

Relatório EP2 - PMR3401

Rafael Sobral Augusto 10337193

Victor Gustavo Moura Santos 8585623

2020

Conteúdo

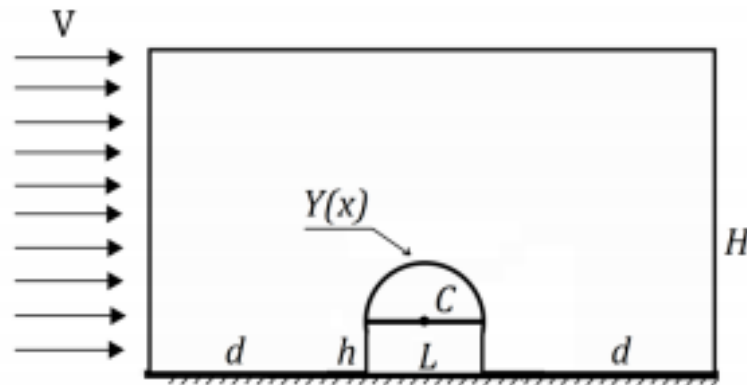
1	Introdução	2
2	Modelagem e Resultados - Parte 1	4
2.1	Corrente Ψ do escoamento	5
2.1.1	Implementação	5
2.1.2	Resultados obtidos	6
2.2	Velocidade do escoamento	9
2.2.1	Implementação	9
2.2.2	Resultados obtidos	10
2.3	Variação de pressão no domínio e no telhado	14
2.3.1	Implementação	14
2.3.2	Resultados obtidos	15
2.4	Força resultante no telhado	23
2.4.1	Implementação	23
2.4.2	Resultados obtidos	23
3	Modelagem e Resultados - Parte 2	26
3.1	Distribuição de temperatura do ar	26
3.1.1	Implementação	26
3.1.2	Resultados obtidos	28
3.2	Taxa de calor retirada do prédio	31
3.2.1	Implementação	31
3.2.2	Resultados obtidos	32
3.3	Conclusões	33
A	Função implementada para o Item A da Parte 1 - <i>ItemA.m</i>	34
B	Função implementada para o Item B da Parte 1 - <i>ItemB.m</i>	36
C	Função implementada para o Item C da Parte 1 - <i>ItemC.m</i>	38
D	Função implementada para o Item D da Parte 1 - <i>ItemD.m</i>	39
E	Função implementada para o Item E da Parte 1 - <i>ItemE.m</i>	40

F	Função implementada para o Item A da Parte 2 - <i>Item2a.m</i>	41
G	Função implementada para o Item B da Parte 2 - <i>Item2b.m</i>	44
H	<i>Script</i> geral responsável por chamar as funções e plotar resultados - <i>EP2Main.m</i>	46

1 Introdução

Este Exercício-Programa tem como objetivo realizar uma análise, por meio do aprendizado acerca do *Método das Diferenças Finitas*, do esforço gerado por um vento no telhado de uma construção e como podemos utilizar os métodos numéricos aprendidos para aproximar a dinâmica de diversos fenômenos físicos representados por equações diferenciais. Para isso, será analisado o problema no domínio definido abaixo:

Figura 1: Domínio considerado para análise do problema.



Fonte: Enunciado do Exercício-Programa.

Onde as dimensões da imagem são:

- $h=3\text{m}$
- $d=15\text{m}$
- $L=6\text{m}$
- $H=24\text{m}$
- $V=100\text{ km/h}$

Estamos considerando a presença de uma única construção que é afetada pelo vento, o que seria o caso para a primeira construção frente ao vento em uma série de silos ou hangares lado a lado. No caso desse problema em especial, para sua modelagem, considerou-se um modelo bidimensional e o escoamento sendo irrotacional, invíscido, em regime permanente e de ar como fluido compressível, como descrito no enunciado, podemos utilizar a equação abaixo para descrever a dinâmica do fluido:

$$(1 - M_x^2) \frac{\partial^2 \Phi}{\partial x^2} + (1 - M_y^2) \frac{\partial^2 \Phi}{\partial y^2} - 2M_x M_y \frac{\partial^2 \Phi}{\partial x \partial y} = 0 \quad (1)$$

Onde M_x e M_y são as componentes do número de Mach. Para o escoamento que estamos lidando, o número de Mach pode ser considerado desprezível, e dessa forma teremos a nova equação

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \quad (2)$$

Onde Φ é o **potencial de velocidade**. Com isso, seu gradiente será a velocidade do fluido:

$$\nabla \Phi = \mathbf{u} = u\vec{i} + v\vec{j} \quad (3)$$

Para tornar a imposição de condições de contorno mais simples, o problema foi implementado utilizando-se a **função de corrente** Ψ , onde:

$$\frac{\partial \Psi}{\partial x} = -\frac{\partial \Phi}{\partial y} = -v \quad (4)$$

$$\frac{\partial \Psi}{\partial y} = \frac{\partial \Phi}{\partial x} = u \quad (5)$$

E assim, pode-se encontrar a solução da equação

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = 0 \quad (6)$$

Para encontrar as condições que delimitam a velocidade e a pressão ao longo do domínio e principalmente, na borda do telhado da construção analisada. Já quanto à distribuição de temperaturas, podemos utilizar da equação:

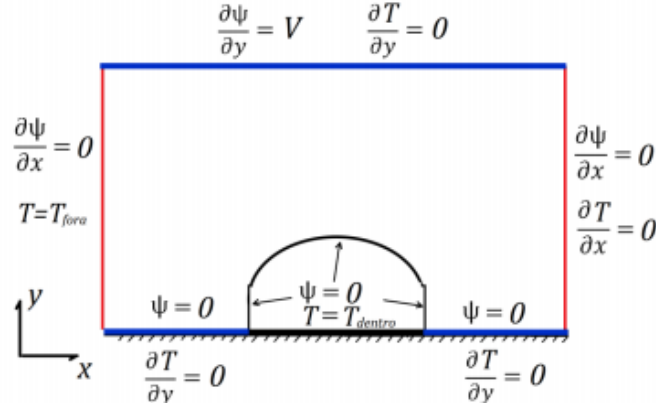
$$k\nabla^2 T - \rho c_p \mathbf{u} \nabla T = 0 \quad (7)$$

E com isso, estimar qual a temperatura ao longo do domínio e qual a transferência de calor entre o prédio e o ar que flui por cima do mesmo. As variáveis mencionadas na equação acima são propriedades do ar, e são:

- $\rho = 1.25 \text{ kg/m}^3$
- $\gamma_{ar} = 1,4$
- $k_{ar} = 0.026 \text{ W/m/K}$
- $c_{par} = 1002 \text{ J/kg/K}$

Para solucionar ambas as equações propostas, considerando que $T_{dentro} = 40^\circ\text{C}$ e $T_{fora} = 20^\circ\text{C}$, as condições de contorno impostas são:

Figura 2: Condições de contorno.



Fonte: Enunciado do Exercício-Programa.

A única consideração que deve ser feita é que, para facilitar a resolução do problema frente a indexação do software *Matlab*, foi adotado um sistema de coordenadas diferente do sugerido, com o eixo y apontando **para baixo**. Isso gera uma pequena alteração nas condições de contorno, pois teremos que, no topo do domínio,

$$\frac{\partial \Psi}{\partial y} = -V \quad (8)$$

Ademais, as condições de contorno adotadas são as mesmas e pode-se aplicar métodos numéricos para aproximar a dinâmica do problema utilizando uma malha quadrada com $\Delta x = \Delta y$.

2 Modelagem e Resultados - Parte 1

A **Parte 1** do Exercício-Programa consiste em resolver, por meio do *Método das Diferenças Finitas*, a equação diferencial

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = 0 \quad (9)$$

E com os resultados encontrados, como parte do pós-processamento do problema, calcular os valores da velocidade de escoamento e da pressão ao longo do domínio, junto com a força vertical resultante sobre o telhado da construção analisada.

Para esse problema, temos condições de contorno **simétricas** ao longo do domínio e, com isso, podemos lidar com somente metade da malha, uma vez que o resultado encontrado será simétrico, reduzindo também pela metade o tempo necessário para o processamento das aproximações e a quantidade de casos que devem ser considerados. Dessa forma, todos os itens na **Parte 1** foram implementados considerando somente a metade da esquerda da malha e os valores foram ajustados para a segunda metade por causa da simetria do problema.

2.1 Corrente Ψ do escoamento

2.1.1 Implementação

Primeiramente, foi implementada a lógica de programação para resolver o principal problema, a equação diferencial para a função de corrente Ψ . Para os pontos internos da malha, longe dos pontos nos quais é necessária uma aproximação diferente devido à natureza das condições de contorno, foi adotada a aproximação usual do *Método das Diferenças Finitas* - fazendo uso da segunda diferença central - para as derivadas parciais:

$$\frac{\partial^2 \Psi}{\partial x^2} = \frac{\Psi_{i-1,j} - 2\Psi_{i,j} + \Psi_{i+1,j}}{\Delta x^2} \quad (10)$$

$$\frac{\partial^2 \Psi}{\partial y^2} = \frac{\Psi_{i,j-1} - 2\Psi_{i,j} + \Psi_{i,j+1}}{\Delta y^2} \quad (11)$$

E assim, a equação geral utilizada para todos os pontos nesse interior foi

$$\Psi_{i,j} = \frac{\Psi_{i+1,j} + \Psi_{i-1,j} + \Psi_{i,j+1} + \Psi_{i,j-1}}{4} \quad (12)$$

Por ser um problema em **regime permanente**, utilizou-se o **Método da Sobre-relaxação**, com $\lambda = 1.85$ para aproximar os valores encontrados para $\Psi_{i,j}$. O critério de parada utilizado foi a diferença máxima entre duas iterações consecutivas de 0.01, como sugerido pelo enunciado.

Para encontrar aproximações para a função corrente Ψ nos pontos nas bordas do domínio, foi necessário fazer uso das condições de contorno fornecidas. Assim, como na borda esquerda a condição de contorno é de $\frac{\partial \Psi}{\partial x} = 0$, a aproximação para a segunda derivada parcial em relação a x , expandindo o *polinômio de Taylor*, foi

$$\frac{\partial^2 \Psi}{\partial x^2} = \frac{2}{\Delta x^2} (\Psi_{i+1,j} - \Psi_{i,j}) \quad (13)$$

E por sua vez, como no topo do domínio a condição de contorno a ser utilizada é $\frac{\partial \Psi}{\partial y} = -V$ e estamos iterando do topo do domínio até o chão, a aproximação da segunda derivada parcial foi

$$\frac{\partial^2 \Psi}{\partial y^2} = \frac{2}{\Delta y^2} \left(\Psi_{i,j+1} - \Psi_{i,j} - \frac{\partial \Psi}{\partial y} \right) = \frac{2}{\Delta y^2} (\Psi_{i,j+1} - \Psi_{i,j} + V) \quad (14)$$

Por fim, o único caso em que devemos analisar as condições de contorno é com relação ao telhado da construção - uma vez que, para realizar a discretização da malha, foram escolhidos somente passos que dariam uma distância exata de Δx para a parede lateral da construção.

Como estamos lidando somente com o lado esquerdo do contorno irregular (uma vez que é um problema simétrico), utilizou-se as seguintes fórmulas para obter o fator a que multiplica Δx ou b que multiplica Δy , para representar a verdadeira distância entre o ponto da malha e o telhado circular da construção:

$$a = \frac{d + \frac{L}{2} - x_{\text{ponto}} - \sqrt{(\frac{L}{2})^2 - y_{\text{ponto}}^2}}{\Delta x} \quad (15)$$

$$b = \frac{y_{\text{ponto}} - \sqrt{\left(\frac{L}{2}\right)^2 - \left(d + \frac{L}{2} - x_{\text{ponto}}\right)^2}}{\Delta y} \quad (16)$$

Onde $(x_{\text{ponto}}, y_{\text{ponto}})$ são as coordenadas dos pontos da malha quadrada a uma distância do contorno irregular menor que Δx ou Δy . Para cada ponto dentro das coordenadas [15, 18]x[3, 6], foi analisado se a distância entre eles e o contorno do círculo era menor que Δx ou Δy . Caso fosse menor que somente um dos dois, era calculada a segunda derivada parcial com relação a somente essa coordenada pelo *Polinômio de Taylor*, e utilizada na equação diferencial. Dessa forma, as aproximações utilizadas foram:

$$\frac{\partial^2 \Psi}{\partial x^2} = \frac{2}{\Delta x^2} \frac{\Psi_{i+1,j} + a\Psi_{i-1,j} - (1+a)\Psi_{i,j}}{a + a^2} \quad (17)$$

$$\frac{\partial^2 \Psi}{\partial y^2} = \frac{2}{\Delta y^2} \frac{\Psi_{i,j+1} + b\Psi_{i,j-1} - (1+b)\Psi_{i,j}}{b + b^2} \quad (18)$$

Assim, pôde-se isolar $\Psi_{i,j}$ na equação diferencial fornecida para obter as aproximações sucessivas do mesmo em todos os pontos (x_1, y_j) da malha quadrada do enunciado pela equação diferencial fornecida, até que fosse atingido o critério de parada. A função *ItemA.m*, presente nos **Apêndices** deste relatório, é a responsável por fazer com que todas essas aproximações sejam feitas, de acordo com a posição do ponto na malha quadrada. Já quanto à definição das variáveis utilizadas e aos trechos de código responsáveis por realizar a exibição dos gráficos demonstrados, ambos estão presentes no arquivo *EP2Main.m*.

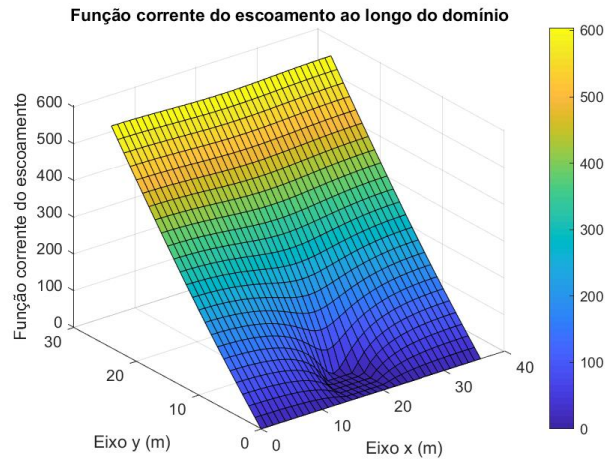
2.1.2 Resultados obtidos

Para realizar os testes deste e de todos os itens posteriores, foram realizados testes com os passos $\frac{h}{3}, \frac{h}{6}, \frac{h}{8}, \frac{h}{10}, \frac{h}{12}$ e $\frac{h}{15}$. A partir desses 6 passos distintos, serão mostrados os gráficos obtidos - em cada item para um conjunto de passos diferentes. Para passos muito pequenos, alguns gráficos ficaram muito poluídos devido às linhas de separação entre os elementos.

Para exibir os resultados obtidos para a função de corrente Ψ , foram usados os comandos *surf* e *contour* do *Matlab*. O primeiro é útil para observar o módulo dos valores dessa função e o segundo, mais relevante para a análise de Ψ , uma vez que este permite com que sejam visualizadas as linhas de corrente das partículas de ar.

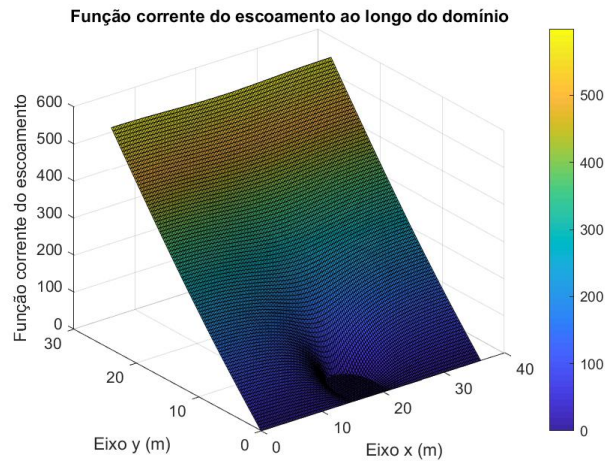
Abaixo, podemos visualizar os resultados obtidos pelo comando *surf* para Ψ para os passos $\Delta x = \frac{h}{3} = 1$ e $\Delta x = \frac{h}{12} = 0.25$:

Figura 3: Função corrente Ψ do escoamento para $\Delta x = 1$.



Fonte: Autoria própria.

Figura 4: Função corrente Ψ do escoamento para $\Delta x = 0.25$.

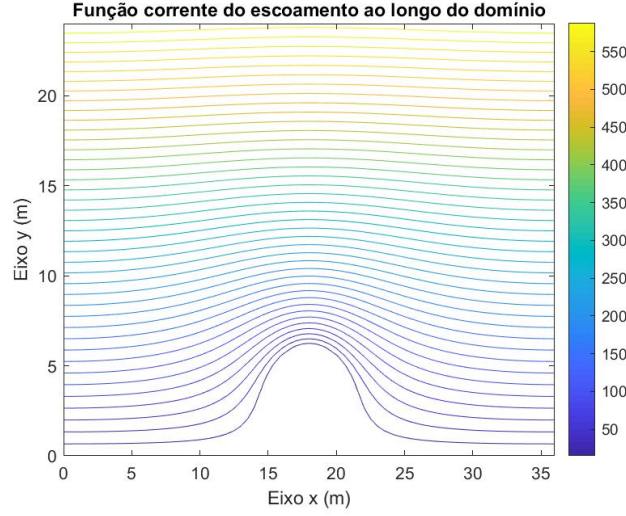


Fonte: Autoria própria.

Dessa forma, podemos ver que as linhas de corrente Ψ são pouco distorcidas para grandes alturas, ou seja, as partículas se deslocam quase sem alteração na direção de seu deslocamento, e possuem grande alteração para alturas mais próximas a da construção, uma vez que devem desviar da mesma. Os valores não são alterados pela diminuição do passo, porém a malha fica mais detalhada.

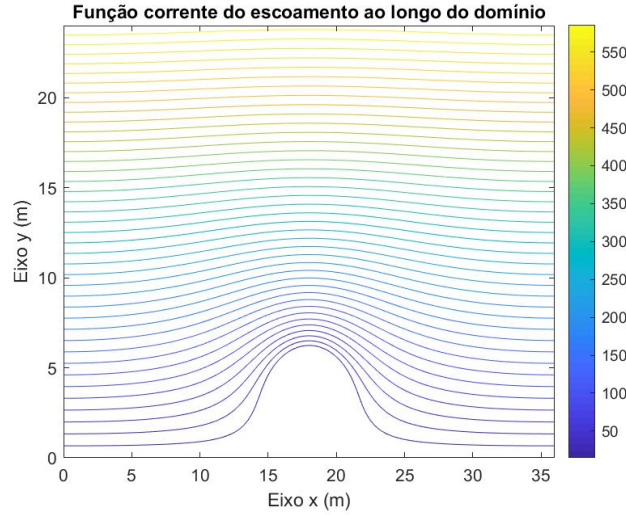
Já com o comando *contour*, com 40 linhas de corrente, pudemos visualizar de forma mais clara esse fenômeno, para os passos $\Delta x = \frac{h}{6} = 0.5$ e $\Delta x = \frac{h}{10} = 0.3$

Figura 5: Função corrente Ψ do escoamento para $\Delta x = 0.5$.



Fonte: Autoria própria.

Figura 6: Função corrente Ψ do escoamento para $\Delta x = 0.3$.



Fonte: Autoria própria.

Com esses gráficos, podemos ver mais claramente como há um grande distúrbio nas linhas de corrente por causa do silo presente no meio do caminho, deslocando de forma abrupta todas as linhas de corrente em uma altura inferior a 10 metros e de forma mais suave as linhas de corrente em alturas superiores, sendo praticamente inalteradas no limite do domínio.

Isso mostra que há uma massa de ar muito maior deslocando-se por uma área menor sobre a construção, o que irá alterar a velocidade e a pressão do ar nesses pontos, como será analisado nos itens posteriores.

A mudança dos passos não provocou grande alteração nas linhas de corrente, estas mantiveram-se praticamente as mesmas independente do passo utilizado. Entretanto, houve uma grande alteração na quantidade de iterações necessárias para a convergência. Para os 6 passos testados, o número de iterações necessário para a convergência para cada um deles foi:

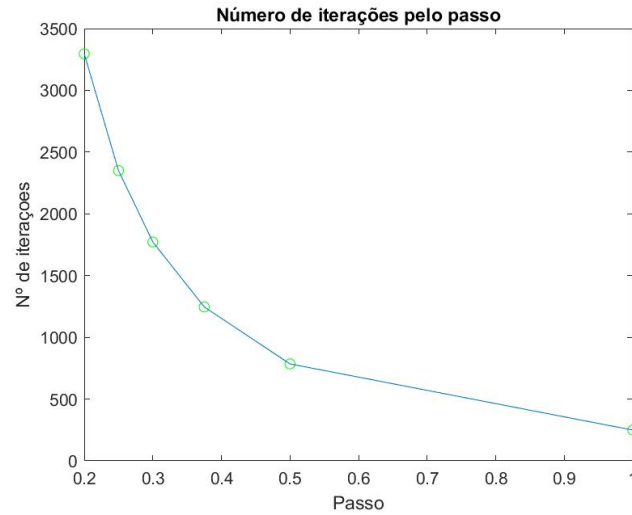
Tabela 1: Tabela que relaciona o passo Δx com o número de iterações necessárias para convergência.

$\Delta x = \Delta y$	Iterações
$\frac{h}{3} = 1$	249
$\frac{h}{6} = 0.5$	785
$\frac{h}{8} = 0.375$	1246
$\frac{h}{10} = 0.3$	1772
$\frac{h}{12} = 0.25$	2350
$\frac{h}{15} = 0.2$	3296

Fonte: Autoria própria

Gerando um gráfico das iterações em função do passo, podemos ver como a diminuição do passo resulta em um crescimento exponencial do número de iterações necessárias para a convergência:

Figura 7: Número de iterações em função do passo Δx .



Fonte: Autoria própria.

2.2 Velocidade do escoamento

2.2.1 Implementação

Já para calcular a velocidade do escoamento do ar ao longo de todo o domínio, foram utilizados os valores já calculados para Ψ , utilizando a relação mencionada anteriormente, de que

$$\mathbf{u} = u\vec{i} + v\vec{j} \quad (19)$$

$$u = \frac{\partial \Psi}{\partial y} \quad (20)$$

$$v = -\frac{\partial \Psi}{\partial x} \quad (21)$$

Entretanto, como na nossa implementação adotamos uma orientação para o eixo y **oposta** à do enunciado, teremos que na verdade, o cálculo da nossa velocidade foi implementado como

$$u = -\frac{\partial \Psi}{\partial y} \quad (22)$$

$$v = \frac{\partial \Psi}{\partial x} \quad (23)$$

Uma vez que essa inversão de sinal proporciona tanto uma inversão no sinal de v , por estar na direção do eixo y , quanto no sinal de $\frac{\partial \Psi}{\partial y}$.

A implementação foi feita inteiramente como um *pós-processamento* após o cálculo de Ψ pelo *Método das Diferenças Finitas*. Utilizando os valores encontrados por essa aproximação, pudemos aproximar os valores das derivadas pela **1ª diferença central**, nos casos dos pontos interiores da malha:

$$\frac{\partial \Psi}{\partial x} = \frac{\Psi_{i+1,j} - \Psi_{i-1,j}}{2\Delta x} \quad (24)$$

$$\frac{\partial \Psi}{\partial y} = \frac{\Psi_{i,j+1} - \Psi_{i,j-1}}{2\Delta y} \quad (25)$$

Já para os valores próximos ao contorno irregular, novamente foi necessário utilizar de *expansões de Taylor*. Para isso, foi feita a expansão até a derivada de segunda ordem de cada qual das variáveis - tanto para os pontos anteriores quanto posteriores das malhas - e eliminou-se o termo da segunda derivada, para posteriormente se isolar o termo da primeira derivada desejado. Assim, as equações para aproximar as primeiras derivadas desejadas - do lado direito do telhado da construção - foram

$$\frac{\partial \Psi}{\partial x} = \frac{\Psi_{i+1,j} - a^2\Psi(i-1,j) - (1-a^2)\Psi(i,j)}{(a+a^2)\Delta x} \quad (26)$$

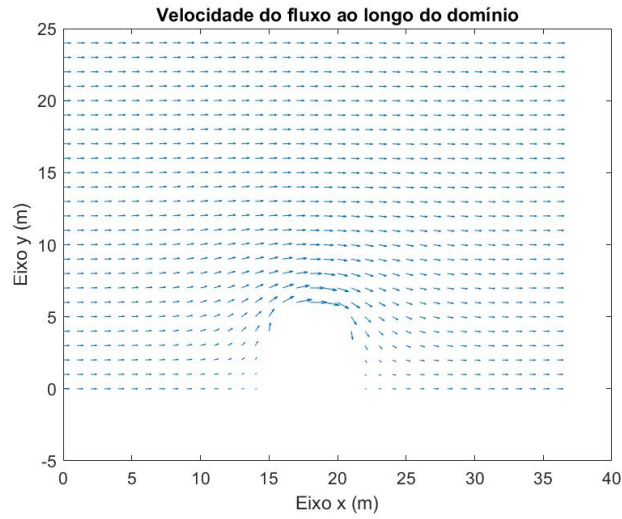
$$\frac{\partial \Psi}{\partial y} = \frac{\Psi_{i,j+1} - b^2\Psi(i,j-1) - (1-b^2)\Psi(i,j)}{(b+b^2)\Delta y} \quad (27)$$

Com isso, pudemos calcular as derivadas parciais de primeira ordem da função corrente Ψ em todo o domínio e, conseqüentemente, obter a velocidade do ar no mesmo. A implementação completa foi feita na função *ItemB.m* presente nos **Apêndices** deste mesmo relatório.

2.2.2 Resultados obtidos

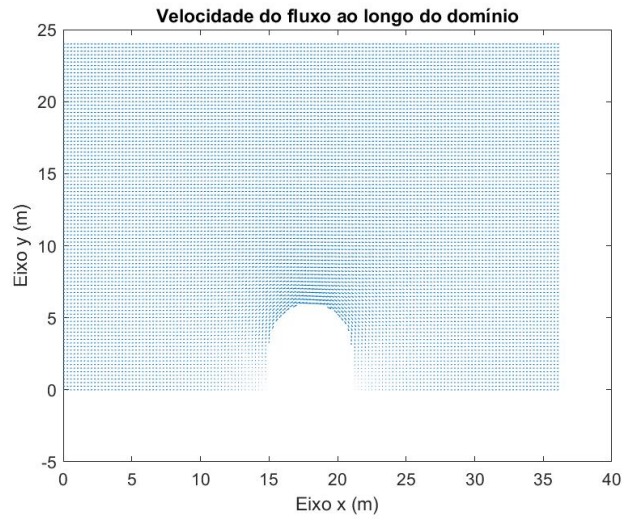
Assim como no **Item A** e em todos os itens posteriores, foram realizados testes com os passos $\frac{h}{3}, \frac{h}{6}, \frac{h}{8}, \frac{h}{10}, \frac{h}{12}$ e $\frac{h}{15}$. Com isso, foi possível obter gráficos que representam a distribuição do vetor da velocidade ao longo do domínio, utilizando o comando *quiver* do *Matlab*. Todos os gráficos obtidos tinham a mesma forma, com a única diferença da quantidade de vetores representados, conforme aumentava-se a discretização. Abaixo, são exibidos os gráficos para os passos $\Delta x = \frac{h}{3} = 1$ e $\Delta x = \frac{h}{12} = 0.25$:

Figura 8: Vetores da velocidade no domínio para $\Delta x = 1$.



Fonte: Autoria própria.

Figura 9: Vetores da velocidade no domínio para $\Delta x = 0.25$.

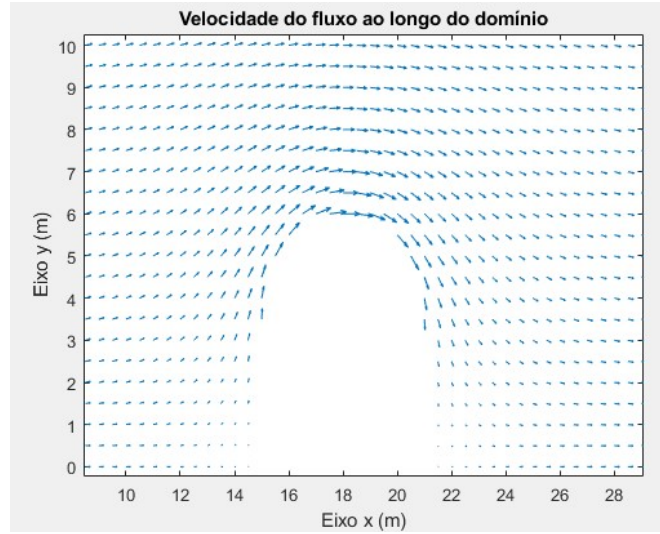


Fonte: Autoria própria.

Como podemos ver pelas imagens, o modelo confirma que o vento ocorre do lado esquerdo do domínio para o lado direito, como era esperado uma vez que fora do domínio é isso que ocorria. Também podemos concluir que, para distâncias grandes o suficiente do silo - alturas maiores que 10 metros - a velocidade do vento permanece praticamente inalterada, demonstrando que a velocidade de movimentação do fluido é perturbada somente em um domínio mais próximo da construção.

Para fazer essa análise mais próxima, abaixo exibimos um gráfico com dimensões mais próximas da do silo para que fosse possível analisar melhor como a velocidade do ar é influenciada por essa interface.

Figura 10: Velocidade do ar próxima da construção para $\Delta x = 0.5$.



Fonte: Autoria própria.

Com essa imagem, podemos confirmar como tanto a direção e o módulo da velocidade são influenciados pela construção. Naturalmente, a velocidade do ar passa a demonstrar uma influência maior de sua componente vertical, uma vez que ele passa a se deslocar para cima para desviar da obstrução em seu caminho. Ao mesmo tempo, esse deslocamento faz com que a mesma massa de ar que estava se deslocando em uma área muito maior tenha que se estreitar para passar por cima do silo.

Esse fenômeno é análogo ao *Efeito Venturi*, o qual explicita que, caso haja o estreitamento de uma seção tubular pela qual um fluido se desloca, sua velocidade deve aumentar na mesma seção. Esse fenômeno é explicado pelo **princípio de continuidade da massa**, pois uma vez que estamos em regime permanente, todo o ar que entra no nosso domínio deverá sair ao fim dele - podemos assumir isso uma vez que em nosso problema, o ar desloca-se somente horizontalmente ao entrar no domínio e as alterações proporcionadas pela construção no fluido no limite do domínio são praticamente imperceptíveis.

Dessa forma, considerando dois trechos distintos do domínio pelo qual o fluido se desloca, 1 e 2, teremos que:

$$\Delta m_1 = \Delta m_2$$

$$\rho_1 A_1 v_1 = \rho_2 A_2 v_2$$

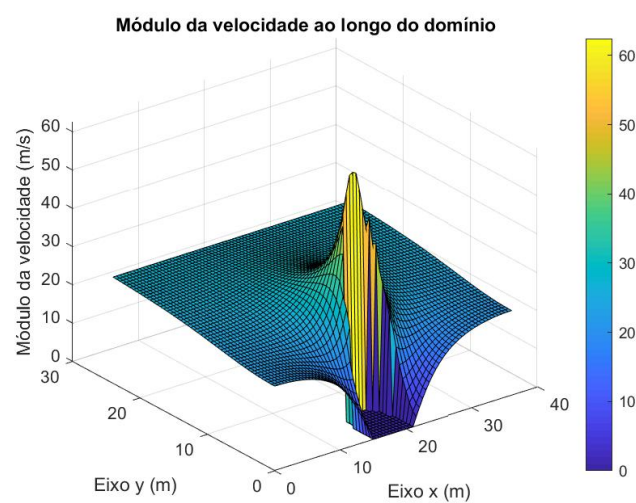
$$\rho_1 = \rho_2 \rightarrow A_1 v_1 = A_2 v_2$$

Ao diminuirmos a área pela qual o ar pode fluir - pela introdução da construção - a sua velocidade naturalmente aumentará, e será maior conforme diminui-se a área. Há muito mais massa de ar fluindo de forma compactada por cima do telhado - como pudemos observar pelas linhas de corrente do **Item A** - e isso ocorre pela diminuição da área que há para ele fluir. Assim, a velocidade aumentará gradativamente e a **máxima velocidade** que o fluido atinge pode ser observada **no topo** do telhado.

Conforme nos afastamos do telhado, esse efeito perde sua influência pois, para as partículas muito acima do telhado, não haverá praticamente nenhum estreitamento de sua seção e continuarão fluindo naturalmente ao longo do domínio. Isso também está de acordo com as linhas de corrente observadas no **Item A**, uma vez que a influência do prédio é cada vez menor no caminho traçado pelas partículas de ar fluindo pelo domínio.

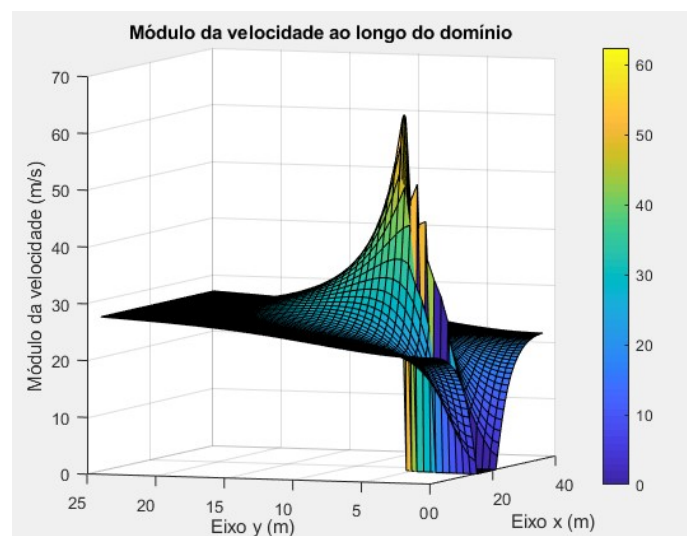
Podemos observar também como, após o silo, a velocidade do ar naturalmente **diminui** conforme a seção se alarga e volta a fluir da mesma maneira que anteriormente. Para analisar de forma mais precisa o módulo da velocidade no domínio, utilizando o comando *surf* do *Matlab*, pudemos observar, para passos de $\Delta x = \frac{h}{6} = 0.5$ e $\Delta x = \frac{h}{10} = 0.3$:

Figura 11: Módulo da velocidade no domínio para $\Delta x = 0.5$.



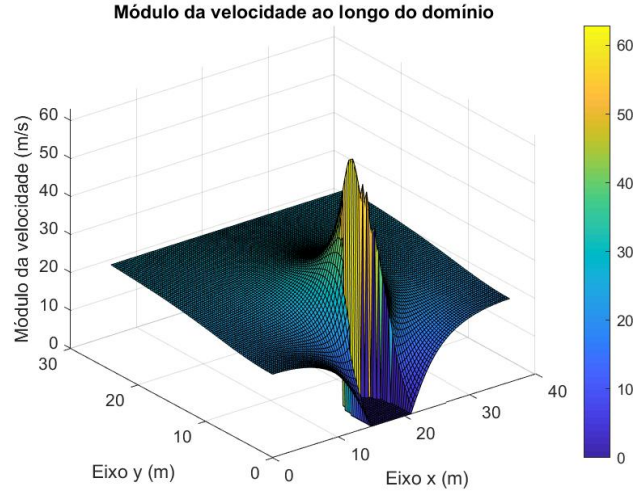
Fonte: Autoria própria.

Figura 12: Módulo da velocidade no domínio para $\Delta x = 0.5$ - Outra visão.



Fonte: Autoria própria.

Figura 13: Módulo da velocidade no domínio para $\Delta x = 0.3$.



Fonte: Autoria própria.

Por esses gráficos, podemos perceber como a velocidade, para uma altura suficientemente grande, se mantém próxima da velocidade $V_{100km/h} = 27.78m/s$ com a qual o fluido entra no domínio. Já próxima do chão há uma pequena diminuição nessa velocidade, uma vez que exatamente no chão a velocidade deve ser zero - devido ao conceito de *camada limite* de Mecânica dos fluidos.

Já próximo do silo, há uma grande alteração no módulo da velocidade, o qual aumenta gradativamente conforme aproxima-se do topo e diminui na mesma escala a partir do mesmo. Além disso, conforme aumenta-se a distância vertical do topo, o módulo da velocidade do fluido passa a ser cada vez menos alterado, comprovando o que foi mencionado anteriormente, até que seja inalterado. Dentro do silo, por sua vez, como não temos nenhuma movimentação de fluido conhecida, a velocidade considerada foi nula.

Apenas para critério de comparação, a velocidade que o vento atinge no domínio, longe do telhado, é comparável com a velocidade que ventos atingem para um furacão *F0* na **Escala Fujita**, escala que vai de 0 a 6 para a medição da gravidade dos tornados, onde 0 é considerado um tornado leve e 6 um tornado surreal. Já no telhado da estrutura, a velocidade atinge cerca de $226km/h$, o que é comparável a um tornado forte, classificado como *F2* na **Escala Fujita**.

Isso nos dá um forte indício que com essa velocidade do vento, há uma grande probabilidade de que caso o telhado não fosse arrancado completamente - uma vez que possui 60 metros de comprimento - com certeza seria gravemente danificado.

2.3 Variação de pressão no domínio e no telhado

2.3.1 Implementação

Para o cálculo da variação da pressão ao longo do domínio inteiro, foi utilizada a equação de Bernoulli fornecida pelo enunciado:

$$p(x, y) - p_{atm} = -\rho \frac{\gamma_{ar} - 1}{\gamma_{ar}} \frac{(V_{fluido}(x, y))^2}{2} \quad (28)$$

Onde

$$V_{fluido}(x, y) = |\mathbf{u}| = \sqrt{u(x, y)^2 + v(x, y)^2} \quad (29)$$

Com isso, pôde-se obter a pressão em todo o domínio. O arquivo *ItemC.m*, contendo o código implementado para esse resultado está presente nos **Apêndices**, como todos os códigos implementados.

Já para encontrar a pressão ao longo do telhado, somente foi necessário fazer uso do vetor *bordairregular*, que salvou os índices de todos os pontos da malha que estavam a uma distância menor que Δx em x e/ou Δy em y do telhado do silo. Com isso, foi possível ordenar esses pontos de acordo com a sua componente x e obter gráficos da evolução da pressão ao longo do telhado.

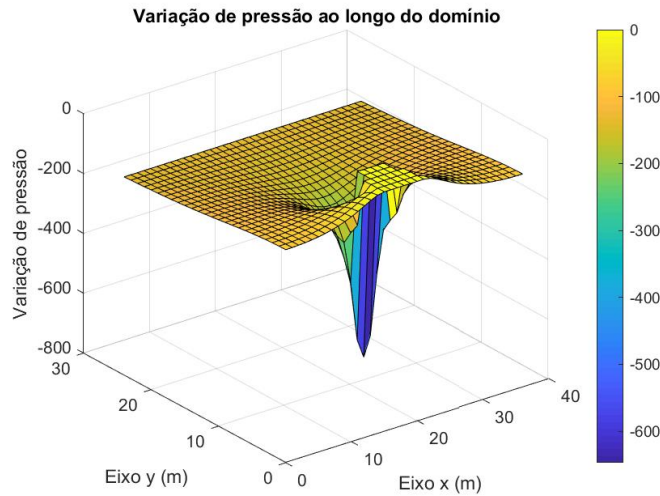
Além disso, foi possível também, fazendo uso do vetor AB - o qual continha os fatores a e b que multiplicados por Δx e Δy davam a real distância entre os pontos da malha e o telhado - aproximar os pontos encontrados por pontos reais do telhado.

Isso foi feito comparando-se os valores de a e b e, utilizando o menor deles (o que indicaria que o ponto está mais próximo do círculo nessa direção), subtraindo-se a distância $a\Delta x$ ou $b\Delta y$, obtendo assim uma projeção de onde no telhado estaria sendo efetivamente aplicada essa pressão. Com certeza não é uma aproximação completamente precisa, principalmente para passos maiores, mas é boa o suficiente para que possamos ter um entendimento maior do problema. O arquivo *ItemD.m* presente em **Apêndices** contém o código responsável pelas implementações mencionadas.

2.3.2 Resultados obtidos

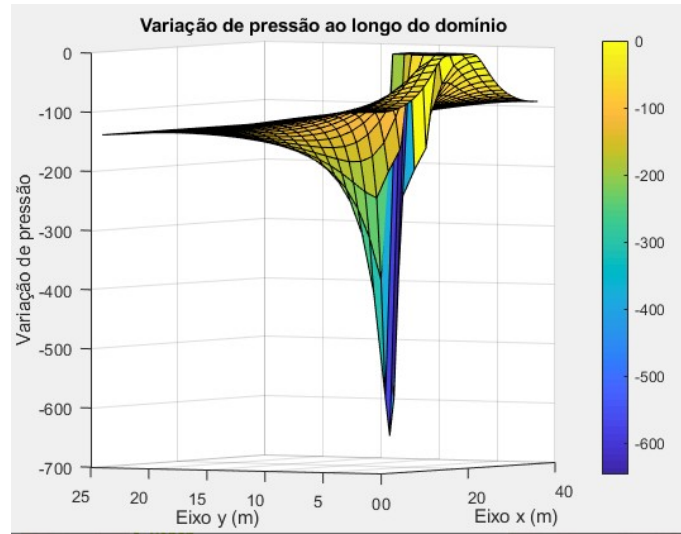
Quanto aos resultados obtidos para a pressão, vamos primeiro analisar o resultado no domínio inteiro e depois refletir sobre a pressão no telhado da estrutura. Dessa forma, utilizando novamente o comando *surf* do *Matlab*, podemos concluir que a pressão se distribui ao longo do domínio, para os passos $\Delta x = 1$ e $\Delta x = 0.2$ da seguinte forma:

Figura 14: Pressão ao longo do domínio para $\Delta x = 1$.



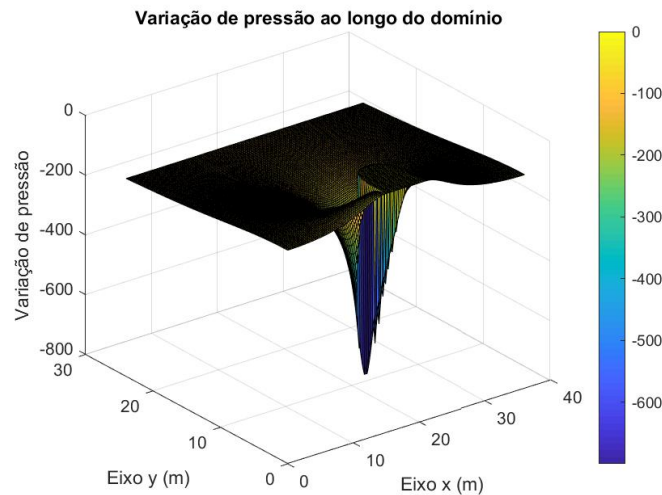
Fonte: Autoria própria.

Figura 15: Pressão ao longo do domínio para $\Delta x = 1$. - Outra visão.



Fonte: Autoria própria.

Figura 16: Pressão ao longo do domínio para $\Delta x = 0.2$.



Fonte: Autoria própria.

Uma vez que a pressão é inversamente proporcional ao módulo da velocidade ao quadrado, esse gráfico condiz com o esperado, uma vez que, nos pontos em que a velocidade atinge seu máximo - em torno do telhado - a pressão atinge seu mínimo. A diferença de pressão $p(x, y) - p_{atm}$ estabiliza-se em torno de $-137Pa$ - ou seja, a pressão é próxima o suficiente da pressão atmosférica, entretanto é um pouco inferior devido à grande velocidade do vento, de $100km/h$.

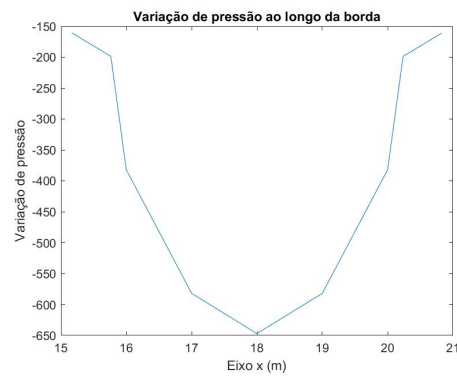
Os resultados encontrados vão completamente de encontro com o esperado, com a pressão aumentando conforme se afasta do silo - novamente, de forma inversamente proporcional à velocidade - até o valor estável numa altura grande o suficiente. Além disso, isso condiz com o observado na vida real, uma vez que o fenômeno da diminuição da pressão em conjunto com o aumento da velocidade é o esperado também pelo *Efeito Venturi* e pela própria conservação de energia,

uma vez que como a energia cinética aumentou, deve haver diminuição na parcela da energia pela qual a pressão é responsável no equacionamento de *Bernoulli*.

Já para o pedido no **Item D**, utilizando os índices armazenados dos pontos que estão em contato com a borda do telhado, pudemos realizar um *plot* simples com os valores da pressão ao longo do telhado em função de sua posição em x e pudemos também fazer um *scatter* no *Matlab*, assim unindo as informações da pressão na borda com as coordenadas x e y conjuntas.

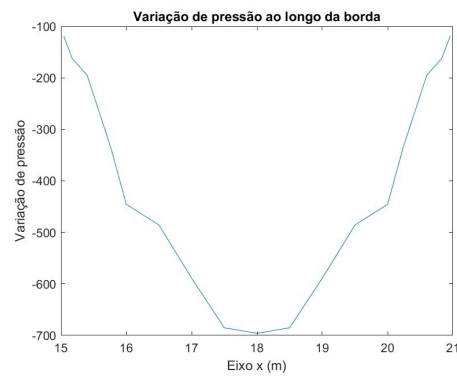
Acreditamos que ambos esses gráficos apresentaram diferenças relevantes que valem serem analisadas com os gráficos de todos os passos. Dessa forma, para o gráfico da pressão em função da posição x , encontramos:

Figura 17: Pressão ao longo do telhado em função de x para $\Delta x = 1$.



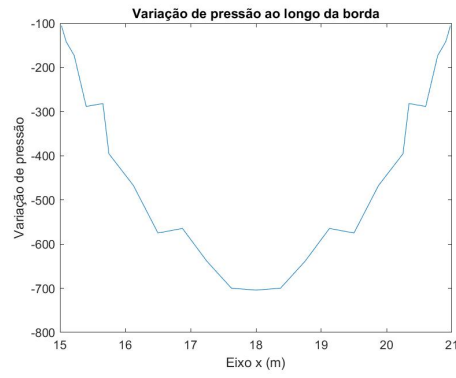
Fonte: Autoria própria.

Figura 18: Pressão ao longo do telhado em função de x para $\Delta x = 0.5$.



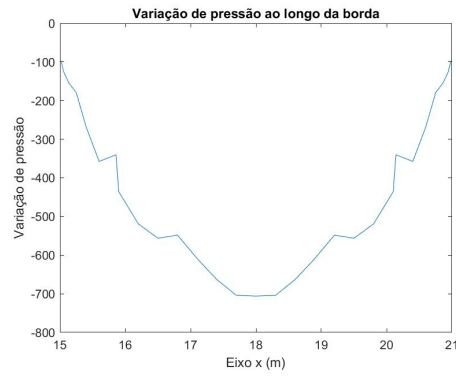
Fonte: Autoria própria.

Figura 19: Pressão ao longo do telhado em função de x para $\Delta x = 0.375$.



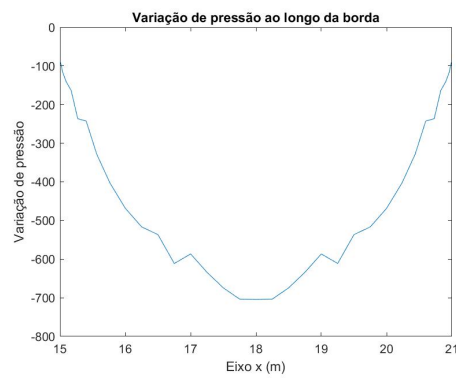
Fonte: Autoria própria.

Figura 20: Pressão ao longo do telhado em função de x para $\Delta x = 0.3$.



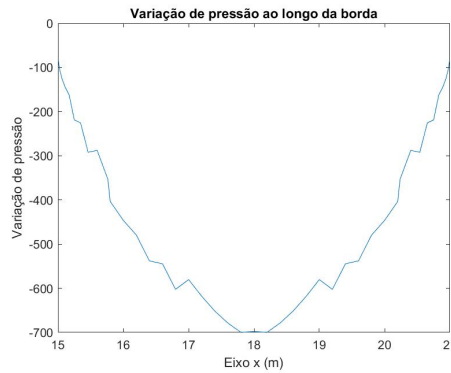
Fonte: Autoria própria.

Figura 21: Pressão ao longo do telhado em função de x para $\Delta x = 0.25$.



Fonte: Autoria própria.

Figura 22: Pressão ao longo do telhado em função de x para $\Delta x = 0.2$.



Fonte: Autoria própria.

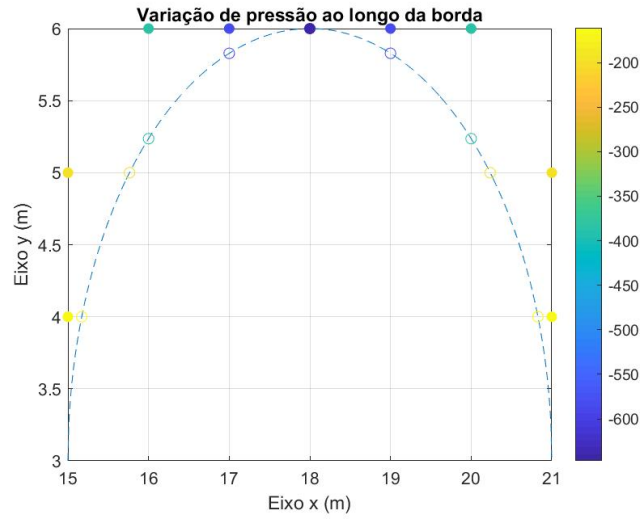
Podemos assim, com esses gráficos, atestar como, além de a pressão também variar de forma simétrica em torno do telhado circular, como ela diminui até o topo do telhado - em $x = 18m$ e então volta a aumentar. Em todos os gráficos, o mínimo da variação da pressão é atingido por volta de $-700Pa$, e a diminuição de pressão é praticamente constante ao longo do telhado.

Há pequenas incoerências como uma pressão um pouco maior em pontos menores de x , mas acreditamos que isso aconteça principalmente devido ao fato de a malha fazer com que a distância em x e y do telhado seja irregular, por ser um contorno curvo, o que faz com que a medição da pressão no telhado não seja completamente precisa. No geral, a distribuição da pressão ao longo do telhado parece corroborar com o que esperado.

Agora, pudemos também fazer um *scatter* dos dados da pressão junto de uma representação do telhado para que pudessemos analisar a que distância essa pressão estava sendo medida em relação ao telhado real. Além disso, utilizamos desse gráfico para plotar os pontos 'aproximados' no telhado onde essas pressões seriam medidas, como foi explicado na parte de *Implementação* acima.

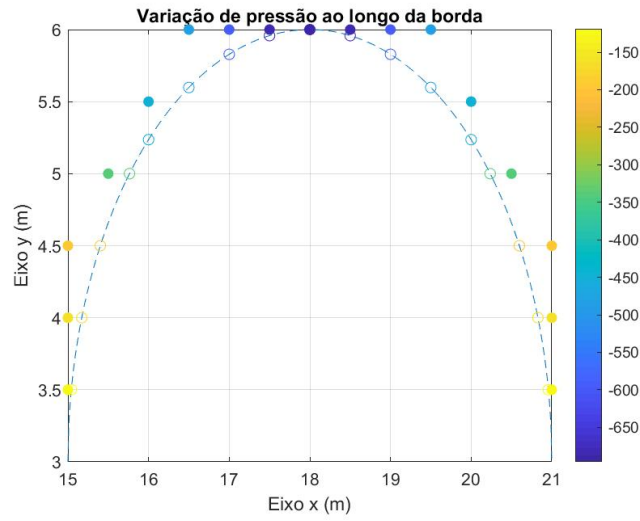
Dessa forma, os pontos coloridos vazados representam pontos hipotéticos, mais próximos de onde teriam sido medidas as pressões nos telhasdos, e os pontos coloridos completamente são os pontos da malha. Já o traçado pontilhado determina o telhado real:

Figura 23: Pressão ao longo do telhado em função de x e y para $\Delta x = 1$.



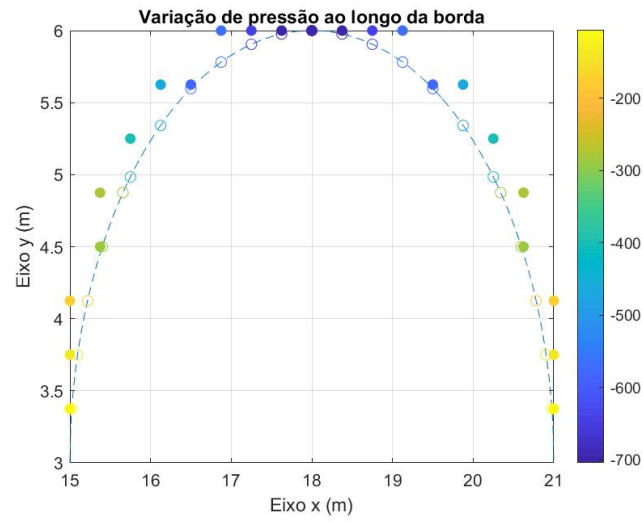
Fonte: Autoria própria.

Figura 24: Pressão ao longo do telhado em função de x e y para $\Delta x = 0.5$.



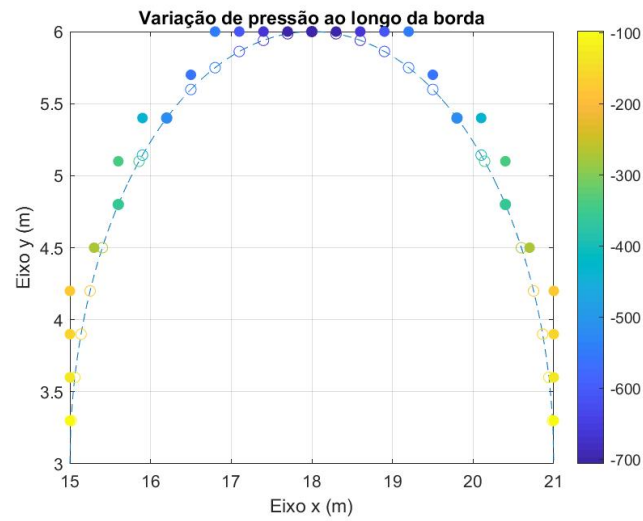
Fonte: Autoria própria.

Figura 25: Pressão ao longo do telhado em função de x e y para $\Delta x = 0.375$.



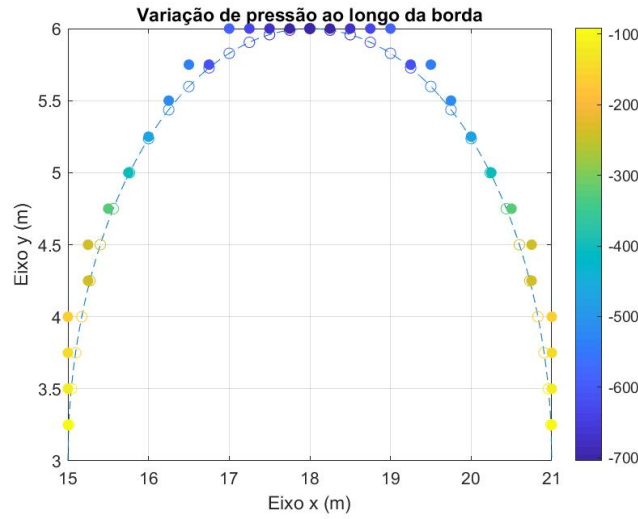
Fonte: Autoria própria.

Figura 26: Pressão ao longo do telhado em função de x e y para $\Delta x = 0.3$.



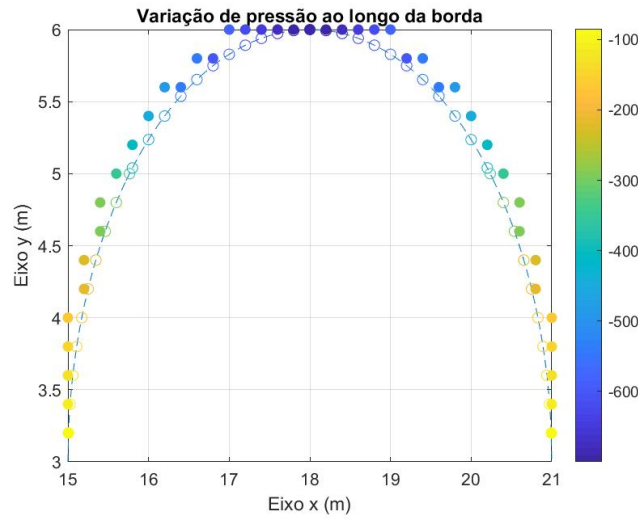
Fonte: Autoria própria.

Figura 27: Pressão ao longo do telhado em função de x e y para $\Delta x = 0.25$.



Fonte: Autoria própria.

Figura 28: Pressão ao longo do telhado em função de x e y para $\Delta x = 0.2$.



Fonte: Autoria própria.

Com o uso desses gráficos, pudemos observar com o o passo influencia na medição mais precisa das pressões e influencia no cálculo da força, uma vez que os vetores normais a superfície serão muito mais próximos da realidade para um passo pequeno como $\Delta x = 0.2$ do que para um passo como $\Delta x = 1$, tanto caso fossem usados os pontos originais - cheios - quanto os pontos ajustados ao contorno do telhado - vazados - que foram usados para calcular os vetores normais para o cálculo da força.

Pudemos observar também com esses gráficos, mais uma vez, como a pressão aumenta ao longo do telhado até o seu topo, e depois volta a diminuir da mesma forma do outro lado conforme o vento possui mais espaço para se distribuir.

2.4 Força resultante no telhado

2.4.1 Implementação

Para implementar as forças resultantes, foram utilizadas as pressões calculadas e os pontos ajustados para caírem sobre o comprimento do telhado circular, como foi mencionado no item acima. Então, são calculados os vetores normais de cada um dos pontos nos quais conhecemos as pressões, que serão os vetores normalizados ortogonais ao vetor que liga o ponto anterior e o ponto posterior ao ponto que está sendo analisado.

Já os pontos médios são calculados utilizando-se dois pontos consecutivos que estão na borda do telhado, para poder identificar onde acaba a ação de uma pressão e começa a ação da próxima. Conforme diminui-se o passo, essa aproximação é mais válida, uma vez que os pontos passam a aproximar cada vez melhor o contorno do telhado - para $\Delta x = \Delta y = 0.5$, isso nos dá um comprimento aproximado do semicírculo que representa o telhado de $9.378m$, enquanto o seu comprimento real é de 9.4248 .

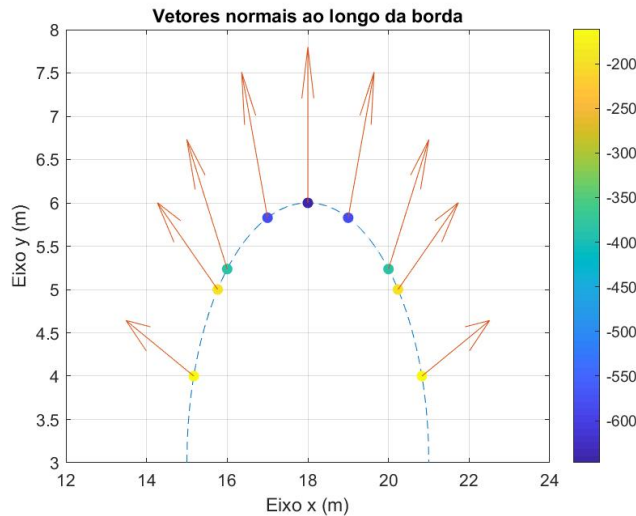
Então, são calculadas as áreas de ação de cada pressão multiplicando-se o comprimento total - 60 metros - do telhado pelo tamanho do lado ligando dois pontos médios consecutivos. Depois, multiplica-se cada área pela pressão que atua naquele retângulo, obtendo o valor da força resultante ali. Entretanto, essa força resultante estará na direção normal ao retângulo utilizado, de forma que para obter a força resultante em y , multiplicamos esse valor pelo componente y do vetor normal e somamos todos esses componentes.

2.4.2 Resultados obtidos

A execução do código da força passou por duas etapas: a primeira, que foi a obtenção dos vetores normais, como descrito no tópico acima, e a obtenção da força resultante em y com esses valores.

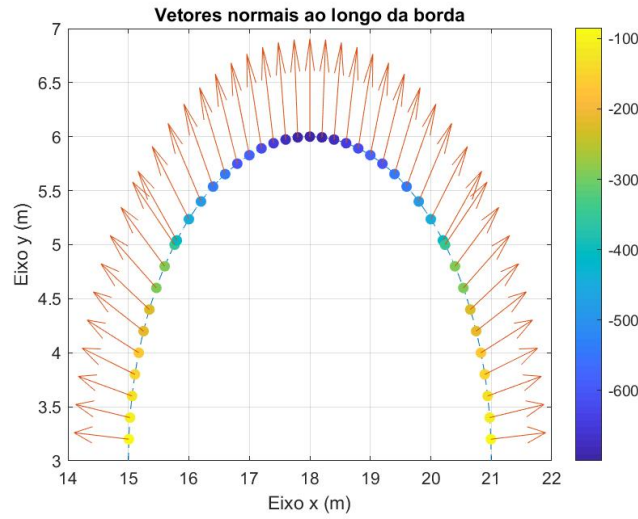
Os gráficos obtidos para os vetores normais normalizados (com módulo igual a 1) para os passos $\Delta x = \frac{h}{3} = 1$ e $\Delta x = \frac{h}{15} = 0.2$ foram:

Figura 29: Vetores normais para $\Delta x = 1$.



Fonte: Autoria própria.

Figura 30: Vetores normais para $\Delta x = 0.2$.

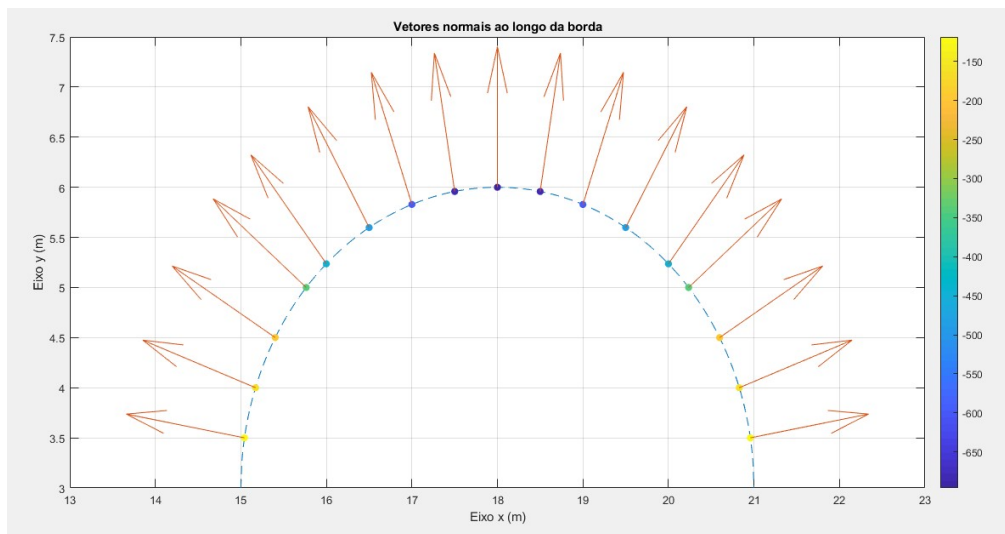


Fonte: Autoria própria.

Podemos observar que, conforme há uma diminuição do passo utilizado, os vetores normais tornam-se cada vez mais verossímeis. Ainda não podemos afirmar que a representação é 100% correta, inclusive pela forma do *Matlab* de representar a imagem sem ser um círculo perfeito, o que distorce um pouco a angulação real entre o vetor e a superfície, mas podemos atestar que com certeza há uma melhoria na forma de representação.

Ampliando o gráfico gerado pelo *Matlab* para que a representação do telhado seja mais próxima de um círculo, podemos atestar como essa representação das normais se mostra mais apropriada, como no gráfico abaixo para $\Delta x = \frac{h}{6} = 0.5$:

Figura 31: Vetores normais para $\Delta x = 0.5$.

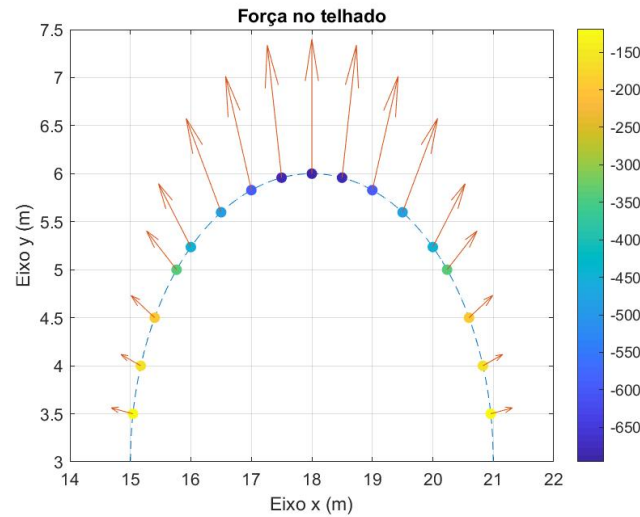


Fonte: Autoria própria.

Após isso, pudemos obter a intensidade das forças resultantes na direção normal - por meio da multiplicação entre a pressão e a área dos retângulos considerados

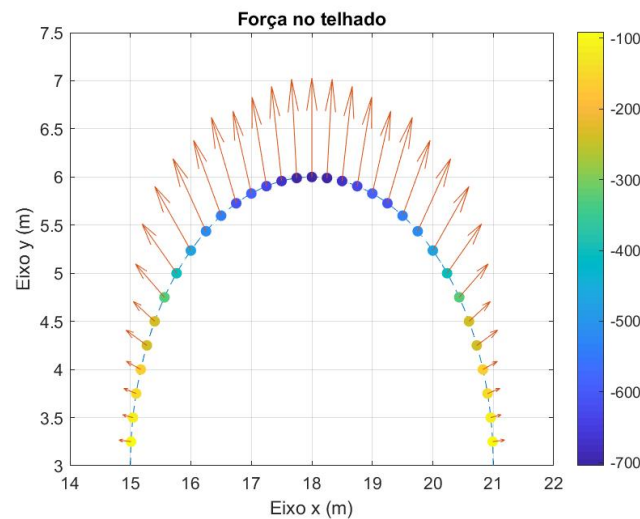
mencionados no t3pico acima. Essa intensidade, como esperado, aumenta conforme nos aproximamos do topo do telhado - da mesma forma que o m3dulo da press3o tamb3m aumenta - e a for7a est3 na dire73o normal ao ret3ngulo. Podemos atestar isso pelos gr3ficos abaixo, gerados com os passos $\Delta x = \frac{h}{6} = 0.5$ e $\Delta x = \frac{h}{12} = 0.25$.

Figura 32: For7a resultante no telhado para $\Delta x = 0.5$.



Fonte: Autoria pr3pria.

Figura 33: For7a resultante no telhado para $\Delta x = 0.25$.



Fonte: Autoria pr3pria.

Assim, podemos ver que a for7a puxando o telhado para cima aumenta conforme atingimos o seu 3pice. Como o c3lculo da for7a resultante 3 influenciado pela 3rea de a73o de cada uma das press3es calculadas e h3 diversas imprecis3es tanto nisso quanto no pr3prio c3lculo das normais e das press3es devido ao passo utilizado.

Portanto, os resultados encontrados para a for7a resultante em y - uma vez que a for7a resultante em x 3 zero, por equil3brio de for7as dado que as press3es

são simétricas com relação ao centro do círculo - para os diferentes passos usados, foram:

Tabela 2: Força resultante no telhado para cada passo Δx .

$\Delta x = \Delta y$	Força resultante
$\frac{h}{3} = 1$	$1.5773 \cdot 10^5$
$\frac{h}{6} = 0.5$	$1.7045 \cdot 10^5$
$\frac{h}{8} = 0.375$	$1.7871 \cdot 10^5$
$\frac{h}{10} = 0.3$	$1.7886 \cdot 10^5$
$\frac{h}{12} = 0.25$	$1.8106 \cdot 10^5$
$\frac{h}{15} = 0.2$	$1.7678 \cdot 10^5$

Fonte: Autoria própria.

Assim, podemos ver que, para a maioria dos passos, a força resultante encontrada gira entorno de $F_R = 1.78 \cdot 10^5 N = 1.78 \cdot 10^2 kN$, o que é uma força bastante considerável e seria capaz de arrancar telhados de construções dependendo das condições em que estivessem presos. No caso do modelo de simulação, por ter um grande comprimento, o telhado inteiro ser arrancado parece improvável, entretanto seriam provocados grandes danos estruturais com certeza - não somente pelo vento, mas por possíveis projéteis que estariam sendo carregados pelo vento também.

Vale lembrar que, como já foi mencionado, um vento com as velocidades que atingimos nesse Exercício-Programa são comparáveis às velocidades que atinge em um tornado. Portanto, essa força está adequada para a velocidade dos ventos aplicados.

3 Modelagem e Resultados - Parte 2

3.1 Distribuição de temperatura do ar

3.1.1 Implementação

Agora, a equação diferencial que desejamos resolver neste Exercício-Programa é outra. Além da velocidade e da pressão do ar em torno do teto, é interessante fazer uma análise acerca da *temperatura* do fluido e da troca de calor entre a estrutura e o ar. Para isso, a equação diferencial que será analisada por meio do *Método das Diferenças Finitas* é

$$k\nabla^2 T - \rho c_p \mathbf{u} \nabla T = 0 \quad (30)$$

Uma vez que o gradiente da temperatura se caracteriza como o vetor

$$\nabla T = \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right) \quad (31)$$

Os componentes da equação acima serão o produto escalar entre os vetores mencionados, ou seja,

$$\nabla^2 T = \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right) \cdot \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right) = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \quad (32)$$

$$\mathbf{u} \cdot \nabla T = (u, v) \cdot \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right) = u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \quad (33)$$

E com isso, podemos utilizar o *Método das Diferenças Finitas* para aproximar as derivadas parciais a serem utilizadas. Nesse problema, entretanto, uma vez que as condições de contorno não são simétricas - o que é esperado, uma vez que a construção está numa temperatura diferente do ar ventando da esquerda para a direita, então sua temperatura deve mudar após passar o silo - não se pôde implementar o problema de forma simétrica.

Além disso, como mencionado pelo enunciado do Exercício-Programa, devido à instabilidade de diferenças finitas centrais para essa equação, aproximou-se a primeira derivada parcial em relação a x ou y por *1ª diferença progressiva* ou *1ª diferença regressiva*, dependendo do sentido da componente da velocidade naquele eixo. Entretanto, uma vez que a velocidade na direção x era sempre positiva, isso facilitou a implementação.

Como haviam muitos casos para serem considerados, cada uma das equações necessárias foi deduzida analiticamente antes de ser utilizada no exercício programa. Para um ponto no meio da malha, sem nenhuma condição de contorno a ser considerada a sua volta, e com $u > 0$ e $v > 0$, foi utilizada a seguinte equação:

$$T_{i,j} = \frac{\frac{k}{\Delta x} (T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}) + \rho c_p (u(i,j)T_{i-1,j} + v(i,j)T_{i,j-1})}{\frac{4k}{\Delta x} + \rho c_p (u(i,j) + v(i,j))} \quad (34)$$

Ao mesmo tempo, caso fosse calculado em um ponto onde $v < 0$, a equação utilizada foi

$$T_{i,j} = \frac{\frac{k}{\Delta x} (T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}) + \rho c_p (u(i,j)T_{i-1,j} - v(i,j)T_{i,j+1})}{\frac{4k}{\Delta x} + \rho c_p (u(i,j) - v(i,j))} \quad (35)$$

Porque só precisamos alterar de *1ª diferença regressiva* para *1ª diferença progressiva* no termo multiplicado por v e manter o resto da equação. Já para o contorno irregular, os fatores a e b continuaram sendo os mesmos em ambos os lados, entretanto a equação fica ligeiramente diferente entre os dois lados, uma vez que em um lado o ponto **anterior** é o ponto a uma distância inferior a Δx e do outro lado o ponto **posterior** que possui essa distância menor.

Foram deduzidas todas as equações necessárias e para todos os casos, novamente - tanto quando o ponto está a uma distância do telhado somente inferior a Δx , somente inferior a Δy ou menor que ambos. Para o caso em que estamos a uma distância do telhado inferior a discretização da malha em ambas as direções e $u > 0$ e $v < 0$ - o que ocorria no lado esquerdo do telhado, devido a inversão da direção do nosso eixo y - encontramos a seguinte equação para ser utilizada:

$$T_{i,j} = \frac{\frac{2k}{\Delta x} \left(\frac{T_{i+1,j}}{a+a^2} + \frac{T_{i-1,j}}{1+a} + \frac{T_{i,j+1}}{b+b^2} + \frac{T_{i,j-1}}{b+1} \right) + \rho c_p (u(i,j)T_{i-1,j} - v(i,j)\frac{T_{i,j+1}}{b})}{\frac{2k}{\Delta x} \left(\frac{1}{a} + \frac{1}{b} \right) + \rho c_p (u(i,j) - \frac{v(i,j)}{b})} \quad (36)$$

E formas equivalentes para as demais condições. Vale ressaltar, que como foi observado no cálculo das velocidades, o componente da velocidade em x é **sempre**

maior que zero, logo nunca foi feita essa checagem na implementação do código da temperatura.

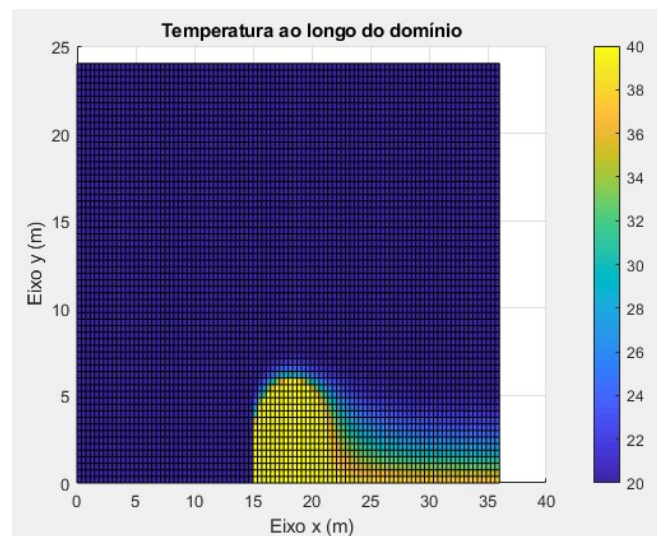
A condição de parada para esse problema foi a mesma da **Parte 1 - Item A**, entretanto, o coeficiente de sobrerelaxação sugerido pelo enunciado foi de $\lambda = 1.15$ agora. Entretanto, para esse valor, não foi possível fazer com que a nossa aproximação convergisse, e resolvemos adotar $\lambda = 1$, o que é equivalente a não utilizar a sobrerelaxação e usar somente o método tradicional de *Gauss-Seidel* para aproximar a solução de sistemas lineares.

De qualquer forma, foi possível fazer com que o problema convergisse e fossem encontradas as soluções para o problema desejado. A implementação completa foi feita na função *Item2a.m*, presente nos **Apêndices**.

3.1.2 Resultados obtidos

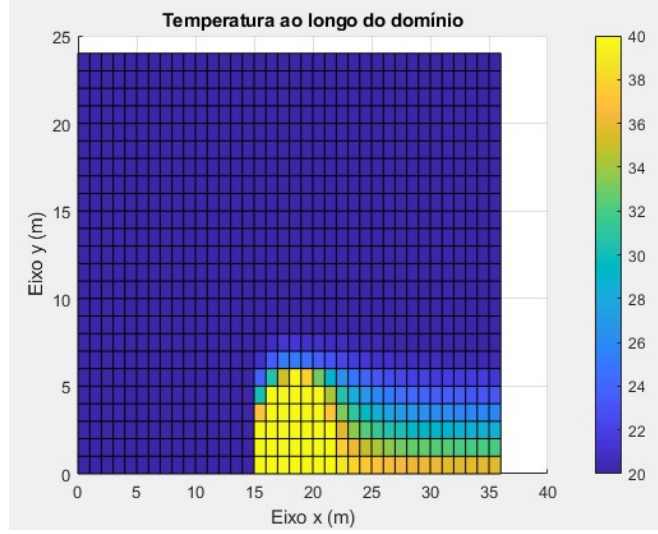
Após realizar as simulações para os mesmos passos utilizados na **Parte 1** e plotar os resultados obtidos com os comandos *surf* e *contourf* do *Matlab* novamente. Usando o primeiro comando, pudemos encontrar os seguintes gráficos para os passos $\Delta x = \frac{h}{8} = 0.375$ e $\Delta x = \frac{h}{3} = 1$:

Figura 34: Temperatura ao longo do domínio para $\Delta x = 0.375$.



Fonte: Autoria própria.

Figura 35: Temperatura ao longo do domínio para $\Delta x = 1$.

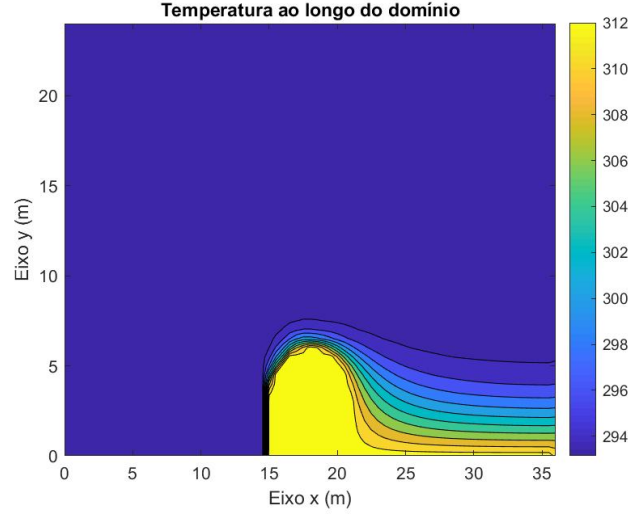


Fonte: Autoria própria.

Considerando que as condições de contorno definem somente que a temperatura do vento entrando no domínio deve ser $T_{fora} = 20^{\circ}\text{C}$ e que a temperatura dentro do silo deve ser $dentro = 40^{\circ}\text{C}$, os resultados desta simulação parecem bastante satisfatórios. A temperatura permanece 20°C para todo o domínio antes de chegar perto da construção, começa a esquentar conforme o vento entra em contato com a estrutura e continua aquecido após isso, porém o calor se dissipa conforme se afasta da construção. Já para o ar longe o suficiente do prédio ao longo do domínio, não há nenhuma alteração relevante, de forma a ser praticamente imperceptível essa alteração para essas moléculas.

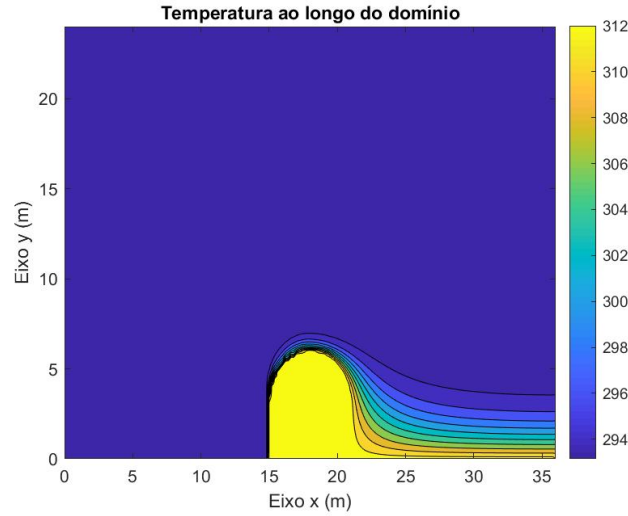
Visando visualizar de uma forma mais clara as linhas de corrente que representam a temperatura ao longo do domínio, utilizamos o comando *contourf* para plotar a temperatura - entretanto, tivemos que passar os valores para *Kelvin*, pois por motivos desconhecidos, em *Celsius* apareciam linhas sem sentido no meio do domínio, apesar de ter a mesma temperatura do resto. Assim, para os passos $\Delta x = \frac{h}{6} = 0.5$ e $\Delta x = \frac{h}{15} = 0.2$, obtivemos os gráficos:

Figura 36: Temperatura ao longo do domínio para $\Delta x = 0.5$ - Utilizando *contourf*.



Fonte: Autoria própria.

Figura 37: Temperatura ao longo do domínio para $\Delta x = 0.2$ - Utilizando *contourf*.



Fonte: Autoria própria.

Os resultados parecem coerentes com o esperado pelos nossos aprendizados em *Transferência de Calor e Mecânica dos Fluidos*. A única incoerência parece ser que, para qualquer passo utilizado, o chão após passar pela construção passa a assumir uma temperatura próxima de 40°C , o que é discordante com a realidade, mas esperado pelas condições de contorno impostas. Caso fosse imposta uma condição de contorno de *Dirichlet*, onde a temperatura do chão devesse ser fixa, possivelmente o calor seria dissipado também por influência da temperatura do chão.

Por fim, como novamente utilizamos o *Método das Diferenças Finitas*, pudemos analisar novamente a quantidade de iterações necessárias para a conversão conforme muda-se o passo. Entretanto, a matriz inicial de temperatura já foi iniciada em 20°C e poucas posições realmente são alteradas para a convergência, o

que diminuiu drasticamente o número de iterações com relação ao implementando em **Parte 1 - Item A**. Assim, tivemos que:

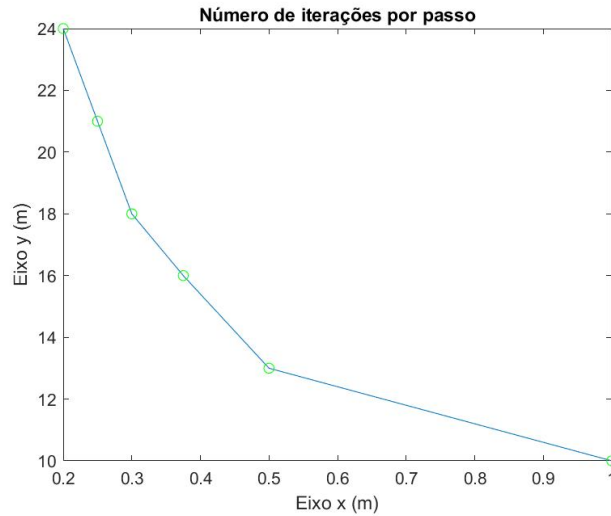
Tabela 3: Número de iterações de acordo com o passo Δx usado.

$\Delta x = \Delta y$	Iterações
$\frac{h}{3} = 1$	10
$\frac{h}{6} = 0.5$	13
$\frac{h}{8} = 0.375$	16
$\frac{h}{10} = 0.3$	18
$\frac{h}{12} = 0.25$	21
$\frac{h}{15} = 0.2$	24

Fonte: Autoria própria.

Curiosamente, apesar do número de iterações ter sido muito menor do que no **Parte 1 - Item A**, também podemos ver que conforme há uma diminuição do passo, o número de iterações necessárias para a convergência também cresce de maneira exponencial:

Figura 38: Número de iterações em função do passo Δx .



Fonte: Autoria própria.

3.2 Taxa de calor retirada do prédio

3.2.1 Implementação

Para a implementação do cálculo da quantidade de calor retirada do prédio, foi necessário utilizar a fórmula do fluxo de calor $\vec{Q}|_{prédio}$ ao longo do prédio inteiro. Podemos calculá-lo por:

$$\vec{Q}|_{prédio} = -k \frac{\partial T}{\partial n} \vec{n} = -(k \nabla T \cdot \vec{n}) \vec{n} \quad (37)$$

Esse valor representa o calor que flui por cada retângulo aproximado de nossa estrutura - e nesse caso, devemos considerar não somente o telhado como o prédio

inteiro, uma vez que as paredes laterais também trocam calor com o ar fluindo ao seu redor - e para obter a quantidade de calor total, em W , é necessário realizar a integração desse valor ao longo da área total:

$$q = \int_A \vec{Q}|_{prédio} \cdot \vec{n} dA = - \int_A k \nabla T \cdot \vec{n} dA \quad (38)$$

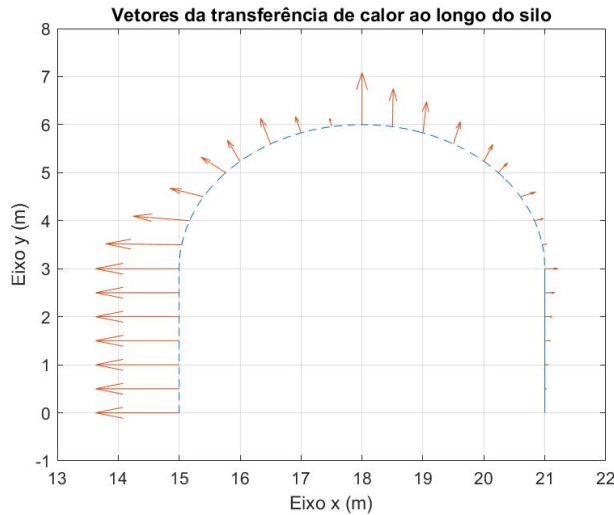
Assim, para calcular o calor total trocado entre a superfície e o meio, é necessário fazer o **produto escalar** entre o gradiente da temperatura ∇T e o vetor normal \vec{n} à superfície em cada um dos pontos, para obter a troca de calor local, e então somar todos esses valores ponderando-os pela área em que agem - o que é representado pela integral na fórmula.

Como estamos lidando com um problema discretizado, somente realizar a soma finita desses valores é suficiente para representar de forma satisfatória a troca de calor realizada. Utilizamos as mesmas normais calculadas para a **Parte 1 - Item E**, mas foi necessário acrescentar também os vetores normais às laterais do prédio, o que foi simples uma vez que estão na vertical. A implementação completa foi feita com a função *Item2b.m*, a qual está presente nos **Apêndices**.

3.2.2 Resultados obtidos

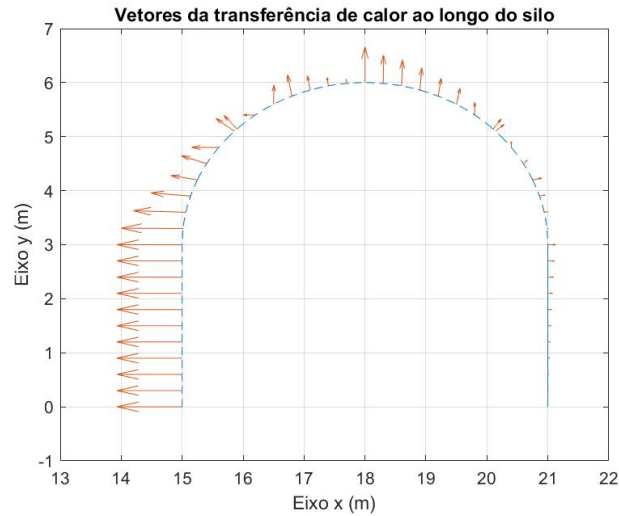
Novamente, assim como mencionado acima em *Implementação*, tal qual no cálculo da força em **Parte 1 - Item E**, encontramos pontos que estariam sobre o círculo e os utilizamos para traçar as normais. Da mesma forma que na força, houve diferenças entre os valores encontrados de acordo com o passo utilizado, porém essas diferenças foram maiores do que com a força. Os gráficos encontrados para a transferência de calor, pra cada posição da construção, para passos de $\Delta x = \frac{h}{6} = 0.5$ e $\Delta x = \frac{h}{10} = 0.3$ foram:

Figura 39: Transferência de calor ao longo da construção para $\Delta x = 0.5$.



Fonte: Autoria própria.

Figura 40: Transferência de calor ao longo da construção para $\Delta x = 0.3$.



Fonte: Autoria própria.

Com isso, pôde-se concluir que a maior parte da transferência de calor ocorre no lado esquerdo da construção, onde há um contato direto da construção à $T_{dentro} = 40^{\circ}\text{C}$ com o ar, que ainda está a $T_{fora} = 20^{\circ}\text{C}$ e vai esquentando conforme se locomove ao longo do telhado. No geral, a transferência de calor total encontrada, em módulo, de acordo com o passo utilizado, foi:

Tabela 4: Condução de calor de acordo com o passo Δx usado.

$\Delta x = \Delta y$	q (W)
$\frac{h}{3} = 1$	464.94
$\frac{h}{6} = 0.5$	528.69
$\frac{h}{8} = 0.375$	634.19
$\frac{h}{10} = 0.3$	745.87
$\frac{h}{12} = 0.25$	837.76
$\frac{h}{15} = 0.2$	925.29

Fonte: Autoria própria.

Acreditamos que essa grande diferença entre os passos ocorreu, principalmente, devido ao aumento do número de pontos em contato com a lateral esquerda da estrutura, o que fará surgirem mais vetores nessa superfície e dará a impressão, para o programa, de uma transferência de calor maior ocorrendo. Vale lembrar que, apesar de exibido em módulo, o calor é transferido da construção para o fluido, uma vez que ele se encontra mais quente do que sua vizinhança.

3.3 Conclusões

Por meio desse Exercício-Programa, pudemos atestar o poder do *Método das Diferenças Finitas* para solucionar problemas com equações diferenciais das mais diversas áreas da engenharia de um modo numérico. Além disso, pudemos analisar a influência dos diferentes passos na aplicação desse método.

Pudemos concluir, acerca do problema proposto, que as velocidades utilizadas são bastante altas e, por isso, haveria uma grande chance da força resultante gerada levantar o teto, ao menos parcialmente, da estrutura analisada.

Apêndices

A Função implementada para o Item A da Parte 1 - *ItemA.m*

```

1 function [psi,bordairregular,AB]=ItemA(psi,H,h,d,L,dx,dy,V,w1,erroel)
2 %Resolve o Item A da Parte 1 do EP2
3
4 bordairregular=[0 0];
5 contador=0;
6 AB=[0 0];
7
8 contiteracoes=0;
9 itera=1;
10
11 while itera==1
12     errovelho=0; %Zera o erro para comparar a cada iteração
13     erronovo=0;
14     for j=1:(length(psi(:,1))-1)
15         %Não precisa ir até o final porque a última linha possui ...
            condição de
16         %Dirichlet, fixa em psi=0, então não itera nela
17         for i=1:(length(psi(1,:))+1)/2
18             %Vai somente até o meio porque é simétrico o problema
19             psivelho=psi(j,i);
20             if i==1
21                 %Devemos aplicar as condições de contorno onde ...
                    dpsi/dx=0
22             if j==1
23                 %Condição de contorno para a quina superior ...
                    esquerda, que
24                 %possui duas condições de Neumann
25                 psi(j,i)=(psi(j,i+1)+psi(j+1,i)+V*dx)/2;
26             else
27                 psi(j,i)=(2*psi(j,i+1)+psi(j-1,i)+psi(j+1,i))/4;
28             end
29         elseif j==1
30             %Devemos aplicar as condições de contorno onde ...
                    dpsi/dy=-V
31             psi(j,i)=(2*psi(j+1,i)+psi(j,i-1)+psi(j,i+1)+2*V*dx)/4;
32         elseif i>=((1/dx)*d+1) && j>=((1/dx)*(H-h)+1
33             %Confere se está dentro do silo (parte retangular)
34             psi(j,i)=0;
35         elseif j<((1/dx)*(H-h)+1 && j>=((1/dx)*(H-L/2-h)+1 && ...
            i>=((1/dx)*d+1
36             %Estamos no retângulo no qual o semicirculo do ...
                silo está
37             %inscrito. O cume, em L/2, terá uma distância dx ...
                com relação ao

```

```

38     %último ponto acima dele sempre, pois serão ...
        escolhidos valores
39     %de dx e dy para tal, assim como a distância para ...
        a lateral do
40     %silo
41     if (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j-1))^2<=(0.5*L)^2
42         %Estamos dentro do semicirculo, e portanto, ...
            devemos definir
43         %esses pontos como zero
44         %Checar direitinho se ta pegando tudo que era ...
            pra ta aqui
45         %dentro
46         psi(j,i)=0;
47         if contiteracoes==0 && ...
            (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j-1))^2==(0.5*L)^2
48             bordairregular(end+1,:)= [j,i];
49             %Salva os índices dos valores que serão ...
                necessários
50             %e as distâncias A e B deles para futuro ...
                uso nos
51             %itens 1D, 1E e 2B
52             AB(end+1,:)= [2,2]; %distancia do ponto a ...
                esquerda
53             contador=contador+1;
54         end
55     elseif ...
        (dx*(i)-d-0.5*L)^2+(H-h-dx*(j-1))^2<(0.5*L)^2 ...
        || ...
56         (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j))^2<(0.5*L)^2
57         %Checa se o pr ximo ponto vai cair no (tanto ...
            em x quanto em
58         %y) cai dentro do círculo. Se for cair, temos ...
            que usar o MDF
59         %para bordas irregulares para calcular a ...
            iteração neles.
60         a=(d+L/2-dx*(i-1)-sqrt((L/2)^2-(H-h-dx*(j-1)).^2))/dx;
61         b=(H-h-dx*(j-1)-sqrt((L/2)^2-(d+0.5*L-dx*(i-1))^2))/dx;
62         if contiteracoes==0
63             %Salva os índices dos valores que serão ...
                necessários
64             %e as distâncias A e B deles para futuro ...
                uso nos
65             %itens 1D, 1E e 2B
66             bordairregular(end+1,:)= [j,i];
67             bordairregular(end+1,:)= [j,length(psi(1,:))+1-i];
68             contador=contador+1;
69             AB(end+1,:)= [a,b]; %distancia do ponto a ...
                esquerda
70             AB(end+1,:)= [a,b]; %distancia do ponto a ...
                direita
71         end
72         if a<1 && b<1
73             %Ponto está mais perto que dx e que dy do ...
                circulo
74             psi(j,i)=(psi(j,i+1)/(a*(a+1))+psi(j,i-1)/(1+a)+psi(j+1,i)/(b*(b+1))+psi(j-1,i)/(1+b))/(1/a+1/b);
75             +psi(j-1,i)/(1+b))/(1/a+1/b);
76         elseif a<1
77             %Ponto está a uma distância maior ou ...

```

```

            igual que dy
            %mas menor que dx do círculo
78         psi(j,i)=(2*psi(j,i+1)/(a+a^2)+2*psi(j,i-1)/(1+a)+psi(j+1,i).
79             +psi(j-1,i))/(2+2/a);
80     elseif b<1
81         %Ponto está a uma distância maior ou ...
            igual que dx
82         %mas menor que dy do círculo
83         psi(j,i)=(2*psi(j+1,i)/(b+b^2)+2*psi(j-1,i)/(1+b)+psi(j,i+1).
84             +psi(j,i-1))/(2+2/b);
85     end
86     else
87         psi(j,i)=(psi(j+1,i)+psi(j-1,i)+psi(j,i+1)+psi(j,i-1))/4;
88     end
89     else
90         psi(j,i)=(psi(j+1,i)+psi(j-1,i)+psi(j,i+1)+psi(j,i-1))/4;
91     end
92     psi(j,i)=w1*psi(j,i)+(1-w1)*psivelho;
93     if i<=(length(psi(1,:)))/2-1
94         psi(j,length(psi(1,:))+1-i)=psi(j,i);
95     elseif i>=length(psi(1,:))/2
96         psi(j,length(psi(1,:))+2-i)=psi(j,i-1);
97     end
98     erronovo=abs(psi(j,i)-psivelho);
99     if erronovo>errovelho
100         %Armazena o maior valor do erro (em m dulo)
101         errovelho=erronovo;
102     end
103 end
104 end
105 end
106 if errovelho<=erroel
107     %Caso o máximo erro calculado seja menor que o aceitavel, ...
        encerra o
108     %loop
109     itera=0;
110 end
111 contiteracoes=contiteracoes+1;
112 %Conta quantas iterações foram necessárias pra convergir
113 end
114
115 display('O total de iterações necessário para esse passo foi')
116 contiteracoes=contiteracoes
117 end

```

B Função implementada para o Item B da Parte 1 - *ItemB.m*

```

1 function [u,v]=ItemB(psi,H,h,d,L,dx,dy,V)
2 %Resolve o Item B da Parte 1 do EP2
3
4 %Inicializa as matrizes das derivadas
5 dpsidx=zeros([(H/dy)+1 1+(2*d+L)/dx]));
6 dpsidy=zeros([(H/dy)+1 1+(2*d+L)/dx]));
7
8 for j=1:length(psi(:,1))

```

```

9     for i=1:(length(psi(1,:))+1)/2
10         if i==1 || j==1
11             %Confere se está em alguma das laterais
12             if j~=1
13                 dpsidx(j,i)=0;
14                 if j~=length(psi(:,1))
15                     dpsidy(j,i)=(psi(j+1,i)-psi(j-1,i))/(2*dx);
16                 else
17                     %Está na quina superior esquerda
18                     dpsidy(j,i)=(psi(j,i)-psi(j-1,i))/dx;
19                 end
20             elseif i~=1
21                 dpsidx(j,i)=(psi(j,i+1)-psi(j,i-1))/(2*dx);
22                 dpsidy(j,i)=-V;
23             else
24                 dpsidx(j,i)=0;
25                 dpsidy(j,i)=-V;
26             end
27         elseif j==length(psi(:,1))
28             dpsidx(j,i)=(psi(j,i+1)-psi(j,i-1))/(2*dx);
29             dpsidy(j,i)=(psi(j,i)-psi(j-1,i))/dx;
30         elseif i>=((1/dx)*d+1) && j>=((1/dx)*(H-h)+1
31             %Confere se está dentro do silo (parte retangular)
32             dpsidx(j,i)=0;
33             dpsidy(j,i)=0;
34         elseif j<((1/dx)*(H-h)+1) && j>=((1/dx)*(H-L/2-h)+1) && ...
35             i>=((1/dx)*d+1
36             %Estamos no retângulo no qual o semicirculo do silo está
37             %inscrito. O cume, em L/2, terá uma distância dx com ...
38             %relação ao
39             %último ponto acima dele sempre, pois serão ...
40             %escolhidos valores
41             %de dx e dy para tal, assim como a distância para a ...
42             %lateral do
43             %silo
44             if (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j-1))^2<=(0.5*L)^2
45                 %Estamos dentro do semicirculo, e portanto, ...
46                 %devemos definir
47                 %esses pontos como zero
48                 %Checar direitinho se ta pegando tudo que era pra ...
49                 %ta aqui
50                 %dentro
51                 if (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j-1))^2==(0.5*L)^2
52                     dpsidx(j,i)=0;
53                     dpsidy(j,i)=(psi(j-2,i)-4*psi(j-1,i)+3*psi(j,i))/...
54                     (2*dx);
55                 else
56                     dpsidx(j,i)=0;
57                     dpsidy(j,i)=0;
58                 end
59             elseif (dx*(i)-d-0.5*L)^2+(H-h-dx*(j-1))^2<(0.5*L)^2 ...
60                 || ...
61                 (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j))^2<(0.5*L)^2
62                 %Checa se o pr ximo ponto vai cair no (tanto em ...
63                 %x quanto em
64                 %y) cai dentro do círculo. Se for cair, temos que ...
65                 %usar o MDF
66                 %para bordas irregulares para calcular a iteração ...

```

```

58         neles.
59         a=(d+L/2-dx*(i-1)-sqrt((L/2)^2-(H-h-dx*(j-1)).^2))/dx;
60         b=(H-h-dx*(j-1)-sqrt((L/2)^2-(d+0.5*L-dx*(i-1))^2))/dx;
61         if a<1 && b<1
62             dpsidx(j,i)=(psi(j,i+1)-a^2*psi(j,i-1)-(1-a^2)*...
63                 psi(j,i))/(a+a^2)*dx;
64             dpsidy(j,i)=(psi(j+1,i)-b^2*psi(j-1,i)-(1-b^2)*...
65                 psi(j,i))/(b+b^2)*dx;
66         elseif a<1
67             dpsidx(j,i)=(psi(j,i+1)-a^2*psi(j,i-1)-(1-a^2)*...
68                 psi(j,i))/(a+a^2)*dx;
69             dpsidy(j,i)=(psi(j+1,i)-psi(j,i))/(dx);
70         elseif b<1
71             dpsidx(j,i)=(psi(j,i+1)-psi(j,i))/dx;
72             dpsidy(j,i)=(psi(j+1,i)-b^2*psi(j-1,i)-(1-b^2)*...
73                 psi(j,i))/(b+b^2)*dx;
74         end
75     else
76         dpsidx(j,i)=(psi(j,i+1)-psi(j,i-1))/(2*dx);
77         dpsidy(j,i)=(psi(j+1,i)-psi(j-1,i))/(2*dx);
78     end
79 else
80     dpsidx(j,i)=(psi(j,i+1)-psi(j,i-1))/(2*dx);
81     dpsidy(j,i)=(psi(j+1,i)-psi(j-1,i))/(2*dx);
82 end
83 if i<=(length(psi(1,:)))/2
84     dpsidx(j,length(psi(1,:))+1-i)=-dpsidx(j,i);
85     %Inverte o sinal pois em vez de o vento estar ...
86     %desviando do silo
87     %está voltando a ocupar o espaço no qual havia o silo.
88     dpsidy(j,length(psi(1,:))+1-i)=dpsidy(j,i);
89 elseif i>=length(psi(1,:))/2
90     dpsidx(j,length(psi(1,:))+1-i)=dpsidx(j,i);
91     dpsidy(j,length(psi(1,:))+1-i)=dpsidy(j,i);
92 end
93 end
94 end
95 %Salva os vetores das derivadas como os componentes das velocidades,
96 %fazendo os ajustes de sinais necessários pela orientação do sistema
97 u=-dpsidy;
98 v=dpsidx;
99 end

```

C Função implementada para o Item C da Parte 1 - *Item C.m*

```

1 function pressaovetor=ItemC(ro,gama,u,v)
2 %Resolve o Item C da Parte 1 do EP2
3
4 pressaovetor=ro*(gama-1)/gama*(0.^2/2-(u.^2+v.^2)/2);
5
6 end

```

D Função implementada para o Item D da Parte 1 - *ItemD.m*

```

1 function ...
    [pressaoarrumada,pontosoriginais]=ItemD(psi,H,h,d,L,dx,dy,pressaovetor,bordairregu
2 %Resolve o Item D da Parte 1 do EP2
3
4 pressao=[0 0 0];
5 pontosoriginais=[0 0];
6 for i=2:length(bordairregular)
7     pontosoriginais(end+1,:)=[dx*(bordairregular(i,2)-1),H-dy*(bordairregular(i,1)-1)
8     if AB(i,1)>1
9         if AB(i,2)<=1
10             pressao(end+1,:)=[dx*(bordairregular(i,2)-1),H-dy*(bordairregular(i,1)-1)
11         else
12             %S vai entrar aqui se for o ponto do topo que ...
13             defini com
14             %distâncias (2,2), e esse não precisa subtrair nada
15             pressao(end+1,:)=[dx*(bordairregular(i,2)-1),H-dy*...
16                 (bordairregular(i,1)-1),pressaovetor(bordairregular(i,1)...
17                 ,bordairregular(i,2))];
18         end
19     elseif AB(i,2)>1
20         if AB(i,1)<=1
21             if bordairregular(i,2)>(length(psi(1,:))-1)/2
22                 pressao(end+1,:)=[dx*(bordairregular(i,2)-1)-AB(i,1)*...
23                     dx,H-dy*(bordairregular(i,1)-1),...
24                     pressaovetor(bordairregular(i,1),bordairregular(i,2))];
25             else
26                 pressao(end+1,:)=[dx*(bordairregular(i,2)-1)+AB(i,1)*dx,...
27                     H-dy*(bordairregular(i,1)-1),...
28                     pressaovetor(bordairregular(i,1),bordairregular(i,2))];
29             end
30         end
31     else
32         if AB(i,1)>=AB(i,2)
33             %Ajusta a distancia de acordo com o valor de b, caso ...
34             seja menor
35             %que a
36             pressao(end+1,:)=[dx*(bordairregular(i,2)-1),H-dy*...
37                 (bordairregular(i,1)-1)-AB(i,2)*dy,...
38                 pressaovetor(bordairregular(i,1),bordairregular(i,2))];
39         else
40             %Ajusta a distancia de acordo com o valor de a, caso ...
41             seja menor
42             %que b
43             if bordairregular(i,2)>(length(psi(1,:))-1)/2
44                 pressao(end+1,:)=[dx*(bordairregular(i,2)-1)-AB(i,1)*dx,...
45                     H-dy*(bordairregular(i,1)-1),...
46                     pressaovetor(bordairregular(i,1),bordairregular(i,2))];
47             else
48                 pressao(end+1,:)=[dx*(bordairregular(i,2)-1)+AB(i,1)*dx,...
49                     H-dy*(bordairregular(i,1)-1),...
50                     pressaovetor(bordairregular(i,1),bordairregular(i,2))];
51             end
52         end
53     end
54 end

```

```

49         end
50     end
51 end
52
53 pressao=[pressao pontosoriginais];
54 [arrumado,indices]=sort(pressao(:,1));
55 pressaoarrumada=pressao(indices,1:3);
56 pontosoriginais=pressao(indices,4:5);
57
58 end

```

E Função implementada para o Item E da Parte 1 - *Item E.m*

```

1 function [FR,Areas,Normais,Forca]=ItemE(pressaoarrumada,Comprimento)
2 %Resolve o Item E da Parte 1 do EP2
3
4 Normais=zeros([length(pressaoarrumada)-1 2]);
5 Pontos_medios=zeros([length(pressaoarrumada) 2]);
6
7 for i=2:(length(pressaoarrumada))
8     if i==2
9         Normais(i-1,:)= [3-pressaoarrumada(i+1,2) ...
10             pressaoarrumada(i+1,1)-...
11             15];
12         Pontos_medios(i-1,:)= [(pressaoarrumada(i,1)+15)/2 (3+...
13             pressaoarrumada(i,2))/2];
14     elseif i==length(pressaoarrumada)
15         Normais(i-1,:)= [pressaoarrumada(i-1,2)-3 21-...
16             pressaoarrumada(i-1,1)];
17         Pontos_medios(i-1,:)= [(pressaoarrumada(i-1,1)+...
18             pressaoarrumada(i,1))/2 (pressaoarrumada(i,2)+...
19             pressaoarrumada(i-1,2))/2 ];
20     else
21         %Gera os vetores que ligam os pontos das pressões em i+1 ...
22         %e i-1,
23         %que teria coordenadas (x,y) e salva em normais como ...
24         %(-y,x), o que
25         %é um vetor normal a esse anterior.
26         Normais(i-1,:)= [pressaoarrumada(i-1,2)-pressaoarrumada(i+1,2),...
27             pressaoarrumada(i+1,1)-pressaoarrumada(i-1,1)];
28         Pontos_medios(i-1,:)= [(pressaoarrumada(i-1,1)+...
29             pressaoarrumada(i,1))/2 (pressaoarrumada(i-1,2)+...
30             pressaoarrumada(i,2))/2];
31     end
32     Normais(i-1,:)=Normais(i-1,:)/sqrt(Normais(i-1,1)^2+Normais(i-1,2)^2);
33 end
34
35 Pontos_medios(end,:)= [(pressaoarrumada(end,1)+21)/2 (3+...
36     pressaoarrumada(end,2))/2];
37 %Calcula as áreas necessárias no EP
38 Areas=Comprimento*(sqrt((Pontos_medios(2:end,1)-...
39     Pontos_medios(1:end-1,1)).^2+(Pontos_medios(2:end,2)-...
40     Pontos_medios(1:end-1,2)).^2));
41 Areas(1)=Areas(1)+Comprimento*(sqrt((15-Pontos_medios(1,1)).^2+...

```



```

39     (3-Pontos_medios(1,2)).^2));
40 Areas(end)=Areas(end)+Comprimento*(sqrt((15-Pontos_medios(1,1)).^2+...
41     (3-Pontos_medios(1,2)).^2));
42 %Calcula a força
43 Forca=Areas.*pressaoarrumada(2:end,3);
44 Componentey=-Forca.*Normais(:,2);
45 %Soma as componentes da força em y para obter a força resultante
46 FR=sum(Componentey);
47
48 end

```

F Função implementada para o Item A da Parte 2 - *Item2a.m*

```

1 function ...
    T=Item2a(T,k,ro,cp,Tdentro,Tfora,dx,dy,H,h,L,d,V,w2,u,v,erroel)
2 %Resolve o Item A da Parte 2 do EP2
3
4 itera=1;
5 contiteracoes=0;
6
7 while itera
8     errovelho=0;
9     erronovo=0;
10    for j=1:(length(T(:,1)))
11        for i=1:(length(T(1,:)))
12            Tvelho=T(j,i);
13            if i==1 || j==1 || i==length(T(1,:)) || j==length(T(:,1))
14                %Condições de contorno do domínio
15                if i==1
16                    T(j,i)=Tfora;
17                elseif j==1
18                    if i~=length(T(1,:))
19                        %Parte superior do domínio
20                        %S consideramos o caso de u>0 pois é o que
21                        %acontece em todo o domínio
22                        T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1)+2*T(j+1,i))...
23                            +ro*cp*(u(j,i)*T(j,i-1)))/...
24                            (4*k/dx+ro*cp*(u(j,i)));
25                    else
26                        %Quina do lado superior direito
27                        T(j,i)=0.5*(T(j,i-1)+T(j+1,i));
28                    end
29                elseif i==length(T(1,:))
30                    %Parte direita do domínio
31                    if j~=length(T(:,1))
32                        %v=0, então vai cancelar independente do ...
33                        sinal
34                        T(j,i)=(k/dx*(2*T(j,i-1)+T(j-1,i)+T(j+1,i))...
35                            )/(4*k/dx);
36                    else
37                        %Quina do lado inferior direito
38                        T(j,i)=0.5*(T(j,i-1)+T(j-1,i));
39                    end
40                elseif j==length(T(:,1))

```

```

40         %Chão do domínio
41         if i<((1/dx)*d+1) || i>((1/dx)*(d+L)+1
42             %Fora do hangar
43             %S    consideramos o caso de u>0 pois é o que
44             %acontece em todo o domínio
45             T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1))+2*T(j-1,i))...
46                 +ro*cp*(u(j,i)*T(j,i-1)))/...
47                 (4*k/dx+ro*cp*(u(j,i)));
48         else
49             T(j,i)=Tdentro;
50         end
51     end
52 elseif i>=((1/dx)*d+1) && j>=((1/dx)*(H-h)+1) && ...
53     i<=((1/dx)*(d+L)+1
54         %Confere se está dentro do silo (parte retangular)
55         T(j,i)=Tdentro;
56 elseif j<((1/dx)*(H-h)+1) && j>=((1/dx)*(H-L/2-h)+1) && ...
57     i>=((1/dx)*d+1) && i<=((1/dx)*(d+L)+1
58         %Estamos no retângulo no qual o semicírculo do ...
59         %inscrito. O cume, em L/2, terá uma distância dx ...
60         %último ponto acima dele sempre, pois serão ...
61         %de dx e dy para tal, assim como a distância para ...
62         %silo
63         if (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j-1))^2<=(0.5*L)^2
64             %Estamos dentro do semicírculo, e portanto, ...
65             %esses pontos como zero
66             T(j,i)=Tdentro;
67         elseif ...
68             (dx*(i)-d-0.5*L)^2+(H-h-dx*(j-1))^2<(0.5*L)^2 ...
69             || ...
70             (dx*(i-2)-d-0.5*L)^2+(H-h-dx*(j-1))^2<(0.5*L)^2
71             %Checa se o próximo ponto – ou anterior – ...
72             %dentro do círculo (tanto em x quanto em y). ...
73             %Se for
74             %cair, temos que usar o MDF para bordas ...
75             %para calcular a iteração neles.
76             if (dx*(i)-d-0.5*L)^2+(H-h-dx*(j-1))^2<(0.5*L)^2
77                 %Proximo ponto em x é o hangar
78                 if ...
79                     (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j))^2<(0.5*L)^2
80                     %Proximo ponto em y é o hangar
81                     a=(d+L/2-dx*(i-1)-sqrt((L/2)^2-...
82                         (H-h-dx*(j-1)).^2))/dx;
83                     b=(H-h-dx*(j-1)-sqrt((L/2)^2-...
84                         (d+0.5*L-dx*(i-1))^2))/dx;
85                     %Usando a aproximação aconselhada ...
86                     para a

```

```

83     %primeira derivada de T, pelos resultados
84     %anteriores sabemos que nesse caso, ...
85     u>0 e v<0
86     %em todos os pontos. Assim, temos que
87     T(j,i)=(2*k/dx*(T(j,i+1)/(a+a^2)+T(j,i-1)/...
88         (1+a)+T(j+1,i)/(b^2+b)+T(j-1,i)/(b+1))-...
89         ro*cp*(-u(j,i)*T(j,i-1)+v(j,i)*...
90         T(j+1,i)/b))/(2*k/dx*(1/a+1/b)+ro*cp*...
91         (u(j,i)-v(j,i)/b));
92     else
93         a=(d+L/2-dx*(i-1)-sqrt((L/2)^2-...
94             (H-h-dx*(j-1)).^2))/dx;
95         T(j,i)=(k/dx*(2*T(j,i+1)/(a+a^2)+2*T(j,i-1)/...
96             (a+1)+T(j+1,i)+T(j-1,i))-ro*cp*(-u(j,i)...
97             *T(j,i-1)+v(j,i)*T(j+1,i)))/(k/dx*(2+2/a)...
98             +ro*cp*(u(j,i)-v(j,i)));
99     end
100 elseif ...
101     (dx*(i-2)-d-0.5*L)^2+(H-h-dx*(j-1))^2<(0.5*L)^2
102     %Ponto anterior em x é o hangar
103     if ...
104         (dx*(i-1)-d-0.5*L)^2+(H-h-dx*(j))^2<(0.5*L)^2
105         %Proximo ponto em y é o hangar
106         a=(dx*(i-1)-(d+L/2)-sqrt((L/2)^2-...
107             (H-h-dx*(j-1)).^2))/dx;
108         b=(H-h-dx*(j-1)-sqrt((L/2)^2-...
109             (d+0.5*L-dx*(i-1))^2))/dx;
110         T(j,i)=(2*k/dx*(T(j,i+1)/(a+1)+T(j,i-1)/...
111             (a+a^2)+T(j+1,i)/(b+1)+T(j-1,i)/(b^2+b))-...
112             ro*cp*(-u(j,i)*T(j,i-1)/a-v(j,i)*T(j-1,i)))/...
113             (2*k/dx*(1/a+1/b)+ro*cp*(u(j,i)/a+v(j,i)));
114     else
115         a=(dx*(i-1)-d-L/2-sqrt((L/2)^2-...
116             (H-h-dx*(j-1)).^2))/dx;
117         T(j,i)=(k/dx*(2*T(j,i+1)/(a+1)+2*T(j,i-1)/...
118             (a+a^2)+T(j+1,i)+T(j-1,i))-ro*cp*(-u(j,i)*...
119             T(j,i-1)/a-v(j,i)*T(j-1,i)))/(k/dx*(2+2/a)+...
120             ro*cp*(u(j,i)/a+v(j,i)));
121     end
122 else
123     %Proximo ponto em y (somente) é o hangar
124     if v(j,i)<=0
125         b=(H-h-dx*(j-1)-sqrt((L/2)^2-...
126             (d+0.5*L-dx*(i-1))^2))/dx;
127         T(j,i)=(k/dx*(2*T(j+1,i)/(b+b^2)+...
128             2*T(j-1,i)/(b+1)+T(j,i+1)+T(j,i-1))-...
129             ro*cp*(-u(j,i)*T(j,i-1)+v(j,i)*T(j+1,i)/b))/...
130             (k/dx*(2+2/b)+ro*cp*(u(j,i)-v(j,i)/b));
131     else
132         T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1)+T(j-1,i)+...
133             T(j+1,i))+ro*cp*(u(j,i)*T(j,i-1)+v(j,i)*...
134             T(j-1,i)))/(4*k/dx+ro*cp*(u(j,i)+v(j,i)));
135     end
136 end
137 else
138     %Pontos internos sem problemas no domínio
139     %Consideramos u>0 sempre pois é o que ...
140     acontece em todo

```

```

137         %o domínio
138         if v(j,i)>=0
139             T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1)+T(j-1,i)+T(j+1,i))...
140                 +ro*cp*(u(j,i)*T(j,i-1)+v(j,i)*T(j-1,i)))/...
141                 (4*k/dx+ro*cp*(u(j,i)+v(j,i)));
142         else
143             T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1)+T(j-1,i)+T(j+1,i))...
144                 +ro*cp*(u(j,i)*T(j,i-1)-v(j,i)*T(j+1,i)))/...
145                 (4*k/dx+ro*cp*(u(j,i)-v(j,i)));
146         end
147     end
148 else
149     %Pontos internos sem problemas no domínio
150     %u>0 em todo o domínio
151     if v(j,i)>=0
152         T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1)+T(j-1,i)+T(j+1,i))...
153             +ro*cp*(u(j,i)*T(j,i-1)+v(j,i)*T(j-1,i)))/...
154             (4*k/dx+ro*cp*(u(j,i)+v(j,i)));
155     else
156         T(j,i)=(k/dx*(T(j,i+1)+T(j,i-1)+T(j-1,i)+T(j+1,i))...
157             +ro*cp*(u(j,i)*T(j,i-1)-v(j,i)*T(j+1,i)))/...
158             (4*k/dx+ro*cp*(u(j,i)-v(j,i)));
159     end
160 end
161 T(j,i)=w2*T(j,i)+(1-w2)*Tvelho;
162 erronovo=abs(T(j,i)-Tvelho);
163 if erronovo>errovelho
164     %Armazena o maior valor do erro (em m dulo)
165     errovelho=erronovo;
166 end
167 end
168 end
169 if errovelho<=erroel
170     %Caso o máximo erro calculado seja menor que o aceitavel, ...
171     encerra o
172     %loop
173     itera=0;
174 end
175 contiteracoes=contiteracoes+1;
176 %Conta quantas iterações foram necessárias pra convergir
177 end
178 display('O total de iterações necessário para esse passo foi')
179 contiteracoes
180 end

```

G Função implementada para o Item B da Parte 2 - *Item2b.m*

```

1 function ...
2     [calor,gradientesarrumado,Normais]=Item2b(T,bordairregular,AB,psi,dx,dy,H,h,L,d,A,
3
4     %Resolve o Item B da Parte 2 do EP2
5
6     gradientes=[0 0 0 0]; %x, y, dT/dx dT/dy
7
8     for i=2:length(bordairregular)
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

6     if bordairregular(i,2)>(length(psi(1,:))+1)/2
7         %u>0 e v>0
8         if AB(i,1)<1
9             dTdx=(T(bordairregular(i,1),bordairregular(i,2)-1)-...
10                T(bordairregular(i,1),bordairregular(i,2)))/dx;
11         else
12             dTdx=(T(bordairregular(i,1),bordairregular(i,2)-1)-...
13                T(bordairregular(i,1),bordairregular(i,2)))/dx;
14         end
15         dTdy=(T(bordairregular(i,1),bordairregular(i,2))-...
16            T(bordairregular(i,1)-1,bordairregular(i,2)))/dx;
17         if AB(i,1)>=AB(i,2)
18             gradientes(end+1,:)=[dx*(bordairregular(i,2)-1),...
19                H-dy*(bordairregular(i,1)-1)-AB(i,2)*dy,dTdx,dTdy];
20         else
21             gradientes(end+1,:)=[dx*(bordairregular(i,2)-1)-AB(i,1)*dx,...
22                H-dy*(bordairregular(i,1)-1),dTdx,dTdy];
23         end
24     elseif bordairregular(i,2)<(length(psi(1,:))+1)/2
25         %u>0 e v<0
26         if AB(i,1)<1
27             dTdx=(T(bordairregular(i,1),bordairregular(i,2))-...
28                T(bordairregular(i,1),bordairregular(i,2)-1))/dx;
29         else
30             dTdx=(T(bordairregular(i,1),bordairregular(i,2))-...
31                T(bordairregular(i,1),bordairregular(i,2)-1))/dx;
32         end
33         if AB(i,2)<1
34             dTdy=(T(bordairregular(i,1)+1,bordairregular(i,2))-...
35                T(bordairregular(i,1),bordairregular(i,2)))/dx;
36         else
37             dTdy=(T(bordairregular(i,1)+1,bordairregular(i,2))-...
38                T(bordairregular(i,1),bordairregular(i,2)))/dx;
39         end
40         if AB(i,1)>=AB(i,2)
41             gradientes(end+1,:)=[dx*(bordairregular(i,2)-1),...
42                H-dy*(bordairregular(i,1)-1)-AB(i,2)*dy,dTdx,dTdy];
43         else
44             gradientes(end+1,:)=[dx*(bordairregular(i,2)-1)+AB(i,1)*dx,...
45                H-dy*(bordairregular(i,1)-1),dTdx,dTdy];
46         end
47     else
48         %Está bem no topo
49         dTdy=(T(bordairregular(i,1)-2,bordairregular(i,2))-...
50            4*T(bordairregular(i,1)-1,bordairregular(i,2))+...
51            3*T(bordairregular(i,2),bordairregular(i,1)))/(2*dx);
52         dTdx=(T(bordairregular(i,1),bordairregular(i,2))-...
53            T(bordairregular(i,1),bordairregular(i,2)-1))/dx;
54         gradientes(end+1,:)=[dx*(bordairregular(i,2)-1),...
55            H-dy*(bordairregular(i,1)-1),dTdx,dTdy];
56     end
57 end
58
59 for j=(length(psi(:,1))-1)+1-3/dy:(length(psi(:,1)))
60     dTdx=(T(j,15/dx+1)-T(j,15/dx))/dx;
61     dTdy=0;
62     gradientes(end+1,:)=[15,H-dx*(j-1),dTdx,dTdy];
63     dTdx=(T(j,21/dx+1)-T(j,21/dx+2))/dx;

```

```

64     Normais=[[-1 0];Normais];
65     gradientes(end+1,:)= [21,H-dx*(j-1),dTdx,dTdy];
66     Normais(end+1,:)= [1 0];
67 end
68
69 [arrumado,indices]=sort (gradientes(:,1));
70 gradientesarrumado=gradientes (indices,:);
71
72
73 produto=gradientesarrumado (2:end,3:4).*Normais(:,1:2);
74 modulo=sqrt (produto(:,1).^2+produto(:,2).^2);
75
76 lateral=zeros (3/dx-1,1);
77 lateral(:,1)=dx*60;
78 %acrescenta nas áreas a parcela relativa aos lados do silo
79 Areas=[0.5*dx*60;lateral;0.5*dx*60;Areas;0.5*dx*60;lateral;0.5*dx*60];
80 calor=-k*sum (Areas.*modulo);
81
82 end

```

H *Script* geral responsável por chamar as funções e plotar resultados - *EP2Main.m*

```

1 clear all
2 clc
3
4 %Definição das variáveis necessárias na primeira tarefa
5 h=3;
6 d=5*h;
7 L=2*h;
8 H=8*h;
9 dx=h/10;
10 dy=dx;
11 V=100/3.6;
12 Comprimento=60;
13 ro=1.25;
14 gama=1.4;
15 w1=1.85; %fator de sobrerelaxação
16 erroel=0.01;
17
18 %Inicializa a matriz de psi
19 psi=zeros (([ (H/dy)+1 1+(2*d+L)/dx])));
20
21 %Define as coordenadas necessárias para os gráficos
22 [X,Y]=meshgrid(dx*((1:length(psi(1,:)))-1),((length(psi(:,1)))-1:1)-1)*dx);
23 x=15:0.01:21;
24
25 %Exercício 1
26 %Item a
27 [psi,bordairregular,AB]=ItemA(psi,H,h,d,L,dx,dy,V,w1,erroel);
28
29 %Plota os gráficos do Item a
30 figure(1)
31 surf(X,Y,psi)
32 title("Função corrente do escoamento ao longo do domínio")

```

```

33 xlabel("Eixo x (m)")
34 ylabel("Eixo y (m)")
35 zlabel("Função corrente do escoamento")
36 colorbar
37
38 figure(2)
39 contour(X,Y,psi,40)
40 title("Função corrente do escoamento ao longo do domínio")
41 xlabel("Eixo x (m)")
42 ylabel("Eixo y (m)")
43 colorbar
44
45 %Item b
46 [u,v]=ItemB(psi,H,h,d,L,dx,dy,V);
47
48 %Plota os gráficos do Item b
49 figure(3)
50 quiver(X,Y,u,-v)
51 title("Velocidade do fluxo ao longo do domínio")
52 xlabel("Eixo x (m)")
53 ylabel("Eixo y (m)")
54
55 figure(4)
56 surf(X,Y,sqrt(u.^2+v.^2))
57 title("Módulo da velocidade ao longo do domínio")
58 xlabel("Eixo x (m)")
59 ylabel("Eixo y (m)")
60 zlabel("Módulo da velocidade (m/s)")
61 colorbar
62
63 %Item c
64 pressaovetor=ItemC(ro,gama,u,v);
65
66 %Plota o gráfico do Item c
67 figure(5)
68 surf(X,Y,pressaovetor)
69 title("Variação de pressão ao longo do domínio")
70 xlabel("Eixo x (m)")
71 ylabel("Eixo y (m)")
72 zlabel("Variação de pressão")
73 colorbar
74
75 %Item d
76 [pressaoarrumada,pontosoriginais]=ItemD(psi,H,h,d,L,dx,dy,pressaovetor,bordairregular)
77
78 %Plota os gráficos do Item d
79 figure(6)
80 plot(pressaoarrumada(2:end,1),pressaaoarrumada(2:end,3))
81 title("Variação de pressão ao longo da borda")
82 xlabel("Eixo x (m)")
83 ylabel("Variação de pressão")
84
85 figure(7)
86 plot(x,sqrt(9-(x-18).^2)+3,'—'),grid
87 hold on
88 scatter(pressaoarrumada(2:end,1),pressaaoarrumada(2:end,2),40,pressaaoarrumada(2:end,3))
89 hold on
90 scatter(pontosoriginais(2:end,1),pontosoriginais(2:end,2),40,pressaaoarrumada(2:end,3))

```

```

91 title("Variação de pressão ao longo da borda")
92 xlabel('Eixo x (m)')
93 ylabel('Eixo y (m)')
94 colorbar
95
96 %Item e
97 [FR,Areas,Normais,Forca]=ItemE(pressaoarrumada,Comprimento);
98
99 %Plota os gráficos do Item e
100 figure(8)
101 plot(x,sqrt(9-(x-18).^2)+3,'—'),grid
102 hold on
103 scatter(pressaoarrumada(2:end,1),pressaoarrumada(2:end,2),40,pressaoarrumada(2:end,3))
104 xlabel('Eixo x (m)')
105 ylabel('Eixo y (m)')
106 colorbar
107 hold on
108 quiver(pressaoarrumada(2:end,1),pressaoarrumada(2:end,2),Normais(:,1),Normais(:,2))
109 title("Vetores normais ao longo da borda")
110
111 figure(9)
112 plot(x,sqrt(9-(x-18).^2)+3,'—'),grid
113 hold on
114 scatter(pressaoarrumada(2:end,1),pressaoarrumada(2:end,2),40,pressaoarrumada(2:end,3))
115 title("Força no telhado")
116 xlabel('Eixo x (m)')
117 ylabel('Eixo y (m)')
118 colorbar
119 hold on
120 quiver(pressaoarrumada(2:end,1),pressaoarrumada(2:end,2),-Forca.*Normais(:,1),-Forca.*Normais(:,2))
121
122 display('A força resultante sobre o telhado, em Newtons, é')
123 FR=FR
124
125 %Exercício 2
126 %Define as variáveis necessárias para a segunda tarefa que ainda ...
    não foram
127 %definidas
128 k=0.026;
129 cp=1002;
130 Tdentro=40;
131 Tfora=20;
132 w2=1; %Ajustado do enunciado pois para 1.15 não convergia
133 T=zeros([ (H/dy)+1 1+(2*d+L)/dx]);
134 T(:,:)=20;
135
136 %Item a
137 T=Item2a(T,k,ro,cp,Tdentro,Tfora,dx,dy,H,h,L,d,V,w2,u,v,erroel);
138
139 %Plota os gráficos do Item a da segunda tarefa
140 figure(10)
141 surf(X,Y,T)
142 title("Temperatura ao longo do domínio")
143 xlabel("Eixo x (m)")
144 ylabel("Eixo y (m)")
145 zlabel("Temperatura ( C )")
146 colorbar
147

```



```

148 figure(11)
149 contourf(X,Y,T+273.15)
150 title("Temperatura ao longo do domínio")
151 xlabel("Eixo x (m)")
152 ylabel("Eixo y (m)")
153 colorbar
154
155 %Item b
156 [calor,gradientesarrumado,Normais]=Item2b(T,bordairregular,AB,psi,dx,dy,H,h,L,d,Areas)
157
158 %Plota o gráfico do Item b da segunda tarefa
159 figure(12)
160 a=zeros([1 301]);
161 a(:,:)=15;
162 c=zeros([1 301]);
163 c(:,:)=21;
164 y=sqrt(9-(x-18).^2)+3;
165 z=[a x c];
166 w=[0:0.01:3 y 0:0.01:3];
167 plot(z,w,'—'),grid
168 hold on
169 quiver gradientesarrumado(2:end,1),gradientesarrumado(2:end,2),gradientesarrumado(2:e
170 title("Vetores da transferencia de calor ao longo do silo")
171 xlabel("Eixo x (m)")
172 ylabel("Eixo y (m)")
173
174 display('O calor total retirado do galpão, em W, é ')
175 calor=calor

```