,0099	-

Há outros parâmetros de qualidade que o *checkMesh* calcula, cerca de outros dez. No entanto, para esta simulação eles são irrelevantes em relação às razão de aspecto, não-ortogonalidade e assimetria. Nos esquemas de discretização, estes três parâmetros são os mais influentes na convergência.

Com estes valores, a malha pode ser classificada como ótima, visto que não apresenta ortogonalizadas, a assimetria é muito baixa e a razão de aspecto é relativamente pequena. A malha tem esta qualidade pois é estruturada. O *blockMesh* cria malhas estruturadas. Para geometrias simples como a deste caso, a obtenção de malhas com boa qualidade é relativamente fácil.

Foram obtidos seis perfis de velocidade para as seis simulações. Os perfis são retirados na saída (OUTLET). Os perfis de velocidade obtidos no OpenFOAM são apresentados nas figuras a seguir, juntamente com os resultados de Fodor(2009), que utilizou o *COMSOL Multiphysics*:

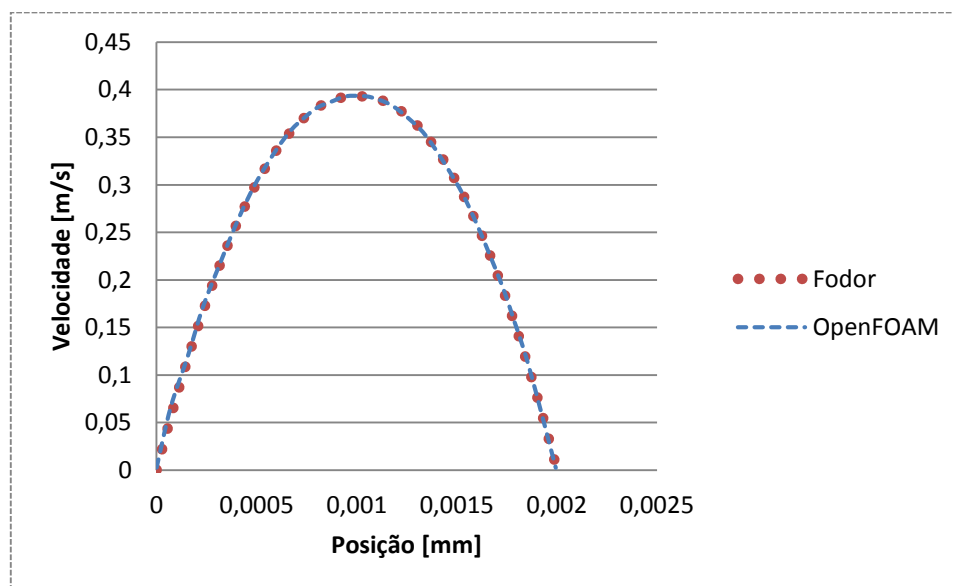


Figura 6-8: Comparação do perfil de velocidade na saída utilizando o modelo Carreau-Yasuda e velocidade de entrada de 0,8 m/s

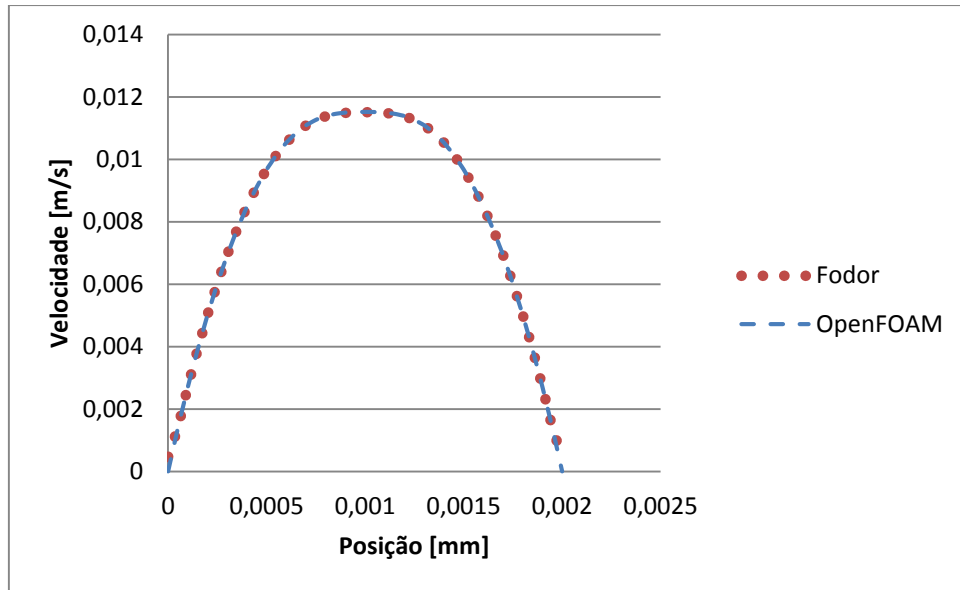


Figura 6-9: Comparação do perfil de velocidade na saída utilizando o modelo Carreau-Yasuda e velocidade de entrada de 0,025 m/s

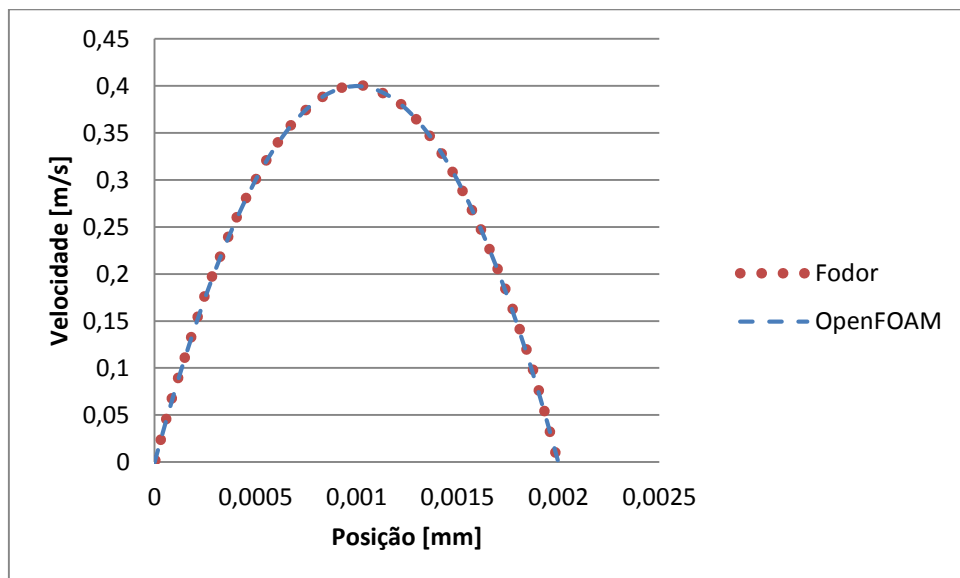


Figura 6-10: Comparação do perfil de velocidade na saída utilizando o modelo Newtoniano e velocidade de entrada de 0,8 m/s

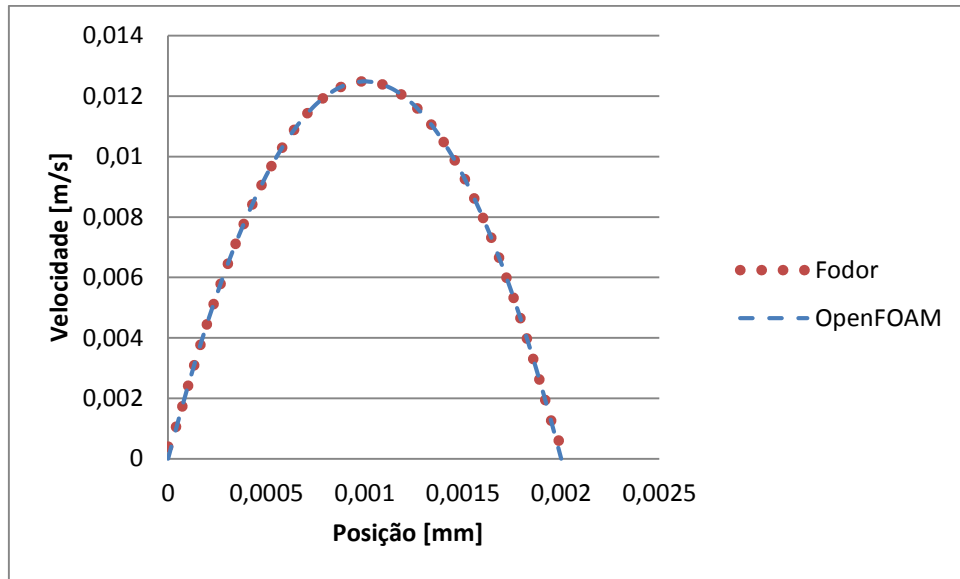


Figura 6-11: Comparação do perfil de velocidade na saída utilizando o modelo Newtoniano e velocidade de entrada de 0,025 m/s

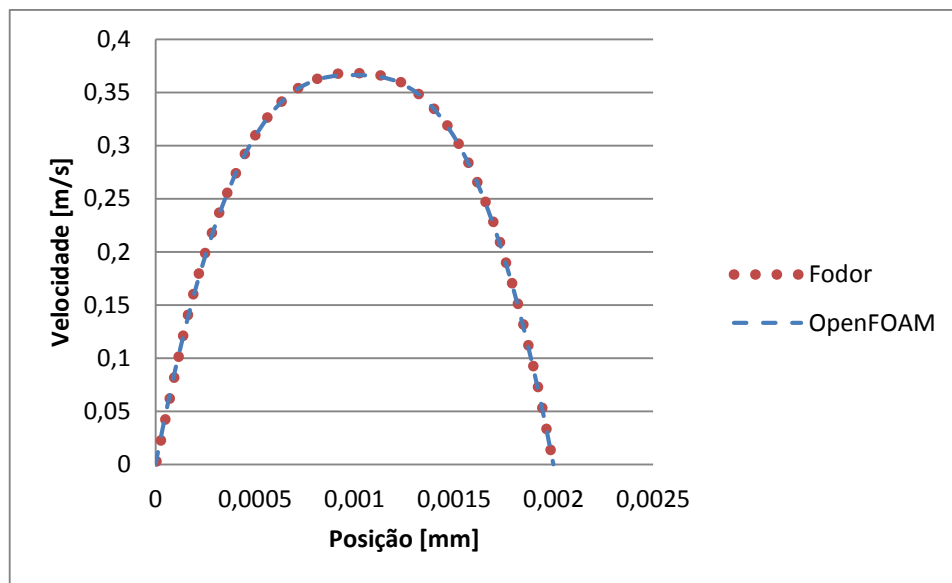


Figura 6-12: Comparação do perfil de velocidade na saída utilizando o modelo Power-Law e velocidade de entrada de 0,8 m/s

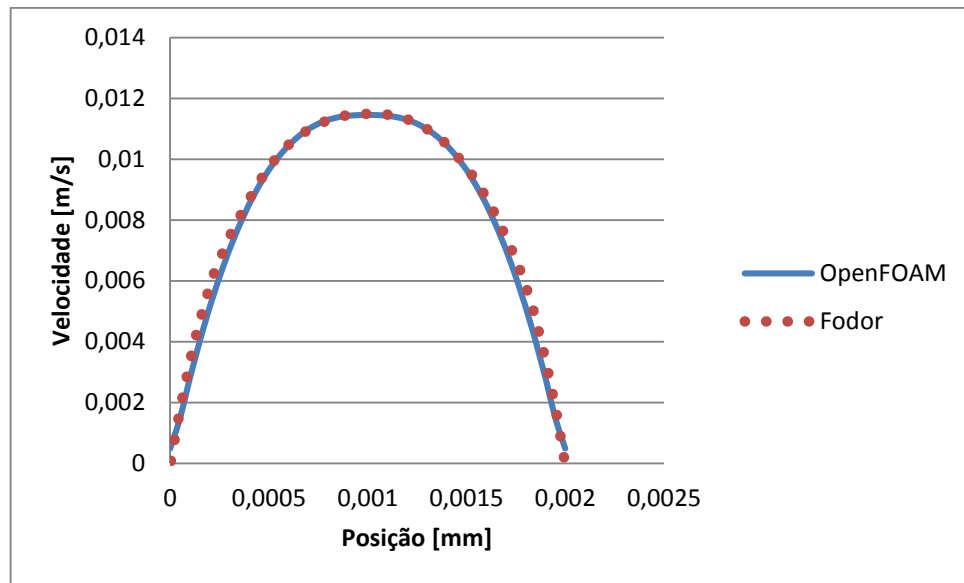


Figura 6-13: Comparação do perfil de velocidade na saída utilizando o modelo Power-Law e velocidade de entrada de 0,025 m/s

Em vista dos resultados gráficos de perfis de velocidade na saída (OUTLET), percebe-se que os resultados estão bem próximos. Nota-se suave variação nos perfis com velocidade de entrada máxima igual a 0,0025 m/s para o modelo de Power-Law.

As linhas de corrente (*streamlines*) obtidas pelo OpenFOAM são apresentadas a seguir, juntamente com as linhas de corrente de Fodor (na mesma proporção da imagem):

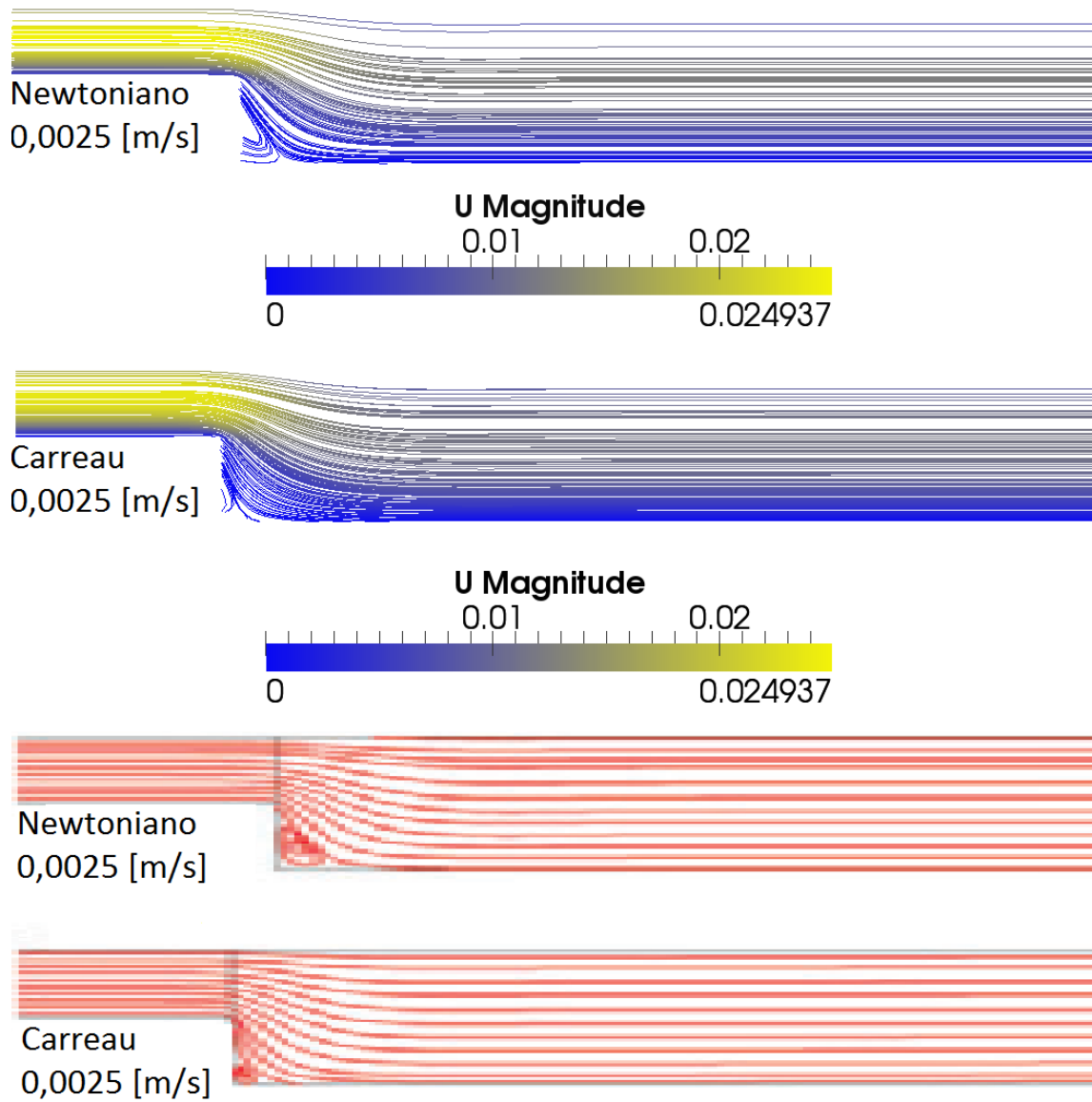


Figura 6-14: Comparativo entre linhas de corrente obtidas na simulação e obtidas por Fodor para velocidade máxima de 0,0025 m/s

Foram também simulados outros 2 casos para comparação com as linhas de corrente de Fodor para velocidade de entrada máxima igual a 0,025m/s:

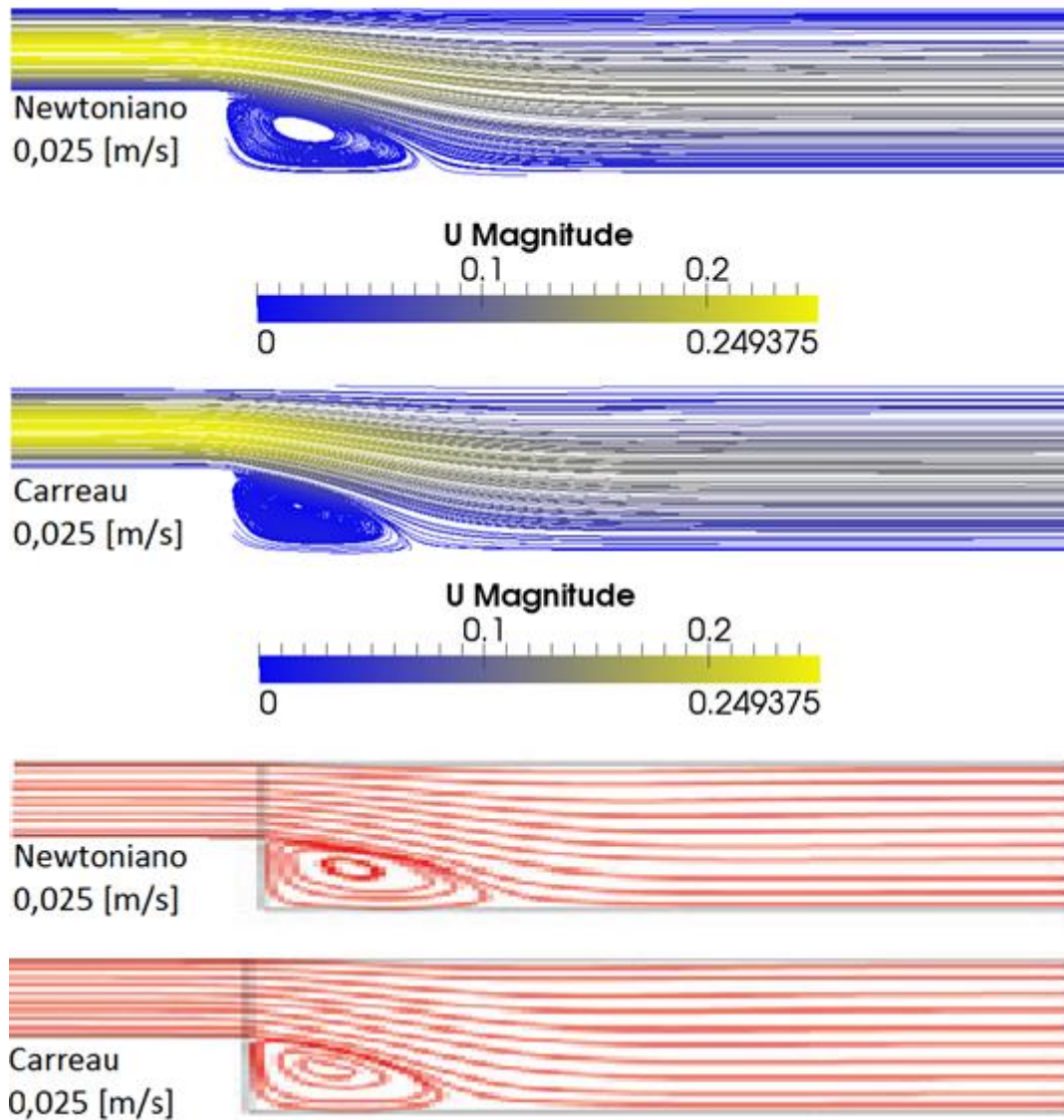


Figura 6-15: Comparativo entre linhas de corrente obtidas na simulação e obtidas por Fodor para velocidade máxima de 0,025 m/s

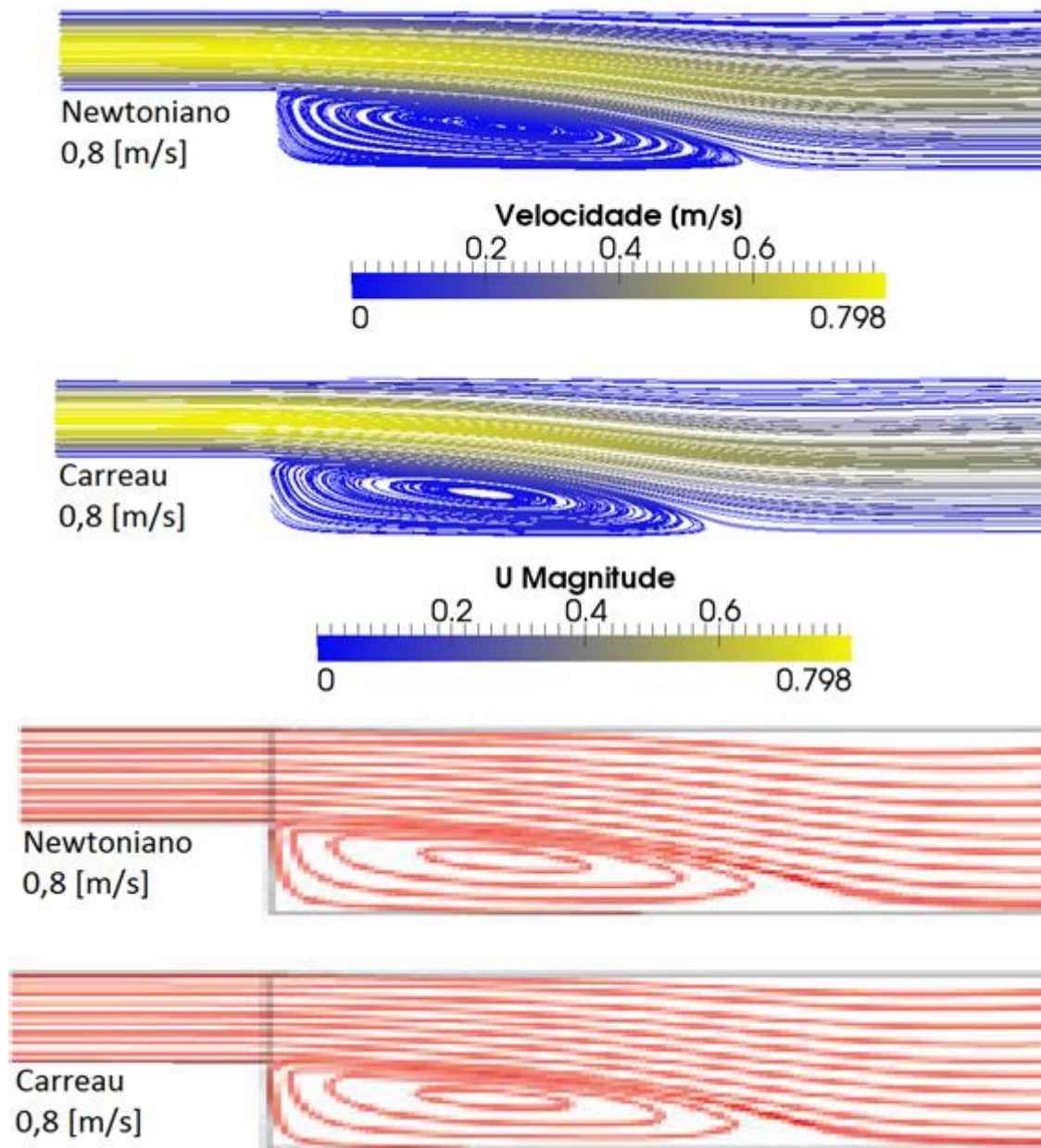


Figura 6-16: Comparativo entre linhas de corrente obtidas na simulação e obtidas por Fodor para velocidade máxima de 0,8 m/s

As linhas de corrente ficaram muito parecidas, a olho nu.

6.2 CASO II

6.2.1 Descrição do caso

Neste caso, objetivou-se reproduzir as simulações de Chen (2006). O fluxo de sangue foi investigado em uma anastomose vascular distal, geometria na qual há uma passagem secundária de fluxo e uma artéria coronária anfitriã. Na artéria anfitriã, inseriu-se uma estenose de 75% (redução percentual de área). O modelo utilizado foi o de Carreau-Yasuda. A geometria é considerada um modelo simplificado. Diferentes locais para a artéria secundária foram estudados. É dada especial atenção para o efeito não-newtoniano do sangue sobre os padrões de fluxo de primárias e secundárias na artéria coronária de acolhimento.

Embora tenha sido utilizados dados numéricos provenientes de simulações feitas por Chen(2006), este autor validou suas simulações de Bertolotti and Deplano (2000). Posteriormente, Moura(2011) utilizou o Software Ansys CFX para reproduzir um caso da publicação de Chen(2006). Os resultados entre os três trabalhos foram extremamente próximos.

A geometria adotada pelo autor é mostrada na Figura a seguir:

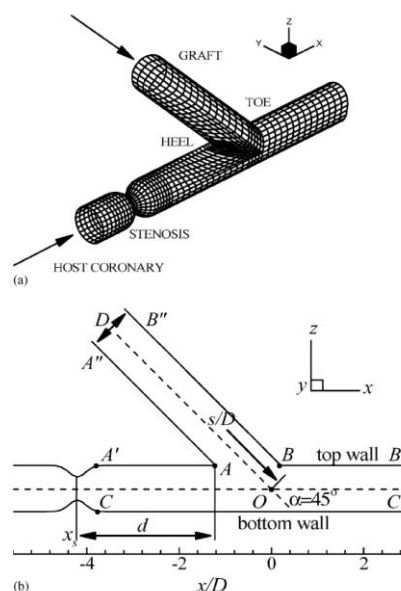


Figura 6-17: Anastomose vascular distal. O diâmetro D da artéria anfitriã é o mesmo da artéria secundária. A distância de A até o centro da estenose é d . (a) Perspectiva; (b) Vista do plano x-z de simetria. Fonte: Chen(2006)

Com referências secundárias, Chen(2006) adotou uma estenose de 75%. Foi utilizado um diâmetro D de 3m. O curvatura da estenose foi equacionada da seguinte forma:

$$\frac{R_x}{R_0} = 1 - \frac{1}{2} e^{\left(\frac{-4(x-x_s)^2}{R_0^2}\right)} \quad (6)$$

Onde R_x é o raio da parte contraída na coronária anfitriã de R_0 igual a 1,5 mm. x_s representa a posição central da estenose. A estenose toma diferentes posições dentre várias simulações d=1,5D, 2D e 3D. Para referência nas seções seguintes, cada uma dessas simulações serão denotadas por D1.5 , D2 e D3.

Além de se alterar a geometria entre simulações, muda-se também a vazão Q em alguns valores. Os valores e suas respectivas denotações para as seções seguintes apresentados a seguir:

Tabela 6.3: Descrição das vazões adotadas para as simulações

Referência	Vazão artéria anfitriã [m³/s]	Vazão artéria secundária [m³/s]
Q12	Q/2	Q/2
Q34	3Q/4	Q/4
Q78	7Q/8	Q/8

O valor de Q é $1,708 \times 10^{-3}$ kg/s. O perfil de entrada é completamente desenvolvido.

A densidade adotada para o fluido foi igual a 1410 kg/m³.

Diante disso, a referência das simulações realizadas foram oito: D1.5-Q12, D1.5-Q34, D1.5-Q78, D2-Q12, D2-Q34, D2-Q78, D3-Q12, D3-Q34 e D3-Q78.

6.2.2 Simulação

Neste item, estão contidos os códigos utilizados no OpenFOAM para recriar as simulações feitas por Chen(2006).

6.2.2.1 Geração de malha

Para geração de malha, foi necessário, de antemão, criar a geometria. Para ambos processos, utilizou-se unicamente o SALOME. O SALOME, embora tenha uma interface visual simples de se manusear, apresentou grandes dificuldades na criação da geometria. Foram realizados excessivos testes para se encontrar como se criar qualquer geometria com boa qualidade para se gerar uma boa malha tetraédrica (malhas hexaédricas são extremamente difíceis de gerar). Depois desses testes, concluiu-se que para se criar tal geometria, seria necessário muitas ações de cautela. As mais relevantes foram:

Criar a geometria com o mínimo possível de faces unidas (faces unidas apresentam arestas, que tendem a impedir a livre geração de malha pelo SALOME, diferentemente de faces únicas);

Criar um sólido sem linhas internas (que geralmente provém da união de dois sólidos para a geração de um sólido).

As ações mencionadas acima devem ser utilizadas com intuito de se gerar a superfície externa da geometria. Depois de criada tal superfície. Parte-se para a criação de sólido com base na superfície externa. Se esses procedimentos forem seguidos, a malha tetraédrica (a ser gerada) terá parâmetros de qualidade bons para malhas tetraédricas (quando gerada no SALOME).

Um trecho de uma das malhas geradas é mostrada a seguir:

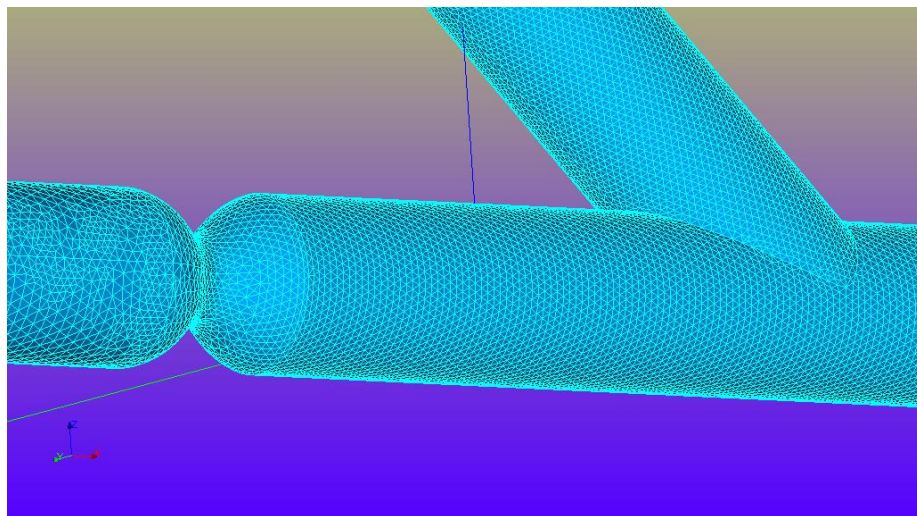


Figura 6-18: Parte da malha para distancia d igual a 3 geradas no SALOME.

Padronizou-se o numero de volumes finitos para cerca de 900000 para todas as malhas deste caso.

Para as três malhas geradas, D1.5, D2 e D3, dos parâmetros de qualidade do *checkMesh*, os mais importantes foram:

Para a malha D1.5:

Tabela 6.4: Parametros de qualidade para a malha do CASO II – malha D1.5

Parâmetro	Valor	Unidade SI
Máxima razão de aspecto	8,61204	-
Máxima não-ortogonalidade	62,1767	°
Máxima assimetria	0,654692	-

Para a malha D2:

Tabela 6.5: Parametros de qualidade para a malha do CASO II – malha D2

Parâmetro	Valor	Unidade SI
Máxima razão de aspecto	6,5017	-
Máxima não-ortogonalidade	61,3822	°
Máxima assimetria	0,842805	-

E para a malha D3:

Tabela 6.6: Parametros de qualidade para a malha do CASO II – malha D3

Parâmetro	Valor	Unidade SI
Máxima razão de aspecto	7,35927	-
Máxima não-ortogonalidade	64,2256	°
Máxima assimetria	0,777459	-

A malha do Caso I era estruturada e basicamente formada por quadrados(a simulação era bidimensional). Neste caso, pela complexidade da geometria, é

necessário recorrer à malhas tetraédricas. Embora tenham menos qualidade, são as mais usadas para geometrias complexas.

Os parâmetros basicamente ficaram próximos. É dada devida atenção ao fato de assimetria e não-ortogonalidade serem relativamente elevadas. Para tais malhas, os esquemas numéricos utilizados no Caso I não são suficientes para obter resultados próximos da exatidão. É necessário aderir a esquemas de discretização mais eficientes.

Para aumentar a velocidade de processamento da solução, qualidade de armazenamento de coordenadas e características da malha também influenciam. Para isto, utilizou-se o comando *renumberMesh* que renumera toda a malha. Após se descobrir este comando, percebeu-se redução de tempos de simulação de aproximadamente 70%.

6.2.2.2 Condições iniciais e de contorno

Embora nas simulações do Caso I tenha se assumido perfil completamente desenvolvido, neste caso, optou-se por alongar as entradas a fim de que o perfil se desenvolvesse completamente. Após a simulação, foi verificado que o perfil se tornou completamente desenvolvido.

Além disso, adotou-se velocidade nula nas paredes da geometria.

O arquivo de condições de contorno e iniciais para o caso D1.5-Q12 são mostrados abaixo:

Para pressão:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       p;
}
// ****

dimensions      [0 2 -2 0 0 0 0];

internalField    uniform 0;

boundaryField
{
    inlet_graft
    {
        type      zeroGradient;
    }
    inlet_host
    {
        type      zeroGradient;
    }
    outlet
    {
        type      fixedValue;
        value      uniform 0;
    }
    wall
    {
        type      zeroGradient;
    }
}
```

Figura 6-19: Condições de contorno para pressão na malha do CASO II

Para velocidade:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    location     "0";
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    inlet_graft
    {
        type      fixedValue;
        value      uniform (0.060599 0 -0.060599);
    }
    inlet_host
    {
        type      fixedValue;
        value      uniform (0.0856 0 0);
    }
    outlet
    {
        type      zeroGradient;
    }
    wall
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }
}
```

Figura 6-20: Condições de contorno para velocidade na malha do CASO II

6.2.2.3 Propriedades físicas

O modelo Carreau aplicado nesta simulação não é o modelo padrão do OpenFOAM.

No modelo de Carreau-Yasuda, o valor de “a” é padronizado como sendo igual a 2 no OpenFOAM. Para a simulação em questão, o valor de “a” deveria ser igual a

0,664. Esta mudança no código foi então realizada, como pode ser confirmado a seguir:

```
Foam::viscosityModels::BirdCarreau::calcNu() const
{
    return
        nuInf_
        + (nu0_ - nuInf_)
        *pow(scalar(1) + pow(k_*strainRate(), a_), (n_ - 1.0)/a_);
}

// ***** Constructors ***** //

Foam::viscosityModels::BirdCarreau::BirdCarreau
(
    const word& name,
    const dictionary& viscosityProperties,
    const volVectorField& U,
    const surfaceScalarField& phi
)
:
    viscosityModel(name, viscosityProperties, U, phi),
    BirdCarreauCoeffs_(viscosityProperties.subDict(typeName + "Coeffs")),
    nu0_(BirdCarreauCoeffs_.lookup("nu0")),
    nuInf_(BirdCarreauCoeffs_.lookup("nuInf")),
    k_(BirdCarreauCoeffs_.lookup("k")),
    n_(BirdCarreauCoeffs_.lookup("n")),
    a_
    (
        BirdCarreauCoeffs_.lookupOrDefault
        (
            "a",
            dimensionedScalar("a", dimless, 0.644) // a_default = 2 , foi alterado para 0.644
        )
    ),
    .
```

Figura 6-21: Trecho do código BirdCarreau.C modificado

A alteração aconteceu no código de modelo de viscosidade Carreau, o BirdCarreau.C. O mesmo foi então recompilado através do comando *wmake libso*, a fim de que os novos valores fossem atualizados. O complemento *libso* é um necessário para interligar dinamicamente todos os códigos do OpenFOAM que têm conexão com o arquivo BirdCarreau.C.

Os demais parâmetros utilizados para definir o modelo de viscosidade utilizado por Chen(2006) são η_0 igual a 0,0022, η_∞ valendo 0,022, λ igual a 0,11s, a sendo igual a 0,644 e n igualado a 0,392. A densidade do sangue é igual a 1410 kg/m³.

Em *transportProperties*, este modelo de viscosidade é ativado:

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       transportProperties;
}
// *****

transportModel BirdCarreau;

nu          nu [ 0 2 -1 0 0 0 0 ] 0;

BirdCarreauCoeffs
{
  nu0          nu0 [ 0 2 -1 0 0 0 0 ] 1.560283688e-05; //μ0 = 0.022 Pa.s >> divided by density of 1410 kg/s = 1.56e-05
  nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1.560283688e-06; //μ∞ = 0.0022 Pa.s >> divided by density of 1410 kg/s = 1.56e-06
  k            k [ 0 0 1 0 0 0 0 ] 0.11; // k = 0.11 Seconds
  n            n [ 0 0 0 0 0 0 0 ] 0.392; //n= 0.392
}
```

Figura 6-22: *transportProperties* do Caso II

6.2.2.4 Esquemas numéricos

Varias simulações foram feitas com intuito de encontrar o melhor conjunto de esquemas numéricos. A possibilidade de combinação é extremamente grande. Apenas para esquemas de interpolação há cerca de cinquenta esquemas. Cada caso requer seu próprio conjunto de esquemas. A única maneira de se selecionar bem estes esquemas é se aprofundando na teoria dos mesmos além de muitas leituras nos fóruns de CFD. Cada usuário do OpenFOAM tem seu próprio esquema para cada tipo de simulação. Os esquemas aqui adotados foram escolhidos com base nessas pesquisas. Alguns foram variados entre simulações a fim de se tentar melhorar um pouco os resultados finais.

Os esquemas de discretização utilizados, localizados em *fvSchemes* são então apresentados:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSchemes;
}
// *****

ddtSchemes
{
    default      steadyState;
}

gradSchemes
{
    default cellLimited leastSquares 1.0;
    grad(U) cellLimited leastSquares 1.0;
}

divSchemes
{
    div(phi,U) bounded Gauss linearUpwind grad(U);
    div(phi,omega) bounded Gauss upwind;
    div(phi,k) bounded Gauss upwind;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default Gauss linear limited 0.5;
}

interpolationSchemes
{
    default      linear;
}

snGradSchemes
{
    default limited 0.3;
}

fluxRequired
{
    default      no;
    p            ;
}
```

Figura 6-23: *fvSchemes* para o Caso II

Assim como no Caso I, para este caso se utiliza um esquema para cada tipo de categoria de termos matemáticos (gradientes, divergentes, etc). Diante dos parâmetros de qualidade das três malhas geradas, é necessário o emprego de esquemas numéricos mais eficientes, com relação à exatidão da solução dos sistemas. Como os parâmetros de qualidade das três malhas têm valores que necessitam maior cuidado, optou-se por esquemas de maior custo computacional.

Para tratar dos gradientes, existe o método de Gauss e o método dos mínimos quadrados. Utilizou-se *cellLimited LeastSquares 1.0*. Este é o esquema de mínimos quadrados, isto é, discretiza em segunda ordem. Dr. Hrvoje Jasak, um dos criadores do OpenFOAM, aconselhou no Oitavo Workshop OpenFOAM, realizado na Universidade da Coréia em Jeju, Coréia do Sul, a utilizar o esquema *leastSquares* para o gradiente em malhas compostas por tetraedros. Segundo Murthy *et al.* (HANDBOOK OF NUMERICAL HEAT TRANSFER, 2006, pág. 25), o método *leastSquares* não impõe restrições sobre o formato das células e não requer uma malha estruturada. Por isso, *leastSquares* foi escolhido para o *gradSchemes*. A palavra-chave *cellLimited* complementa o esquema de mínimos quadrados. Esta função determina o gradiente limitado ao longo de uma linha que liga os centros de células adjacentes. No caso específico, está definido para 1.0. Este valor limita ao máximo os gradientes. Na prática, quando este valor tende a 1, aumenta-se a estabilidade e diminui-se a exatidão. Do contrário, se este valor tender a 0, significa que a estabilidade da solução diminuirá, mas a exatidão aumentará, visto que os gradientes não serão limitados. O ideal, portanto, seria obter o menor valor possível para que a exatidão da solução aumentasse, mas que ainda continuasse estável. Optou-se em manter o mesmo em 1 visto que demandaria muito tempo para tentar otimiza-lo.

Para lidar com laplacianos, utiliza-se o método de Gauss com interpolação linear e limitado a 0,5. Para esta faixa de valores de não-ortogonalidade, é recomendável se utilizar estes valores.

É importante ressaltar que há outro esquema muito importante. Se trata do esquema escolhido para o *snGradSchemes*. Este tipo de termo é tratado de forma especial no Openfoam. Ele é responsável por tratar dos gradientes normais de superfície. Ora, diante dos altos ângulos de não-ortogonalidades medidos pelo *checkMesh*, é indubitável que se apliquem esquemas mais rígidos para estes gradientes de superfícies. É interessante limitar esses gradientes em 0,5 (em média) para não-ortogonalidade entre 60 e 70, 0,7 (em média) para não-ortogonalidade entre 60 e 70. No trabalho se superestimou este valor a fim de alcançar uma estabilidade melhor na solução.

Embora se tenha várias simulações distintas, manteve-se o mesmo *fvSchemes* para todos os casos simulados.

6.2.2.5 Controle da solução

No arquivo *fvSolution*, estão definidos os critérios de convergência, além de outros dados importantes:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSolution;
}
// *****

solvers
{
    p
    {
        solver      GAMG;
        tolerance    1e-05;
        relTol       0.1;
        smoother     GaussSeidel;
        nPreSweeps    0;
        nPostSweeps   2;
        cacheAgglomeration on;
        agglomerator  faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels   1;
    }

    "(k|epsilon|R|nuTilda)"
    {
        solver      smoothSolver;
        smoother     symGaussSeidel;
        tolerance    1e-05;
        relTol       0.1;
    }

    U
    {
        solver PBICG;
        preconditioner DILU;
        tolerance 1e-8;
        relTol 0.1;
    }
}
SIMPLE
{
    nNonOrthogonalCorrectors 4;

    residualControl
    {
        p      1e-5;
        U      1e-5;
        "(k|epsilon|omega)" 1e-3;
    }
}
relaxationFactors
{
    fields
    {
        p      0.3;
    }
    equations
    {
        U      0.7;
        k      0.7;
        epsilon 0.7;
        R      0.7;
        nuTilda 0.7;
    }
}
```

Figura 6-24: *fvSolution* para o Caso II

Entre valores de não-ortogonalidade, de 40 até 60, geralmente se usa *nNonOrthogonalCorrectors* igual a 1. De 60 até 70, é recomendável definir o numero *nNonOrthogonalCorrectors* para 2. E, entre 70 até 80, muitos usuários do OpenFOAM utilizam *nNonOrthogonalCorrectors* igual a 3.

6.2.2.6 Resultados e comentários adicionais

Para o Caso II, foram realizadas muitas simulações, após se conseguir encontrar um conjunto de esquemas numéricos razoavelmente bons para as malhas em questão (convergindo para resíduos de 0,00001), foram então feitas mais cinquenta e quatro simulações. Cada uma requisitou em média 35 minutos para simular (utilizando simulação em paralelo com 8 núcleos de um processador core i7, com frequência de 3.2 Ghz). Das cinquenta e quatro simulações finais, dezoito delas são apresentadas neste trabalho. As outras 36 foram realizadas para modelos Newtonianos, considerando dois valores de viscosidade também simulados por Chen(2006). Para o modelo newtoniano, os resultados obtidos ficaram bem próximos. Apenas estão apresentados neste trabalho os resultados obtidos com o modelo Carreau-Yasuda, por ser muito utilizado em artigos científicos concernentes ao assunto.

Foram colhidos três perfis de velocidade para cada simulação para serem comparados com Chen(2006). O autor não revela a magnitude das velocidades, não obstante apresenta as formas dos perfis. Foi feita uma avaliação qualitativa, ajustando-se as curvas para larguras e alturas aproximadas. Pode-se se fazer isto visto que as vazões são iguais, devendo todas as curvas serem bem próximas. Diante da questão de não serem reveladas magnitudes de velocidade, poderia-se descobri-las apenas se uma condição fosse satisfeita: se os perfis forem idênticos em formato. Caso o perfil de velocidade fosse simétrico na direção de rotação do eixo da artéria, também poderia ser feita uma integração dupla na área da seção a fim de se obter a magnitude dos perfis de velocidade. Como não há simetria, foi necessário se restringir a comparação apenas ao critério qualitativo.

Como mencionado nos tópicos anteriores, utilizou-se nomes para referência de cada simulação. Relembrando-as: D1.5-Q12, D1.5-Q34, D1.5-Q78, D2-Q12, D2-Q34, D2-Q78, D3-Q12, D3-Q34 e D3-Q78.

A partir daqui, segue-se os perfis de velocidade obtidos na simulação para o modelo Carreau:

Os perfis de velocidade foram retirados de linhas que passam pelos pontos Pa, Pb e Pc. $Pa = x/D=0$, $Pb = x/D=1$ e $Pc = x/D=2$. D é o diâmetro. O eixo x/D é visto na Figura abaixo:

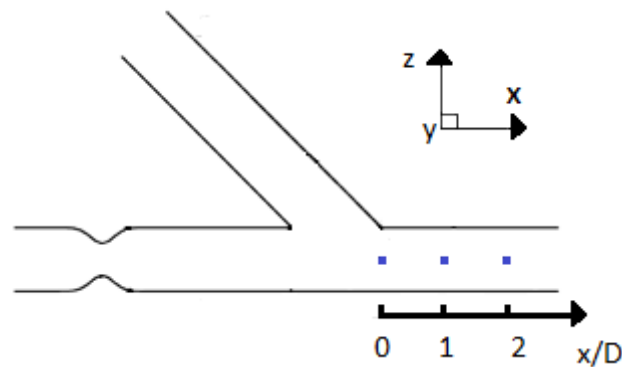


Figura 6-25: Localização dos pontos Pa, Pb e Pc para o CASO II

Os pontos Pa, Pb e Pc estão localizados na linha de raio da artéria, isto é, no centro. Da Figura 6.26 até a Figura 6.34 estão apresentados os gráficos comparativos para linhas cruzando os pontos Pa, Pb e Pc na direção z (vertical). Da Figura 6.35 até a Figura 6.43 estão apresentados os gráficos comparativos para linhas cruzando os pontos Pa, Pb e Pc na direção y (horizontal). Este mesmo sistema será utilizado para o CASO III, apresentado logo após estes resultados.

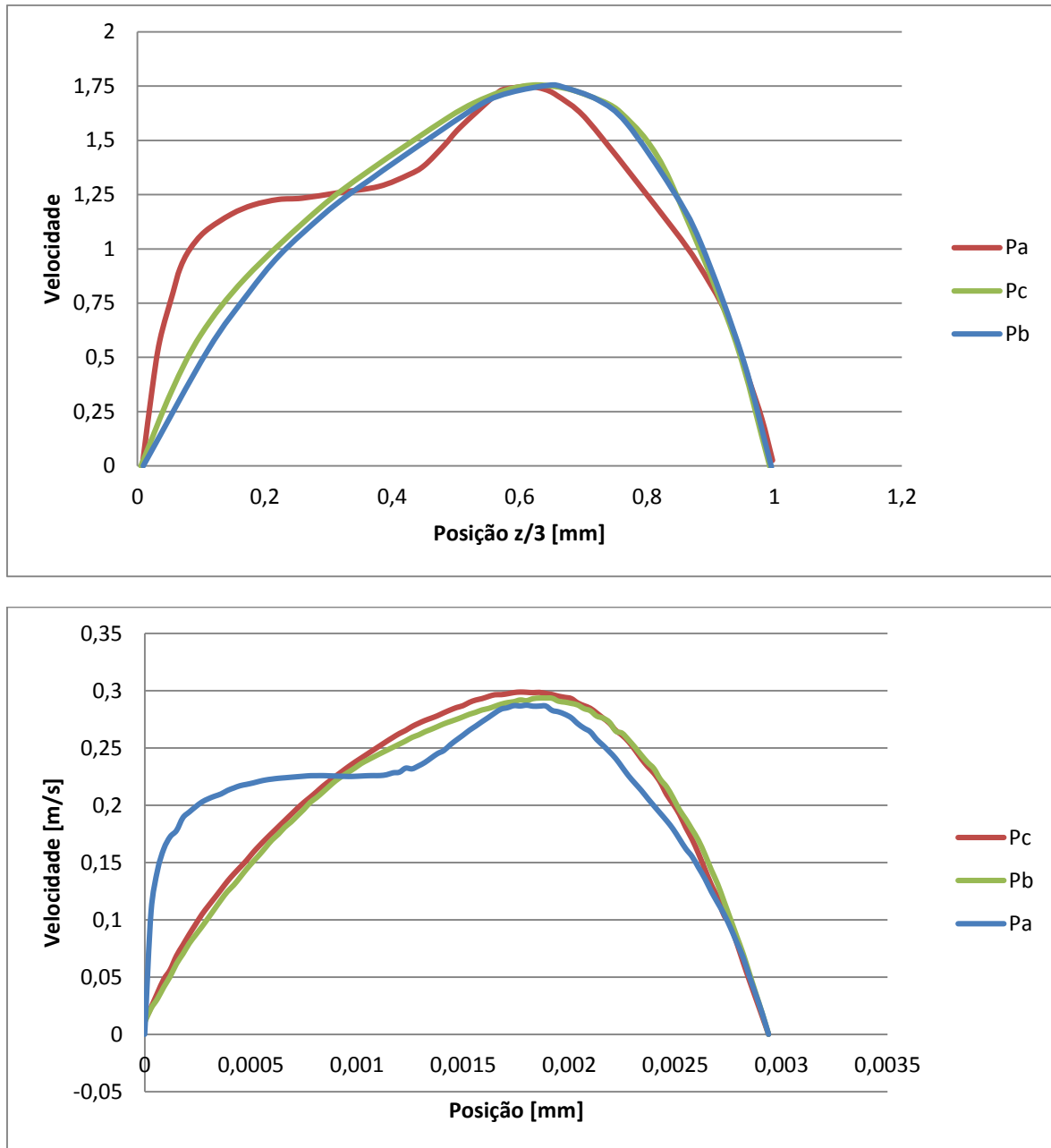


Figura 6-26: Comparação entre os perfis de velocidade D1.5-Q12 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

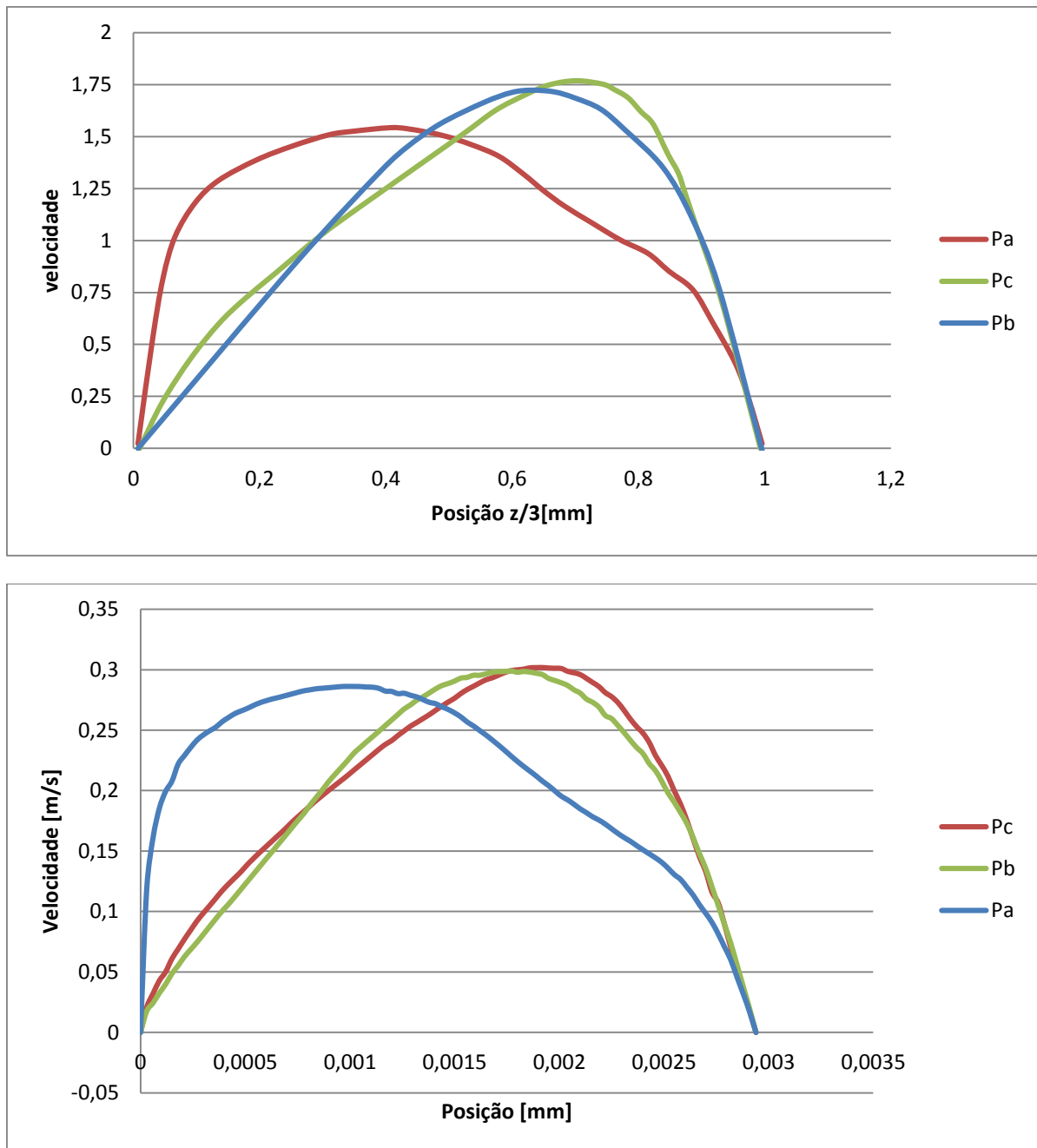


Figura 6-27: Comparação entre os perfis de velocidade D1.5-Q34 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

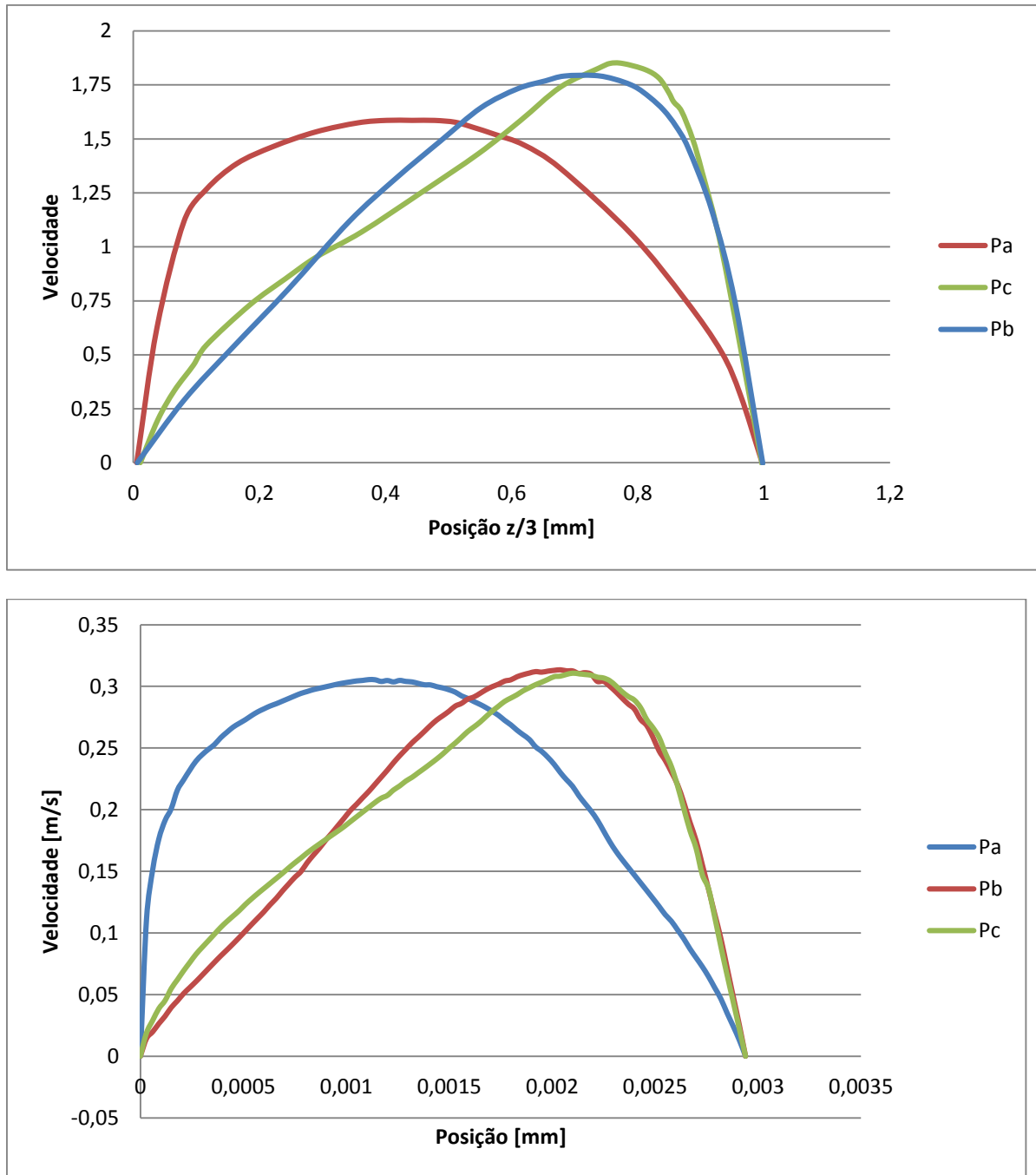


Figura 6-28: Comparação entre os perfis de velocidade D1.5-Q78 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

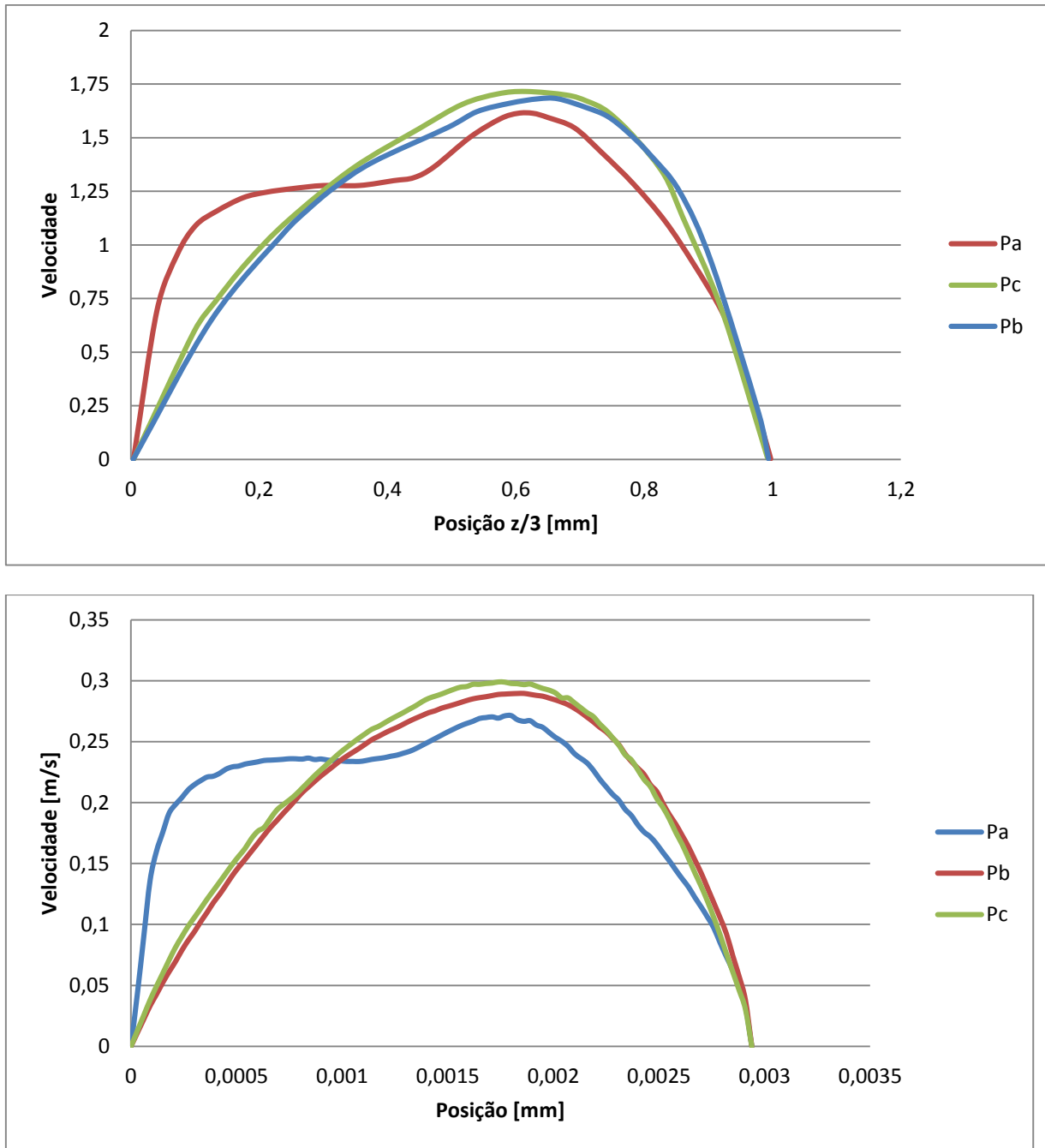


Figura 6-29: Comparação entre os perfis de velocidade D2-Q12 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

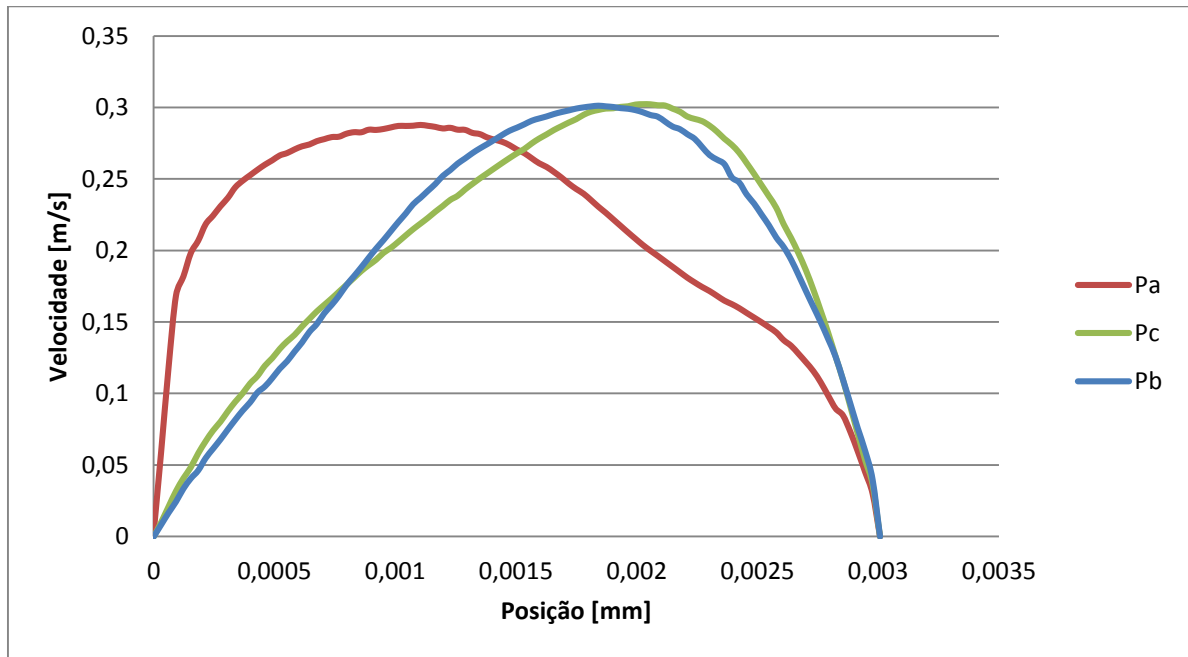
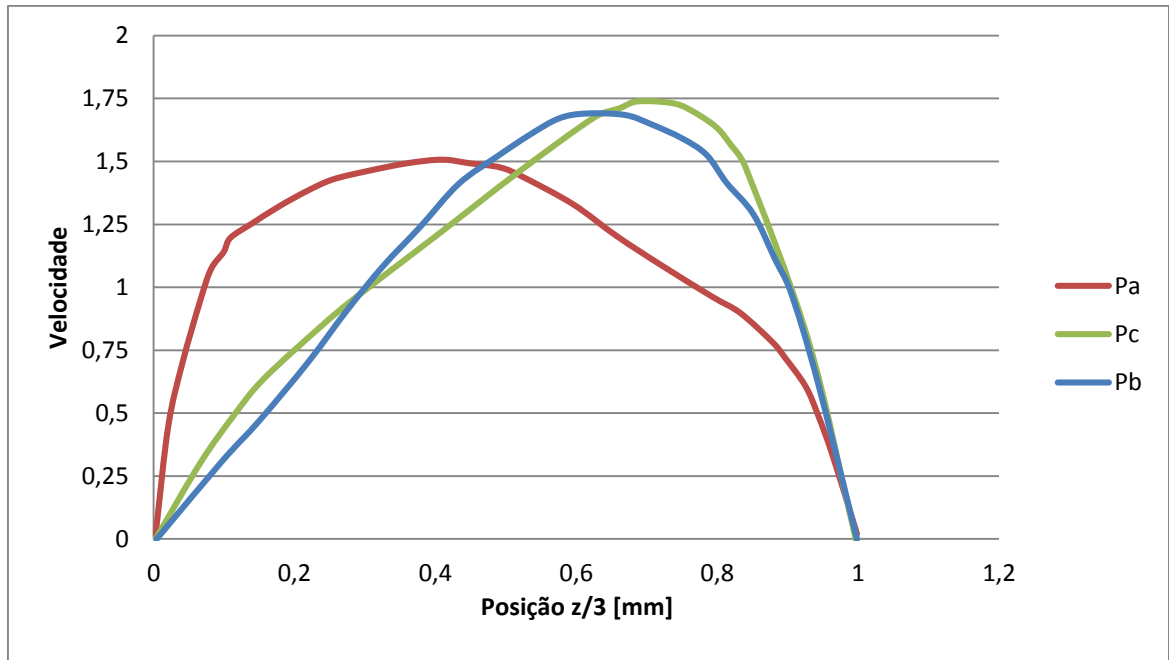


Figura 6-30: Comparação entre os perfis de velocidade D2-Q34 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

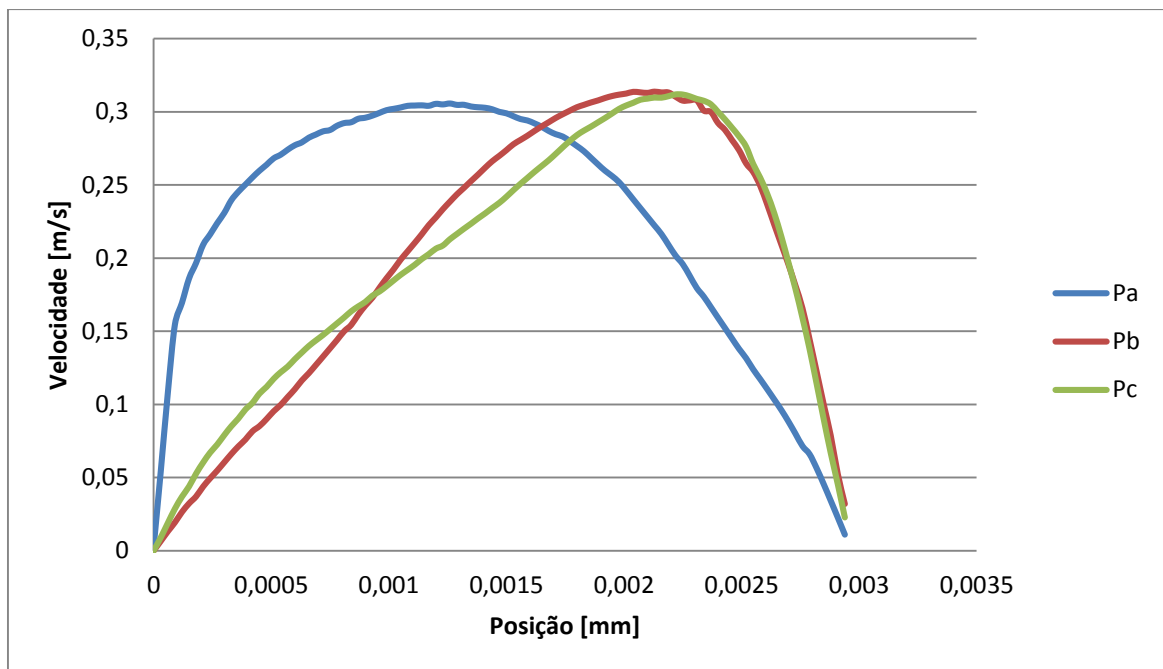
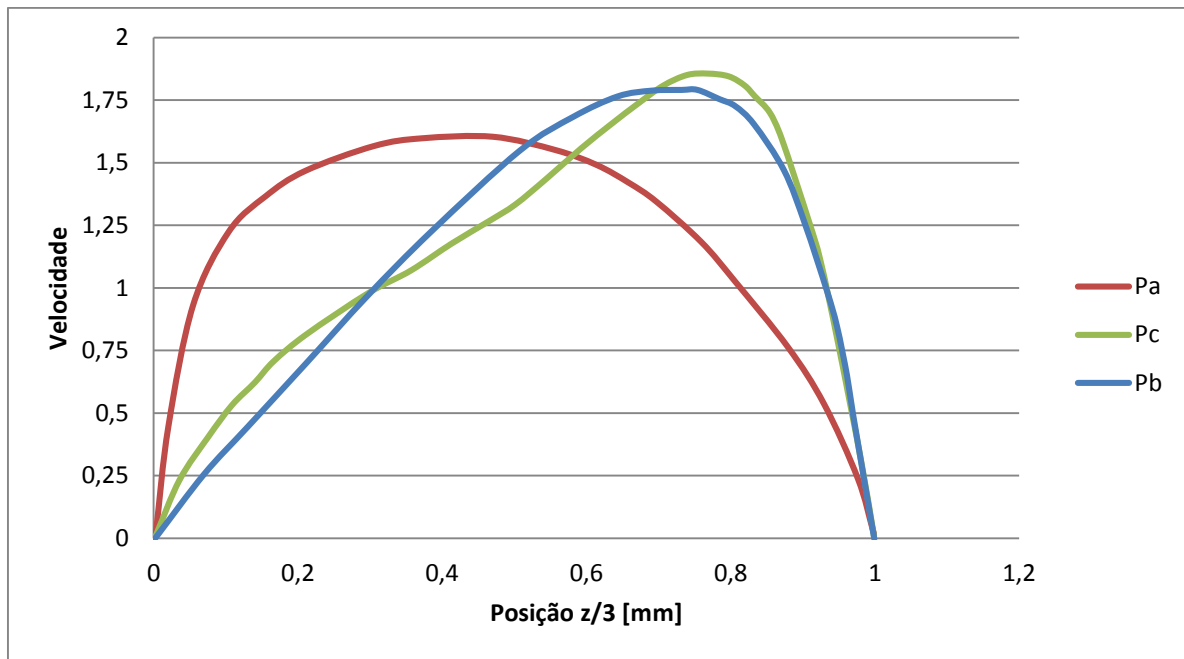


Figura 6-31: Comparação entre os perfis de velocidade D2-Q78 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

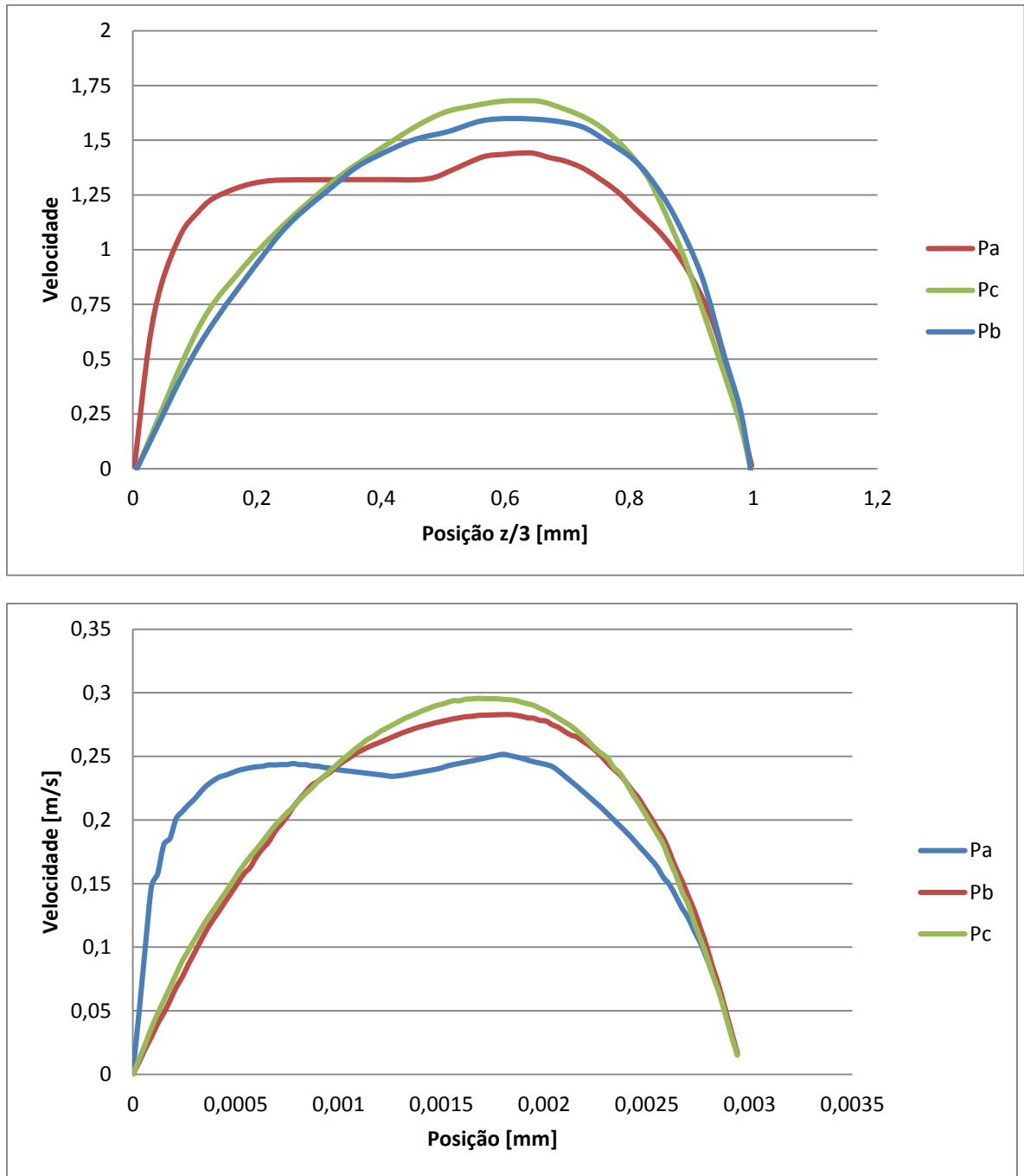


Figura 6-32: Comparação entre os perfis de velocidade D3-Q12 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

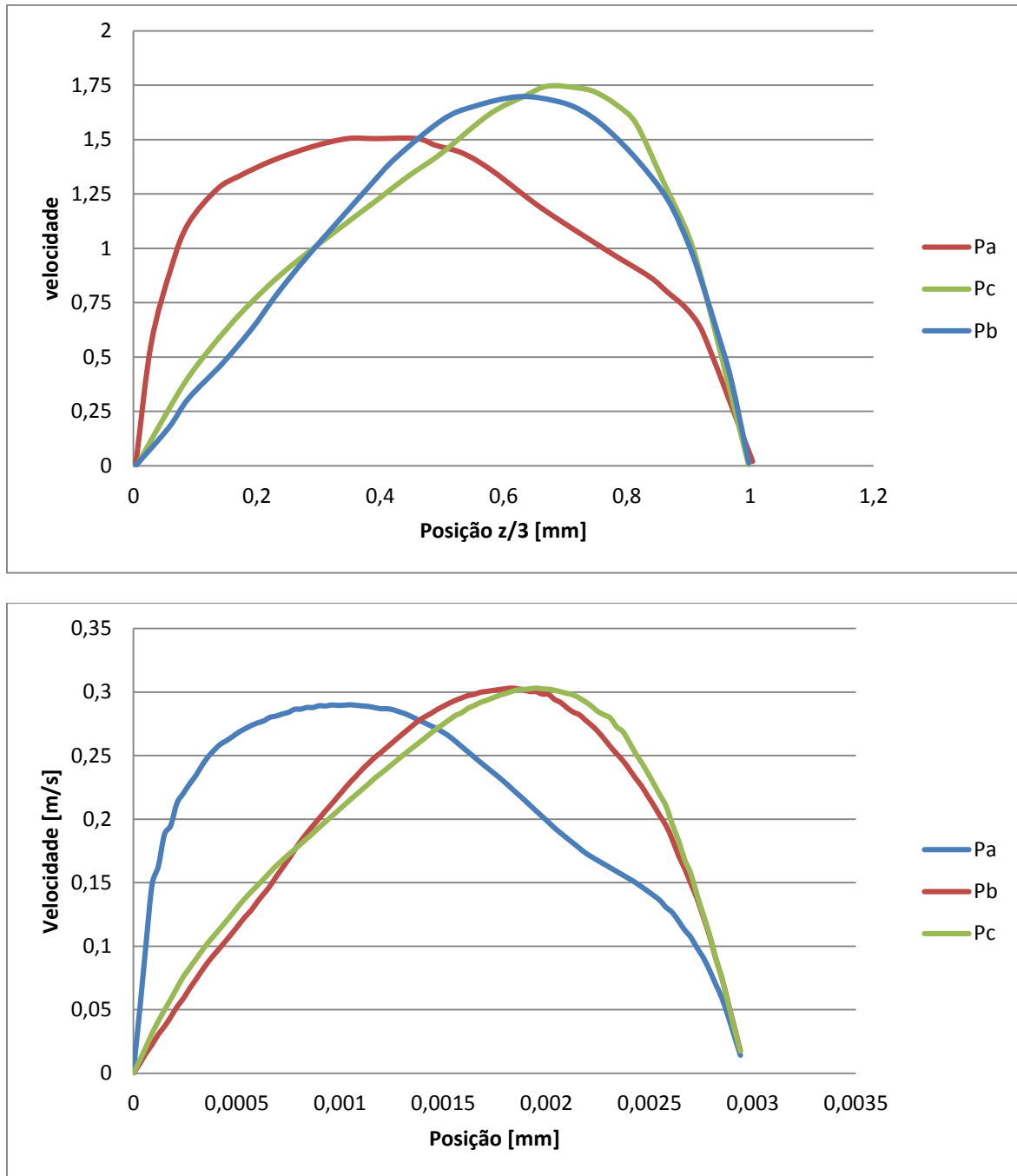


Figura 6-33: Comparação entre os perfis de velocidade D3-Q34 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

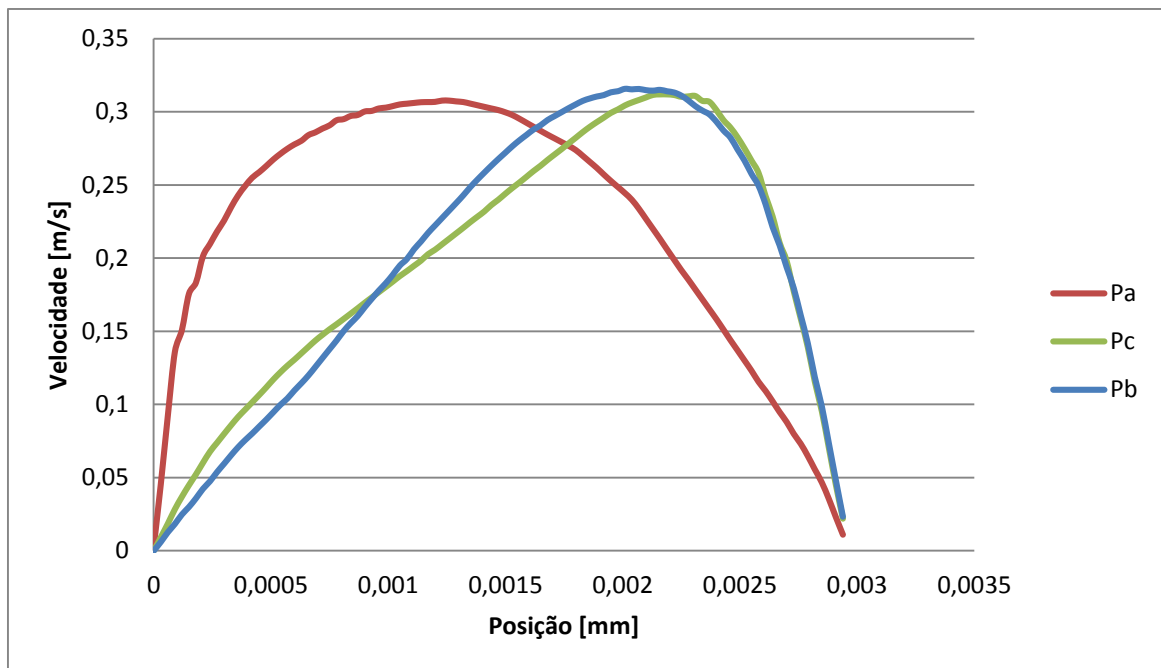
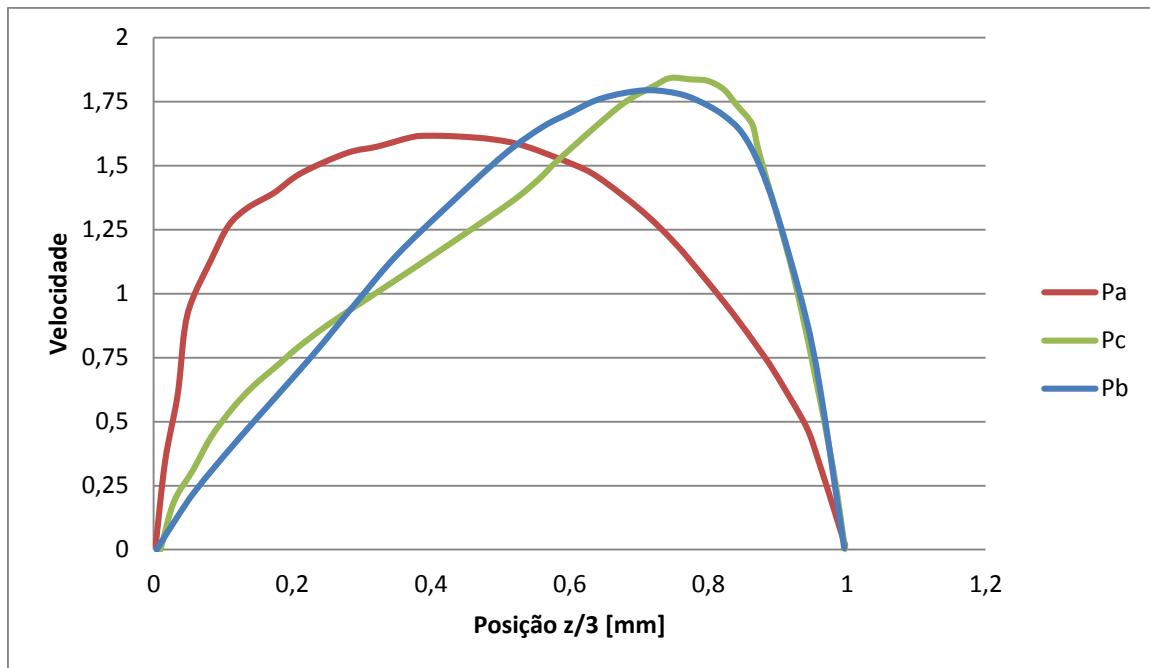


Figura 6-34: Comparação entre os perfis de velocidade D3-Q78 das linhas na direção z passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

Da Figura 6.35 até a Figura 6.43 estão apresentados os gráficos comparativos para linhas cruzando os pontos Pa, Pb e Pc na direção y (horizontal):

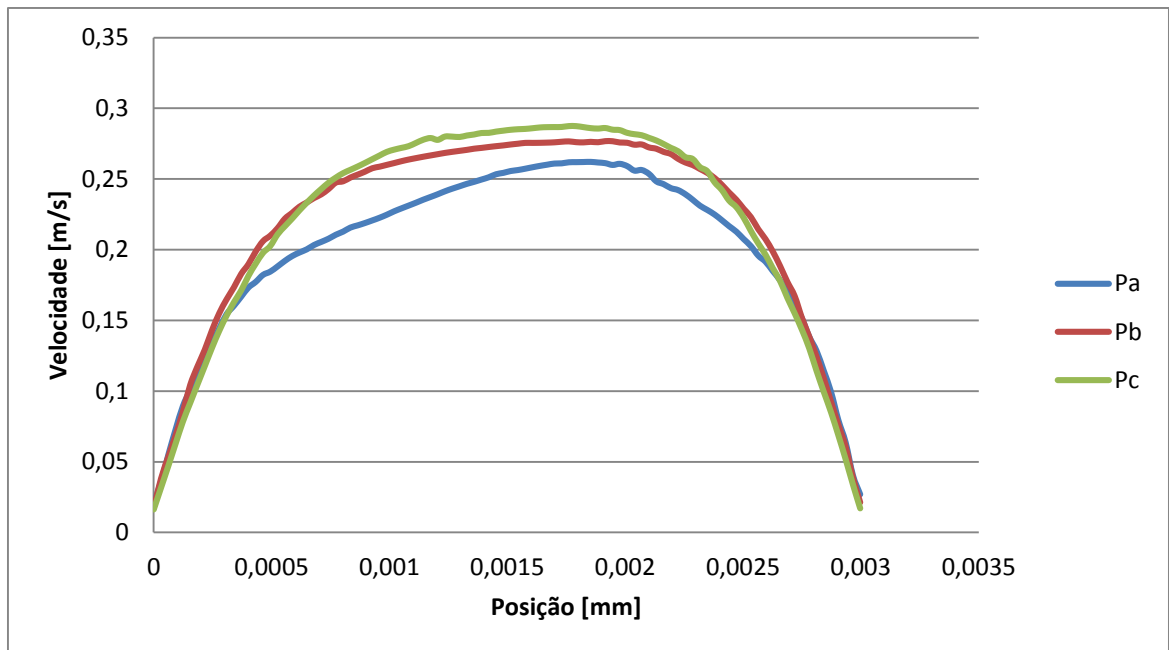
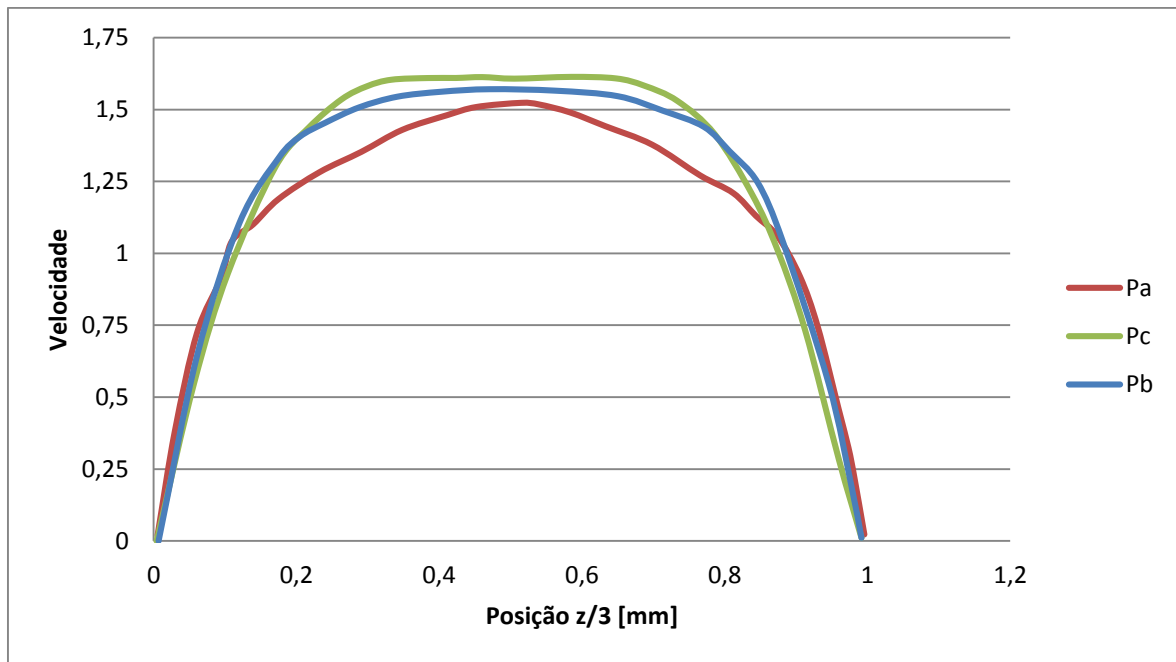


Figura 6-35: Comparação entre os perfis de velocidade D1.5-Q12 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

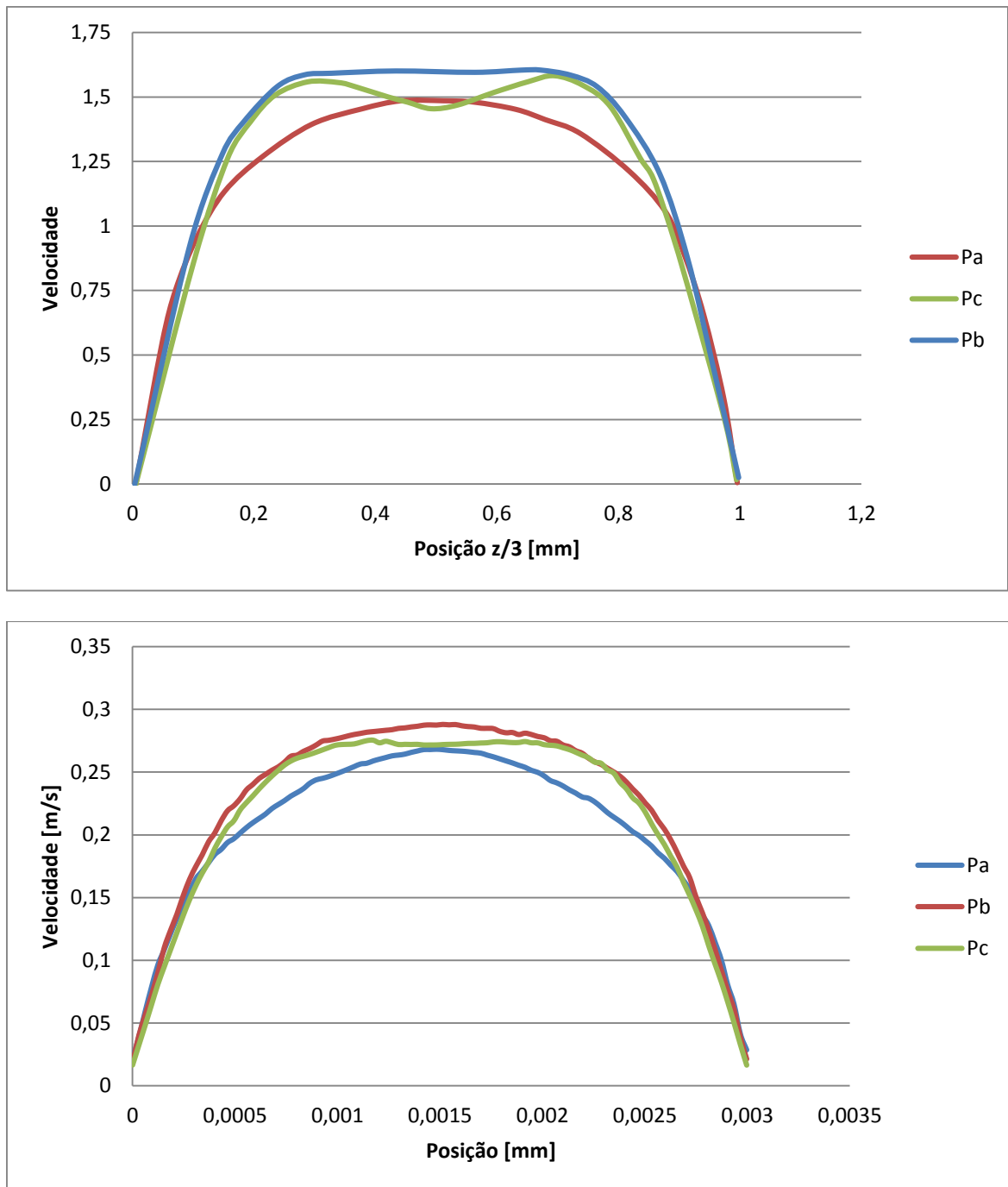


Figura 6-36: Comparação entre os perfis de velocidade D1.5-Q34 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

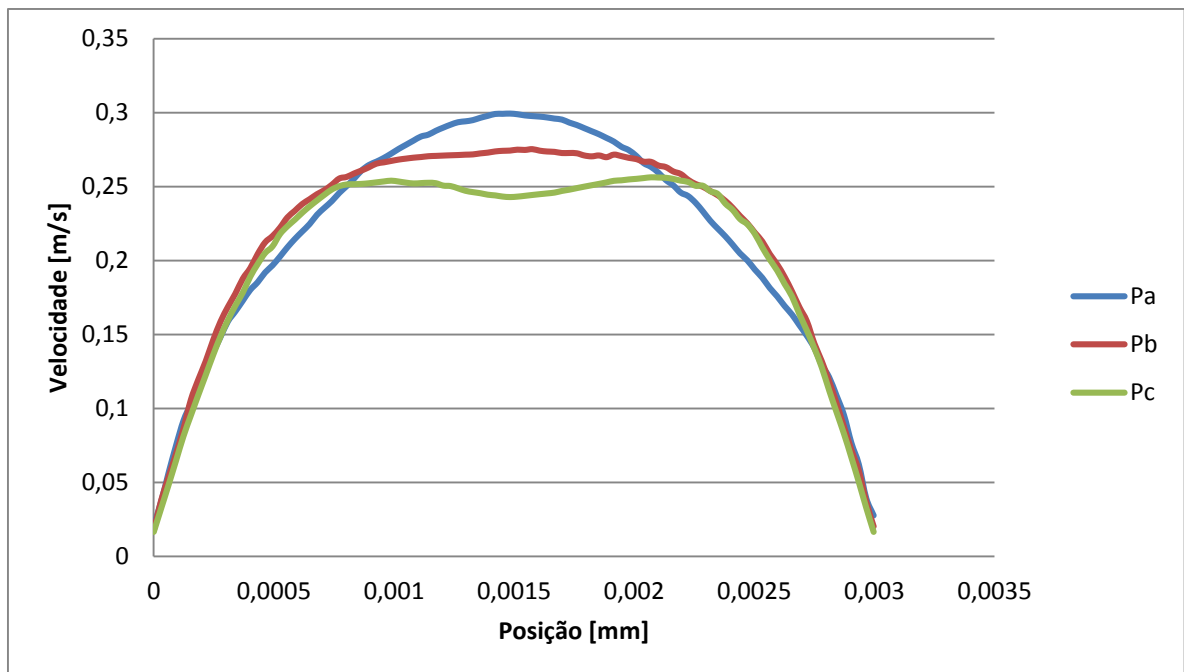
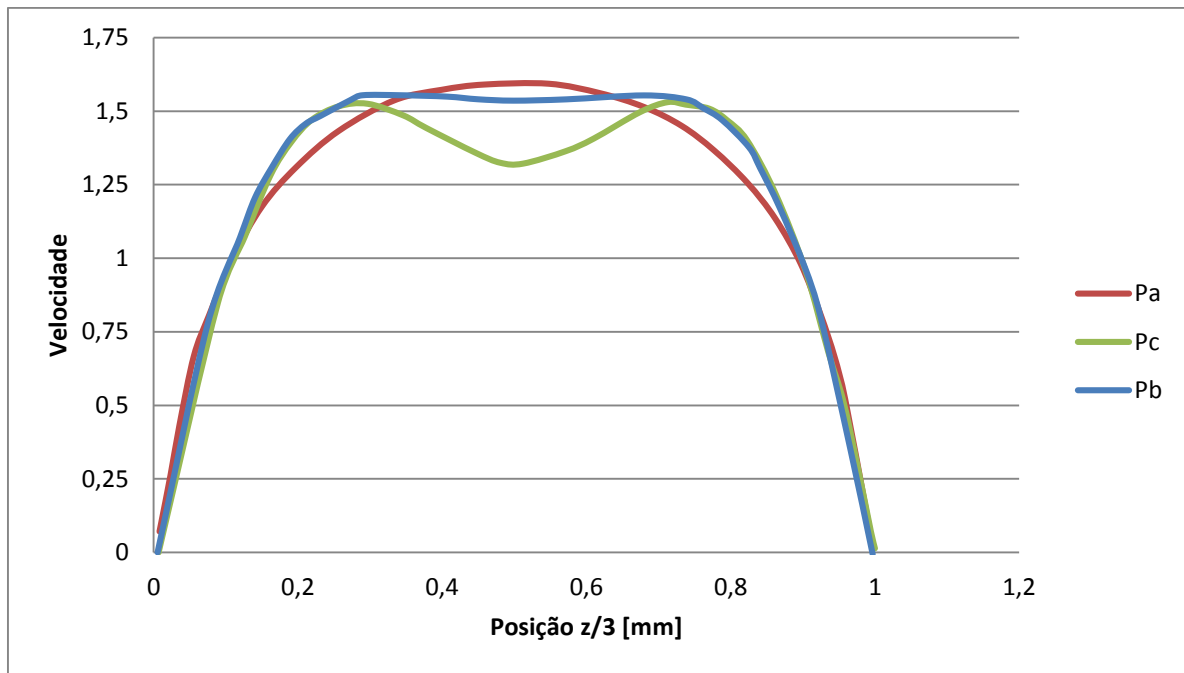


Figura 6-37: Comparação entre os perfis de velocidade D1.5-Q78 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

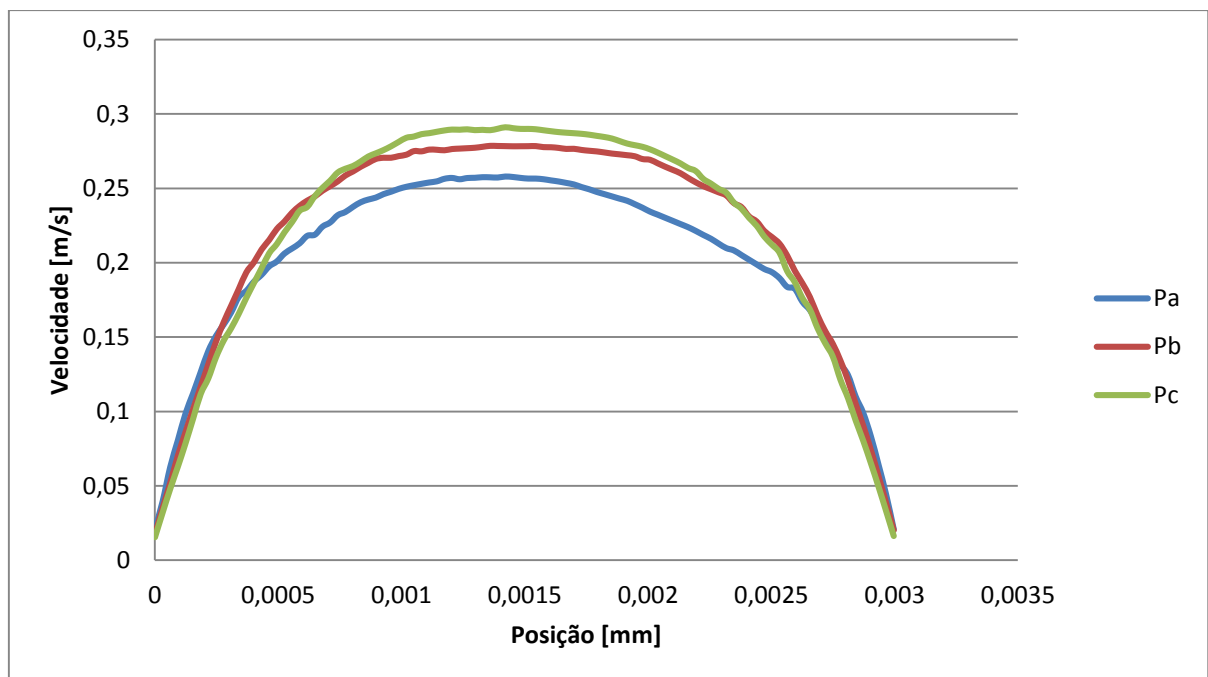
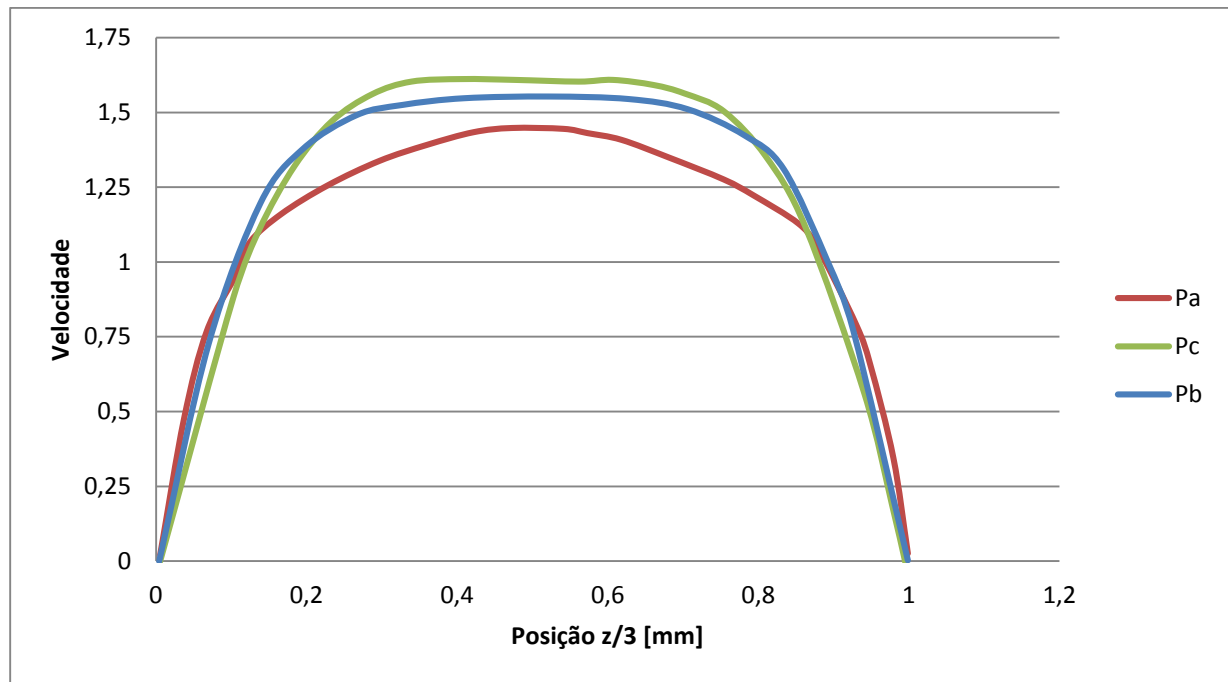


Figura 6-38: Comparação entre os perfis de velocidade D2-Q12 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

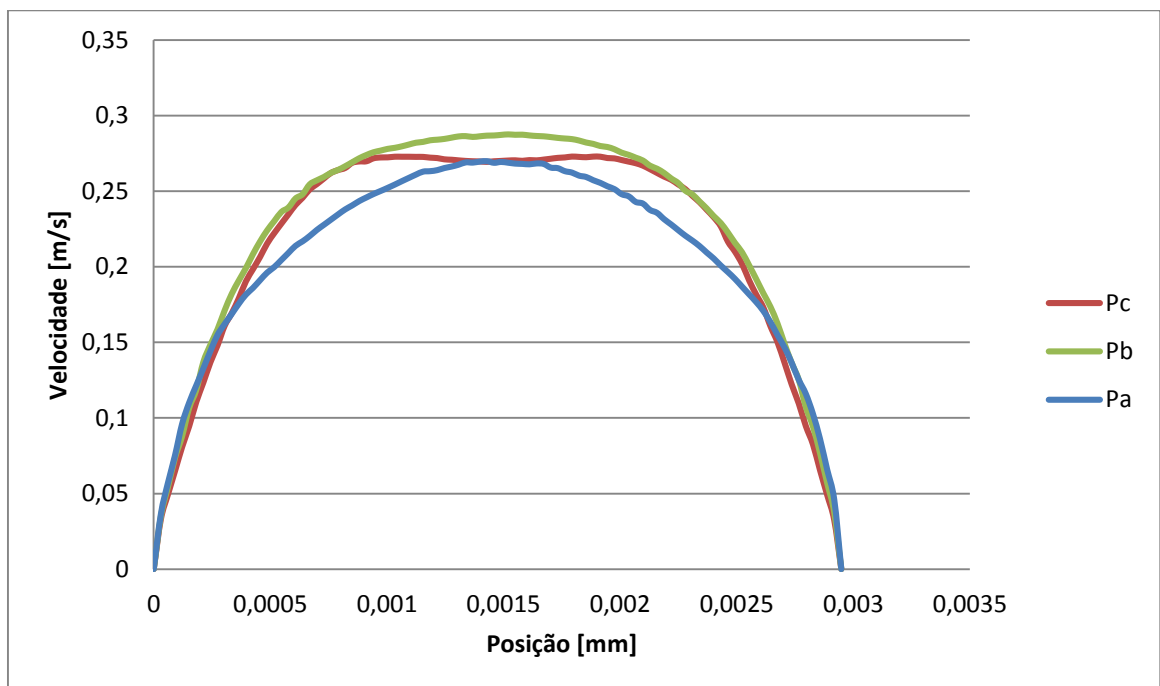
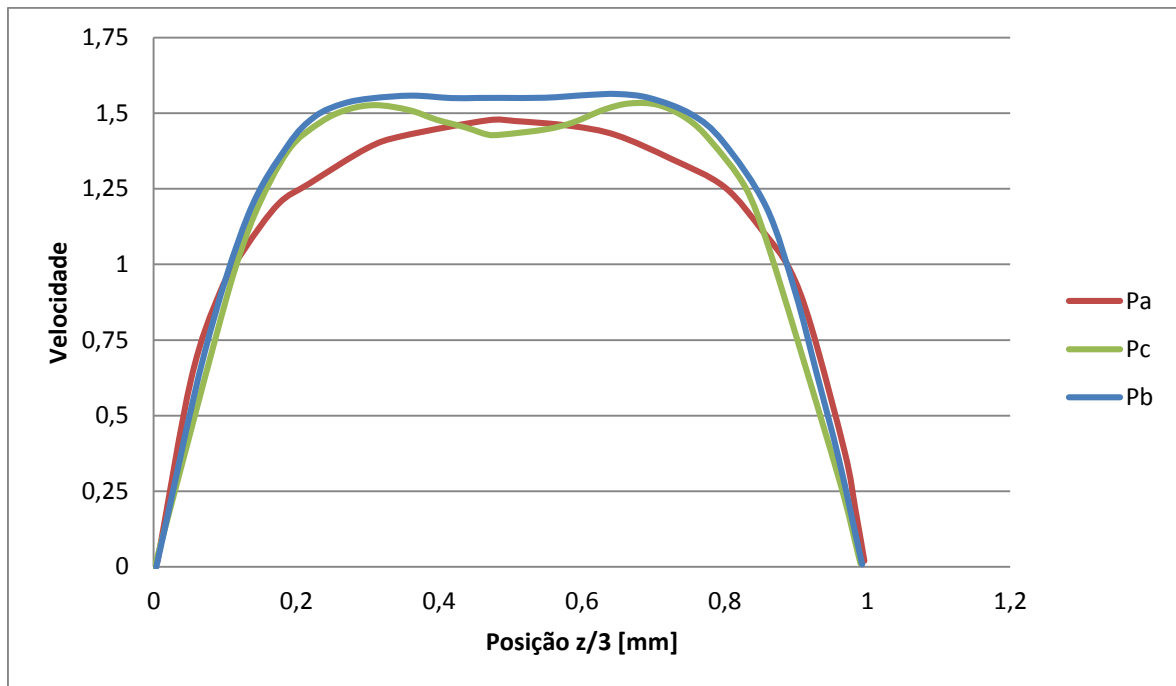


Figura 6-39: Comparação entre os perfis de velocidade D2-Q34 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

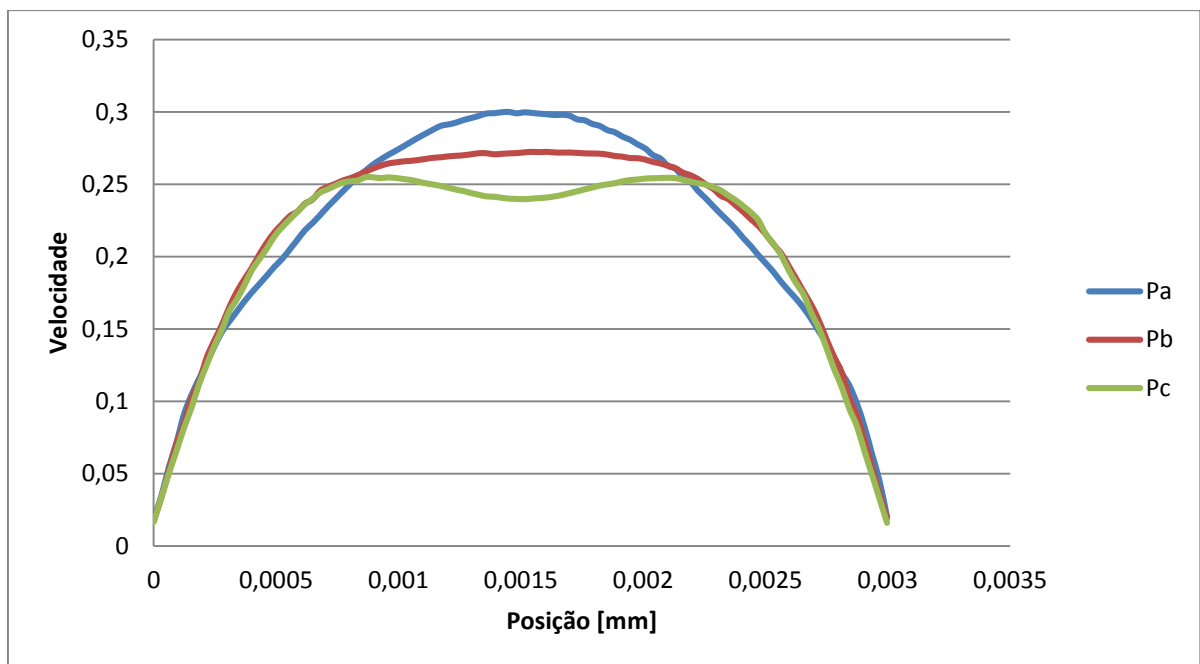
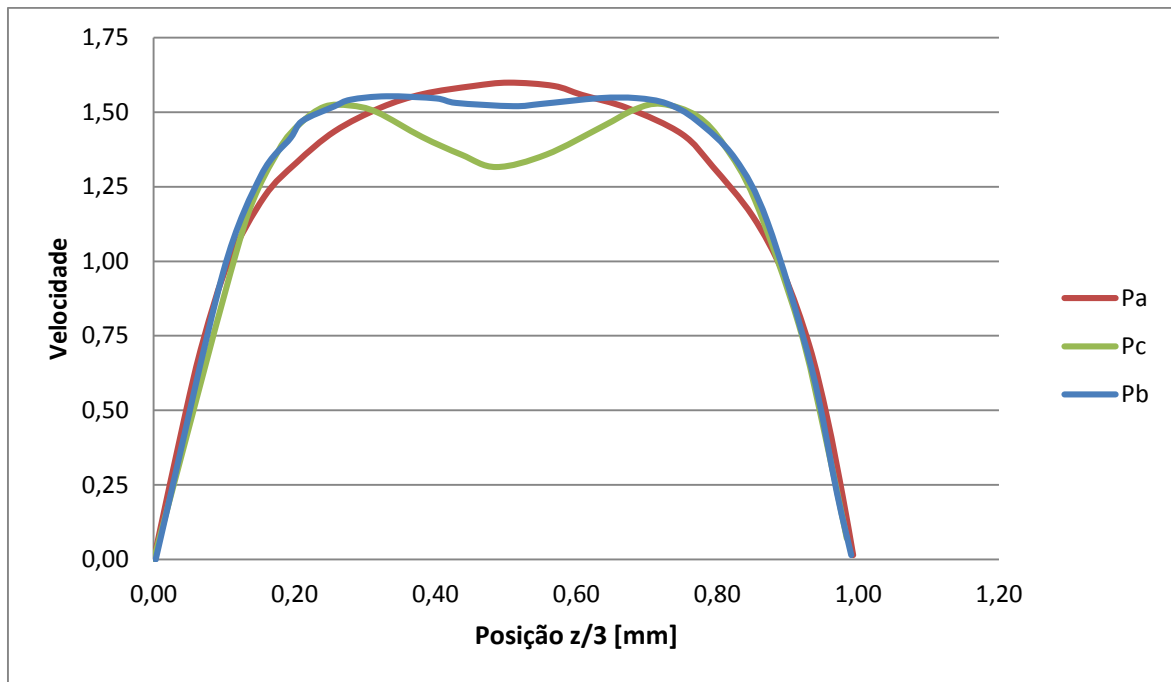


Figura 6-40: Comparação entre os perfis de velocidade D2-Q78 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

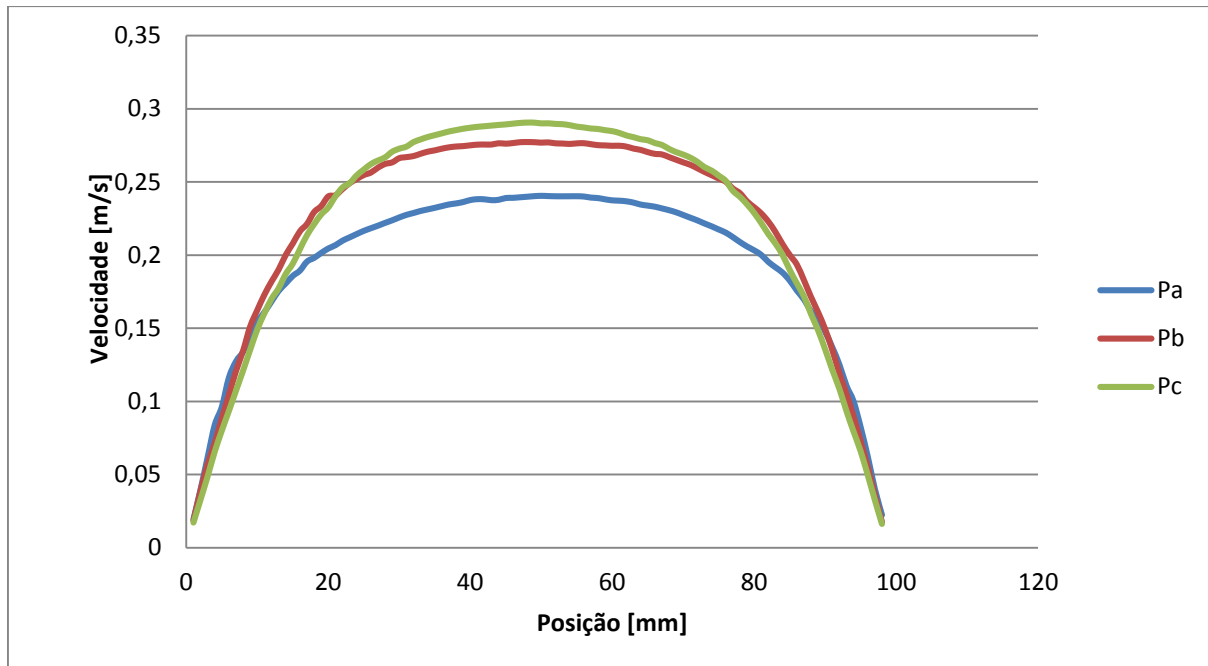
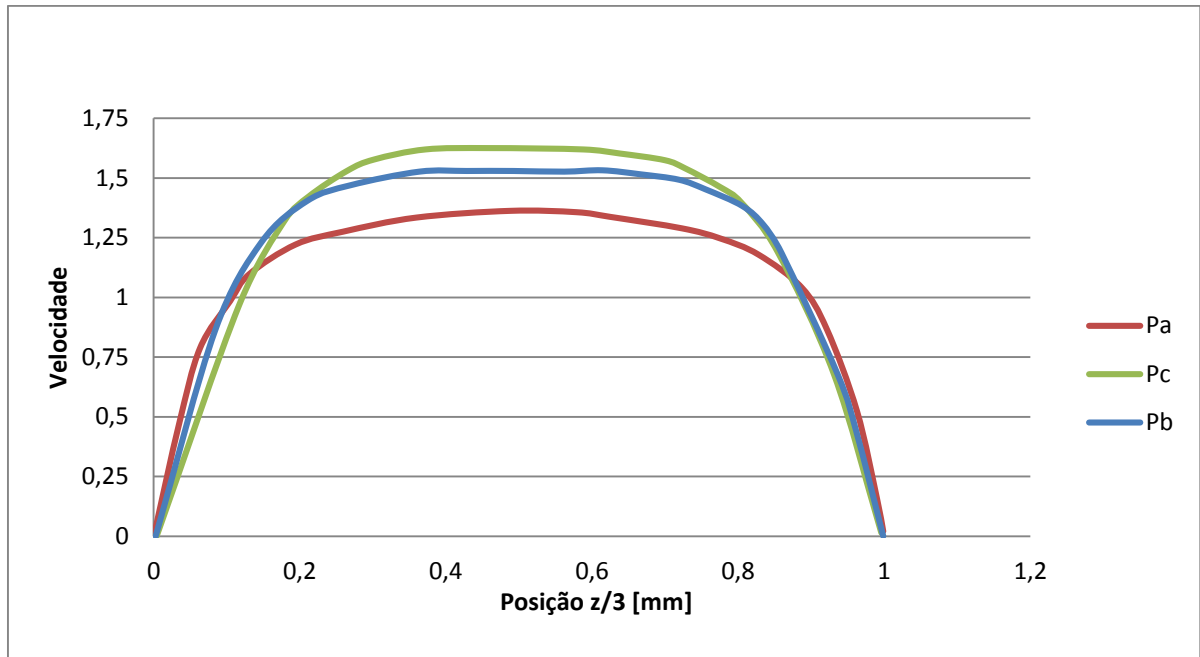


Figura 6-41: Comparação entre os perfis de velocidade D3-Q12 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

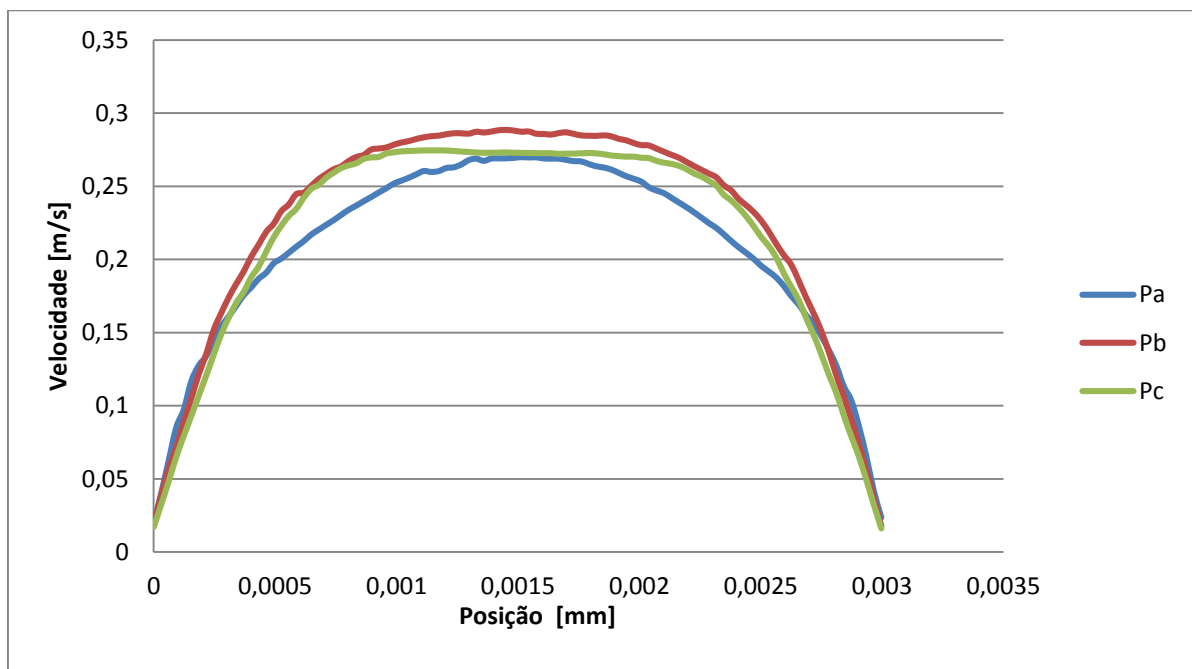
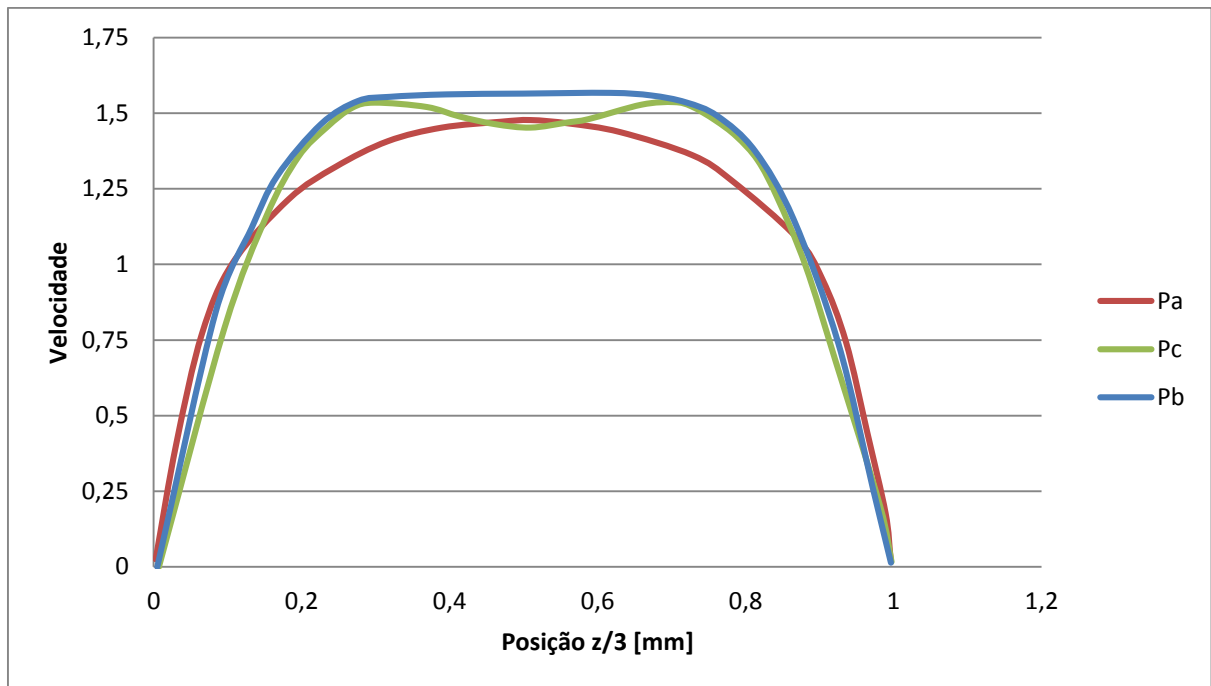


Figura 6-42: Comparação entre os perfis de velocidade D3-Q34 das linhas na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

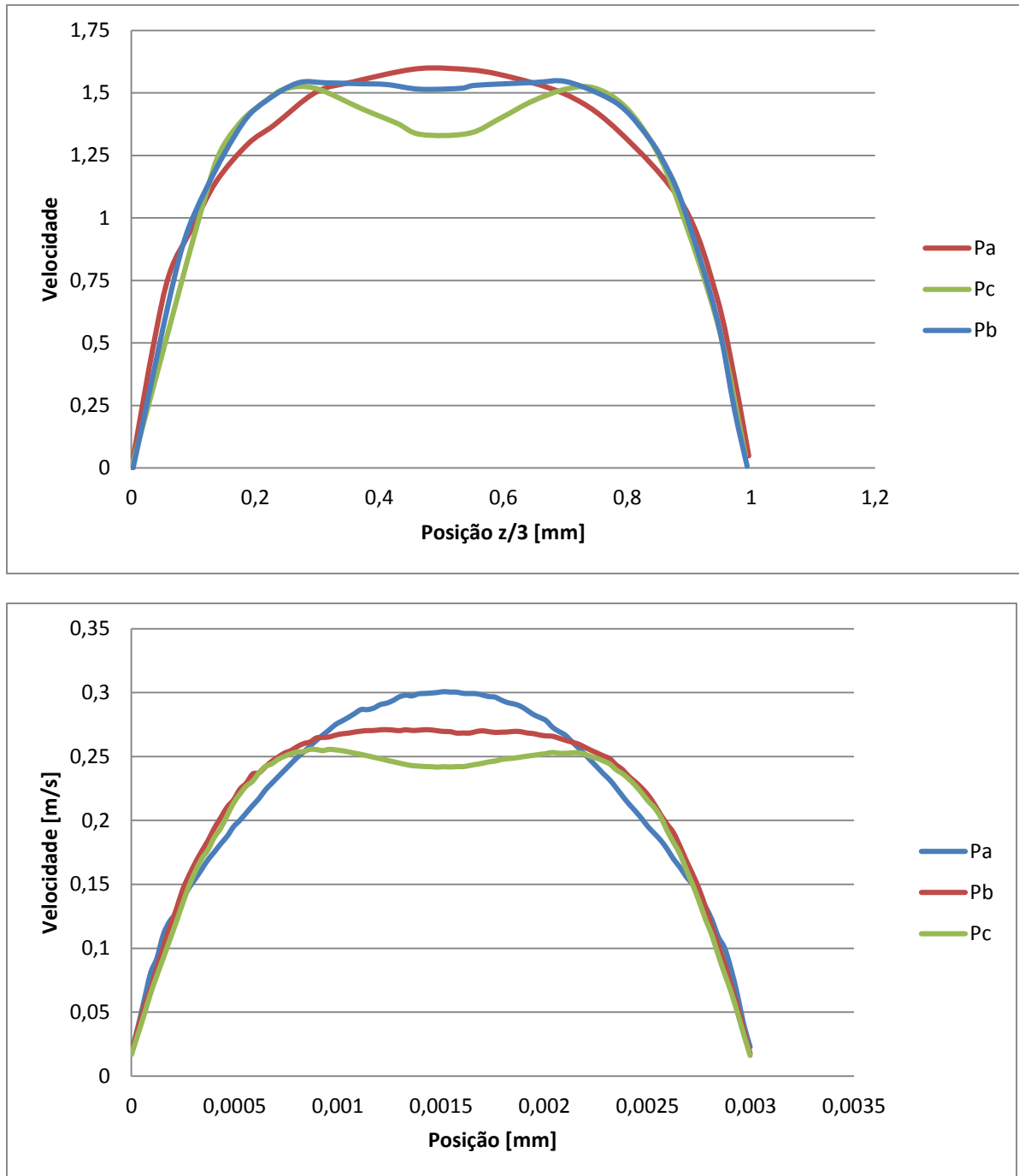


Figura 6-43: Comparação entre os perfis obtidos para D3-Q78 na direção y passando pelos pontos Pa Pb e Pc. Os resultados qualitativos de Chen são mostrados acima. Abaixo estão os resultados deste trabalho

Os resultados estão bem próximos para o perfil vertical (perfis na direção z). Para perfis na direção y , os resultados foram mais incertos. Por, exemplo, um comportamento não esperado de assimetria surgiu para o caso D1.5-Q12. Para esta malha em especial, um possível corte na geometria em simetria no plano z - x poderia

solucionar o problema (aplicando condições de contorno de gradiente de pressão nulo e gradiente de velocidade nulo).

Dois fatores podem ser considerados como possíveis principais para a falta de exatidão das soluções. O primeiro, é a própria malha, que apresenta ângulos de não-ortogonalidade intermediários. Valores considerados altos geralmente estão acima de 80 graus. O segundo, são os parâmetros para os esquemas numéricos. Alguns utilizam a função *limited*, que embora sejam muito bons, há a necessidade de se conhecer bem os valores utilizados (valores entre 0 e 1). Somente com testes pode-se conseguir encontrar acurácia maior. Mesmo diante das incertezas, ainda se pode considerar que os resultados foram muito próximos em várias simulações.

Percebe-se também que o perfil “M”, que é a curva gerada na posição “Pc”, foi atenuada (por conta de altos valores de velocidade ao redor do centro - posição “Pa”. Este perfil “M” dá a forma de uma curva com baixas velocidades perto do centro. Esta curva tem este formato já que durante o encontro dos dois fluxos, há a redução de velocidade na região de encontro. A redução se dá pelo fato das velocidades das duas entradas entrarem em contato com vetores de velocidades diferentes. Por conta da componente *z* do fluxo proveniente da artéria secundária ser negativo, este fluxo que deveria passar pela região central é deslocado para a região inferior da artéria(o que pode ser notado nos perfis “Pb” e “Pc” na direção *y*).

6.3 CASO III

6.3.1 Descrição do caso

A geometria da artéria deste caso foi semelhante à do CASO II, exceto pelo fato desta ter o dobro do diâmetro, isto é, 6 milímetros. A simulação realizada é a de Bertolotti e Deplano(2000).

A viscosidade adotada foi de 0,0072 Pa.s, isto é, Newtoniano.

O gráfico de fluxo fornecido por Bertolotti e Deplano(2000) é mostrado a seguir:

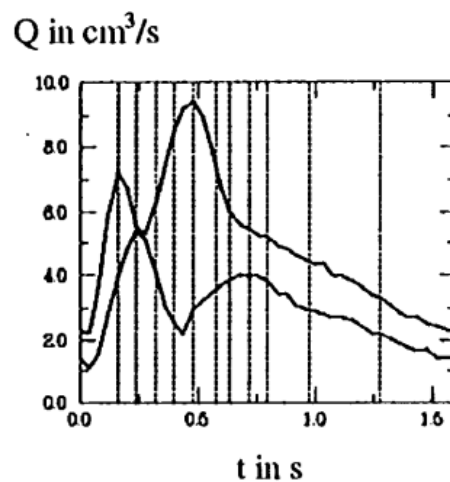


Figura 6-44: Perfil de velocidades instantâneas para as duas entradas. A curva superior é o fluxo instantâneo na artéria secundária. A curva inferior corresponde ao fluxo instantâneo na artéria anfitriã.
Fonte: Bertolotti e Deplano(2000)

Na saída se considerou pressão relativa nula.

6.3.2 Simulação

6.3.2.1 Geração de malha

O processo de geração de malha foi idêntico ao CASO II. Por coincidência, a malha era exatamente a mesma, diferindo apenas pelo fato da mesma ter o dobro do

tamanho, isto é, 6 milímetros de diâmetro. No CASO II, a malha tinha 3 milímetros de diâmetro.

6.3.2.2 Condições iniciais de contorno

O arquivo de condições de contorno para velocidade foi o seguinte:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    location     "0";
    object       U;
}
// *****
dimensions      [0 1 -1 0 0 0];
internalField    uniform (0 0 0);
boundaryField
{
    inlet_host
    {
        type            uniformFixedValue;
        uniformValue     tableFile;
        tableFileCoeffs
        {
            outOfBounds    repeat;
            fileName        "host_profile";
        }
    }
    inlet_graft
    {
        type            uniformFixedValue;
        uniformValue     tableFile;
        tableFileCoeffs
        {
            outOfBounds    repeat;
            fileName        "graft_profile";
        }
    }
    wall
    {
        type            fixedValue;
        value            uniform (0 0 0);
    }
    outlet
    {
        type            zeroGradient;
    }
}

// *****
```

Figura 6-45: Condições de contorno para velocidade

Para este caso, especificamente, houve a necessidade de criar condições de contorno que variassem com o tempo. Duas alternativas estiveram presentes: A primeira era utilizar a biblioteca GroovyBC. A segunda era utilizar a própria linguagem do Openfoam. Optou-se em criar essas condições variantes no tempo com a linguagem do OpenFOAM. Com algumas pesquisas nos fóruns de CFD encontrou-se o código aplicado na figura acima. Foram gerados arquivos onde foram depositos os vetores de velocidade para cada instante de tempo: *host_profile* para a entrada na artéria anfitriã e *graft_profile* para as velocidades na artéria secundária.

Do perfil de vazão, por meio da lei da conservação de massa calculou-se a velocidade média de entrada para as artérias anfitriã e secundária:

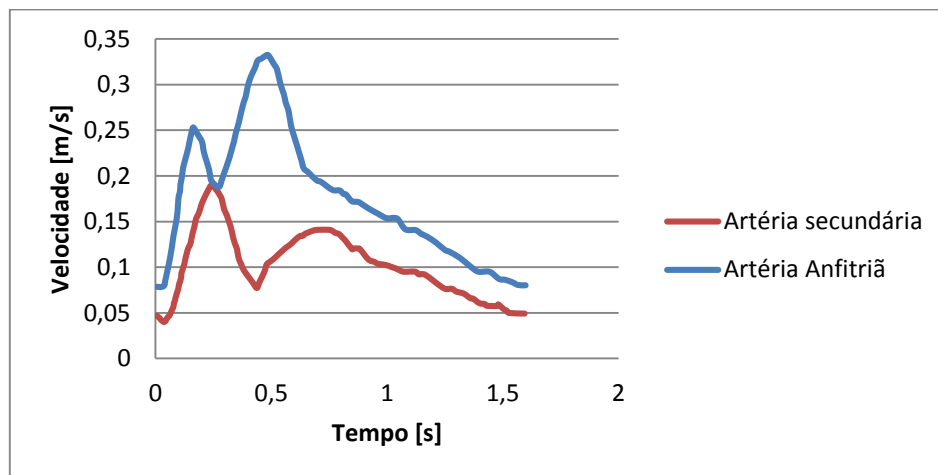


Figura 6-46: Perfis de velocidade média instantânea nas seções de entrada.

Nas entradas aplicou-se então um perfil constante da velocidade média das entradas. Este perfil variava conforme o gráfico da Figura acima. Foi gerado um alongamento nas entradas e saídas a fim de ser alcançar o perfil de velocidades completamente desenvolvido.

O arquivo de condições de contorno para pressão foi o seguinte:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       p;
}
// *****

dimensions      [0 2 -2 0 0 0 0];
internalField    uniform 0;

boundaryField
{
    inlet_host
    {
        type      zeroGradient;
    }
    inlet_graft
    {
        type      zeroGradient;
    }
    wall
    {
        type      zeroGradient;
    }
    outlet
    {
        type      fixedValue;
        value      uniform 0;
    }
}
```

Figura 6-47: Condições de contorno para pressões no CASO III

6.3.2.3 Propriedades físicas

O fluido é considerado Newtoniano. O arquivo *transportProperties* fica sendo então:

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       transportProperties;
}
// *****

transportModel  Newtonian;

nu              nu [ 0 2 -1 0 0 0 ] 6.85714e-06; // viscosity=0.0072 Pa.s --->
//                                                    density assumption=1050 kg/m³

// *****
```

Figura 6-48: *transportProperties* para o CASO III

6.3.2.4 Esquemas numéricos

Os esquemas numéricos utilizados aqui foram os mesmos aplicados para o CASO II, diferindo apenas pelo fato deste ser transiente. Para discretização utilizou-se o método de segunda ordem de Chank-Nicolson. Este esquema é de segunda ordem e implícito. É um dos melhores, se não for o melhor. Embora seja muito bom, tem um alto custo computacional. O valor utilizado para *CrankNicolson* pode variar entre 0 e 1. Quando definido em 1, significa que será usado como um puro Crank-Nicolson (segunda ordem), que dará maior acurácia e mais oscilações no processo de convergência. Quando definido em 0, significa que o esquema rodará como um esquema de Euler puro (primeira ordem). O valor utilizado foi igual a 0,5, a fim de se obter o melhor dos dois.

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSchemes;
}
// *****

ddtSchemes
{
    default      CrankNicolson 0.5;
}
gradSchemes
{
    default      cellLimited leastSquares 1.0;
    grad(U)      cellLimited leastSquares 1.0;
}

divSchemes
{
    div(phi,U)   bounded Gauss linearUpwind grad(U);
    div(phi,omega) bounded Gauss upwind;
    div(phi,k)   bounded Gauss upwind;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default      Gauss linear limited 0.5;
}

interpolationSchemes
{
    default      linear;
}

snGradSchemes
{
    default      limited 0.3;
}

fluxRequired
{
    default      no;
    p            ;
}

```

Figura 6-49: fvSchemes para o CASO III

6.3.2.5 Controle da Solução

Para controle da solução, adotaram-se o mesmos critérios do CASO II. Como este é o único caso transiente simulado neste trabalho, é a primeira vez que se utiliza o número de Courant. Abaixo se encontra sua definição:

$$C = \frac{u\Delta t}{\Delta x} \quad (8)$$

Onde u é a velocidade, Δt é o passo de tempo (Δt) e Δx é o espaçamento de grades da malha.

O número de Courant é um dos parâmetros mais importantes em simulações transientes, pois através destes simples valor, se consegue relacionar a velocidade

de escoamento com a resolução da malha. Na prática, é recomendável C menor ou igual a 1.

Como a malha não tem os volumes uniformes, surge a dificuldade de se medir os comprimentos dos volumes finitos, o Δx .

Diante disso, o OpenFOAM disponibiliza a função *adjustableRunTime* para alguns solvers. Quando não há esta opção, dependendo do solver, é possível modificar o código do OpenFOAM para se acrescentar este controle do número de Courant. Esta palavra-chave permite que o OpenFOAM ajuste o *deltaT* a fim de que o número de Courant não ultrapasse o valor de *MaxCourant*. No caso, este caso foi simulado com número de Courant máximo igual a 1, que significa na prática que onde houver a maior velocidade na malha, a informação não ultrapassará um volume finito, mantendo a igualdade de *Courant* igual a 1. Caso este valor ultrapassar 1, a solução pode divergir. Pela relação da fórmula, quanto mais se refina uma malha, mais se deve diminuir o *deltaT*.

6.3.2.6 Resultados e comentários adicionais

Nas simulações feitas, o número de Courant permaneceu em 1. O ponto negativo disto é que o *deltaT* diminuiu drasticamente, culminando em 75 horas de simulação para alcançar o tempo de 3,2 segundos.

Os perfis de velocidade foram retirados de linhas que passam pelos pontos P_a , P_b e P_c . $P_a = x/D=0$, $P_b = x/D=1$ e $P_c = x/D=2$. D é o diâmetro. O eixo x/D é visto na Figura abaixo:

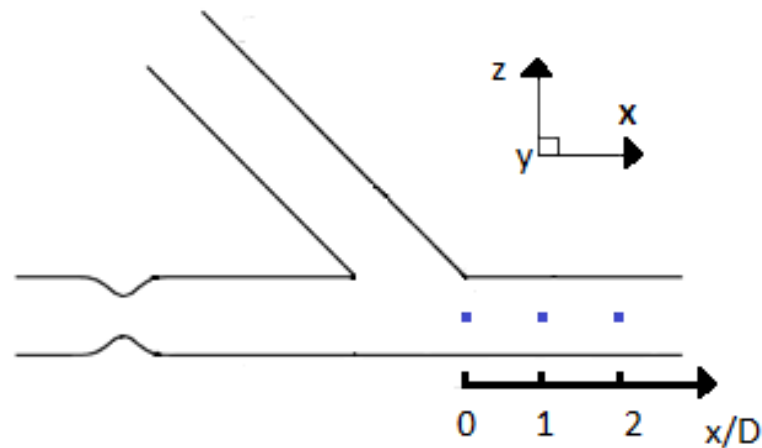


Figura 6-50: Localização dos pontos Pa, Pb e Pc para o CASO III

Os pontos Pa, Pb e Pc estão localizados na linha de raio da artéria, isto é, no centro. Os perfis foram gerados para as linhas que cruzam estes pontos. Foram traçados gráficos para linhas verticais (que passam pelos pontos seguindo a direção do eixo z) e também para linhas horizontais (que passam pelos pontos seguindo a direção do eixo y). Da Figura 6.51 até a Figura 6.54, são traçados os perfis verticais. Da Figura 6.55 até a Figura 6.58, são traçados os perfis horizontais.

Ambos os perfis foram traçados para os instantes 0,24s, 0,48s, 0,64s e 0,88s do segundo ciclo (Bertolotti verificou que os perfis se estabilizavam no segundo ciclo):

Comparação de perfis para o tempo: 0,48 segundos:

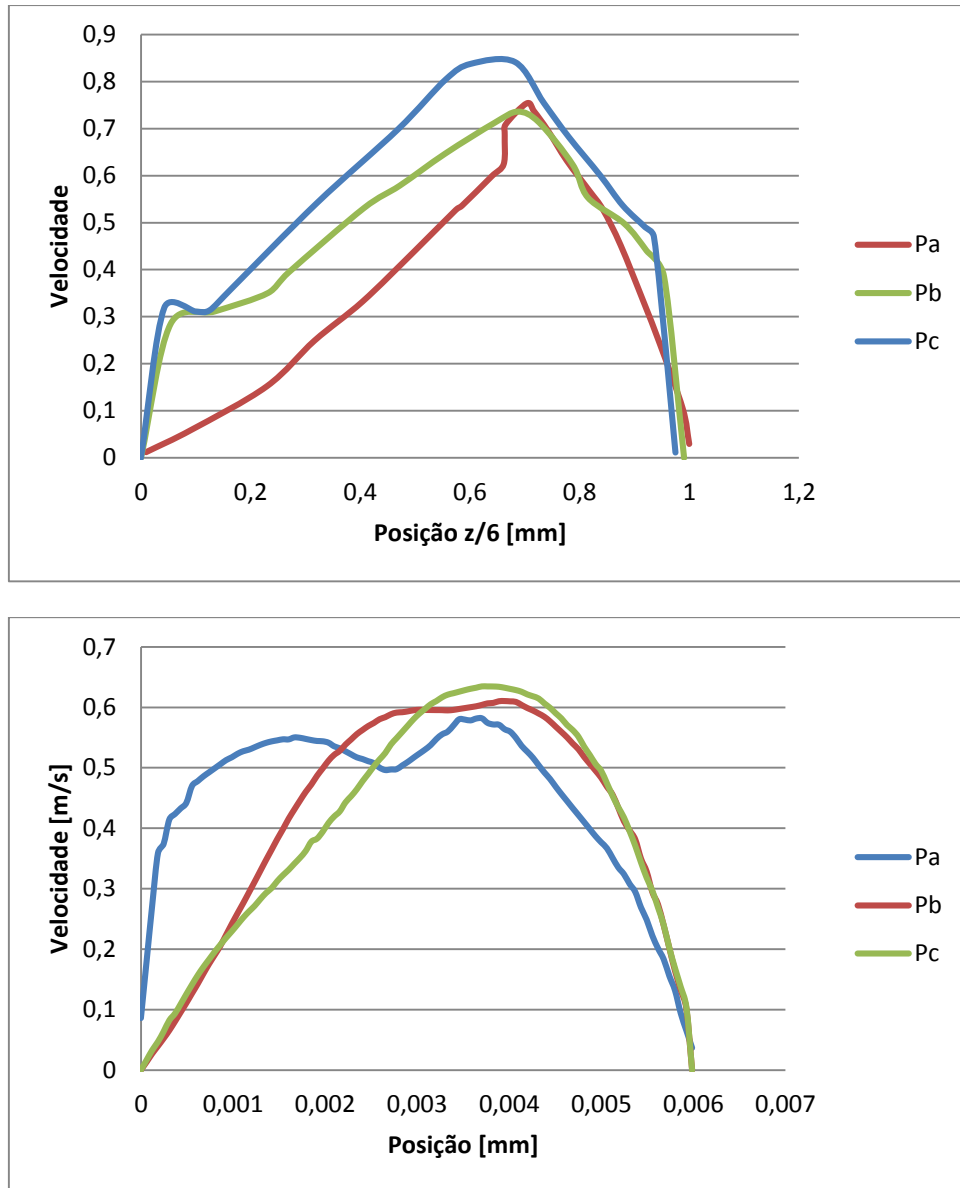


Figura 6-51: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,24 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Comparação de perfis para o tempo: 0,48 segundos

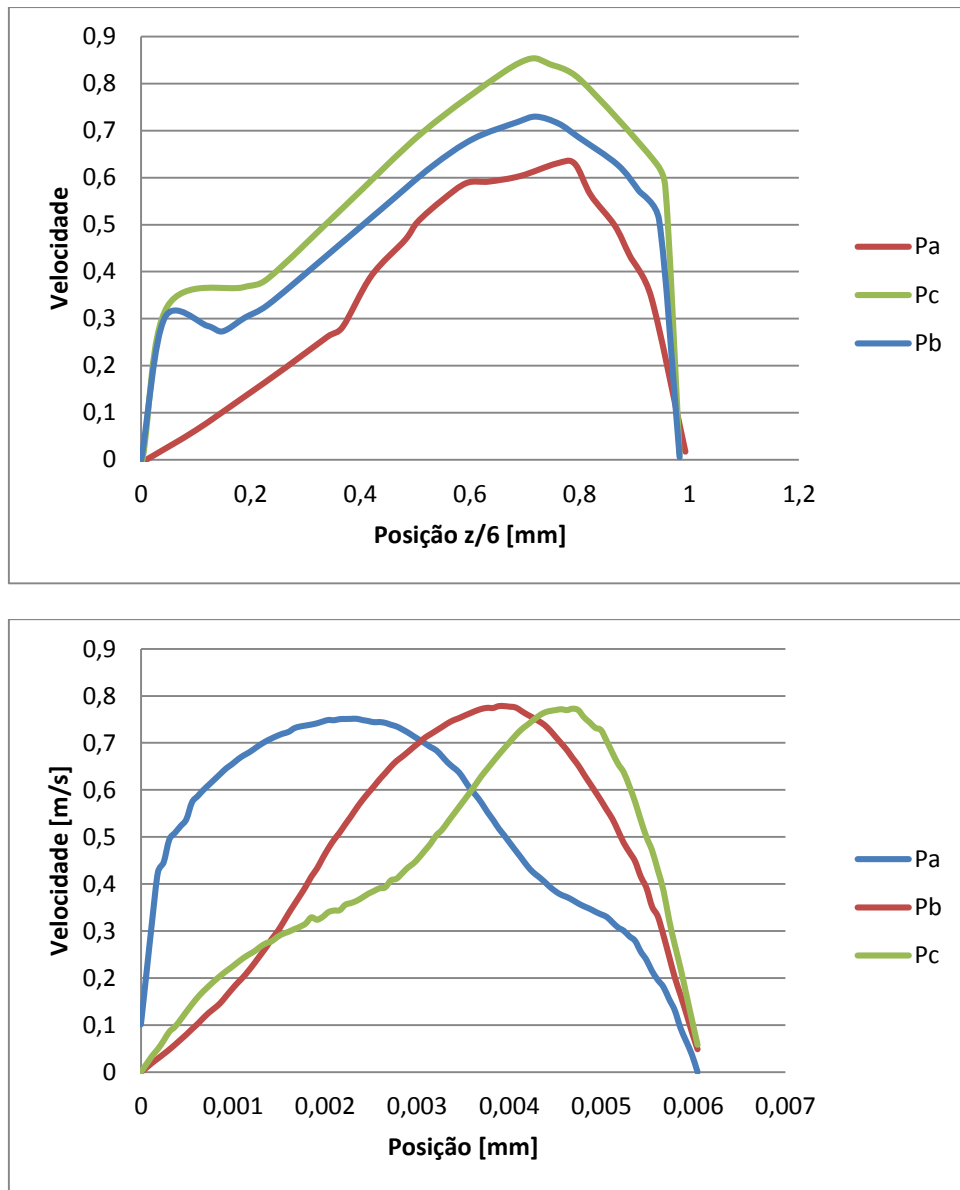


Figura 6-52: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,48 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Comparação de perfis para o tempo: 0,64 segundos

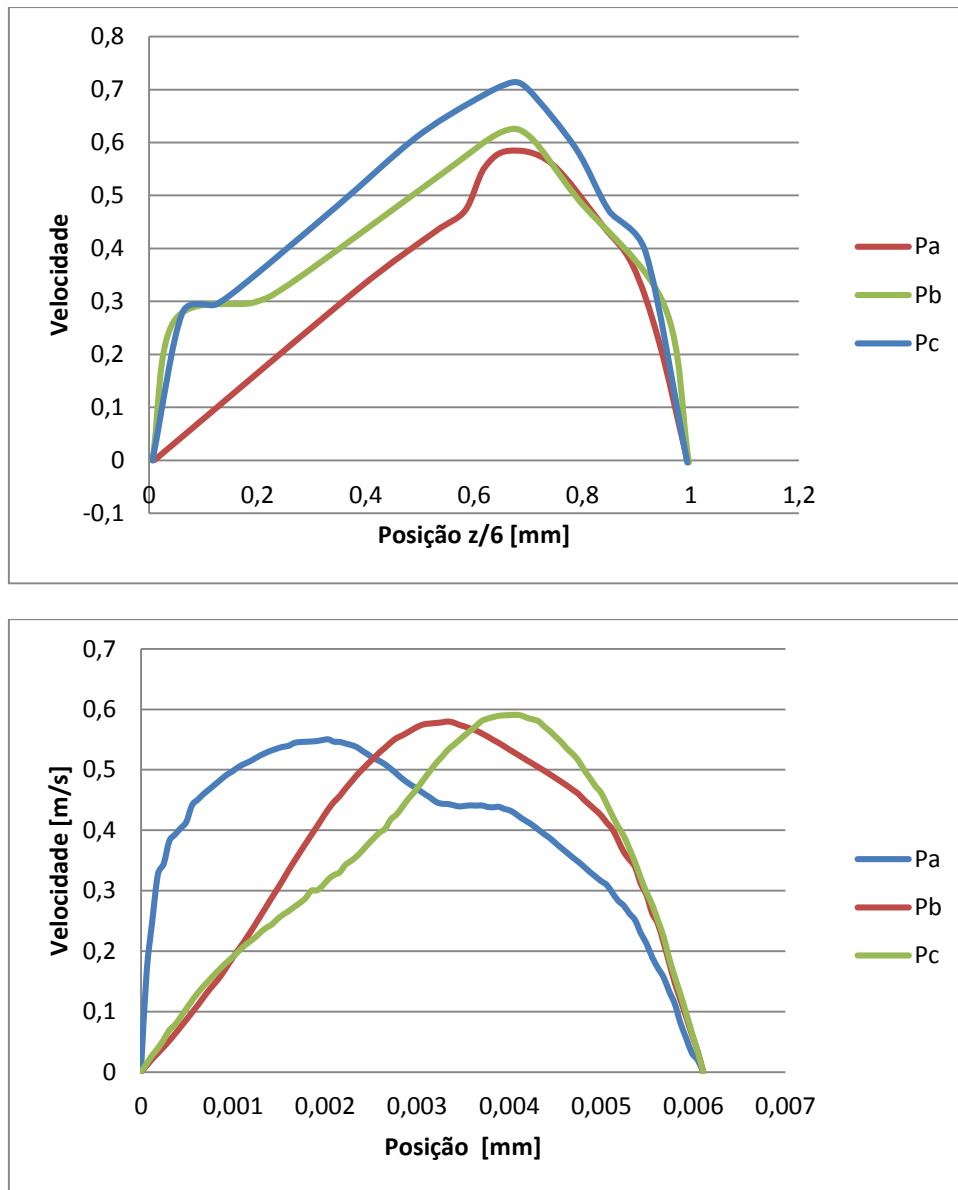


Figura 6-53: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,64 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Comparação de perfis para o tempo: 0,88 segundos

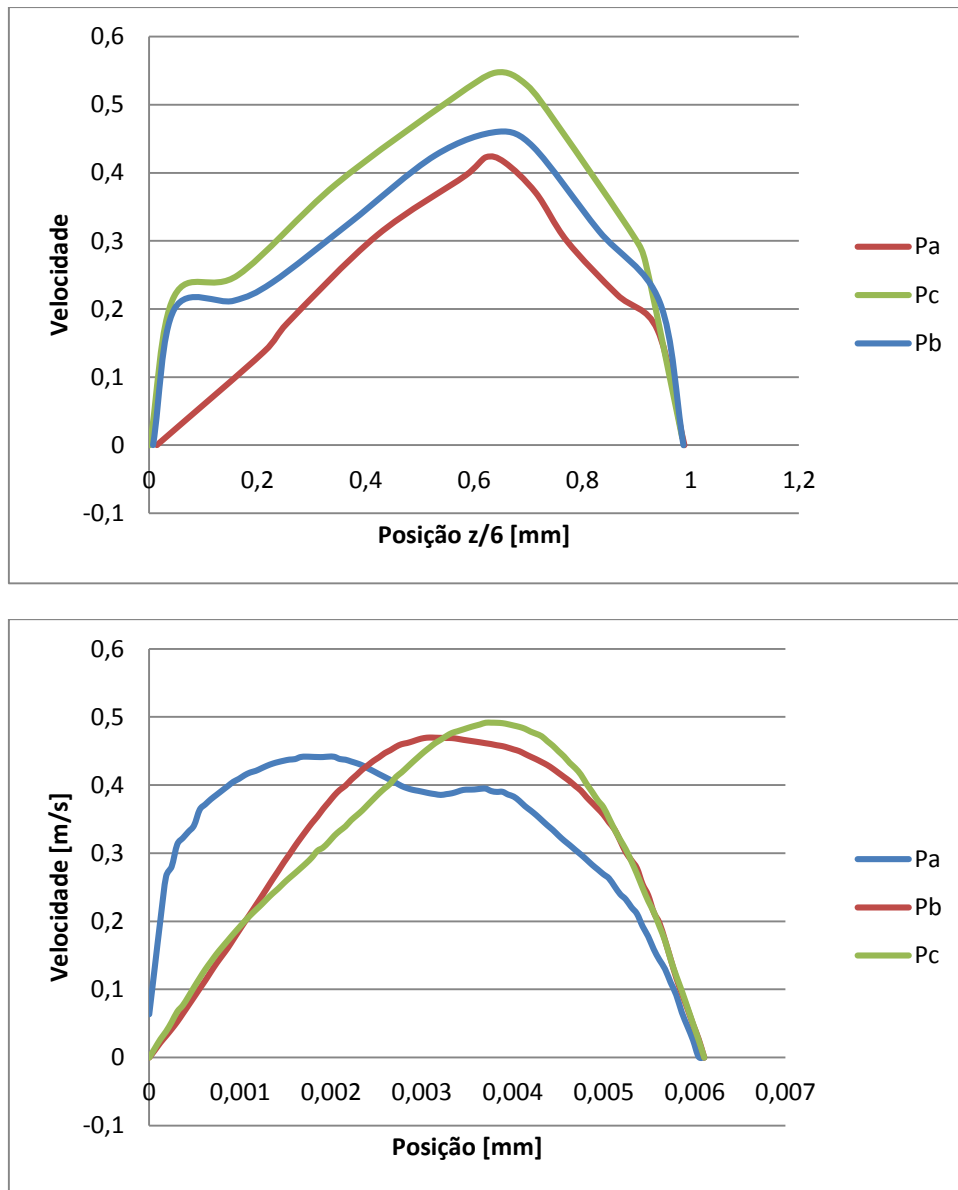


Figura 6-54: Comparação entre perfis de velocidade nas linhas verticais passando por Pa, Pb e Pc para o tempo de 0,88 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Acima foram apresentados os perfis nas linhas que cortam os pontos Pa, Pb e Pc na direção vertical, z. Agora seguem os mesmos perfis, mas na direção horizontal, isto é, direção y:

Comparação de perfis para o tempo de 0,24 segundos:

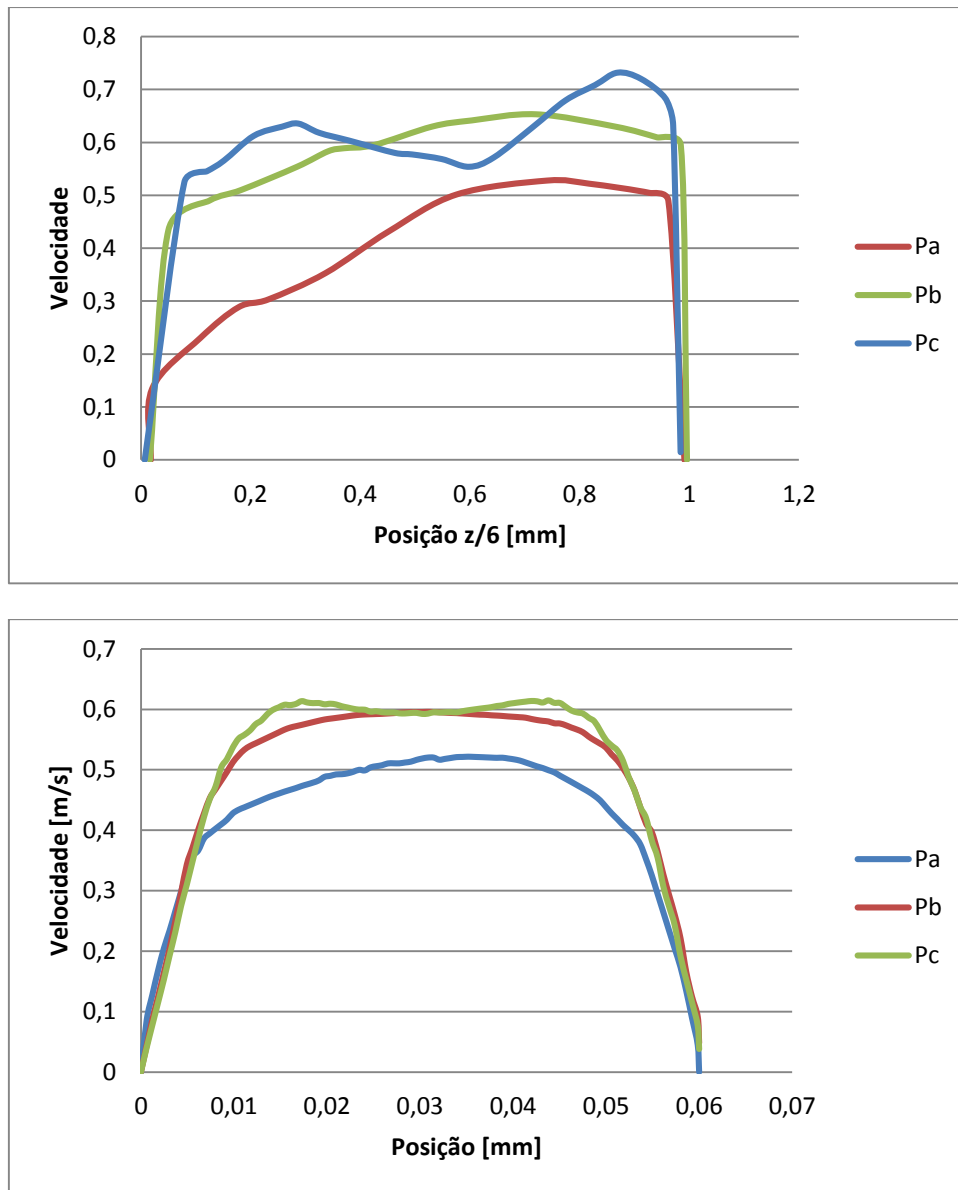


Figura 6-55: Comparação entre perfis de velocidade nas linhas horizontais passando por Pa, Pb e Pc para o tempo de 0,24 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Comparação de perfis para o tempo de 0,48 segundos:

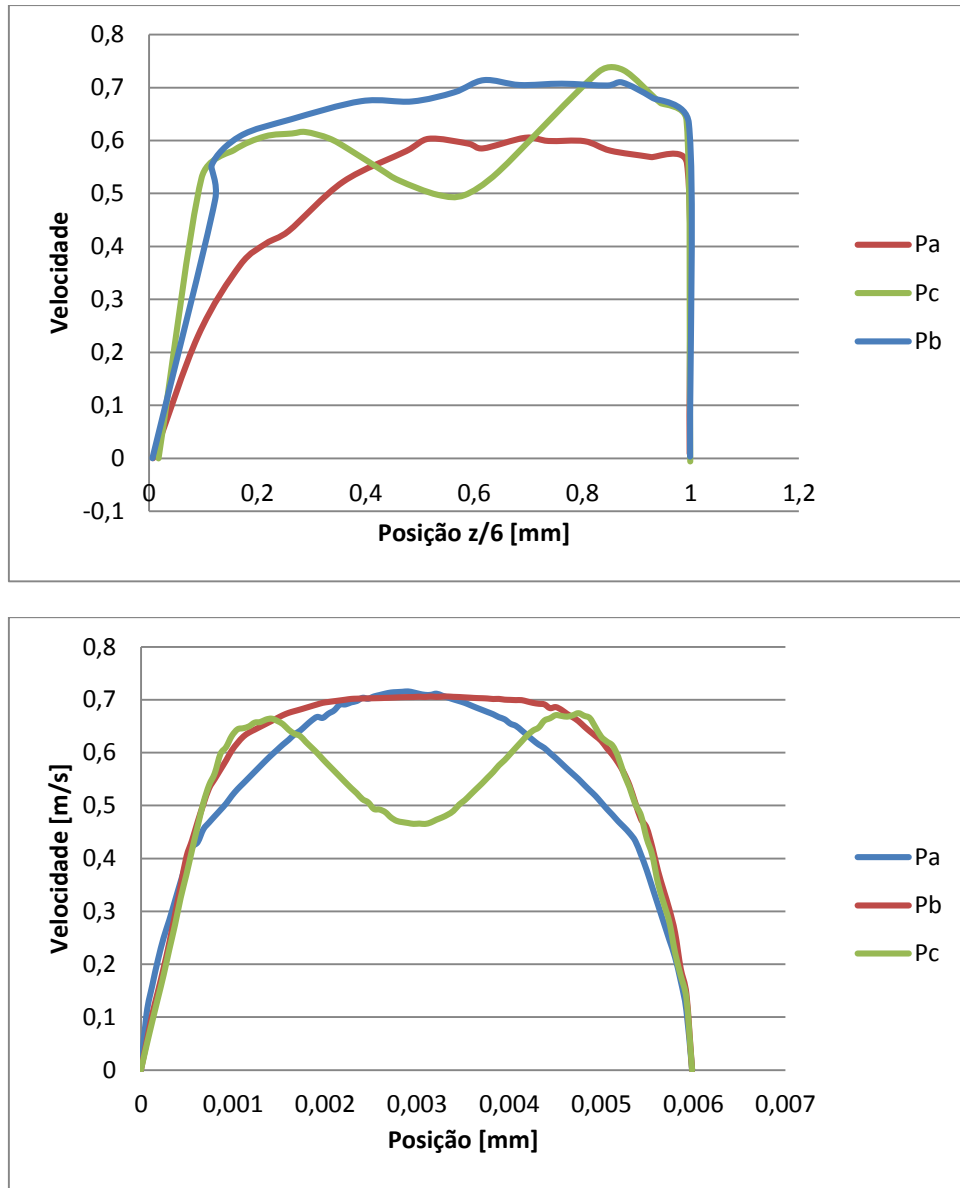


Figura 6-56: Comparação entre perfis de velocidade nas linhas horizontais passando por Pa, Pb e Pc para o tempo de 0,48 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Comparação de perfis para o tempo de 0,64 segundos:

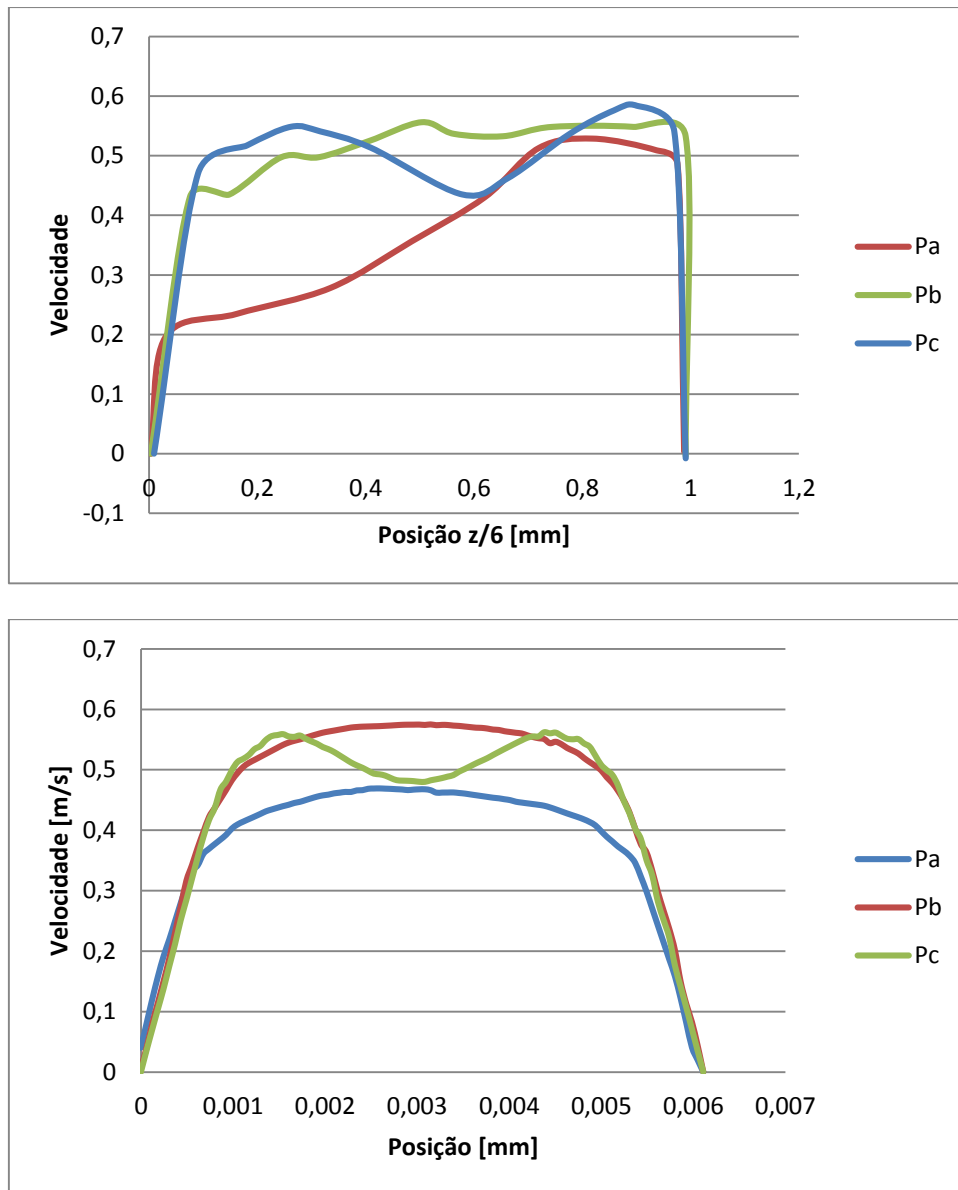


Figura 6-57: Comparação entre perfis de velocidade nas linhas horizontais passando por Pa, Pb e Pc para o tempo de 0,64 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Comparação de perfis para o tempo de 0,88 segundos:

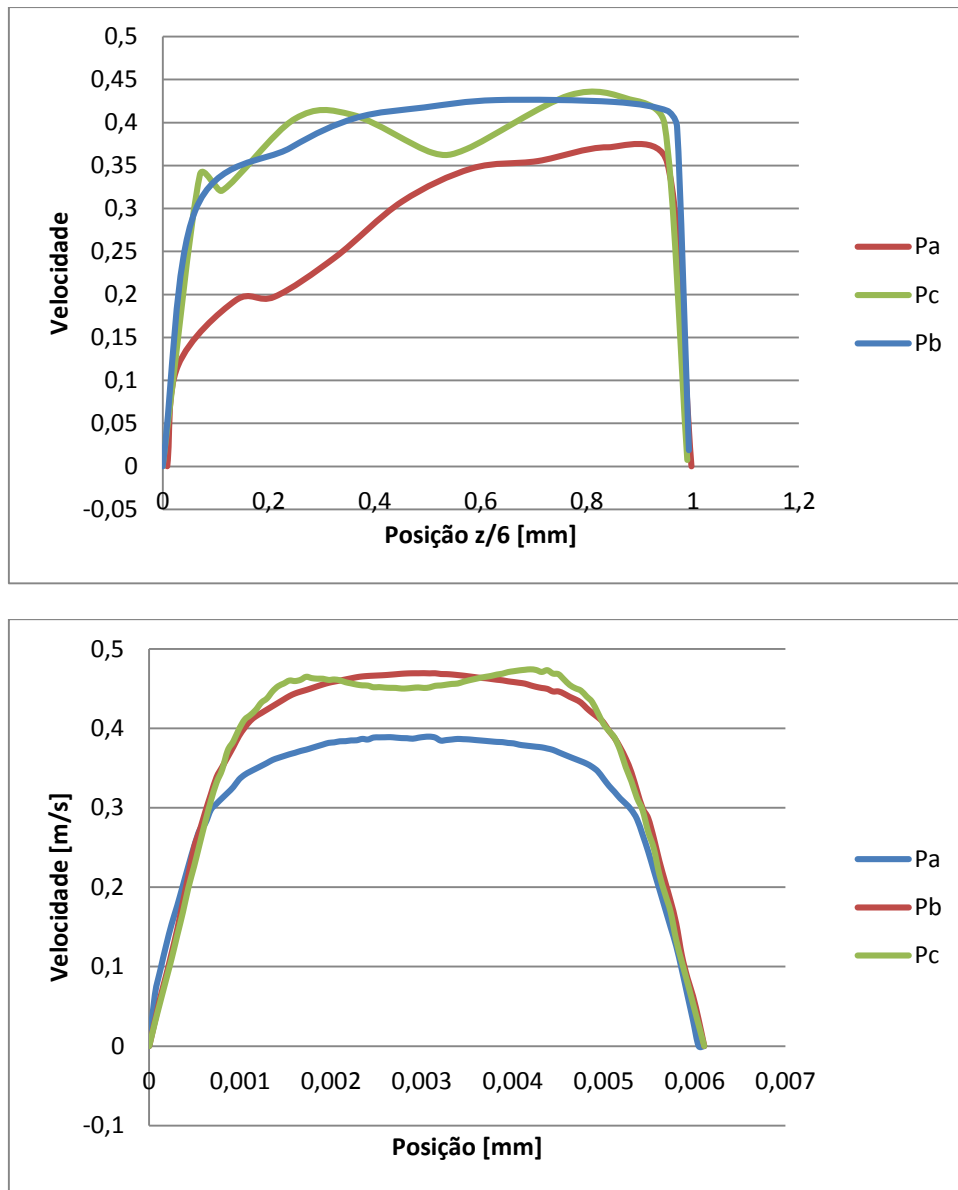


Figura 6-58: Comparação entre perfis de velocidade nas linhas horizontais passando por Pa, Pb e Pc para o tempo de 0,88 segundos. O gráfico superior é o experimental de Bertolotti e Deplano(2000). O gráfico inferior é o obtido pelo OpenFOAM

Embora os gráficos comparativos não tenham sido semelhantes aos experimentais de Bertolotti de Deplano(2000), estes tiveram comportamentos parecidos com o CASO II. Os gráficos experimentais aparentam ter comportamentos anormais. Embora o escoamento seja laminar, não se sabe se este tipo de comportamento é normal no caso em estudo.

7 CONSIDERAÇÕES FINAIS E SUGESTÕES

O OpenFOAM apresentou um bom desempenho diante das exigências. Quanto à sua praticidade e interatividade com o usuário, tem suas desvantagens. O usuário precisa se atentar para muitos detalhes importantes da simulação. Muitas vezes se simulava uma situação em que apenas ao concluí-la se percebia que faltavam detalhes, visto que a estrutura de dados se torna complicada de manejar, pela quantidade de arquivos que devem ser administrados pelo usuário. Esta situação não ocorre no CFX da ANSYS por exemplo, visto que o programa possui interface gráfica que facilita todo o processo de configuração do caso a se simular, além do programa iniciar apenas depois que o usuário entra com todas as variáveis e condições.

Quanto capacidade OpenFOAM, percebeu-se que para situações simples onde não há complexidades extremas, o software se comporta muito bem. Para se obter os mesmos resultados para simulações de casos mais complexos, não basta um conhecimento superficial apenas, como o CFX por exemplo. É necessário ao menos ter um conhecimento genérico a respeito do funcionamento interno do OpenFOAM. Isso vai desde o conhecimento da estrutura do programa a até mesmo conhecer os esquemas numéricos de discretização.

Em vista dos resultados serem muito densos, com muitas tabelas, não houve tempo para mais simulações. Seria ainda necessário se testar mais esquemas numéricos, a fim de se encontrar um conjunto de esquemas com boa acurácia para a simulação de sangue utilizando o modelo de Carreau-Yasuda.

A visão geral que se teve do Software foi que o mesmo tem muitas capacidades que se equiparam aos programas proprietários existentes no mundo. Há uma relação simples entre o OpenFOAM e os softwares proprietários: Enquanto o Openfoam é gratuito, exige mais conhecimento técnico do que os softwares com licença paga. Este conhecimento, se sólido, pode fazer com que se consiga resultados bons a custo zero, como no CFX por exemplo.

Estudou-se por alguns dias a versão de extensão do OpenFOAM, a versão FOAM-Extend 3.1. O intuito foi de descobrir se havia algum *solver* nesta versão que simulasse Interação Fluido-Estrutura (do inglês, *Fluid Structure Interaction - FSI*), isto é, a malha da artéria sofre deformação durante a simulação em função da pressão

total que atua na mesma. Este *solver* foi encontrado. Foram feitas algumas simulações a fim de se compreender e averiguar suas capacidades. Todavia, o *solver* não é tão simples de se configurar para determinadas situações, uma vez que não tem a arquitetura padrão de diretórios do OpenFOAM. O contato com o criador do *solver* foi feito, todavia, o mesmo não respondeu mais às mensagens e por conta disso não foi possível descobrir algumas informações importantes, como a interpretação de alguns códigos adicionais.

Enquanto várias sugestões possam ser feitas, prioriza-se a necessidade de futuros estudos dos esquemas de discretização para bio-escoamentos, a fim de se melhorar ainda mais as soluções. A sugestão secundária é a comparação do *solver* de Fluido-Estrutura com softwares comerciais. Para comparação de resultados, pode-se utilizar os de YAMABE(2013), que simulou uma estenose pelo método Fluido-Estrutura.

8 REFERENCIAS BIBLIOGRÁFICAS

- [1] MOURA, João Pedro Carvalho Rêgo de Serra e .“Analysis and Simulation of Blood Flow in the Portal Vein with Uncertainty Quantification”. Universidade Técnica de Lisboa. 2011.
- [2] BERTOLOTI V. DEPLANO,"Numerical and experimental models of post-operative realistic flows in stenosed coronary bypasses", Jornal of Biomechanics, pag. 1050, 2001.
- [3] CARDIOVASCULAR PHYSIOLOGICS CONCEPTS. Disponível em < <http://www.cvphysiology.com/Hemodynamics/H006.htm> >. Acessado em 8 de dezembro de 2014.
- [4] CFD Online, Forum. Disponível em < <http://www.cfd-online.com/Forums/openfoam/> >. Acessado em 5 de dezembro de 2014.
- [5] CFD Online, Validation and Verification Forum < <http://www.cfd-online.com/Forums/openfoam/> > . Acessado em 5 de dezembro de 2014.
- [6] CHEN, Jie, WANG, Wen. “Non-Newtonian Effects of blood flow on hemodynamics in distal vascular graft anastomoses”, Jornal of Biomechanics, pag. 1984, 2001.
- [7] “Fluid Structure Interaction - OpenFOAM Library for 9th OpenFOAM Workshop” - Zagreb, Croatia, 2011.
- [8] Foam-Extend, OpenFOAM Extend Project. Disponível em: < <http://www.extend-project.de/> > . Acessado em 1 de novembro de 2014.
- [9] FODOR P.S. e SIEBERT ,M.W. , “Newtonian and Non-Newtonian Blood Flow over a Backward-Facing Step: Steady-State Simulation”.Physics Department, Cleveland State University, Cleveland, Ohio, USA, 2009.
- [10] Handbook of Numérica Heat Transfer, 2006, pág. 25.
- [11] SALOME, software. Disponível em < <http://www.salome-platform.org/>>. Acessado em 5 de dezembro de 2014.
- [12] SALOME, User Guide. Disponível em < <http://www.salome-platform.org/user-section/documentation/current-release> >.Acessado em 5 de dezembro de 2014.
- [13] YAMABE, Paulo Vinicius Miyuki e Diana, Rodrigo Eiji Yamagata.“SIMULAÇÃO COMPUTACIONAL 3D DE SISTEMAS ARTERIAIS” Universidade de São Paulo,2013.

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

VITOR SPADETO VENTURIN

**SIMULAÇÃO COMPUTACIONAL DE BIO-ESCOAMENTOS
UTILIZANDO O OPENFOAM**

VITÓRIA
2014