Implementação do ESSPP no ALG

Equações principais:

Continuidade:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho y) = 0$$

Momentum:

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p - \tau_{xx}) + \frac{\partial}{\partial y}(\rho u v - \tau_{xy}) = 0$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v - \tau_{xy}) + \frac{\partial}{\partial y}(\rho v^2 + p - \tau_{yy}) = 0$$

Energia:

$$\frac{\partial}{\partial t}(E_t) + \frac{\partial}{\partial x}\left[(E_t + p)u + q_x - u\tau_{xx} - v\tau_{xy}\right] + \frac{\partial}{\partial y}\left[(E_t + p)v + q_y - u\tau_{yx} - v\tau_{yy}\right] = 0$$

Equações complementares:

Energias:

$$E_t = \rho\left(e+\frac{V^2}{2}\right)$$

Fluxos de calor:

$$q_x = -k\frac{\partial T}{\partial x} \qquad q_y = -k\frac{\partial T}{\partial y}$$

Tensões:

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \qquad \tau_{xx} = \lambda(\nabla\cdot\boldsymbol{V})+2\mu\frac{\partial u}{\partial x} \qquad \tau_{yy} = \lambda(\nabla\cdot\boldsymbol{V})+2\mu\frac{\partial v}{\partial y}$$

Onde uma boa aproximação de lambda, para gases (de acordo com Schlichting, 1979), permite $\lambda = -\frac{2}{3}\mu$

Equações de estado e demais:

$$p = \rho R T \qquad T = \frac{e}{c_v} \qquad k = \frac{\mu c_p}{\boldsymbol{Pr}} \qquad \mu = \mu_0\left(\frac{T}{T_0}\right)^{\frac{3}{2}}\left(\frac{T_0+110}{T+110}\right)$$

Algoritmo principal da resolução numérica

Inicia malha, condições iniciais e condições de contorno

Para cada instante de tempo Δt:

·Valores das variáveis no tempo anterior = valores das variáveis atuais

Enquanto solução não converge de acordo
com uma certa margem de erro (iteração de Picard):

·Valores das variáveis na iteração anterior = valores das variáveis atuais

·Resolver densidade
·Resolver velocidades horizontal e vertical
·Resolver energia
·Resolver temperatura
·Resolver equações de estado
(pressão, viscosidade e condutividade térmica)

Fim Enquanto

Fim Para

Algoritmo para resolver densidades

Resolver densidade()

·Atualiza todas as variáveis, cada célula de acordo com seus vizinhos

Para cada célula da malha:

·Computa propriedades dos vizinhos

·Calcula derivadas parciais por diferenças finitas

·Calcula densidade explicitamente em relação a uma iteração de Picard e implicitamente em relação ao tempo

Fim Para

Fim densidade()

Algoritmo para resolver velocidades

Resolver velocidades()

·Atualiza todas as variáveis, cada célula de acordo com seus vizinhos

Para cada célula da malha:

· Computar tensões e suas derivadas parciais

(?)

Fim Para

Fim velocidades()

Algoritmo para resolver energia

Resolver energia()

·Atualiza todas as variáveis, cada célula de acordo com seus vizinhos

Para cada célula da malha:

(?)

Fim Para

Fim energia()

Algoritmo para resolver temperatura

Resolver energia()

·Atualiza todas as variáveis, cada célula de acordo com seus vizinhos

Para cada célula da malha:

·Computar c_v

(?)

Fim Para

Fim energia()