

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
*CAMPUS FRANCISCO BELTRÃO*  
CURSO DE ESPECIALIZAÇÃO EM MÉTODOS  
MATEMÁTICOS APLICADOS

Leandro Luis Michelson

**Verificação de Solvers do OpenFOAM para o  
Escoamento Supersônico Sobre Cones**

Francisco Beltrão, Paraná

2019

Leandro Luis Michelson

## **Verificação de Solvers do OpenFOAM para o Escoamento Supersônico Sobre Cones**

Trabalho de Conclusão apresentado ao  
Curso de Especialização em Métodos Ma-  
temáticos Aplicados como requisito parcial  
para a obtenção do título de Especialista.

Universidade Tecnológica Federal do Paraná

Orientador: Jonas Joacir Radtke

Coorientador: Guilherme Bertoldo

Francisco Beltrão, Paraná

2019



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Campus Francisco Beltrão  
Diretoria de Pesquisa e Pós-Graduação  
Especialização em Métodos Matemáticos Aplicados



## **TERMO DE APROVAÇÃO**

### **Trabalho de Conclusão de Curso de Especialização**

### **VERIFICAÇÃO DE SOLVERS DO OPENFOAM PARA O ESCOAMENTO SUPERSÔNICO SOBRE CONES**

por

**LEANDRO LUIS MICHELSON**

Trabalho de Conclusão de Curso de Especialização apresentado às 13 horas e 30 min. do dia 26 de outubro de 2019, como requisito parcial para obtenção do grau de especialista em Métodos Matemáticos Aplicados, da Universidade Tecnológica Federal do Paraná, *Campus* Francisco Beltrão. O candidato foi arguido pela Banca Avaliadora composta pelos professores que abaixo assinam este Termo. Após deliberação, a Banca Avaliadora considerou o trabalho Aprovado.

---

**JONAS JOACIR RADTKE**

Professor Orientador

---

**GUILHERME BERTOLDO**

Professor Coorientador

---

**VILMAR STEFFEN**

Membro da Banca

---

**Prof. Vilmar Steffen**

Responsável pela Coordenação do CEMMA  
Curso de Especialização em Métodos Matemáticos Aplicados

**A FOLHA DE APROVAÇÃO ORIGINAL (ASSINADA) ENCONTRA-SE NA COORDENAÇÃO DO  
CURSO DE ESPECIALIZAÇÃO EM MÉTODOS MATEMÁTICOS APLICADOS.**

Dedico este trabalho a minha família  
e as pessoas que me transformam no que sou.

## **AGRADECIMENTOS**

Agradeço a UTFPR e aos professores pela oferta do curso e em especial aos orientadores Jonas Joacir Radtke e Guilherme Bertoldo.

*"A ciência nunca resolve um problema sem criar pelo menos outros dez".(George Bernard Shaw).*

## RESUMO

Este trabalho apresenta a verificação do software OpenFOAM aplicado ao escoamento de fluido com velocidade supersônica sobre cones. Utilizando três solvers do OpenFOAM, são feitas simulações para obter o valor do coeficiente de arrasto frontal. Para verificar as soluções encontradas pelos *solvers* usa-se a solução de Taylor-Maccoll e os estimadores de erro GCI e Richardson. Os resultados que os três *solvers* encontram não diferem do resultado de Taylor-Maccoll em mais de 0,069% e tem a mesma precisão para resultado dos estimadores. Com os resultados encontrados é possível dizer que, para simulações similares e com grau de precisão igual ou inferior aos deste trabalho, o *solver* mais indicado é o rhoCentralFoam, mas que também leva mais tempo.

**Palavras-chave:** OpenFOAM. Estimadores. Cone. Escoamento.

## **ABSTRACT**

This work presents the verification of using OpenFOAM software applied to supersonic velocity fluid flow over cones. Using three OpenFOAM solvers simulations are made to find the value of the frontal drag coefficient. To verify the solutions found by the solvers used the Taylor-Maccoll solution and the error estimators GCI and Richardson. The results that the three solvers find do not differ from the Taylor-Maccoll result by more than 0,069% and have the same accuracy for the estimators result. With the results found it is possible to say that for similar simulations and with accuracy equal to or lower than the ones in this work the rhoCentralFoam is the most suitable solver, but it also took longer.

**Keywords:** OpenFOAM. Estimators. Cone. Flow.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Cone onde ocorrem as simulações . . . . .	15
Figura 2 – Vista do cone e da região de escoamento em 2D . . . . .	21
Figura 3 – Estrutura representando os volumes por onde escoo o fluido . . . . .	27
Figura 4 – Coeficiente de arrasto frontal . . . . .	34
Figura 5 – Tempo de simulação x Refino de malha . . . . .	37

## LISTA DE TABELAS

Tabela 1 – Propriedades do fluido . . . . .	26
Tabela 2 – Coeficiente de arrasto frontal . . . . .	33
Tabela 3 – Diferença entre Taylor-Maccoll e OpenFOAM . . . . .	35
Tabela 4 – Ordem aparente $pu$ . . . . .	35
Tabela 5 – Ordem efetiva $pe$ . . . . .	35
Tabela 6 – Incerteza GCI usando $pl$ . . . . .	36
Tabela 7 – Incerteza de Richardson usando $pl$ . . . . .	36
Tabela 8 – Tempo de simulação . . . . .	37

## **LISTA DE ABREVIATURAS E SIGLAS**

CFD            Computed Fluid Dynamics

OpenFOAM    Open Source Field Operation and Manipulation

## LISTA DE SÍMBOLOS

$A$	Área
$C$	Coeficiente independente
$Cd$	Coeficiente de arrasto frontal
$D$	Arrasto
$E$	Erro
$E_i$	Energia total
$e$	Energia
$F_s$	Fator de segurança
$h$	Comprimento característico de volume
$Ma$	Número de Mach
$n$	Número de mols
$p$	Pressão
$pe$	Ordem efetiva
$pl$	Ordem assintótica
$pV$	Ordem verdadeira
$pu$	Ordem aparente
$q$	Razão de refino
$T$	Temperatura
$V$	Volume
$U$	Incerteza
$U_{GCI}$	Estimativa de erro GCI
$U_{Ri}$	Estimativa de erro de Richardson
$u$	Velocidade do fluido
$\delta t$	Intervalo de tempo

$\varepsilon_\tau$	Erro de truncamento
$\Theta_{GCI}$	Efetividade do estimador GCI
$\theta$	Efetividade
$\Lambda$	Variável dependente
$\mu$	Viscosidade dinâmica do fluido
$\rho$	Massa específica do fluido
$\Phi$	Solução analítica
$\phi$	Solução numérica
$\phi_1$	Solução na malha fina
$\phi_\infty$	Solução analítica estimada
$\psi_u$	Razão de convergência

## SUMÁRIO

1	INTRODUÇÃO . . . . .	14
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	17
3	METODOLOGIA . . . . .	20
3.1	MODELAGEM MATEMÁTICA . . . . .	20
3.2	CONFIGURAÇÃO DOS <i>SOLVERS</i> . . . . .	25
3.3	VERIFICAÇÃO . . . . .	28
4	RESULTADOS . . . . .	33
5	CONCLUSÕES . . . . .	38
5.1	CONSIDERAÇÕES FINAIS . . . . .	39
6	TRABALHOS FUTUROS . . . . .	40
	REFERÊNCIAS . . . . .	41
	APÊNDICES . . . . .	43
	APÊNDICE A – ARQUIVO UTILIZADO PARA DEFINIR A GEOMETRIA E AS CONDIÇÕES DE CONTORNO . . . . .	44
	APÊNDICE B – EXEMPLO DE CONFIGURAÇÃO DO ARQUIVO FVS-CHEMES . . . . .	46

## 1 INTRODUÇÃO

O aperfeiçoamento das técnicas de aproximação numérica e a melhor compreensão dos fenômenos físicos (o que facilita a modelagem dos problemas) propiciou o melhoramento dos softwares na área de engenharia. Isso se refletiu na maior eficiência de processamento de modelos matemáticos, que se tornaram mais complexos e demorados mas que conseguem resultados de grande precisão. Além disso é possível determinar erros que podem ocorrer com antecedência, o que reduz custos e acelera o processo (TANNEHILL; ANDERSON; PLETCHER, 1997).

Este trabalho utiliza dados gerados pelo software OpenFOAM aplicado ao escoamento de fluidos sobre cones, onde por meio de comparações foi realizada a verificação e avaliada a eficiência de alguns *solvers*. Neste caso *solvers* são programas solucionadores de problemas de fluidodinâmica.

Modelos matemáticos semelhantes aos usados neste trabalho são utilizados para resolverem problemas como a simulação de escoamento de fluidos sobre superfícies, testes de aerodinâmica de modo geral, em que podem ser verificadas propriedades como temperatura, pressão, coeficiente de arrasto, velocidade do fluido em diferentes regiões do objeto em estudo, etc.

O OpenFOAM de acordo com Greenshields (2018) é uma estrutura em C++ onde são criados aplicativos *solvers* para resolver problemas de mecânica dos fluidos e utilitários que são projetados para executar tarefas de manipulação de dados. O OpenFOAM é totalmente gratuito e conta com um grande suporte online por parte de seus usuários e desenvolvedores, sendo estas as principais razões para sua escolha. Além disto, os *solvers*, os utilitários e as bibliotecas do OpenFOAM podem ser estendidos e modificados de acordo com a necessidade do usuário, desde que este tenha o conhecimento necessário sobre a física e tecnologia de programação nele envolvidos. O OpenFoam conta com aproximadamente 250 aplicativos que acompanham a sua estrutura. Destes *solvers*, três foram utilizados neste trabalho: *sonicFoam*, *rhoSimpleFoam* e *rhoCentralFoam*.

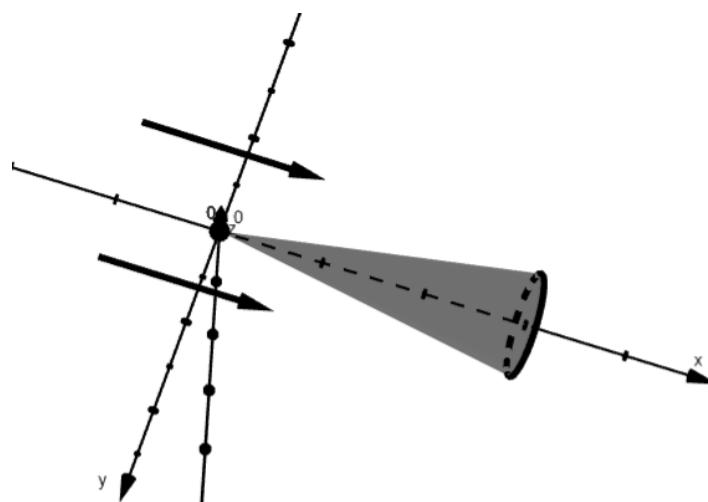
Foi feita a verificação dos dados gerados pelos *solvers*, "O objetivo da verificação é determinar em que medida um modelo matemático é resolvido adequadamente através de um método numérico."(MARCHI, 2001, p. 6) e "O objetivo da validação é

determinar em que medida um modelo matemático representa um determinado fenômeno real."(MARCHI, 2001, p. 6). Neste tipo de trabalho podem ser feitos dois tipos de verificação, a verificação de código e a verificação de solução pois no código podem ocorrer erros de programação, de iteração, e de arredondamento, já para a verificação de solução são utilizados os estimadores e calculada a estimativa de erro da solução encontrada. Como este trabalho aborda os resultados encontrados pelos *solvers* foi realizada somente a verificação de solução. Neste trabalho não nos interessa validar o modelo matemático, como estamos comparando os resultados com soluções encontradas em outro trabalho que já fez isso, faremos a parte que se refere ao teste dos *solvers*.

As simulações feitas com estes *solvers* ocorrem sobre a superfície de um cone (Figura 1) com razão de aspecto 3 (comprimento sobre diâmetro da base). O diâmetro de base do cone é de uma polegada, o escoamento foi Mach 3,5 (quantidade de vezes a velocidade do som) e foi utilizado o modelo de Euler para resolver o problema.

Também foram cronometrados os tempos que cada *solver* levou para chegar as condições impostas para resolver o problema de aproximação de ( $Cd$ ). Tais resultados possibilitaram avaliar qual o *solver* mais eficiente em resolver o problema abordado.

**Figura 1** – Cone onde ocorrem as simulações



Fonte – Autoria própria (2019)

Para esse trabalho ser possível é necessário que se tenha a solução analítica ou alguma solução numérica confiável em mãos para que ser possível comparar os resultados. A escolha do cálculo de  $Cd$  para o escoamento sobre um cone com as características descritas acima se refere justamente a isso, pois no processo de simulação computacional podem ocorrer alguns erros. Os erros que podem ocorrer são



o de modelagem que é a diferença entre a solução analítica e o valor verdadeiro de uma variável de interesse e o erro numérico, que é a diferença entre a solução analítica exata e a solução numérica (MARCHI, 2001 apud FERZIGER; PERIĆ, 1999). O erro numérico ocorre principalmente por erro de discretização, erro de iteração, erro de arredondamento, etc. Como temos a solução de Taylor-Maccoll, que é uma solução numérica mas com grande exatidão, tanto que pode ser considerada uma solução analítica, pode-se comparar os resultados e verificar se os *solvers* utilizados levam a resultados satisfatórios, ou se possuem algum erro.

Dependendo da precisão desejada, nem sempre existe a necessidade de se utilizar modelos que conseguem descrever com a maior precisão possível problemas reais. Por exemplo, sabemos que em problemas reais todos os fluidos são viscosos, mas em determinadas regiões onde temos o número de Reynolds ( $Re$ ) elevado os efeitos viscosos podem ser desconsiderados. As regiões próximas a parede não podem ter a viscosidade desconsiderada, mas fora desta camada a viscosidade pode ser desconsiderada e portanto não é necessário mais resolver as equações de Navier-Stokes e pode-se utilizar a Equação de Euler, ou seja, equação de quantidade de movimento sem atrito.

A escolha dos *solvers* é realizada de acordo com as características do problema, por exemplo, pode-se definir o nível de turbulência do escoamento, ou seja, se é laminar ou turbulento. Neste trabalho o problema é tratado como sendo laminar e com fluxo transiente, onde as propriedades variam com o tempo. No entanto, existe interesse no estado estacionário, ou seja, na situação em que as propriedades não variam mais com o tempo.

Com os *solvers* e entradas de dados bem definidas foram efetuadas simulações computacionais seguindo recomendações de Marchi (2001), onde foi feito o refino de malhas garantindo comparações de resultados mais confiáveis. A utilização de mais de uma malha é necessária quando se faz o uso dos estimadores. No presente trabalho foram usados dois estimadores, Richardson e GCI, onde foi feita a verificação da solução numérica, para assim termos condições de verificar os resultados fornecidos pelos três *solvers*.

## 2 FUNDAMENTAÇÃO TEÓRICA

Historicamente o desenvolvimento de tecnologias relacionadas a dinâmica de fluidos mostrou-se importante na trajetória do desenvolvimento da humanidade, no livro *Fundamentals of Aerodynamic* é destacado um exemplo:

O voo supersônico de alta velocidade havia se tornado uma característica dominante da aerodinâmica até o final da Segunda Guerra Mundial, nessa época os aerodinâmicos apreciavam as vantagens de usar formas esbeltas e pontiagudas do corpo para reduzir o atrito de veículos supersônicos. Quanto mais aguçado e esbelto o corpo, mais fraca é a onda de choque presa ao nariz e, portanto, menor o arrasto da onda (JUNIOR, 2001, p. 6).

Como a tecnologia de construção de aeronaves e a produção de materiais melhores para construí-las foi aumentando, também foi necessário que se criassem melhores formas de testá-los, sempre buscando o menor custo possível. Segundo Versteeg e Malalasekera (2007) o desenvolvimento de computadores mais potentes foi o responsável por possibilitar o desenvolvimento de ferramentas capazes de fazer simulações da dinâmica dos fluidos (CFD - Computer Fluid Dynamics). Os métodos computacionais tinham a desvantagem em relação a ferramentas como o CAE que fazem testes de estresse, de serem mais complexos, o que exigia um grande processamento. Como essa desvantagem foi resolvida, tais métodos passaram a ser mais usados, tendo seu grande desenvolvimento na década de 90.

Um dos *softwares* que vem sendo bastante usado é o OpenFOAM (*Open Source Field Operation and Manipulation*), que é gratuito e possui bastante material sobre sua utilização. No OpenFOAM tem-se a vantagem de poder editar qualquer parte da simulação que será feita, tanto na criação da malha, que já vem integrada no OpenFOAM não necessitando de *software* adicional, tanto quanto na configuração de entradas de dados e configurações dos pacotes de CFD que estão inseridos no OpenFOAM. Nos *softwares* pagos isto é um ponto limitante pois o usuário não tem acesso a todas as configurações do programa ou o *software* apresenta dados incompletos sobre a modelagem e implementação numérica. Segundo Greenshields (2018), os usuários podem estender a coleção de pacotes CFD, utilitários e bibliotecas do OpenFOAM, tendo somente o pré-requisito de terem um conhecimento sobre a matemática do problema, física e tecnologia de programação envolvidos.

Outros trabalhos já fizeram o uso do OpenFOAM para simular escoamentos. O trabalho de Correia (2016) faz a simulação de um leito fluidizado e avalia o comportamento hidrodinâmico de partículas de cera de carnaúba que se encontram neste fluido.

O OpenFOAM também foi utilizado para fazer a simulação de um vazamento de gás com formação de numvem inflamável no trabalho de Fiates (2015), onde se utiliza o *solver ReactingFoam* no problema de dispersão de gás.

Um dos métodos mais usados pelo OpenFOAM é o método de volumes finitos. A abordagem pelo método de volumes finitos feito pelos *solvers* pode ser considerada um grande avanço nos métodos de resolução de equações diferenciais, no livro *Computational Fluid Mechanics And Heat Transfer* é destacado o seguinte:

Olhando para trás sobre a metodologia dos métodos de volumes finitos e séries de Taylor, podemos observar que o método da série de Taylor forneceu prontamente aproximações de diferença para as derivadas e a representação para a EDP (Equação Diferencial Parcial) completa foi composta a partir da adição de várias dessas representações. Por outro lado, o método de volumes finitos emprega a declaração de conservação ou lei física (geralmente invocada de forma integral) correspondente a todo EDP. A característica distintiva da abordagem de volumes finitos é que um "equilíbrio" de alguma quantidade física é feita na região (volume de controle) na vizinhança de um ponto da malha (TANNEHILL; ANDERSON; PLETCHER, 1997, 75)

Utilizando o método de volumes finitos descrito anteriormente são resolvidas equações de dinâmica de fluidos como as de Navier-Stokes, Euler, entre outras derivações. Utilizando descrições feitas por Junior (2001) e Tannehill, Anderson e Pletcher (1997), percebe-se que existem diferentes representações e formas de abordar o mesmo problema, além de existirem muitas diferenças nos problemas que acabam gerando variações nos modelos. Para este trabalho buscou-se uma explicação abrangente mas que ao mesmo tempo não se torna-se complexa demais, sendo esta uma das principais razões pela escolha dos materiais citados acima.

Para poder comparar resultados que o OpenFOAM gera com outros resultados, se faz necessária a utilização de estimadores, que são ferramentas matemáticas utilizadas para estimar a magnitude do erro gerado pelos programas de simulação computacional. Como descrito por Marchi (2001), existem dois grandes grupos de métodos que podem ser empregados para realizar estimativas de erro. O primeiro é realizado sobre uma única malha, ou seja, são realizadas simulações sobre um objeto em que não se modifica sua estrutura, e o segundo onde a estrutura é dividida em partes cada vez menores, o que aumenta a precisão nas simulações.

Para fazer as comparações entre os *solvers* é necessário que haja uma solução como referência, pois apesar de ser possível calcular estimativas de erro usando os estimadores sem ter uma solução analítica, não seria a melhor forma de avaliá-los. Para isso, foi usada a solução de Taylor-Maccoll, apresentada no trabalho Taylor e Maccoll (1933).

Com o objetivo de comparar os *solvers*, faz-se a verificação da solução. Segundo (ROACHE, 1998 apud BLOTTNER, 1990) a verificação e a validação podem ser resumidas como: verificação é resolver as equações corretamente e validação é resolver as equações corretas. Neste trabalho não será realizada a validação, pois o objetivo é comparar os *solvers* e não os métodos usados para resolvê-lo.

Segundo Babuska e Oden (2004), se o processo de verificação e validação são separados é necessário tratar erros de modelagem e erros de discretização de forma independente. Qualquer exercício de validação baseado em modelo computacional em que o erro de discretização não é quantificado é inútil. Por tanto o processo de verificação pode ocorrer sem que o processo de validação ocorra.

Para calcularmos a estimativa do erro, que também é chamada de incerteza por alguns autores, utilizam-se os estimadores. Neste trabalho são apresentados os estimadores de Richardson e GCI (MARCHI, 2001 apud ROACHE, 1998). Segundo Marchi (2001) ambos apresentam resultados ruins em malhas grossas, o que é um problema para simulações que não dispõem de grande capacidade de processamento para fazer um refino de malha que garanta malhas mais finas.

### 3 METODOLOGIA

Neste trabalho de verificação de *solvers* do OpenFOAM são realizadas simulações computacionais, onde os dados referentes ao coeficiente de arrasto frontal são tabelados e comparados entre si. Após a realização das simulações são feitos testes de determinação de erro numérico utilizando os estimadores de Richardson e GCI.

Para que as comparações entre os *solvers*, rhoSimpleFoam, rhoCentralFoam e sonicFoam sejam possíveis, é preferível que elas resolvam o mesmo problema e que este tenha uma solução analítica conhecida, no caso estaremos usando um resultado confiável. Para isto foi verificado o coeficiente de arrasto frontal ( $C_d$ ) de um escoamento supersônico de fluido compressível de regime permanente sobre um cone. As entradas de dados dos três *solvers* usados neste trabalho são as mesmas que foram utilizadas em outros trabalhos para se chegar ao resultado de referência.

Nos tópicos a seguir estes dados são apresentados juntamente com os métodos utilizados para resolver o problema, o domínio em que ocorrem as simulações, configurações do *solver* e utilização dos estimadores.

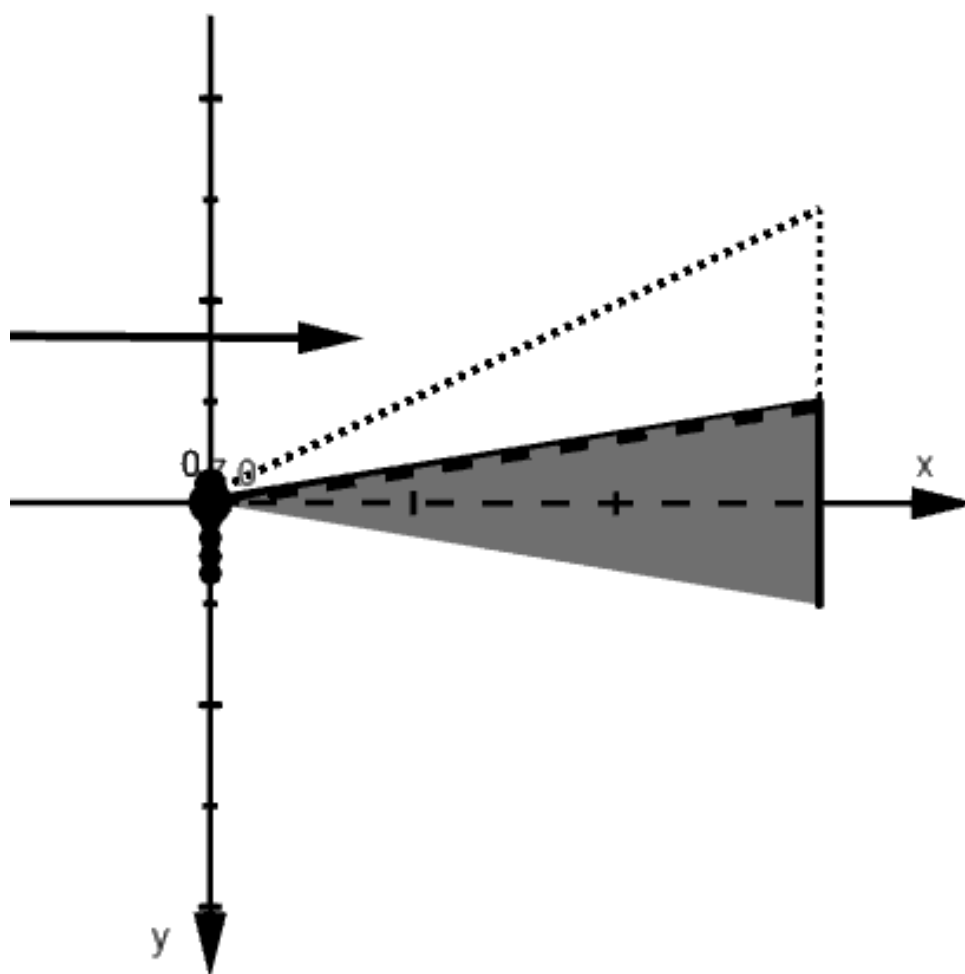
#### 3.1 MODELAGEM MATEMÁTICA

A superfície considerada para o cálculo de  $C_d$  está representada na Figura 2. A seta aponta a direção em que ocorre o escoamento, a região cinza representa o cone e a área pontilhada representa a região em que é simulado o escoamento. Sobre a superfície do cone é simulado o escoamento de um fluido compressível.

O fluxo compressível é rotineiramente definido como densidade variável, isso está em contraste com o fluxo incompressível, onde se supõe que a densidade seja constante o tempo todo. Obviamente na vida real, todo fluxo de todo fluido é compressível a algo maior ou menor extensão; portanto, um fluxo verdadeiramente constante de densidade (incompressível) é um mito. (JUNIOR, 2001, p 12)

O fluido é uma mistura tratada pelo *solver* como um gas ideal de massa molar igual a  $28,969\text{g/mol}$ , de calor específico ( $C_p$ ) igual a  $1004\text{J/Kg.K}$  que é a capacidade térmica por unidade de massa da substância, de viscosidade ( $\mu$ ) igual a 0 e o número de Prandtl ( $Pr$ ) igual a  $0,72449$ .

Neste trabalho foi usado o modelo de Euler para encontrar o valor de  $C_d$ , já

**Figura 2** – Vista do cone e da região de escoamento em 2D

Fonte – Autoria própria (2019)

que o resultado que dispomos para fazer as comparações também é baseado neste modelo. O que diferencia o modelo de Euler do modelo de Navier-Stokes é que no de Euler não é considerada a viscosidade do fluido, por tanto foi simulado sobre uma superfície sem atrito, simplificando bastante os cálculos e por consequência reduzindo o tempo de processamento.

A vantagem de se usar o modelo de Euler para resolver o problema é a simplificação que este traz aos cálculos, possibilitando uma maior velocidade ao *solver* para encontrar o resultado que se busca. Por outro lado, o modelo de Euler não é o mais próximo da realidade, visto que problemas reais sempre tem viscosidade, e como

apresentado em Bertoldo e Marchi (2017) podem ocorrer diferenças nos resultados de Euler e de Navier-Stokes. Como este é um trabalho de comparação entre os *solvers* e possuímos o resultado confiável podemos usar o modelo de Euler.

Segundo Tannehill, Anderson e Pletcher (1997) o escoamento do fluido sobre o cone pode ser simulado pelas equações de conservação de massa, de momento e de energia.

A equação de conservação de massa é dado por:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0, \quad (3.1)$$

onde  $\rho$  é a densidade do fluido e  $u$  é a velocidade do fluido. O primeiro termo representa a taxa de aumento da densidade no volume de controle e o segundo termo representa a taxa de fluido de massa que sai da superfície de controle por unidade de volume.

Para a equação de conservação de momento linear, aplicando-se a segunda lei de Newton infinitesimal a um fluido que passa pelo volume de controle encontra-se a seguinte equação:

$$\frac{\partial}{\partial t}(\rho u) + \nabla \cdot (\rho u u) = 0, \quad (3.2)$$

em que o primeiro termo representa a taxa de aumento de momento (por unidade de volume) no volume de controle e o segundo termo representa a taxa de momento perdido por convecção (por unidade de volume) através da superfície de controle.

A primeira lei da termodinâmica aplicada a um fluido que passa por um volume de controle fixo infinitesimal produz a equação de conservação de energia:

$$\frac{\partial E_i}{\partial t} + \nabla \cdot E_i u = 0 \quad (3.3)$$

onde  $E_i$  é a energia total por unidade de volume dada como

$$E_i = \rho \left( e + \frac{u^2}{2} + \text{energia potencial} + \dots \right) \quad (3.4)$$

sendo  $e$  a energia interna por unidade de massa. O primeiro termo no lado esquerdo da Equação 3.3 representa a taxa de aumento de energia no volume de controle, enquanto o segundo termo representa a taxa de energia perdida por convecção através da

superfície do volume de controle.

Para determinar a solução do sistema de equações de conservação é utilizada a equação de estado, equação da lei dos gases ideais:

$$pV = nRT \quad (3.5)$$

sendo  $p$  a pressão,  $V$  o volume,  $n$  o número de mols da substância,  $R$  a constante dos gases ideais (8,314472 J/(mol·K)) e  $T$  a temperatura.

Assim, para simular o escoamento considerado neste trabalho são resolvidas seis equações, a conservação da massa, os três componentes do momento, a energia e a equação de estado.

Podemos representar as equações de conservação anteriores em termos de integrais (Equações 3.6, 3.7, 3.8). As equações escritas na forma de integrais são usadas no método de volumes finitos, onde os princípios de conservação são aplicados a uma região no espaço (volume de controle). A abordagem por este método faz um equilíbrio de alguma quantidade física na vizinhança de um ponto na malha.

Segundo Versteeg e Malalasekera (2007) existem três principais técnicas de solução numérica, diferenças finitas, volumes finitos e métodos espectrais. Em resumo, o algoritmo numérico de volumes finitos funciona do seguinte modo, inicialmente ocorre a integração das equações dominantes do fluxo de fluido sobre todos volumes de controle, depois a discretização que consiste na conversão das equações integrais resultantes em um sistema de equações algébricas, e por último, solução das equações algébricas por um método iterativo. O que distingue o método de volumes finitos dos outros é o primeiro passo. A conservação de uma variável de fluxo geral, por exemplo, a velocidade, dentro de um volume de controle finito, pode ser expressa como um equilíbrio entre os vários processos que tendem a aumentá-lo ou diminuí-lo.

A seguir são apresentadas as equações que são usadas neste trabalho usando o método de volumes finitos.

A equação de continuidade é definida como "o princípio físico de que a massa é conservada, ou seja, declara que o fluxo mássico líquido no volume de controle deve ser igual à taxa de aumento de massa dentro do volume de controle."(JUNIOR, 2001, p.



46). Matematicamente esta igualdade pode ser representada por:

$$-\iint_s \rho u dS = \frac{\partial}{\partial t} \iiint_V \rho dV \quad (3.6)$$

"A taxa de mudança de momento do fluido que está fluindo através do controle do volume a qualquer instante é igual à força líquida exercida sobre o fluido dentro do volume"(JUNIOR, 2001, p. 49). Tal comportamento é descrito pela equação de momento:

$$\iint_s (\rho u dS) u + \iiint_V \frac{\partial(\rho u)}{\partial t} dV = - \iint_s \rho dS \quad (3.7)$$

A equação da energia (3.8) descreve que "A taxa de calor adicionado ao fluido mais a taxa de trabalho realizado no fluido é igual à taxa de mudança de energia do fluido à medida que flui através do volume de controle, ou seja, a energia é conservada"(JUNIOR, 2001, p. 52).

$$-\iint_s \rho u dS = \iiint_V m \frac{\partial}{\partial t} \left[ \rho \left( e + \frac{u^2}{2} \right) \right] dV + \iint_s \rho \left( e + \frac{u^2}{2} \right) u dS \quad (3.8)$$

Descrições completas destas equações encontram-se no capítulo 2 do livro Junior (2001).

Os resultados que os *solvers* nos fornecem são o coeficiente de arrasto total no objeto, mas o que temos são os dados do coeficiente de arrasto frontal disponíveis para comparação. Sabendo que o valor do coeficiente de arrasto de base é de 0,116612527962822 e considerando que a pressão na base do cone é igual a pressão da corrente livre, podemos subtrair esse valor de cada resultado que o *solver* nos fornecer e comparar com 0,075479996996251, que é o valor de Taylor-Maccoll (BERTOLDO; MARCHI, 2017) que utilizamos como *benchmark*.

Partindo da equação do cálculo do arrasto  $D$ , dada por

$$D = Cd \frac{1}{2} \rho u^2 A \quad (3.9)$$

encontra-se a equação do cálculo de coeficiente de arrasto  $Cd$

$$Cd = \frac{D}{\frac{1}{2} \rho u^2 A} \quad (3.10)$$

em que  $\rho$  é a massa específica do ar,  $u$  é a velocidade do escoamento e  $A$  a área de referência. A área de referência varia de acordo com o objeto a ser estudado, no caso de um automóvel é a área de projeção frontal dele, para uma asa de avião é a

superfície alar dela. O coeficiente de arrasto pode ser interpretado como a resistência de um objeto durante o escoamento de um fluido, é um número adimensional e serve para quantificar essa resistência durante o escoamento de acordo com a superfície.

### 3.2 CONFIGURAÇÃO DOS SOLVERS

Para cada *solver* foram utilizados os mesmos arquivos com a mesma ordem de refino de malhas, por tanto o que se espera que diferencie os resultados são as características de cada *solver*:

- RhoSimpleFoam: Solucionador de estado estacionário para fluxo turbulento de fluidos compressíveis.
- RhoCentralFoam: Solucionador de fluxo compressível baseado em densidade e em esquemas central-upwind de Kurganov e Tadmor.
- SonicFoam: Solucionador transitório compressível para fluxos transônicos ( $0,8 < Ma^1 < 1,2$ ) ou supersônicos com turbulência.

Para configurar o OpenFoam é necessário editar alguns arquivos que estão dentro de pastas desta plataforma. Para cada malha é criada uma pasta com o nome da malha, por exemplo  $10 \times 10$ , dentro desta pasta estarão as pastas 0, constant e system. Dentro da pasta 0 estão os arquivos P, T e U. Dentro da pasta constant ficam os arquivos thermophysicalProperties e turbulenceProperties, e dentro da pasta system ficam os arquivos *blockMeshDict*, *controlDict*, *fvSchemes* e *fvSolution*. No decorrer deste capítulo é explicada a função de cada arquivo e quais valores usados.

Inicialmente, é preciso definir no arquivo blockMesh (apêndice Apêndice A), a forma do objeto, suas dimensões e as propriedades de cada face, pois estas podem ser com ou sem atrito e se permitem passagem de fluido, qual o tamanho da malha e se esta possui regiões com espessura diferente. Neste trabalho é considerado um cone onde o fluido toca somente a superfície deste, sem que a estrutura onde o escoamento acontece atrapalhe a passagem do fluido. As malhas tem o número de volumes multiplicado por dois em cada direção (x e y) para cada nova simulação, ou seja, serão refinadas de  $10 \times 10$ ,  $20 \times 20$ ,  $40 \times 40$  e seguem assim até  $640 \times 640$ , resultando que cada célula terá um quarto do tamanho da célula da simulação anterior. No eixo z que define

<sup>1</sup> Número de Mach:  $Ma = \frac{\text{Velocidade média do objeto}}{\text{Velocidade do som}}$

o grau de abertura do cone ocorre o mesmo, inicialmente o cone terá 5° de abertura, na segunda 2.5° e assim sucessivamente.

Em *fvSchemes* são definidos os métodos de resolução das equações e as discretizações utilizadas (Apêndice B).

Na pasta 0 encontram-se os arquivos P, T e U, onde são definidas as entradas de valores de pressão, temperatura e velocidade do escoamento respectivamente Tabela 1.

**Tabela 1 – Propriedades do fluido**

<b>Pressão</b>	
Constante	49184,4770563629 Pa
<b>Temperatura</b>	
Uniforme	300 K
<b>Velocidade</b>	
Constante (eixo x)	1215,20356055112 m/s

Na pasta *constant* é definido o regime de escoamento (com ou sem turbulência), descrição completa do fluido e descrição completa da malha (pasta *polyMesh*). Esta pasta é criada após ser executado o comando *blockMesh* no terminal, é uma pasta gerada pelo OpenFoam de acordo com as entradas de dados contidas em *blockMeshDict* dentro da pasta *system*. No Apêndice A são apresentados os pontos que formam o cone utilizado na simulação.

### Tempo total simulado do escoamento

O tempo total do escoamento foi escolhido com base na velocidade do fluido e no comprimento do cone, assim, define-se o *endTime* (tempo total do escoamento) como:

$$\text{ENDTIME} = 10 \times \frac{\text{COMPRIMENTO DO CONE}}{\text{VELOCIDADE DO FLUIDO}} =$$

$$10 \times \frac{0,0762}{1215} = 10 \times 0,000627055 = 0,000627055 \approx 6 \times 10^{-4} \text{s} \quad (3.11)$$

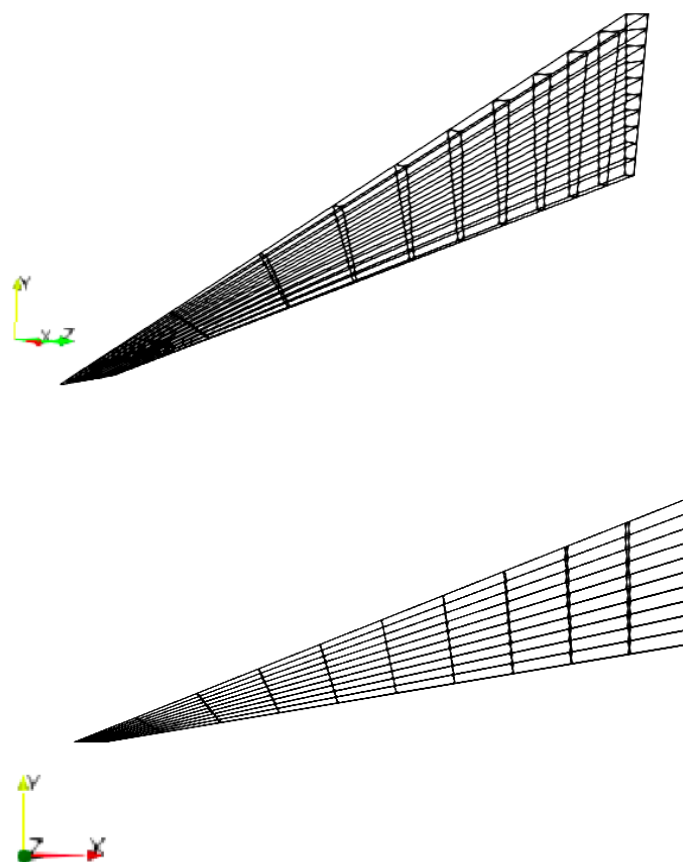
O  $\delta t$  é o passo de tempo usado na discretização da derivada temporal, é definido usando a menor célula possível. Observando a Figura 3 podemos ver que a menor

célula na malha 10x10 é a da ponta do cone. A distância da ponta do cone até o contorno é de 0,00381m. Na malha mais grossa, que tem 10 volumes por direção, a menor largura da célula é de 0,000381m. Como  $\delta t = \text{distância}/\text{velocidade}$ , temos

$$\delta t = \frac{0,000381m}{1215,20356055112m/s} \sim 3 \times 10^{-7}s$$

Estes tempos são inseridos no arquivo controlDict, além de outras informações como intervalo de escrita e precisão. No caso, o solver interrompe a simulação caso atinja dez casas de precisão. Essa precisão é baseada na quantidade de vezes que o solver encontra o mesmo resultado em determinado tempo, ou seja, se para um determinado tempo o *solver* encontra o mesmo resultado com dez casas de precisão a simulação é interrompida.

**Figura 3** – Estrutura representando os volumes por onde escoa o fluido



**Fonte** – Autoria própria (2019)

O próprio *solver* durante a simulação gera os dados que precisamos, tais como tabelas com os valores de tempo  $\delta t$  (definidos em intervalos escolhidos de acordo

com o tamanho da menor célula) e de  $Cd$ , além de outras grandezas que não foram utilizadas neste trabalho.

Após geração e coleta desses dados, é necessário verificar se os resultados convergem para alguma solução e se esta se aproxima da solução analítica, pois se a solução não convergir podem existir erros de entradas de dados ou o *solver* apresenta alguma falha no código. Se a solução converge, é necessário verificar qual é este valor, para futuramente calcular o erro e/ou a precisão do *solver*. Também com este valor é realizada a comparação com os resultados dos outros *solvers*, pois assim é possível determinar qual deles é mais preciso. Além do valor de  $Cd$  também faremos uma comparação entre os tempos que cada *solver* leva para chegar a solução, pois nem sempre precisaremos do *solver* com a melhor solução e sim um que consiga no menor tempo possível uma solução que se considere aceitável.

Os estimadores de Richardson e GCI são usados para estimativas de erro e incertezas, pois estes também são usados em outros trabalhos que abordam este tipo de simulação e são largamente usados pela comunidade científica.

Para o *solver* rhoCentralFoam é necessário fazer uma média dos valores de  $Cd$ , pois apesar de os resultados que o *solver* fornece convergirem para algum ponto, eles acabam tendo oscilações em intervalos de tempo menores, assim, para o valor de  $Cd$  do rhoCentralFoam a média é calculada usando o tempo para o fluido atravessar o cone 3x, ou seja,  $3 \times \frac{0,0762m}{1220m/s} \sim 1,88 \times 10^{-4}s$ .

Com os valores fornecidos pelo *solver* é possível criar gráficos de  $Cd$  x tempo onde é possível analisar a convergência ou não da solução. Também com os valores de  $Cd$  é construída uma tabela para aplicar os estimadores de Richardson e GCI e comparar os estimadores e os resultados apresentados por estes para cada *solver*.

Durante a simulação é possível cronometrar o tempo de execução, assim além dos valores de  $Cd$  que nos interessam, também é possível verificar o tempo que a simulação levou para convergir ou para chegar ao tempo final de simulação.

### 3.3 VERIFICAÇÃO

Como exposto na Introdução, neste trabalho é feita a verificação de solução. A diferença entre verificação de solução e verificação de código é que na verificação

de código temos o objetivo de descobrir se há erros de programação do código, já na verificação de solução utilizando o resultado da simulação e os estimadores calcula-se qual a estimativa de erro desta solução.

Pode-se avaliar a qualidade de uma estimativa de erro através da sua efetividade

$$\theta = \frac{U}{E} \quad (3.12)$$

a efetividade ( $\theta$ ) é a razão entre a incerteza ( $U$ ) e o erro  $E$ . Segundo Marchi (2001, p. 6) "A estimativa de erro da solução numérica de uma variável de interesse é avaliada pela diferença entre a sua solução analítica estimada ( $\phi_\infty$ ) e a própria solução numérica ( $\phi$ )"

$$U(\phi) = \phi_\infty - \phi \quad (3.13)$$

Uma estimativa de erro ideal é aquela cuja efetividade é igual à unidade ( $\theta = 1$ ), isto é, quando a incerteza é igual ao erro ( $U = E$ ), neste caso, a estimativa de erro é confiável e tem a máxima acurácia<sup>2</sup> possível. "Uma estimativa de erro é confiável quando a magnitude da incerteza é maior do que a magnitude do erro de discretização e ambos têm o mesmo sinal."(MARCHI, 2001, p 53).

O erro de discretização é causado apenas por erros de truncamento, portanto, não são considerados os erros de arredondamento, de iteração e de programação. De acordo com o que Marchi (2001) expôs, erro de truncamento ( $\varepsilon_\tau$ ) de uma equação diferencial é o resíduo que resulta quando se substitui a solução analítica exata da variável dependente ( $\Lambda$ ) na equação discretizada do modelo matemático.

Entende-se  $\Lambda$  como sendo a variável dependente nas equações diferenciais dos modelos matemáticos que se utilizam de métodos numéricos para resolver problemas similares aos abordados neste trabalho.

O erro de discretização  $E(\phi)$  é a diferença entre a solução analítica ( $\Phi$ ) e a solução numérica ( $\phi$ )

$$E(\phi) = \Phi - \phi \quad (3.14)$$

<sup>2</sup> Entende-se como exatidão ou precisão

Já a equação geral do erro de discretização é escrita da seguinte forma:

$$E(\phi) = C_1 h^{p_l} + C_2 h^{p_2} + C_3 h^{p_3} + C_4 h^{p_4} + \dots \quad (3.15)$$

em que  $E(\phi)$  é o erro de discretização da solução numérica ( $\phi$ ),  $h$  representa o tamanho dos elementos que compõem a malha,  $p_l$  é a ordem assintótica e  $C$  são coeficientes independentes de  $h$ .

A ordem assintótica  $p_l$  utilizada acima é definida como o menor expoente de  $h$  na equação geral do erro de truncamento Equação 3.16 (MARCHI, 2001 apud FERZIGER; PERIĆ, 1999).

$$\varepsilon(\phi) = C_1 h^{p_l} + C_2 h^{p_2} + C_3 h^{p_3} + C_4 h^{p_4} + \dots \quad (3.16)$$

onde os coeficientes  $C_i$  podem ser positivos ou negativos e podem ser função da variável dependente ( $\Lambda$ ) e de suas derivadas, isto é, podem variar com a coordenada  $x$ , mas independem do tamanho ( $h$ ) dos elementos da malha.

Percebe-se que as equações (3.15) e (3.16) são iguais, mas é importante considerar que os coeficientes  $C_i$  da Equação 3.15 podem ser iguais ou diferentes dos coeficientes  $C_i$  da Equação 3.16. As duas equações podem ser definidas usando as ordens verdadeiras e assintótica.

Segundo Marchi (2001) define-se as ordens verdadeiras ( $pV$ ) como sendo os expoentes de  $h$  dos termos não-nulos na equação do erro de truncamento. No caso da Equação 3.16 são dados por  $pL, p_2, p_3$  e  $pn$ . As ordens verdadeiras seguem a relação  $pL < p_2 < p_3 < p_4 < \dots < pn$ . Já a ordem assintótica é definida como o menor expoente de  $h$  na equação geral do erro de truncamento. É um número inteiro positivo e satisfaz a condição  $pL \geq 1$ .

No tocante ao que foi dito anteriormente sobre a utilização dos estimadores e segundo o que Marchi (2001) expôs em seu trabalho, definimos a estimativa de Erro de Richardson como sendo

$$U_{Ri}(\phi) = \phi_\infty - \phi \quad (3.17)$$

Quando não se dispõe de uma solução analítica é feita uma estimativa do valor da solução analítica ( $\phi_\infty$ ) que pode ser calculada pelo Estimador de Richardson ( $U_{Ri}$ )

$$U_{Ri}(\phi_1) = \frac{(\phi_1 - \phi_2)}{(q^{p_l} - 1)} \quad (3.18)$$

onde  $q$  é a razão de refino das malhas e  $\phi_1$  e  $\phi_2$  que são as soluções numéricas obtidas com malhas fina e grossa respectivamente.

A estimativa de Richardson e a estimativa GCI também podem ser calculadas usando a ordem aparente ( $p_u$ ), por exemplo, para o estimador de Richardson a equação fica da seguinte forma:

$$U_{Ri}(\phi_1) = \frac{(\phi_1 - \phi_2)}{(q^{p_u} - 1)} \quad (3.19)$$

"A ordem aparente ( $p_u$ ) é definida como a inclinação local da curva de incerteza ( $U$ ) da solução numérica ( $\phi$ ) versus o tamanho ( $h$ ) dos elementos da malha num gráfico logarítmico." (MARCHI, 2001, 84). Assim podemos verificar se a ordem da incerteza da solução que encontramos tende a ordem assintótica dos erros de truncamento.

#### *Ordem aparente com refino constante*

$$p_u = \frac{\log \psi_u}{q} \quad (3.20)$$

onde,

$$\psi_u = \frac{(\phi_2 - \phi_3)}{(\phi_1 - \phi_2)} \quad (3.21)$$

em que  $p_u$  é a ordem aparente,  $q$  a razão de refino da malha,  $\phi_1$  a malha fina,  $\phi_2$  a malha intermediária e  $\phi_3$  a malha grossa. Quando temos a razão de refino constante podemos usar a Equação 3.21, onde  $\psi_u$  é definido por Marchi (2001) como sendo a razão de convergência da solução numérica para a solução analítica.

A vantagem de se usar  $p_u$  é que ele pode ser calculado para qualquer problema e variável de interesse, porém são necessárias três soluções numéricas.

Também podemos calcular os estimadores com base ordem efetiva  $p_e$ , definida por Marchi (2001) como "a inclinação local da curva do erro de discretização ( $E$ ) da solução numérica ( $\phi$ ) versus o tamanho ( $h$ ) dos elementos da malha num gráfico logarítmico". Matematicamente  $p_e$  é definido como

$$C_e h^{p_e} = E(\phi) \quad (3.22)$$

em que  $C_e$  é um coeficiente que pode ser independente de  $h$ . Assim, para se calcular o



valor de  $pe$  usa-se a solução analítica ( $\Phi$ ) na equação a seguir

$$pe = \frac{\log\left(\frac{\Phi - \phi_2}{\Phi - \phi_1}\right)}{\log(q)}, \quad (3.23)$$

sendo  $\phi_1$  o resultado na simulação com a malha fina,  $\phi_2$  na malha grossa e  $pe$  é a ordem efetiva.

Quando se tem a efetividade igual a um, temos uma estimativa de erro ideal, ou seja,

$$\frac{U_{Ri}}{E} = 1$$

Segundo (MARCHI, 2001 apud ROACHE, 1998) usando o Estimador GCI calcula-se a incerteza de uma solução numérica usando a equação:

$$U_{GCI}(\phi_1) = F_s \frac{|\phi_1 - \phi_2|}{(q^{pl} - 1)}, \quad (3.24)$$

em que  $F_s$  é um fator de segurança igual a três.

De modo semelhante ao de Richardson a efetividade do estimador GCI ( $\Theta_{GCI}$ ) é definida pela razão entre a sua incerteza ( $U_{GCI}$ ) e o módulo do erro de discretização ( $E$ ),

$$\Theta_{GCI} = \frac{U_{GCI}}{|E|} \quad (3.25)$$

De acordo com Marchi (2001) partindo da equação Equação 3.24, a representação correta da solução numérica ( $\phi$ ) e sua incerteza ( $U_{GCI}$ ) obtida com o estimador GCI é

$$\phi = \phi_1 \pm U_{GCI}(\phi_1) \quad (3.26)$$

em que  $\phi_1$  é a solução numérica na malha fina.

## 4 RESULTADOS

Neste capítulo são apresentados os resultados encontrados pelos *solvers*, os resultados que os estimadores geraram e adaptações em relação a ideia inicial do trabalho.

Após realizar as simulações foi possível perceber que diferente do que foi feito inicialmente, que além do refinamento de malhas nas direções x e y, também é necessário fazer um refino no ângulo da cunha, ou seja, a cada refino de malha 10x10, 20x20, 40x40, ...640x640, também é necessário refinar o ângulo  $5^\circ$ ,  $\frac{5^\circ}{2}$ ,  $\frac{5^\circ}{4}$ ,  $\frac{5^\circ}{8}$ ... . Com estas simulações foi possível obter os valores de  $Cd$  total que estão nas Tabelas 2 e 3.

Também foram realizadas simulações para a malha de 1280 x1280, mas os resultados gerados pelos três *solvers* foram inconclusivos. Uma possível justificativa seria as células serem demasiadamente pequenas em determinadas regiões combinadas com alguma limitação do *solver* fazendo os resultados divergirem. Foram realizados testes mudando parâmetros de parada e refino de malha e mesmo assim os resultados não mudaram, portanto foram desconsiderados os resultados desta malha para os três *solvers*.

A seguir encontra-se a Tabela 2 com o cálculo de  $Cd$  frontal, que é a diferença do resultado que o *solver* fornece ( $Cd$  total) e o coeficiente de arrasto de base (0,116612527962822), ou seja, o resultado disso é o coeficiente de arrasto frontal.

**Tabela 2 – Coeficiente de arrasto frontal**

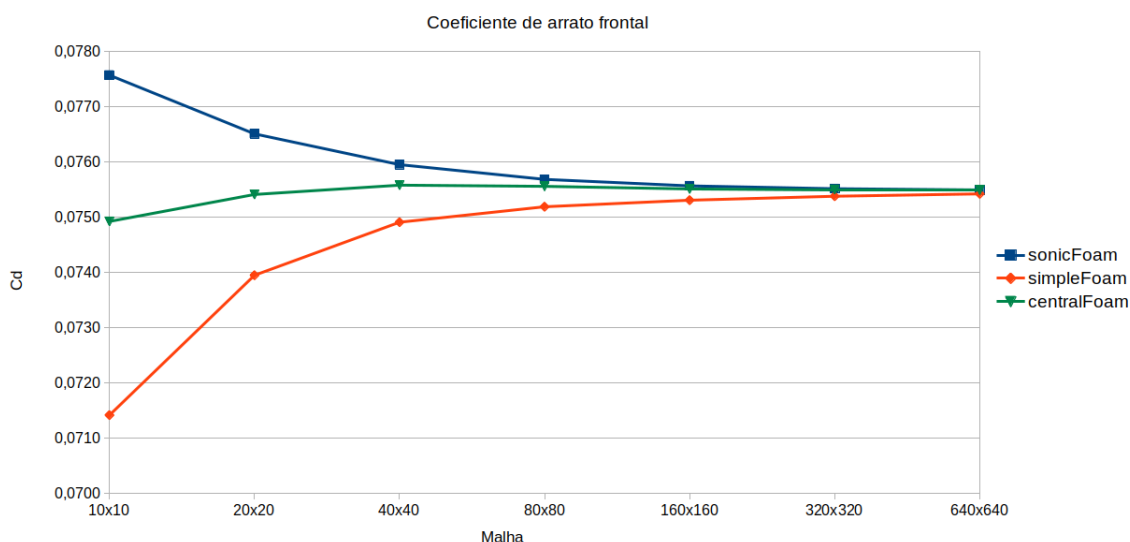
<b>Malha</b>	<b>sonicFoam</b>	<b>rhoSimpleFoam</b>	<b>rhoCentralFoam</b>
10x10	0,077576317094	0,071421239257	0,074928493507
20x20	0,076513823040	0,073953962207	0,075416913098
40x40	0,075956813163	0,074914261562	0,075585508001
80x80	0,075690442702	0,075194210057	0,075564125157
160x160	0,075572426690	0,075313021273	0,075516681790
320x320	0,075521282732	0,075384745155	0,075507141651
640x640	0,075499034830	0,075427678319	0,075495415113

**Fonte –** Autoria própria (2019)

Assim, os valores da Tabela 2 utilizados como base para os demais cálculos.

O gráfico a seguir apresenta o comportamento da convergência de  $C_d$  frontal para cada *solver* durante o refino de malhas.

**Figura 4 – Coeficiente de arrasto frontal**



Fonte – Autoria própria (2019)

Vale considerar que o valor do coeficiente de arrasto na malha mais fina do *solver* rhoCentralFoam não chegou a convergir, a simulação foi interrompida com  $9.844\,278\,96 \times 10^{-5}$  s, pois este leva muito tempo para convergir, não havendo tempo hábil para que isto acontecesse. Como seus resultados são obtidos através de média aritmética dos últimos valores (descrito anteriormente), foi realizada a média das últimas 7250 iterações.

Sobre os resultados de  $C_d$ , sabe-se que o resultado usado como *benchmark* para o problema em questão é  $C_d = 0,075479996996251$ . A seguir está tabelada a diferença entre os resultados de referência e dos *solvers*:

Pelos resultados mostrados na Tabela 3 é possível perceber que todas as simulações melhoraram seu resultado com o refinamento da malha.

Na Tabela 4 estão os resultados dos cálculos de diferença entre malhas e de ordem aparente ( $pu$ ). Seu cálculo permite verificar na prática, isto é, a posteriori das soluções numéricas, se à medida que  $h$  é reduzido, a ordem da incerteza das soluções numéricas tende à ordem assintótica ( $pl$ ) dos erros de truncamento, ordem esta que é um resultado teórico, obtido a priori das soluções numéricas.

Como os resultados apresentados por  $pu$  não seguem uma ordem assintótica,

**Tabela 3** – Diferença entre Taylor-Maccoll e OpenFOAM

	<b>Malha</b>	<b>sonicFoam</b>	<b>rhoSimpleFoam</b>	<b>rhoCentralFoam</b>
m1	10x10	-0,002096	0,004059	0,000552
m2	20x20	-0,001034	0,001526	0,000063
m3	40x40	-0,000477	0,000566	-0,000106
m4	80x80	-0,000210	0,000286	-0,000084
m5	160x160	-0,000092	0,000167	-0,000037
m6	320x320	-0,000041	0,000095	-0,000027
m7	640x640	-0,000019	0,000052	-0,000015

Fonte – Autoria própria (2019)

**Tabela 4** – Ordem aparente  $pu$ 

<b>Malhas</b>	<b>sonicFoam</b>	<b>rhoSimpleFoam</b>	<b>rhoCentralFoam</b>
m1, m2 e m3	0,93	1,40	1,53
m2, m3 e m4	1,06	1,78	—
m3, m4 e m5	1,17	1,24	-1,15
m4, m5 e m6	1,21	0,73	2,31
m5, m6 e m7	1,20	0,74	-0,30

Fonte – Autoria própria (2019)

ou seja, não convergem para um valor, indicando que: (i) a malha ainda não está suficientemente refinada ou (ii) há algum erro de programação. Como não é possível afirmar qual dos dois casos está ocorrendo, não é adequado aplicar  $pu$  para calcular estimativas de erro.

Na Tabela 5 são apresentados os valores de  $pe$ :

**Tabela 5** – Ordem efetiva  $pe$ 

<b>Malhas</b>	<b>sonicFoam</b>	<b>simpleFoam</b>	<b>rhoCentralFoam</b>
m1, m2 e m3	1,02	1,41	3,13
m2, m3 e m4	1,12	1,43	—
m3, m4 e m5	1,18	0,99	0,33
m4, m5 e m6	1,19	0,78	1,20
m5, m6 e m7	1,16	0,81	0,43

Fonte – Autoria própria (2019)

Segundo Marchi (2001) quando o tamanho da célula que compõem a malha tende a zero, a ordem efetiva  $pe$  tende a ordem assintótica ( $pl$ ) pois o primeiro termo do erro passa a dominar o seu valor total. Assim, utilizando a ordem ( $pl$ ) e o estimador GCI, foi gerada a Tabela 6.

Pelos resultados apresentados na Tabela 6 nota-se que os valores do estimador

**Tabela 6** – Incerteza GCI usando  $pl$ 

Malhas	sonicFoam	rhoSimpleFoam	rhoCentralFoam
m1 e m2	0,003187	0,007598	0,001465
m2 e m3	0,001671	0,002881	0,000506
m3 e m4	0,000799	0,000840	0,000064
m4 e m5	0,000354	0,000356	0,000142
m5 e m6	0,000153	0,000215	0,000029
m6 e m7	0,000067	0,000129	0,000035

Fonte – Autoria própria (2019)

GCI melhoram com o refino da malha nos *solvers* sonicFoam e rhoSimpleFoam, convergindo para 0. Já o *solver* rhoCentralFoam apesar de apresentar melhoras com o refinamento de malha não obteve um resultado melhor na última simulação, isso se deve muito provavelmente ao fato da simulação não ter alcançado o critério de parada.

Também foi feito o mesmo cálculo que a tabela anterior mas agora com o estimador de Richardson (Tabela 7).

**Tabela 7** – Incerteza de Richardson usando  $pl$ 

Malha	sonicFoam	rhoSimpleFoam	rhoCentralFoam
20x20	0.001062	0,190566	0,000488
40x40	0,000557	0,191527	0,000169
80x80	0,000266	0,000280	0,000021
160x160	0,000118	0,000119	0,000047
320x320	0,000051	0,000072	0,000010
640x640	0,000022	0,000043	0,000012

Fonte – Autoria própria (2019)

Observando os valores gerados pela Tabela 7, é possível perceber que novamente o intervalo de confiança aumentou no decorrer do refino de malhas, mas que no *solver* rhoCentralFoam esses resultados não seguiram uma melhora constante.

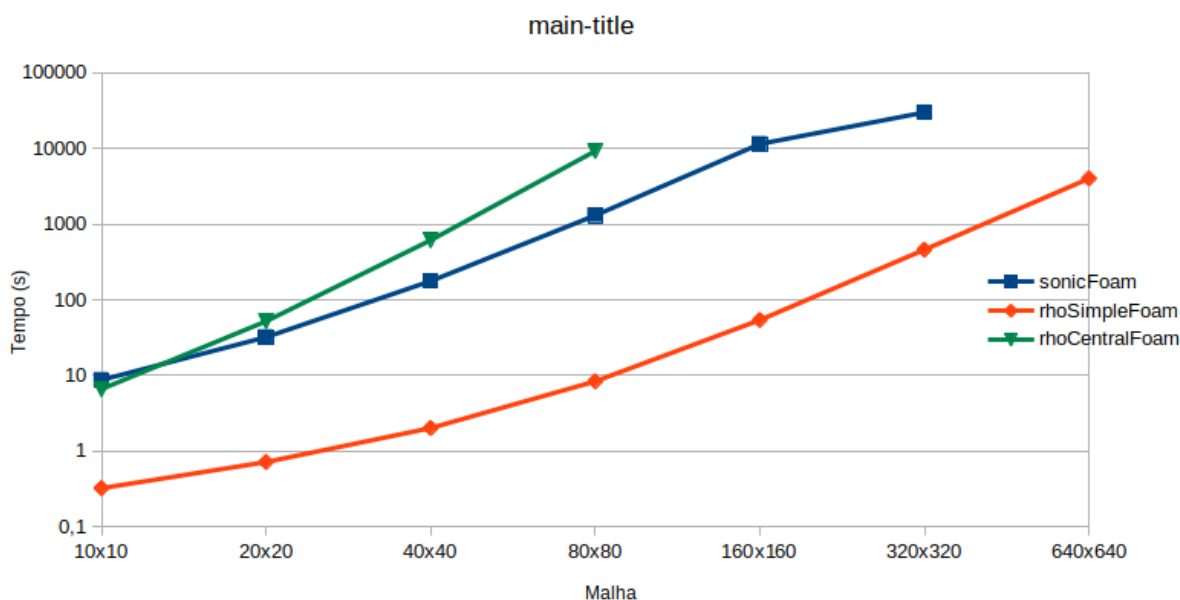
Também foi possível gerar a Tabela 8 com os tempos de cada simulação: No gráfico que segue é possível visualizar a forma com que o tempo de simulação se comporta para cada *solver* no decorrer do refino de malhas. Podemos observar que nem todos os tempos estão tabelados, isto se deve ao fato de que para malhas mais finas onde o tempo de simulação é maior, foi necessária a criação de várias tabelas com os valores de  $Cd$  e de tempo, como cada tabela inicia a contagem do zero algumas malhas criaram muitos arquivos, inviabilizando a contagem.

Tabela 8 – Tempo de simulação

Malha	sonicFoam	rhoSimpleFoam	rhoCentralFoam
10x10	8,66 s	0,32 s	6,56 s
20x20	31,94 s	0,71 s	51,8 s
40x40	176,68 s	2 s	613,24 s
80x80	1304,67 s	8,31 s	9278,85 s
160x160	11517,13 s	53,79 s	
320x320	268974,22 s	459,39 s	
640x640		4023,93 s	

Fonte – Autoria própria (2019)

Figura 5 – Tempo de simulação x Refino de malha



Fonte – Autoria própria (2019)

Percebe-se que de acordo com o refinamento da malha o tempo de simulação aumenta em todos os *solvers*. Também é evidente que o *solver* que atingiu o critério de convergência em menos tempo é o *rhoSimpleFoam*, seguido do *sonicFoam* e por último o *rhoCentralFoam*, isso em todas as simulações cronometradas, mas lembrando que somente o tempo é levado em consideração e não a qualidade dos resultados.

## 5 CONCLUSÕES

Durante as simulações pode-se perceber que é alto o nível de processamento exigido por este tipo de simulação, considerando ainda que este é um problema pequeno se comparado a um problema real de fluxo de ar sobre a superfície de um avião por exemplo, em que o objetivo seria o melhor resultado possível com a maior confiabilidade possível.

Os resultados que foram gerados pelos *solvers* não se distanciaram da solução de referência, a diferença deles para o resultado de Taylor-Maccoll foi da ordem de 0,020% do rhoCentralFoam, 0,069% do simpleFoam e 0,025% para o sonicFoam na simulação da malha mais fina.

Os estimadores obtiveram resultados com confiabilidade de 0,08% para o sonicFoam, 0,17% para o rhoSimpleFoam e 0,046% para o rhoCentralFoam. O resultado obtido pelo rhoCentralFoam é satisfatório considerando que a diferença entre os *solvers* e Taylor-Maccoll também é dessa ordem. Os resultados para as estimativas de erro de rhoSimpleFoam não foram satisfatórias se comparada com os outros dois, portanto a utilização destes resultados varia de acordo com a necessidade do problema. É importante considerar que se esperava que as ordens  $pu$  e  $pe$  convergissem em todos os casos, o que não aconteceu, talvez por algum erro de programação ou a necessidade de um maior refino na malha.

Para resultados melhores seria necessário identificar a razão pela qual não foi possível gerar resultados coerentes com a malha 1280 x 1280. Pois mesmo com um processador mais rápido, o que acabou sendo um problema para o rhoCentralFoam na sua última simulação, não se chegaria a um resultado melhor.

Observando os resultados obtidos é possível perceber que o tempo de processamento do *solver* cresce exponencialmente com o refino da malha. O *solver* rhoCentralFoam é quem possui as melhores aproximações mas também é o que leva mais tempo, seguido do sonicFoam e do rhoSimpleFoam. Logo, levando em consideração que mesmo em malhas mais grossas o rhoCentralFoam encontrou melhores resultados se comparado com malhas mais finas dos outros dois, o que torna seu tempo de melhor resposta menor que os outros dois, seria o *solver* mais indicado para este tipo de simulação.

## 5.1 CONSIDERAÇÕES FINAIS

Este trabalho trouxe uma ótima ferramenta para a realização de simulações, tanto no que se refere ao OpenFOAM quanto aos *solvers*. Apesar de exigir uma grande capacidade de processamento pelos computadores, consegue ser útil em outros tipos de pesquisa também, pois nem sempre é necessária uma solução com alta precisão.

Quando o Greenshields (2018) diz que é necessário ter conhecimento sobre a física do problema, a matemática e a parte de programação para se fazer modificações nos *solvers* realmente é necessário que o usuário tenha esses conhecimentos. Em compensação o OpenFOAM traz vantagens em relação as entradas de dados, o usuário não precisa ter alto grau de conhecimento para isso, e a forma com que ele entrega os resultados também é muito prática.



## 6 TRABALHOS FUTUROS

Como foi possível chegar a resultados de grande precisão com os *solvers*, podem ser feitas simulações usando problemas mais complexos, por exemplo, considerando o fluido como sendo viscoso, os dados podem ser comparados com os deste trabalho e comparados com a solução de Taylor-Maccoll, pois apesar de ser resolvido usando o modelo de Euler, espera-se que com o modelo de Navier-Stokes por exemplo, encontre-se soluções com resultados mais próximos da realidade.

Também podem ser feitos testes com outros tipos de malhas, no caso de malhas finas onde os volumes de controle acabaram se tornando muito pequenos (malha 1280 x 1280) gerando resultados inconclusivos, podem ser feitos testes com malhas geradas por outros programas, o que pode resolver o problema da divergência dos dados.

## REFERÊNCIAS

- BABUSKA, Ivo; ODEN, J. Tinsley. Verification and validation in computational engineering and science: basic concepts. **Science Direct**, n. 3, p. 4057–4066, 2004. páginas 19
- BERTOLDO, Guilherme; MARCHI, Carlos Henrique. Verification and validation of the foredrag coefficient for supersonic and hypersonic flow of air over a cone of fineness ratio 3. **Elsevier**, p. 409–424, 2017. páginas 22, 24
- BLOTTNER, Frederick G. Accurate navier-stokes results for the hypersonic flow over a spherical nosetip. **Journal of Spacecraft and Rockets**, v. 27, p. 113–122, 1990. páginas 19
- CORREIA, Rui Manuel Vieira. **Simulação Computacional em OpenFOAM de um Leito Fluidizado**. Dissertação (Mestrado) — Faculdade de Ciências e Tecnologia - Universidade de Coimbra, 2016. páginas 17
- FERZIGER, Joel H.; PERIĆ, Milovan. **Computational Methods for Fluid Dynamics**. 2. ed. [S.l.]: Berlin : Springer, 1999. páginas 16, 30
- FIATES, Juliane. **Desenvolvimento de uma Metodologia para Simulação de dispersão de gás inflamável por meio de CFD Utilizando OpenFOAM**. Dissertação (Mestrado) — Universidade Estadual de Campinas - INICAMP, 2015. páginas 18
- GREENSHIELDS, Christopher J. **User Guide**. [S.l.], 2018. 237 p. Disponível em: <<http://openfoam.org>>. Acesso em: 21 de maio de 2019. páginas 14, 17, 39
- JUNIOR, John David Anderson. **Fundamentals of Aerodynamics**. 3. ed. Avenue of the Americas, New York: McGraw-Hill, 2001. páginas 17, 18, 20, 23, 24
- MARCHI, Carlos Henrique. **Verificação de Soluções Numéricas Unidimensionais em Dinâmica dos Fluidos**. Julho 2001. 305 p. Tese (Doutor em Engenharia – Engenharia Mecânica) — Universidade Federal de Santa Catarina, Florianópolis, 2001. páginas 14, 15, 16, 18, 19, 29, 30, 31, 32, 35
- ROACHE, P. J. **Verification and Validation in Computational Science and Engineering**. Albuquerque, USA: Berlin : Springer, 1998. páginas 19, 32
- TANNEHILL, John C.; ANDERSON, Dale A.; PLETCHER, Richard H. **Computational Fluid Mechanics and Heat Transfer**. 2. ed. 1101 Vermont Avenue, NW, Suite 200, Washington, DC: Taylor & Francis, 1997. páginas 14, 18, 22
- TAYLOR, Geoffrey Ingram; MACCOLL, J. W. The air pressure on a cone moving at high speeds ii. **Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character**, v. 139, n. 838, p. 298–311, 1933. Disponível em: <<https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1933.0018>>. páginas 18

VERSTEEG, H K; MALALASEKERA, W. **An Introduction to Computational Fluid Dynamics**. 2. ed. Saffron House, Kirby Street, London, England: Pearson Education Limited, 2007. páginas 17, 23

## **APÊNDICES**

## APÊNDICE A – ARQUIVO UTILIZADO PARA DEFINIR A GEOMETRIA E AS CONDIÇÕES DE CONTORNO

```

convertToMeters 1;
vertices
(
  (0 0 0)
  (0.0762 0.0126879124140896 -0.000553966219539767)
  (0.0762 0.031719781035224 -0.00138491554884942)
  (-0.00381 0 0)
  (0.0762 0.0126879124140896 0.000553966219539767)
  (0.0762 0.031719781035224 0.00138491554884942)
);

blocks
(
  hex (0 1 2 3 0 4 5 3) (10 10 1) simpleGrading (1 1 1)
);

edges
(
  arc 1 4 (0.0762 0.0127 0)
  arc 2 5 (0.0762 0.03175 0)
);

boundary
(
  inlet
  {
    type patch;
    faces
    (
      (3 5 2 3)
    );
  }
  outlet
  {
    type patch;
  }
);

```

```
        faces
        (
            (1 2 5 4)
        );
    }
    coneSurface
    {
        type wall;
        faces
        (
            (0 1 4 0)
        );
    }
    symmetryLine
    {
        type symmetry;
        faces
        (
            (3 0 0 3)
        );
    }
    front
    {
        type wedge;
        faces
        (
            (0 4 5 3)
        );
    }
    back
    {
        type wedge;
        faces
        (
            (0 3 2 1)
        );
    }
);
```

**APÊNDICE B – EXEMPLO DE CONFIGURAÇÃO DO ARQUIVO FVSCHEMES**

```
ddtSchemes
{
    default      Euler;
}

gradSchemes
{
    default      Gauss linear;
}

divSchemes
{
    default      none;
    div(phi,U)   Gauss upwind;
    div(phiid,p) Gauss limitedLinear 1;
    div(phi,e)   Gauss limitedLinear 1;
    div(phi,K)   Gauss limitedLinear 1;
    div(phiiv,p) Gauss limitedLinear 1;
    div(((rho*nuEff)*dev2(T(grad(U)))) Gauss linear;
}

laplacianSchemes
{
    default      Gauss linear corrected;
}

interpolationSchemes
{
    default      linear;
}

snGradSchemes
{
    default      corrected;
}
```