

3 Formulação Numérica

3.1 Método de volumes finitos

Para a solução numérica do problema estudado, foi utilizado o programa comercial Fluent (Fluent, Inc.), licenciado pelo Departamento de Engenharia Mecânica da Pontifícia Universidade Católica do Rio de Janeiro. No programa Fluent, as equações de conservação são discretizadas através do método de volumes finitos, descrito por Patankar (1980).

O método de volumes finitos é construído a partir da divisão do domínio de integração das equações em volumes de controles, de forma que para cada célula exista um ponto nodal ou nó. Os nós estão situados no centro dos volumes de controle.

A Fig. 4.1 mostra a representação de um volume de controle e sua vizinhança. O ponto P é o ponto principal, nó, de um volume de controle genérico. Os pontos E, W, N e S são os pontos principais dos volumes de controle vizinhos a leste, oeste, norte e sul respectivamente, sendo δ_x e δ_y as distâncias entre estes pontos e P. As faces dos volumes de controle são representadas pelas linhas pontilhadas e Δ_x e Δ_y são as distâncias entre as faces, ou dimensão do volume de controle. As velocidades axiais nas faces leste e oeste são representadas por u_e e u_w , da mesma forma as velocidades radiais nas faces norte e sul são representadas por v_n e v_s . As velocidades são obtidas nas faces dos volumes de controle, enquanto todas as outras variáveis são obtidas nos nós. Foi utilizada uma malha deslocada para os componentes de velocidade, a fim de evitar campos de pressão irreais.

O método consiste na integração das equações diferenciais de conservação em cada volume de controle. Assumindo perfis para a variação das variáveis entre pontos nodais, obtêm-se as equações discretizadas. Assim, as equações discretizadas expressam o princípio de conservação da mesma forma que a

equação diferencial faz para volumes infinitesimais, garantindo desta forma a conservação integral das grandezas envolvidas.

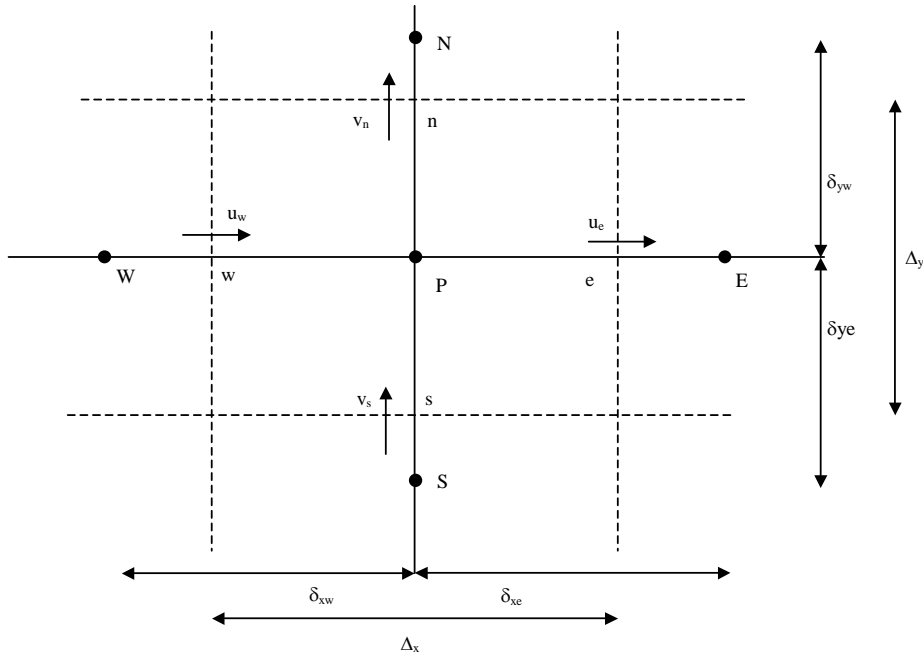


Fig. 3.1 Volume de controle típico

De acordo com o método de volumes finitos de Patankar, as equações diferenciais devem ser representadas em sua forma conservativa como:

$$\frac{\partial}{\partial t}(\rho\phi) + \text{div}(\rho v\phi) = \text{div}(\Gamma \text{grad}(\phi)) + S \quad \text{Eq. (3.1)}$$

onde ρ é a massa específica, v é o vetor velocidade, ϕ a variável dependente, Γ o termo de difusão e S o termo de fonte. A fonte é dada por um termo dependente da variável e outro independente, como podemos ver.

$$S = S_c + S_p\phi \quad \text{Eq. (3.2)}$$

Fazendo $\phi = 1$, $\Gamma = 0$ e $S = 0$, a equação (4.1) reduz-se à equação da continuidade. Nas equações de quantidade de movimento, ϕ representa os componentes da velocidade, e Γ a função viscosidade, η . O termo de fonte, S , agrupa termos devidos aos efeitos de curvatura e termos oriundos do fato da viscosidade variar com a taxa de deformação.

O acoplamento pressão velocidade foi feito com o algoritmo SIMPLEC (Patankar, 1980). O sistema algébrico resultante foi resolvido com o algoritmo TDMA (Tri-Diagonal Matrix Algorithm) linha-por-linha (Patankar, 1980) e com o algoritmo de correção por blocos (Settari e Aziz, 1973) para acelerar o processo de convergência.

O programa Fluent já tem modeladas algumas funções viscosidade não Newtonianas, como as equações Power-Law, Carreau e Herschel-Bulkley. Porém, para a utilização do modelo de média geométrica ponderada considerado para a função viscosidade neste trabalho, foi necessário utilizar um dos recursos adicionais do programa Fluent, as funções a serem definidas pelo usuário (UDF – User Defined Function), que são programadas pelo usuário utilizando as variáveis calculadas e definidas no programa, e então acopladas ao programa principal. Estas funções permitem que o usuário inclua determinadas características específicas de seu problema. Em particular, utilizou-se a sub rotina para cálculo de viscosidade. O Programa permite alterar a viscosidade do fluido ao final de cada iteração, que depende da tensão cisalhante e do classificador R. A tensão cisalhante é fornecida pelo programa mas o classificador R teve que ser calculado em outra subrotina. Assim, foi feita uma subrotina (Apêndice 1) para o cálculo da viscosidade ao longo do processo iterativo de solução das equações de conservação. O parâmetro classificador de escoamento R foi calculado em outra subrotina. Uma descrição mais detalhada destas subrotinas é apresentada no decorrer deste capítulo. Toda a programação foi feita em linguagem C.

3.2 Definição da Malha

Foi utilizada uma malha estruturada, uniforme na direção axial. De forma a definir uma malha que pudesse reproduzir resultados confiáveis com esforço

computacional dentro do viável, foi realizado um teste de malha para definir a malha utilizada neste trabalho.

Os resultados das velocidades médias nas áreas menor e maior foram avaliados para sete malhas diferentes, para uma geometria específica e um tipo de fluido. Os raios R_i e R_0 foram 0.025 m e 0.075 m respectivamente. As viscosidades foram consideradas constantes e iguais a 0.6 e 8 cp para a viscosidade cisalhante e extensional. O erro dos valores encontrados com a pior malha (20x5, 20 volumes na direção axial e 5 na direção radial) em relação aos obtidos com a malha mais refinada (140x35) estavam abaixo dos 5%. Por isso, foi escolhida a malha intermediária (100x25), o qual deu uma diferença de menos de 0.3% em relação à malha mais refinada.

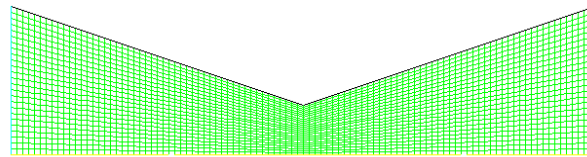


Fig. 3.2 – Malha típica.

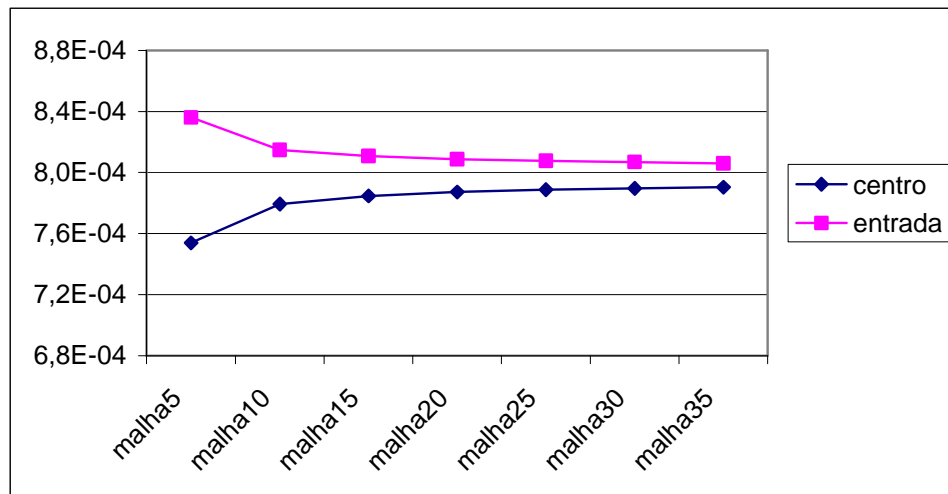


Fig. 3.3 – Gráfico da integral de velocidade.

3.3 Funções definidas pelo usuário (UDF)

Para modelar o comportamento reológico do fluido, em função do classificador de velocidade R , duas funções foram escritas na linguagem de programação C e acopladas ao programa Fluent.

A primeira função definida pelo usuário (UDF – User Defined Function) escrita foi a função para cálculo do classificador R .

Descrição detalhada sobre a implementação numérica da equação constitutiva pode ser encontrada no trabalho publicado por Souza Mendes, Thompson, Carvalho e Naccache (2000) intitulado “*Detailed Derivations for Implementing a Flow-Type-Base Constitutive Equation for Plane and Axisymmetric Flows*”.

Outra função definida pelo usuário utilizada neste trabalho foi a função de cálculo da viscosidade em função do parâmetro R e da taxa de deformação.

Como mencionado anteriormente, a função viscosidade utilizada para modelar o comportamento reológico do fluido neste trabalho é definida da seguinte forma:

$$\eta(\dot{\gamma}, R) = \eta_s (\dot{\gamma})^R \cdot \eta_u (\dot{\epsilon})^{1-R} \quad \text{Eq. (3.3)}$$

Como o foco do nosso estudo é a influência da viscosidade extensional no escoamento, decidimos modelar o fluido de Boger, que considera a viscosidade de cisalhamento constante, e a viscosidade de extensão variando em função da taxa de deformação.

A UDF para cálculo do classificador de escoamento R e da viscosidade extensional utilizada neste trabalho encontra-se no Apêndice 1.

3.4 Acoplamento das UDFs ao Fluent.

Depois que as funções foram escritas elas foram compiladas no Fluent, onde as novas variáveis foram criadas para auxiliar no cálculo da viscosidade.

Como a função para o cálculo do classificador R depende do campo de velocidades, e o campo de velocidades depende da viscosidade do fluido, foi adotado um critério para implementação numérica do problema.

Primeiramente, foi gerado um campo de velocidades considerando fluido Newtoniano. A partir deste campo de velocidades o classificador R foi calculado para todo o domínio. O campo do classificador R inicial é utilizado para cálculo da viscosidade, que é utilizada na primeira iteração. Diversas sub-iterações são realizadas para encontrar uma solução para o campo de viscosidades calculado. Uma vez encontrada uma solução parcial para o campo de velocidades, o classificador é re-calculado gerando um novo campo de viscosidade, que é utilizado na segunda iteração. Normalmente, a partir da terceira iteração os resultados apresentam uma convergência aceitável e o processo iterativo pode ser interrompido.