

Projeto 2 - EM570

Análise da equação da energia
em uma parede plana

Natan Silva Silveira
Rafael Augusto Braga Cota
Paulo Yoshio Kuga

Projeto 2 - EM570

**Prof. William R. Wolf
PED J. Douglas A. de Lira**

Natan S. Silveira - 184975
Paulo Y. Kuga - 204451
Rafael A. B. Cota - 243366

Campinas, Junho de 2021

1 - Introdução

O presente trabalho trata dos casos de condução de calor unidimensional em paredes planas finitas e infinitas. Para o estudo desses casos, foram desenvolvidos dois programas em MATLAB/Octave que permite que encontremos a distribuição de temperatura para qualquer instante de tempo. Nos utilizamos da Equação de Energia unidimensional, no seu caso de ausência de atrito viscoso e incompressibilidade. Nos casos dos resultados, em todos, usamos propriedades do cobre.

Neste programa, utilizamos dois métodos numéricos de resolução: o Explícito e o Implícito. Realizamos também uma análise do erro para os casos estudados, fazendo uma comparação com as soluções analíticas fornecidas no livro, servindo como um meio de validação dos programas. Também apresentamos brevemente uma formulação alternativa para o método Explícito, otimizada para programas como o Octave/Matlab, que são adequados para operações matriciais.

Ao longo do relatório, será realizado um estudo para várias condições de contorno distintas, além de considerarmos dois tipos de distribuição de temperatura inicial, uma uniforme e outra senoidal.

Apresentamos também um caso em que a condição de contorno é variante no tempo, onde podemos observar um comportamento diferenciado em relação às condições de contorno variantes no espaço.

2 - Metodologia

Utilizando-nos de dois métodos distintos de resolução numérica para equações diferenciais parciais, comparamos os resultados do caso de parede plana com convecção nas duas faces com a solução aproximada para o usuário poder decidir qual método utilizar em um outro caso prático. Tomando o número de Fourier numérico como referência (visto que ele depende da discretização do espaço e do tempo), procuramos o método mais sólido e mais genérico.

Buscamos elaborar o código cuja utilização seja a mais fácil possível para um usuário leigo. Desta forma, buscamos estruturar nosso programa de tal forma que, bastando trocar as condições iniciais desejadas, o resultado seria obtido e apresentado. Não obstante, a estruturação em switches dentro do programa, possibilita que um usuário mais experiente possa facilmente implementar novos casos dentro da metodologia.

3 - Desenvolvimento

Para o método explícito, devemos desenvolver a equação em função de apenas um termo avançado no tempo e colocar em um loop espacial para calcularmos cada ponto e depois realizarmos um loop temporal para cada instante de tempo, considerando sempre o instante de tempo anteriormente calculado. Este loop é uma edição constante de um vetor. No caso, esse vetor guarda as informações da distribuição de temperatura no tempo $n+1$. Esta foi a abordagem que utilizamos no código explícito.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \Rightarrow \frac{T_{j+1}^{n+1} - T_j^n}{\Delta t} = \alpha \left(\frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta x^2} \right) \rightarrow F_0 = \frac{\Delta T \cdot \alpha}{\Delta x^2} \Rightarrow$$

$$T_j^{n+1} = F_0 T_{j+1}^n + (1-2F_0) T_j^n + F_0 T_{j-1}^n \rightarrow \text{Loop "for"}$$

Porém, há uma abordagem alternativa que fizemos como teste, que consiste em multiplicar uma matriz de coeficientes pelo vetor já conhecido. Apresentaremos melhor esta abordagem posteriormente e na discussão, vantagens desta.

Para o desenvolvimento do método implícito, nos utilizaremos da mesma ideia, porém discretizando a derivada em x para um tempo n+1:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \alpha \left(\frac{T_{j+1}^{n+1} - 2T_j^{n+1} + T_{j-1}^{n+1}}{\Delta x^2} \right) \rightarrow T_j^{n+1} - T_j^n = F_0 (T_{j+1}^{n+1} - 2T_j^{n+1} + T_{j-1}^{n+1})$$

$$\rightarrow T_j^{n+1} = -F_0 T_{j+1}^{n+1} + (1+2F_0) T_j^{n+1} + F_0 T_{j-1}^{n+1} \rightarrow T_j^{n+1} = \begin{bmatrix} -F_0 & (1+2F_0) & -F_0 \end{bmatrix} \begin{bmatrix} T_{j-1} \\ T_j \\ T_{j+1} \end{bmatrix}^{n+1}$$

Fazendo uma sobreposição, é possível montar um sistema linear e resolvê-lo por iteração para encontrarmos as soluções em todos os instantes de tempo. Mas para a execução correta do problema, precisamos analisar o funcionamento de três condições de contorno que utilizaremos no nosso projeto: temperatura constante em uma face, troca de calor por convecção e borda adiabática.

Analisemos a temperatura constante no ponto final. Note que como já conhecemos as temperaturas no final do domínio (neste caso), simplesmente utilizamos a formulação acima e manipulamos a expressão, de tal forma a nos retornar a seguinte expressão:

$$T_L = cte \rightarrow T_{L-1} + F_0 T_L = \begin{bmatrix} -F_0 & (1+2F_0) \end{bmatrix} \begin{bmatrix} T_{L-2} \\ T_{L-1} \end{bmatrix}^{n+1}$$

Assim, podemos sobrepor na nossa matriz de coeficientes. Note que sempre nossa matriz de coeficientes, sem analisar os graus de liberdade e condições de contorno, terá dimensão n-2 por n, onde n é a discretização no espaço. Esta dimensão muda conforme achamos as condições de contorno. O desenvolvimento seguinte mostra que na verdade, o sistema a ser resolvido tem por sua matriz quadrada um n_q , onde tal número são os graus de liberdade do sistema.

Analisemos a troca de calor adiabática no início do domínio. Observe que criamos um ponto imaginário fora do domínio do problema. Porém, ressaltamos que esta abordagem não perde seu princípio físico, uma vez que no mecanismo de convecção, o calor é transferido de um ponto para fora do sistema para um dentro do sistema ou vice-versa. Desta forma, conseguimos obter um resultado mais realista e achar uma matriz $n_q = n$, pois todo nosso domínio discreto é grau de liberdade. No caso, o desenvolvimento abaixo ilustra uma situação adiabática para um caso no início do contorno.

$$\frac{-k(T_{\phi}^{n+1} - T_{-1}^{n+1})}{dx} = 0 \quad (\Rightarrow T_{-1}^{n+1} = T_{\phi}^{n+1} \rightarrow T_0^{n+1} - T_0^n = F_0(T_1^{n+1} - 2T_0^{n+1} + T_{-1}^{n+1}))$$

$$\Rightarrow T_0^{n+1} - T_0^n = F_0(T_1^{n+1} - 2T_0^{n+1} + T_{-1}^{n+1}) \quad \Rightarrow T_0^{n+1} = -F_0 T_1^{n+1} + (1+F_0) T_0^n$$

$$\Rightarrow [(1+F_0) - F_0] \begin{bmatrix} T_0^{n+1} \\ T_0^n \end{bmatrix} = T_0^n$$

Similar ao caso anterior, observe a troca de calor por convecção Definamos $Bil = h\Delta x k^{-1}$ como Número de Biot local (relativo a discretização) do problema. Note que o procedimento de solução do problema é sempre o mesmo.

$$\frac{-k(T_{L+1}^{n+1} - T_L^{n+1})}{dx} = h(T_L^{n+1} - T_{\infty}) \rightarrow T_{L+1}^{n+1} - T_L^{n+1} = Bil T_{\infty} - Bil T_L^{n+1} \rightarrow$$

$$\Rightarrow T_{L+1}^{n+1} = Bil T_{\infty} + (1-Bil) T_L^{n+1} \rightarrow T_L^n = -F_0 T_{L+1}^{n+1} + (1+2F_0) T_L^{n+1} - F_0 T_{L-1}^{n+1} \rightarrow$$

$$-F_0(Bil T_{\infty} + (1-Bil) T_L^{n+1}) + (1+2F_0) T_L^{n+1} - F_0 T_{L-1}^{n+1} = T_L^n \rightarrow [-F_0 (1+F_0(1+Bil))] \begin{bmatrix} T_{L-1}^{n+1} \\ T_L^{n+1} \end{bmatrix} = T_L^n + F_0 Bil T_{\infty}$$

Desta forma, montamos nosso programa com base nestas premissas. Note que unindo o caso anterior com o atual, temos as condições de contorno do problema da Seção 5.5 do livro, que trata de parede plana finita com convecção em ambas as faces. Com um algebrismo similar ao do caso adiabático, obtemos a condição de contorno para o início do domínio da parede do lado esquerdo para a Seção 5.6 - Parede Infinita.

Antes de concluirmos esta seção, voltemos ao caso explícito. Observe o seguinte desenvolvimento. (Nota: está escrito K, mas é uma matriz $n_q \times n_q$).

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \frac{\alpha(T_m^n - 2T_j^n + T_{j-1}^n)}{\Delta x^2} \Rightarrow T_j^{n+1} = F_0 T_m^n + (1-2F_0)T_j^n + F_0 T_{j-1}^n$$

Suponha o caso convectivo: $T_{-1}^n = (1-F_0(1+B,I))T_0^n + F_0 T_1^n + B,I F_0 T_\infty$
 $T_m^n = F_0 T_{-1}^n + (1-F_0(1+B,I))T_L^n + B,I F_0 T_\infty$

$$\Rightarrow \begin{bmatrix} (1-F_0(1+B,I)) & F_0 & 0 & \dots \\ F_0 & (1-2F_0) & F_0 & \dots \\ 0 & F_0 & \dots & \vdots \\ \vdots & \vdots & \ddots & F_0 \\ 0 & 0 & \dots & F_0 - (1-F_0(1+B,I)) \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ \vdots \\ T_L \end{bmatrix}^n + \begin{bmatrix} B,I F_0 T_\infty \\ 0 \\ \vdots \\ B,I F_0 T_\infty \end{bmatrix} =$$

$$T_{-1}^{n+1} \rightarrow K T_{-1}^n + T_\infty = T_{-1}^{n+1} \xrightarrow{n=1} K T^0 + T_\infty = T^1 \rightarrow$$

$$n=2: K(KT^0 + T_\infty) + T_\infty = T^2 \xrightarrow{n=1} K^2 T^0 + T_\infty(K+I) = T^2 \rightarrow$$

$$T^3 = K(K^2 T^0 + T_\infty(K+I)) + T_\infty = K^3 T^0 + (K^2 + K + I)T_\infty \Rightarrow n=3 \rightarrow$$

$$K^3 T^0 + \left(\sum_{i=0}^2 K^i\right) T_\infty = T^3$$

Em palavras, note que transformamos todas as equações possíveis em n em um sistema matricial de multiplicação e soma de matrizes e vetores. Note que desta forma, obtemos uma equação para o tempo ξ de tal forma que apenas o tendo, podemos obter a distribuição de temperatura no mesmo, pois conhecemos todos os outros fatores. É computacionalmente custoso caso este tempo seja muito grande. Porém observe que ele é independente da iteração e ele executa operações matriciais diretamente. Discutiremos posteriormente possibilidades de implementação deste código e porque ele é importante para a implementação no Octave/MATLAB.

4 - Resultados e Discussões

4.1 - Influência número de Fourier e de seus componentes

Antes de analisarmos os resultados obtidos pelos métodos numéricos na resolução dos problemas propostos, precisamos entender a importância de outros componentes da resolução para melhor entendimento de como interpretar os resultados obtidos posteriormente. Logo, nós iremos analisar qual a influência do número de Fourier, o passo no tempo e a quantidade de pontos dentro da discretização do problema.

Primeiramente, para toda a análise desta seção, nós iremos nos basear no problema transiente com condição de contorno de convecção, o qual temos acesso tanto a soluções numéricas quanto a solução analítica. Sendo que, os parâmetros usados como referência serão difusividade térmica igual a $1,18e-4 \text{ m}^2/\text{s}$, passo no tempo igual a $4,95 \text{ s}$ e distância entre pontos igual a $9,9e-3 \text{ m}$.

Assim sendo, o primeiro componente que iremos analisar é a influência do passo no tempo, os gráficos que estão apresentados logo abaixo demonstram a resolução do

problema usando o método numérico implícito tanto para o caso inicial quanto para um caso no qual o passo no tempo foi alterado para 20 segundos:

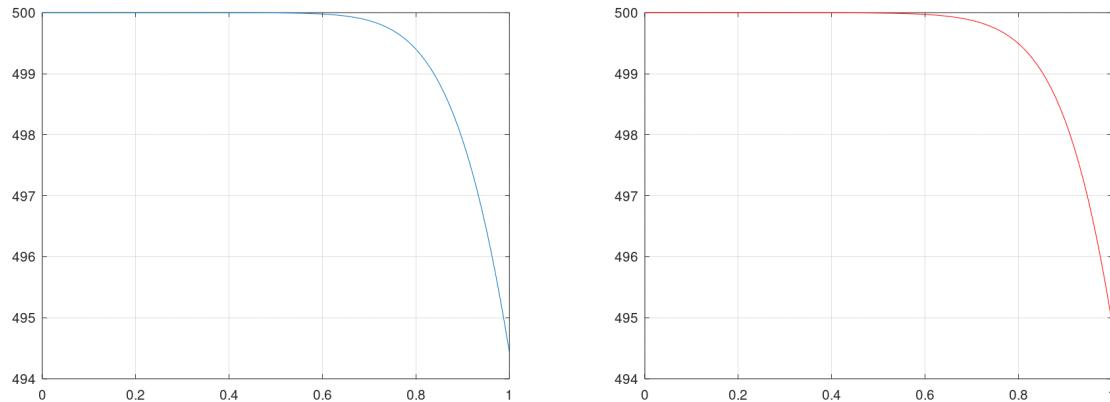


Figura 1 e 2. Gráficos com distribuição de temperatura para passos no tempo 4.95s (linha azul) e 20s (linha vermelha) para método implícito.

Podemos ver que há uma perceptível diferença entre a distribuição de temperatura dos dois gráficos, sendo que foi escolhido arbitrariamente o tempo de 100 s decorridos desde o início da troca de calor. Logo, é possível notar que quanto mais refinado for o passo no tempo, ou seja, quanto menor for esse passo, mais preciso será o resultado obtido. Isso se deve ao fato de quanto maior for o passo menos interações ocorrerão e consequentemente levarão a resultados com aproximações mais grosseiras.

Do mesmo modo, ao analisarmos a influência da quantidade de pontos obteremos resultados bem diferentes, como os apresentados abaixo:

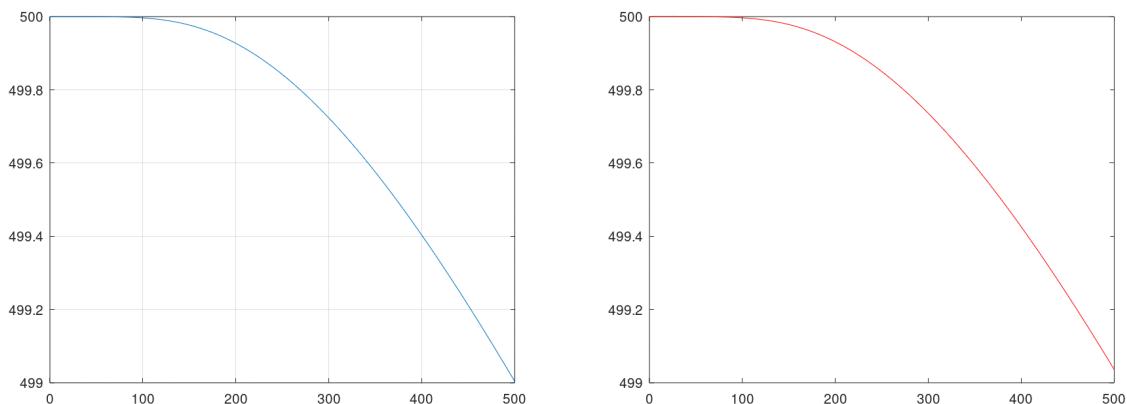


Figura 3 e 4. Gráficos com temperatura ao longo do tempo para ponto no meio da parede para 101 pontos no espaço (linha azul) e 50 pontos no espaço (linha vermelha).

Vemos que a quantidade de pontos altera o resultado obtido, a partir da análise do gráfico, para a mesma situação na qual foi retirada a temperatura de um ponto no meio da parede ao longo do tempo de simulação, podemos afirmar que assim como o passo no

tempo, a quantidade de pontos altera a qualidade dos resultados por causa dos mesmos motivos apresentados acima.

Por outro lado, ao analisarmos a influência do número de pontos no tempo para a solução do método explícito, percebemos que a precisão não é alterada. No exemplo abaixo, podemos ver como a solução para 200 e 400 pontos no tempo é idêntica. A única coisa que perdemos com a utilização de menos pontos no espaço para o método explícito são os detalhes de como é a distribuição de temperatura em outros instantes de tempo.

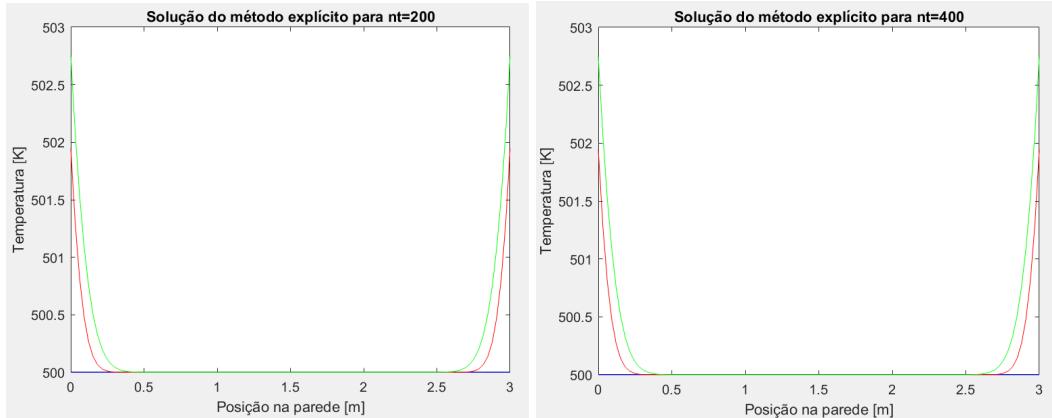


Figura 5 e 6. Gráficos com distribuição de temperatura para 200 e 400 pontos no espaço para o método explícito.

Outra análise que podemos fazer é com relação a influência do número de Fourier na resolução do método numérico explícito. Sabemos que este método só converge para uma solução quando o $Fo < 0,5$, sendo essa uma desvantagem para o uso desse método. Segue abaixo um exemplo de como não obtemos uma solução para $Fo > 0,5$, onde consideramos $Fo = 5,5284$.

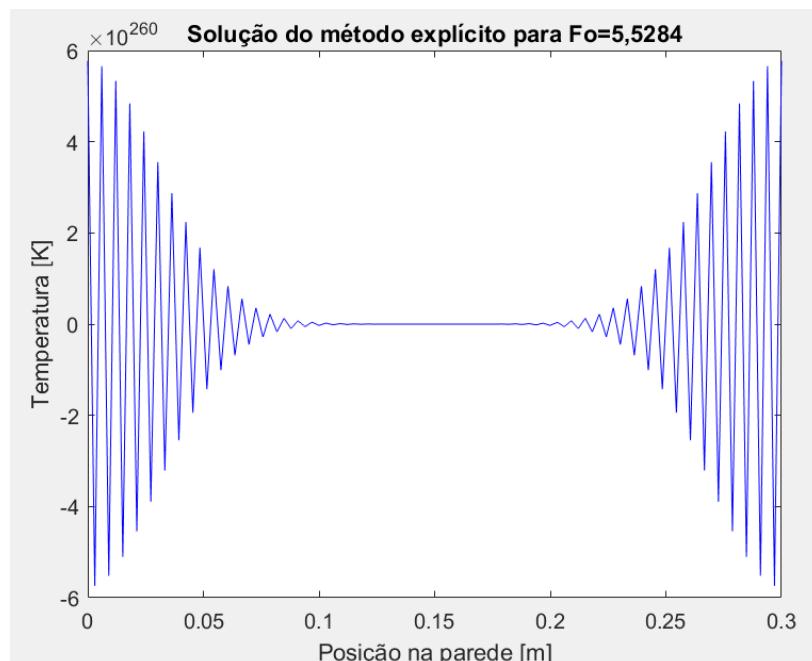


Figura 7: Gráfico mostrando a divergência do método numérico explícito para Fourier maior do que 0,5.

Portanto, é possível concluir que os dois componentes que estudamos acima são de grande proeminência no número de Fourier, tendo um importante papel na qualidade dos resultados obtidos, sendo melhor, sempre que possível, recorrer a componentes mais refinados.

4.2 - Soluções analítica, explícita e implícita para parede finita com convecção nas faces ($x=0$ e $x=L$)

Agora que já é conhecido como outros fatores podem alterar, podemos partir para apresentação dos resultados obtidos em diferentes problemas propostos, sendo eles o problema de convecção na parede e na parede infinita, sendo que os dois problemas serão apresentados os resultados da resolução analítica, explícita e implícita.

Desse modo, a resolução do problema de convecção nas paredes, sendo a parede de cobre e possuindo a distribuição de temperatura uniforme inicialmente gerou estes resultados:

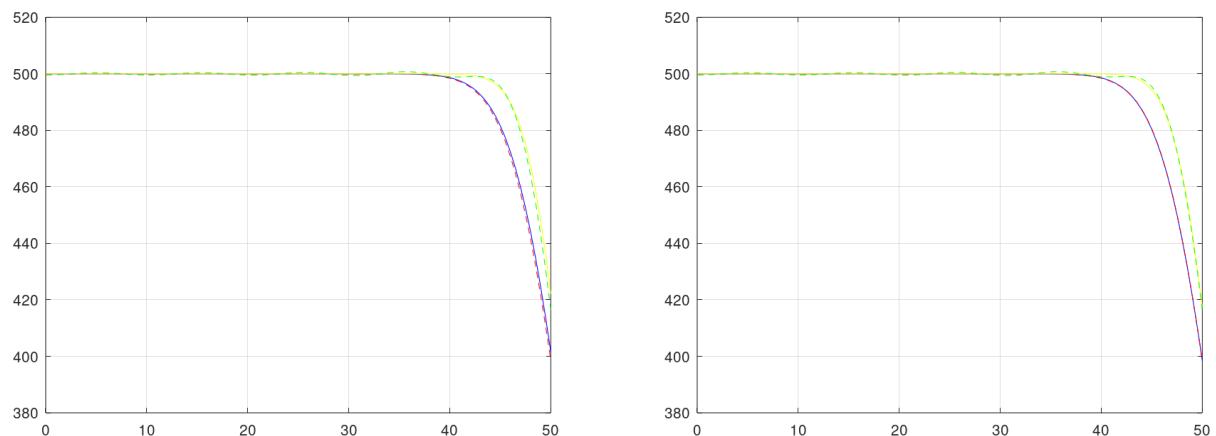


Figura 8 e 9: Distribuição de temperatura para metade da parede com convecção calculada no modo analítico (linha tracejada), implícito (gráfico da esquerda) e explícito (gráfico da direita) para dois instantes de tempo, com temperatura inicial uniforme.

A visualização dos gráficos é realizada somente para metade da parede, pois o problema é de natureza simétrica o que possibilita a simplificação de apresentação dos dados. Ao analisarmos o gráfico é possível perceber que tanto a resolução feita pelo método implícito quanto o método explícito apresentam grande proximidade com a resolução analítica, o que indica que os códigos apresentados têm a capacidade de representar o problema com uma precisão boa. Isso fica mais evidente quando pegamos um quadro mais próximo dos valores obtidos:

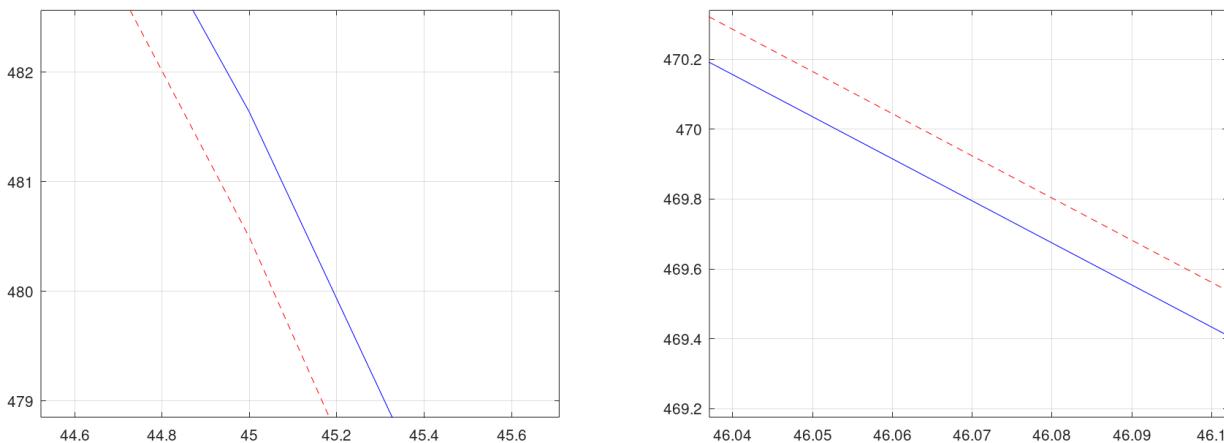


Figura 10 e 11: Pontos de temperatura da parede com convecção calculada no modo analítico (linha tracejada), implícito (gráfico da esquerda) e explícito(gráfico da direita).

No qual o erro geral obtido para resolução implícita foi de 0,004776 % e o erro geral obtido para a resolução explícita foi de 0,00391 %. Logo, os dois métodos de resolução demonstram-se bem fidedignos a proximidade com a resolução analítica.

4.3 - Soluções analítica, explícita e implícita para parede infinita com convecção em sua face ($x=0$)

Para a nossa análise inicial do caso da parede infinita, consideramos que essa parede possui um comprimento de 300 m, com condição de contorno de convecção em $x=0$ e condição de contorno de temperatura sempre constante com a variação do tempo para $x=L$ ($T(x=L,t) = 500 \text{ K}$). Os demais dados de entrada que utilizamos foram: temperatura do fluido igual a 600 K; tempo igual a $8,5 \cdot 10^6 \text{ s}$; temperatura inicial constante em 500 K; propriedades térmicas do cobre.

Os resultados utilizando esses parâmetros podem ser observados abaixo, onde são mostrados os gráficos para os três métodos em três instantes de tempo diferentes.

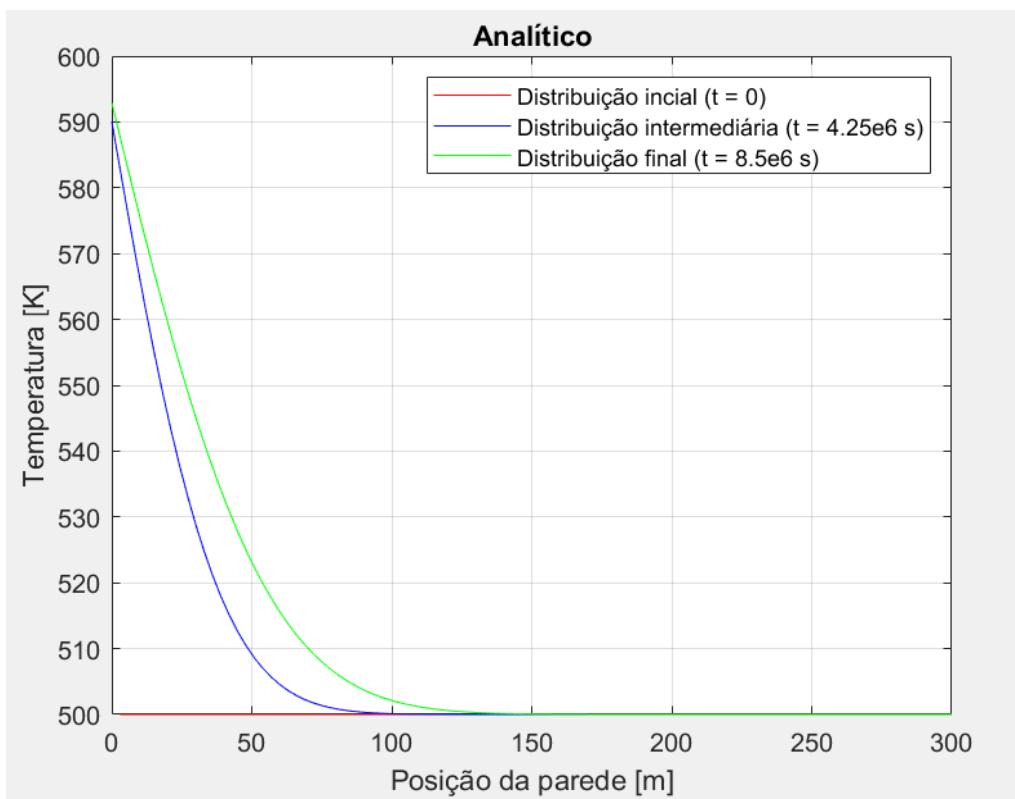


Figura 12: Distribuição de temperatura para a parede infinita calculada analiticamente para três instantes de tempo, com temperatura inicial uniforme.

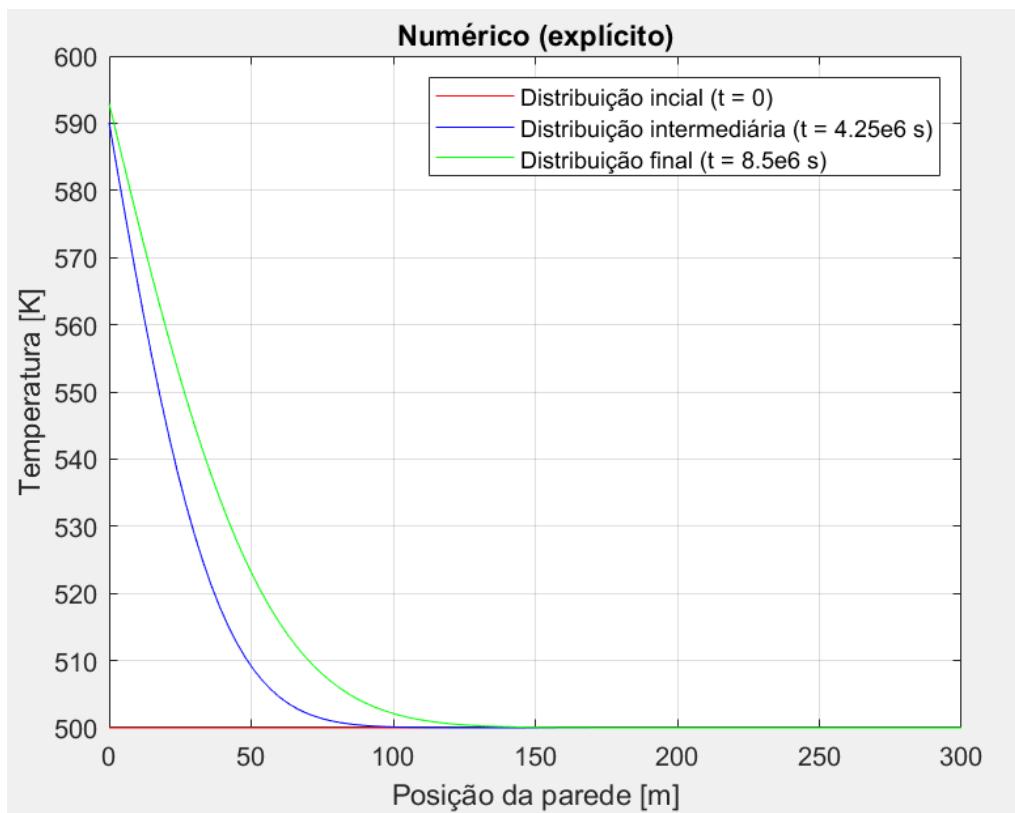


Figura 13: Distribuição de temperatura para a parede infinita calculada numericamente (método explícito) para três instantes de tempo, com temperatura inicial uniforme.

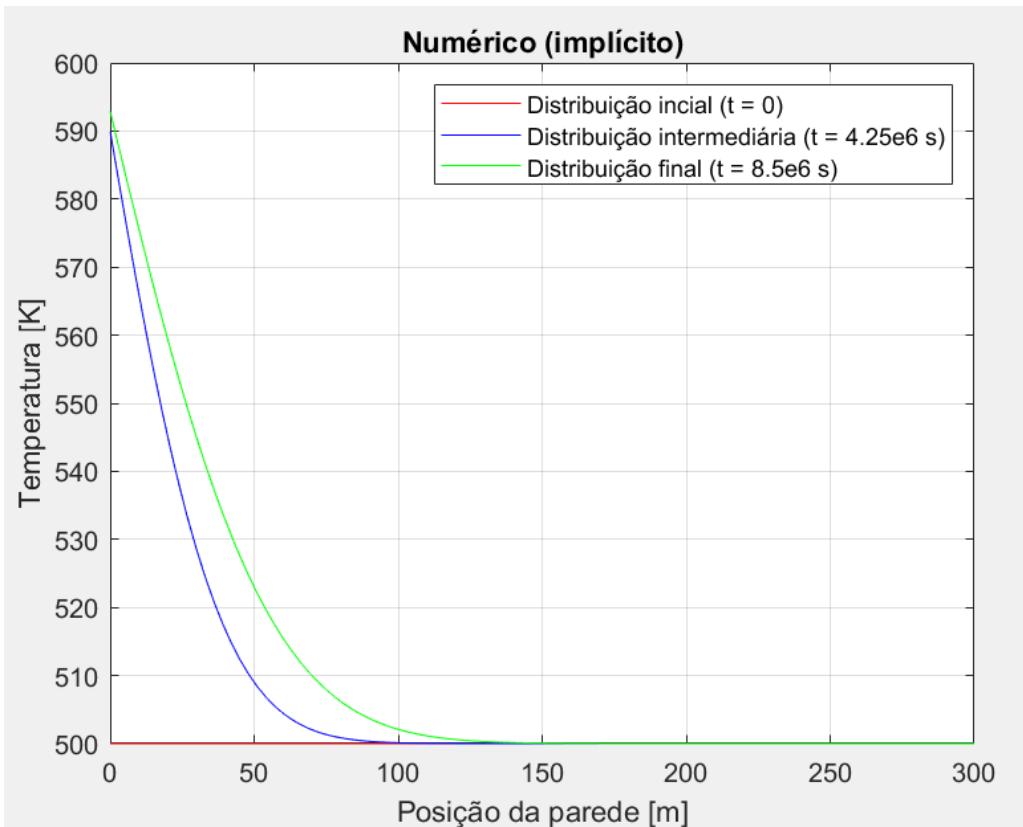


Figura 14: Distribuição de temperatura para a parede infinita calculada numericamente (método explícito) para três instantes de tempo, com temperatura inicial uniforme.

Se realizarmos uma análise do erro global, temos os seguintes valores: para o caso explícito, temos um erro de 0,0010 % para a solução intermediária e 0,0007 % para a solução final, validando assim o método explícito; já para o caso implícito, temos um erro de 0,0099 % para a solução intermediária e 0,0070 % para a solução final, sendo o método implícito também validado.

Outro caso que realizamos o estudo foi o caso em que a temperatura inicial na parede possuía uma distribuição senoidal. Os resultados podem ser vistos nos gráficos abaixo.

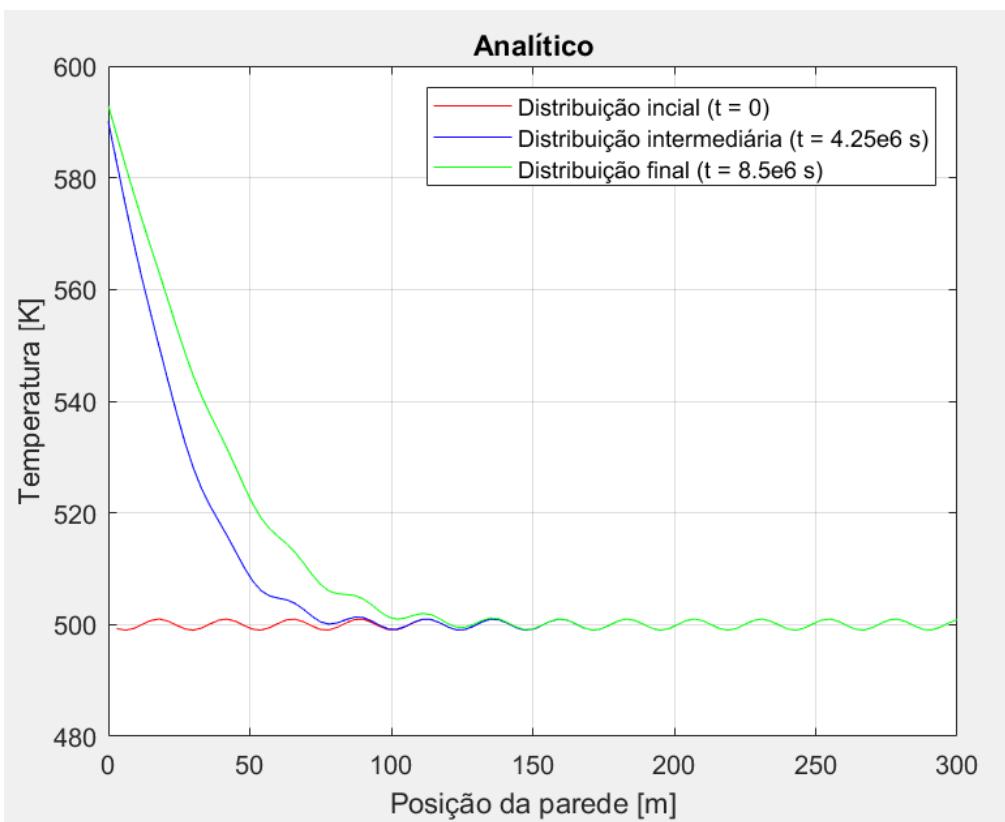


Figura 15: Distribuição de temperatura para a parede infinita calculada analiticamente para três instantes de tempo, com temperatura inicial em forma senoidal.

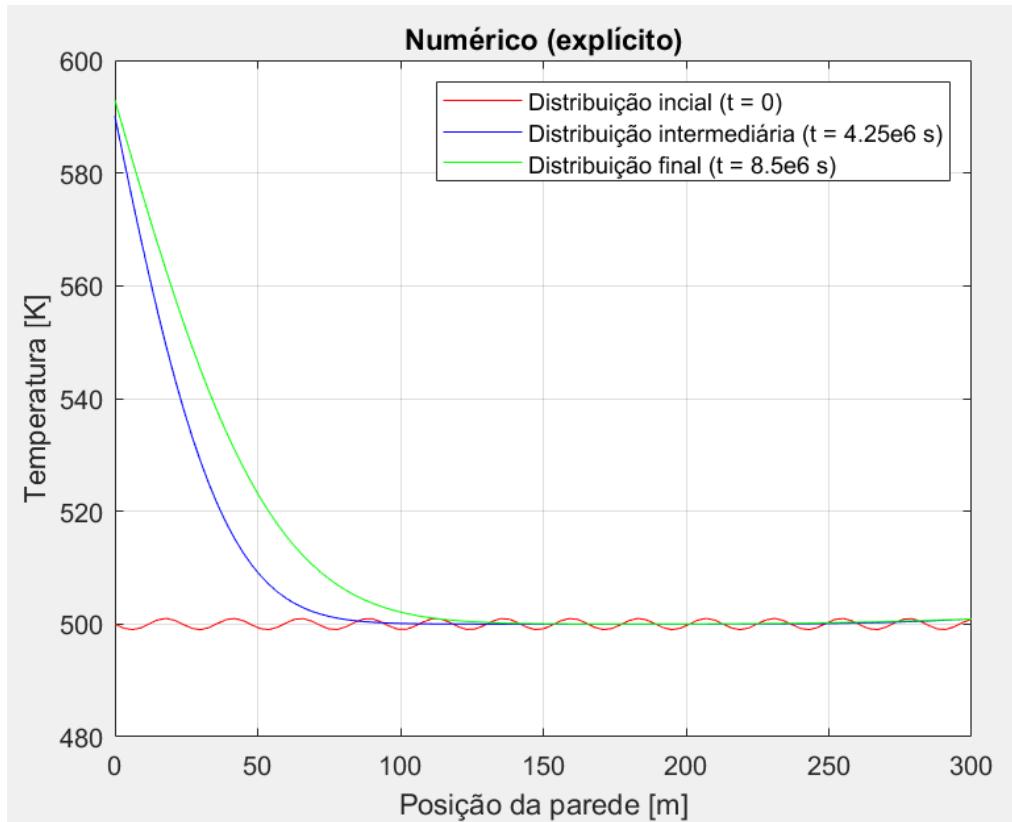


Figura 16: Distribuição de temperatura para a parede infinita calculada numericamente (método explícito) para três instantes de tempo, com temperatura inicial em forma senoidal.

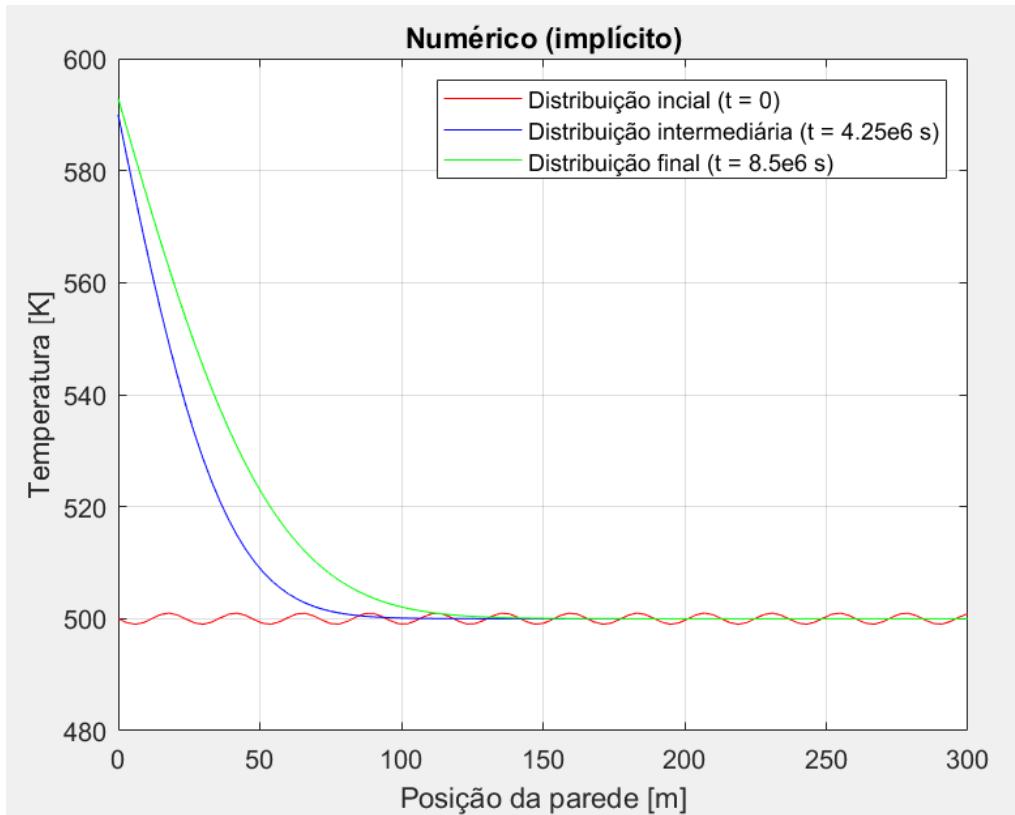


Figura 17: Distribuição de temperatura para a parede infinita calculada numericamente (método implícito) para três instantes de tempo, com temperatura inicial em forma senoidal.

Já para este caso, obtemos a seguinte análise do erro global: para o caso explícito, temos um erro de 0,0163 % para a solução intermediária e 0,0218 % para a solução final, validando assim o método explícito; já para o caso implícito, temos um erro de 0,0100 % para a solução intermediária e 0,0070 % para a solução final, sendo o método implícito também validado. Caso queiramos diminuir esses erros, basta discretizarmos a parede com um número maior de pontos, ou analogamente, diminuir o intervalo entre os pontos.

A distribuição senoidal utilizada segue a seguinte função: $T(x, t = 0) = \sin(m * x) + T_1 + (T_2 - T_1) * (x/(2 * L))$, onde T_1 , T_2 e m são parâmetros pré-definidos que definem a forma da distribuição. Nos casos apresentados acima, utilizamos $T_1=T_2=500$, e $m=50$. Variando T_1 e T_2 , variamos onde se inicia e termina a distribuição, enquanto o m altera o tamanho do comprimento de onda dessa distribuição. Para ilustrar, foi alterado os valores desses parâmetros para: $T_1 = 550$, $T_2=500$ e $m=500$. O resultado pode ser observado no gráfico abaixo.

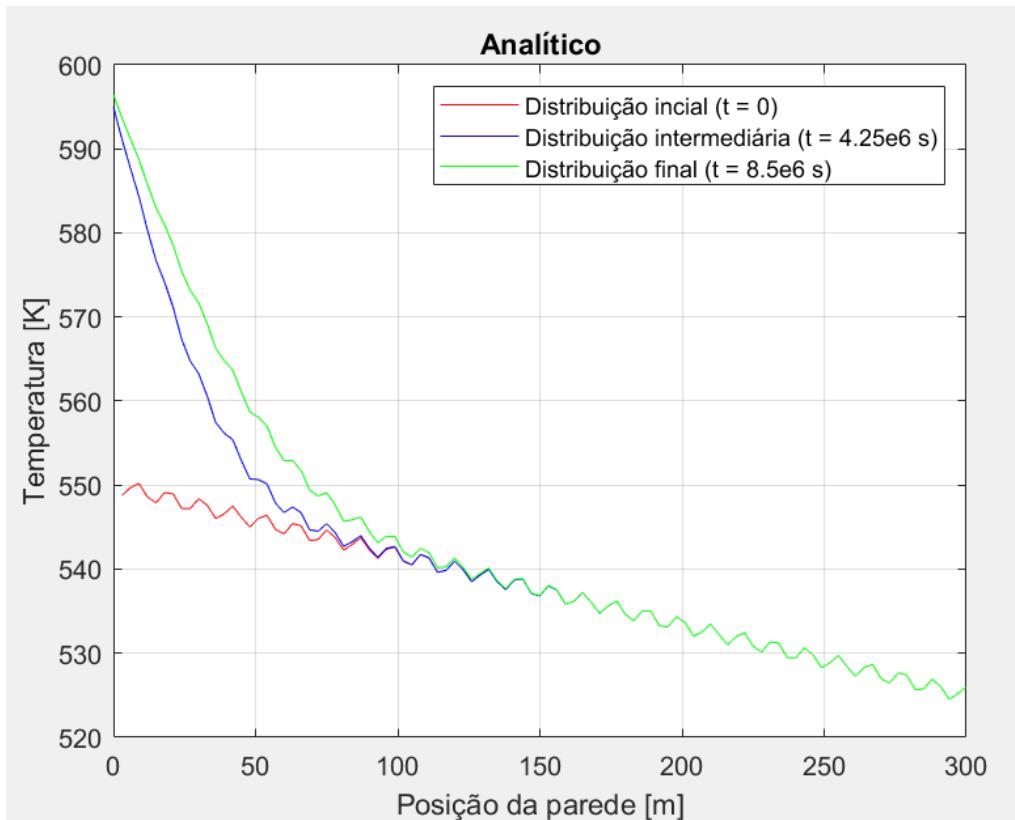


Figura 18: Distribuição de temperatura para a parede infinita calculada analiticamente para três instantes de tempo, com temperatura inicial em forma senoidal com diferentes valores de T_1 , T_2 e m .

Após validarmos os programas numéricos comparando-os com o programa analítico, podemos variar as condições de contorno do problema com uma certa confiança de que a solução será bem próxima da realidade.

O próximo caso analisado difere apenas na condição de contorno em $x=0$, onde passa de convecção para temperatura constante ($T(x=0,t) = 300$ K). Os resultados podem ser observados a seguir, para distribuição inicial uniforme e senoidal.

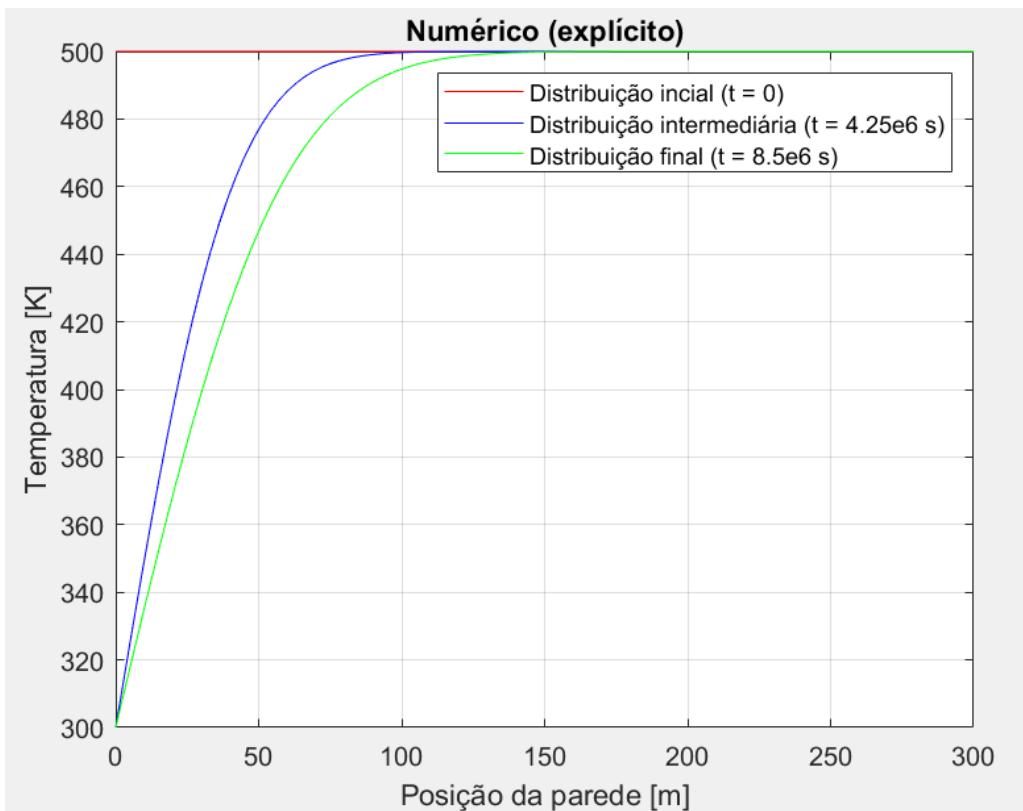


Figura 19: Distribuição de temperatura para a parede infinita calculada numericamente (método explícito) para três instantes de tempo, com temperatura inicial uniforme.

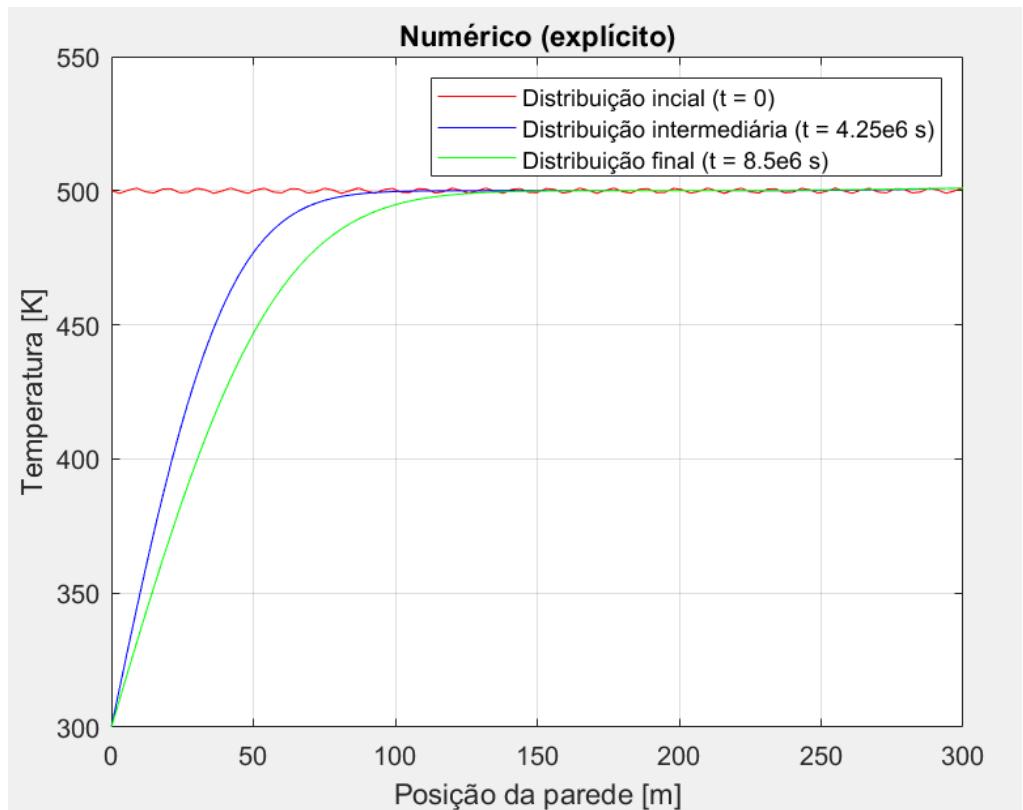


Figura 20: Distribuição de temperatura para a parede infinita calculada numericamente (método explícito) para três instantes de tempo, com temperatura inicial em forma senoidal.

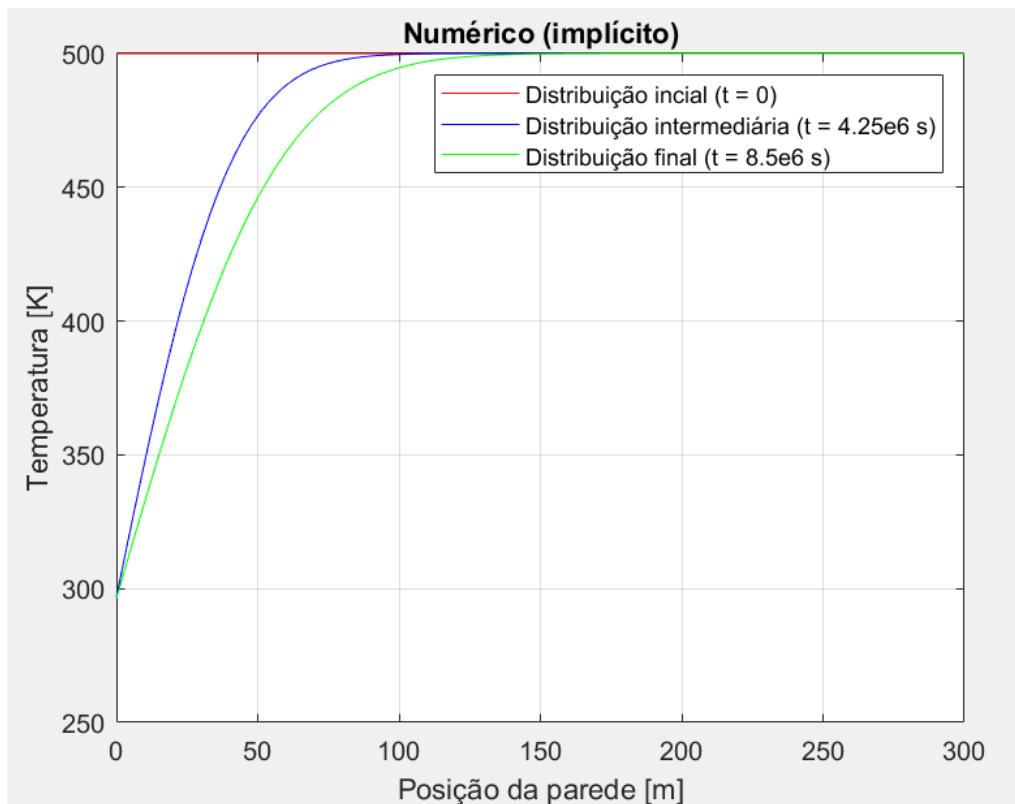


Figura 21: Distribuição de temperatura para a parede infinita calculada numericamente (método implícito) para três instantes de tempo, com temperatura inicial uniforme.

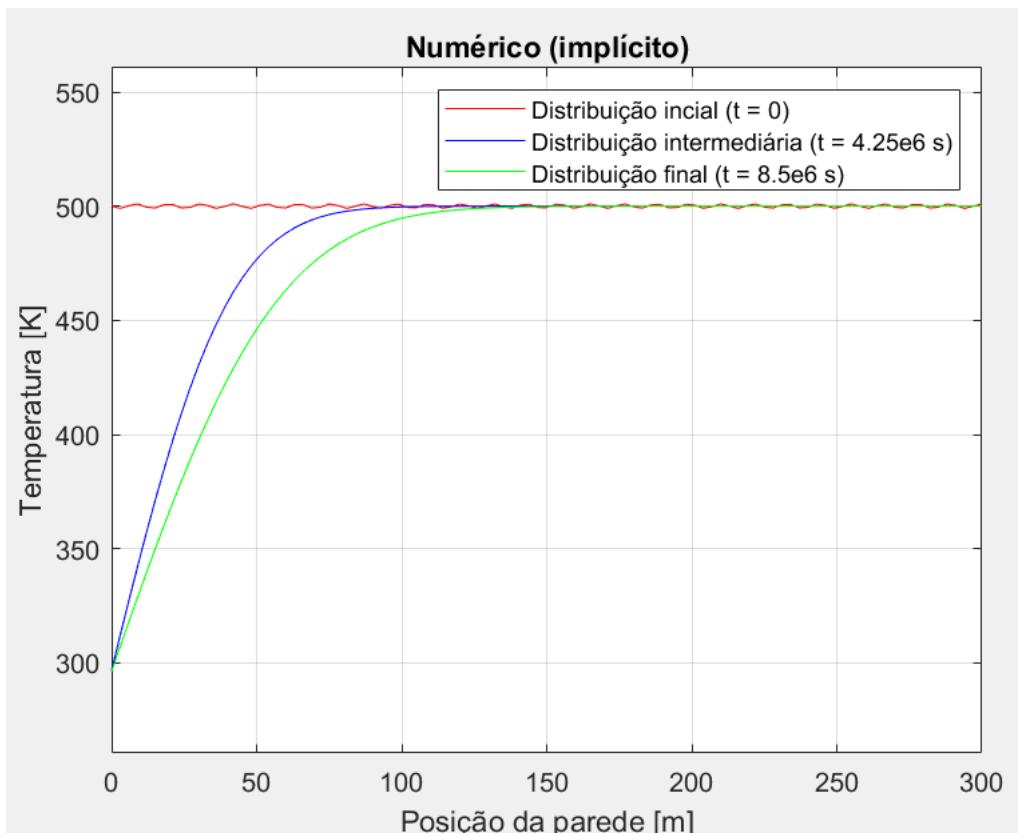


Figura 22: Distribuição de temperatura para a parede infinita calculada numericamente (método implícito) para três instantes de tempo, com temperatura inicial em forma senoidal.

4.4 - Outros testes

Para efeito de compreensão de outros efeitos, realizamos outros testes para executar o algoritmo implícito. Escolhemos este por ser mais robusto e suportar soluções para qualquer Fo .

4.4.1 - Transição - Regime transiente para regime permanente

A primeira solução é uma parede de temperatura constante em ambos os lados. Porém a distribuição de temperatura nela é diferente das temperaturas nas pontas.

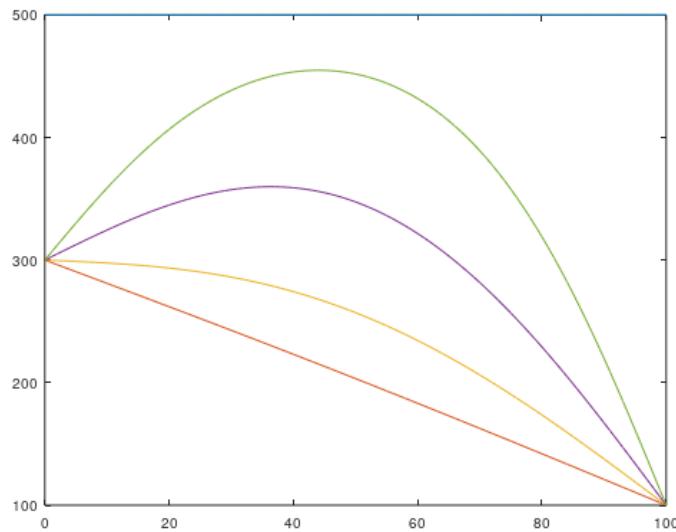


Figura 23: Caso de resfriamento de parede plana com duas temperaturas diferentes no contorno.

Observe que o caso converge para o caso de distribuição de calor em parede plana com temperatura constante, como foi estudado na matéria. Em outras palavras, o sistema atinge o regime permanente.

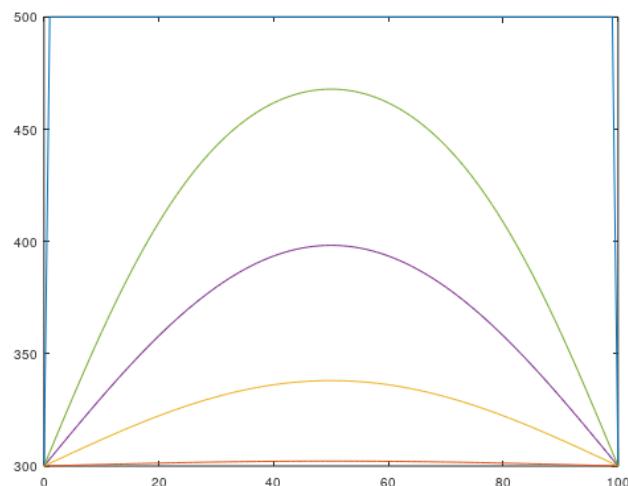


Figura 24: Caso de temperatura constante nas duas pontas.

A segunda solução é caso a barra tenha sido colocada dentro de um material de temperatura constante em suas duas pontas.

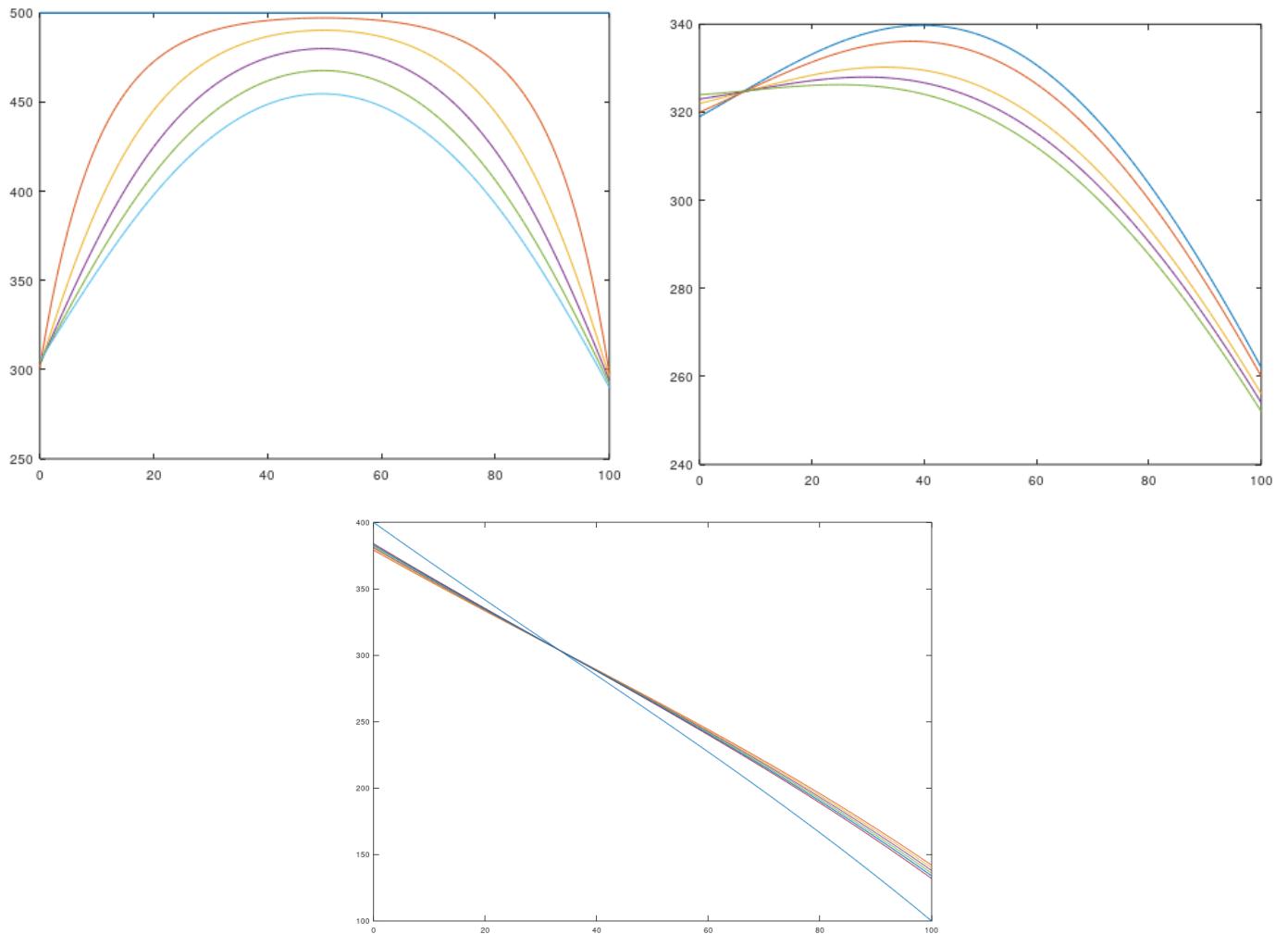
É esperado que a barra, com o passar do tempo, atinja seu resfriamento por completo, de tal forma a todos seus pontos entrarem em equilíbrio térmico com o ambiente.

4.4.2 - Temperatura modelável no tempo

Suponha que a distribuição de uma condição de contorno na borda possa ser modelada no tempo. O programa implícito suporta uma implementação vetorial. Suponha que tenhamos uma ascensão linear. Teremos um resultado da figura 25.

Temos um resultado assim nos primeiros instantes. Observe a temperatura variando nas pontas, junto com o decaimento da temperatura da barra. Após um tempo, observe que a temperatura na ponta esquerda supera o potencial original, como mostra a figura 26.

Próximo ao final da análise, ele começa a se tornar um regime permanente, assim como vimos no primeiro caso de transição, figura 27.



Figuras 25,26 e 27 - Da esquerda para direita, e para baixo. As temperaturas modeláveis no tempo em tempos diferentes.

É possível testar outras condições de contorno, como um decaimento exponencial da temperatura. Aqui iremos plotar um gráfico de superfície, para a melhor visualização total do fenômeno:

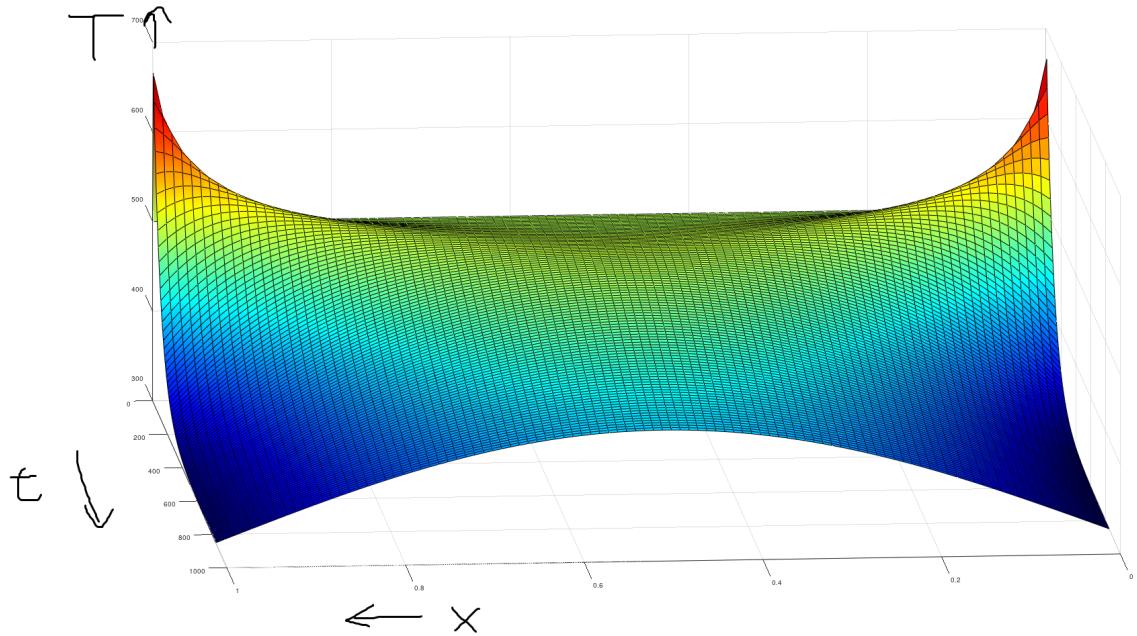


Figura 28 - Eixo a eixo simbolizado. Observe a distribuição no espaço x,t,T (espaço, tempo, temperatura)

Observe que, essa variação no potencial nas bordas ocasiona formas interessantes e complexas a serem resolvidas analiticamente. Observe agora uma condição de contorno em que temos um ruído senoidal conhecido variante no tempo:

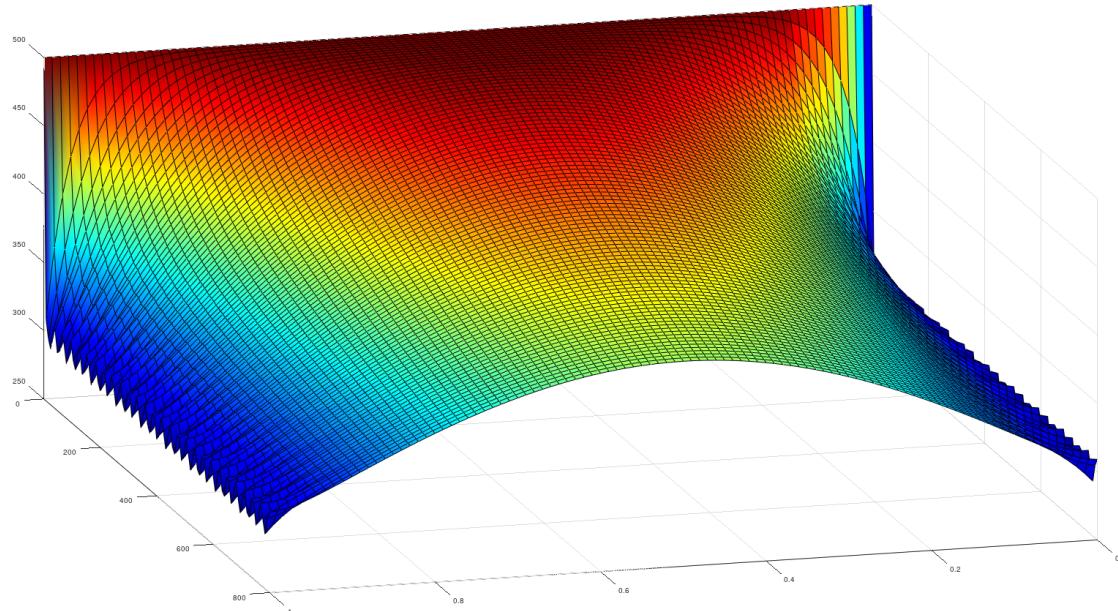


Figura 29 - Caso senoidal de distribuição

Observe em detalhe uma parte da borda:

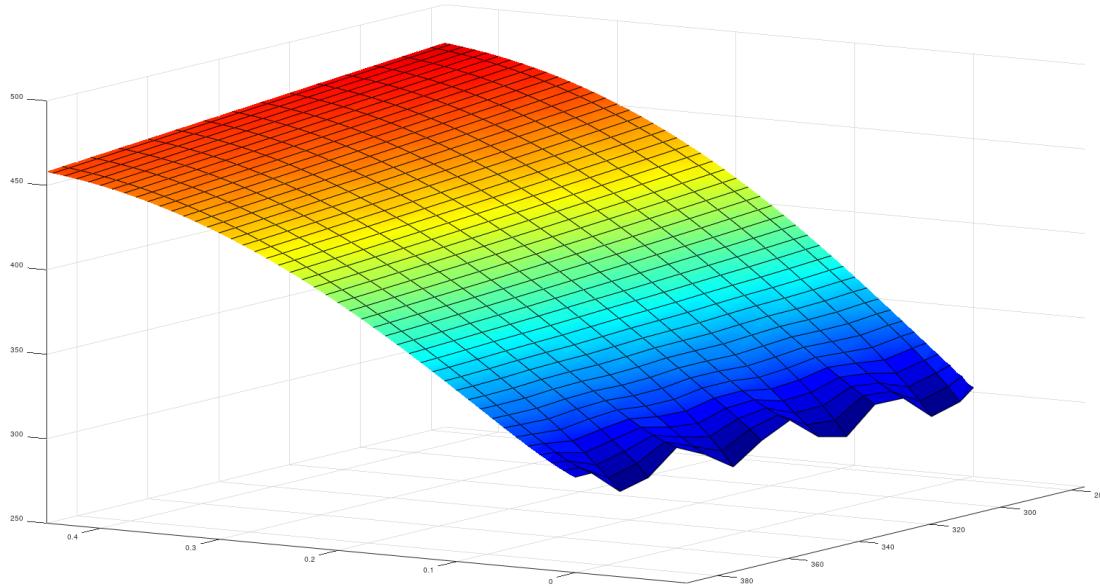


Figura 30 - Borda em detalhe da distribuição.

Desta forma, esta é uma demonstração que é possível utilizar o programa para resolver problemas de contorno que variem no tempo.

4.4.3 - Implementação vetorial

Sabemos que o Octave é otimizado para multiplicações matriciais. O método explícito escrito como multiplicação de matrizes, rende resultados mais rápidos em comparação com o método de edição de array. Implementamos de uma forma recursiva, de tal maneira a ele acessar sempre seu último resultado gerado para computar o próximo.

Desenvolvemos apenas para o caso da convecção o método explícito por multiplicação de matrizes. No mesmo programa que acompanha este relatório, o testeexp.m, demonstra utilizando a função tic-toc para nos retornar o tempo de processamento em segundos.

Para um caso de uma parede com 10 metros, de cobre, discretizada em 101 pontos, em um tempo de processo de 3000 segundos, com 1001 pontos na discretização, temos os seguintes resultados:

```
Fo = 0.035655
tempo_matricial = 0.071381
tempo_array_edit = 0.8479
>> |
```

Figura 31 - Print da tela do testeexp.m após a execução

Para 100 segundos, 1001 pontos de espaço e 3001 pontos de tempo, temos o seguinte resultado:

```

JANELA DE COMANDOS
Fo = 0.039617
tempo_matricial = 9.0074
tempo_array_edit = 25.213
>> |

```

Figura 32 - idem a Figura 31, para o caso 1001x3001

Observe que é quase três vezes mais rápida a abordagem otimizada do que a abordagem de edição de array.

É fora do escopo deste trabalho discutir em detalhes técnicos a implementação do Octave em C++ que permite a agilização de operações matriciais. Porém, em suma, podemos pensar que o número de operações feitas em uma multiplicação matricial é menor que o número de operações feitas em uma edição de busca por ponteiro na memória (imagina-se que seja um ponteiro em virtude da utilização do índice).

Desta forma, apresentamos esta abordagem otimizada para o método explícito, que posteriormente, em trabalhos futuros, poderá ser desenvolvida.

4.4.4 - Hipóteses teóricas sobre paralelização no modo matricial para o caso explícito:

Encontramos algumas possibilidades computacionais não disponíveis para teste, porém aparentam funcionar na teoria. Supomos um conjunto de computadores que possua uma memória e barramento coletivos, de tal forma que eles possuam acesso simultâneo (por threading, provavelmente).

Em todas as nossas implementações, para gerar o gráfico, salvamos as informações que obtivemos em uma matriz. Então, estes computadores teriam acesso a essa matriz pela memória compartilhada.

Uma vez que a fórmula de recorrência permite independência, poderíamos ter este conjunto computacional executando tarefas por passo, e.g. computador 1 executa do 1 ao 3, computador 2 do 4 ao 6, o computador 3 do 7 ao 9 e etc... Isso melhoraria o tempo de resolução do problema consideravelmente. Porém, note que nessa estratégia poderíamos ter um gargalo, pois uma vez que o tempo se torna muito grande, gerar A^k começa a se tornar complicado. Porém, isso depende do tamanho de A.

Entretanto, como computadores são finitos, poderíamos unir uma estratégia de memória compartilhada. Suponha dois computadores. O computador 1 poderia ter acesso ao resultado que o computador 2 gerou, para executar o caso 7, 8, 9, apenas aplicando a primeira recorrência não genérica apresentada no desenvolvimento. Em outras palavras, o computador 2 gerará a matriz A^6 . O computador 1 apenas executa $A^*(A^6)$ e já tem A^7 .

Podemos pensar em outras implementações. Suponha novamente um caso com dois computadores. O 1 inicia executando o tempo 1 e o 2 inicia executando o tempo 3. Uma vez que as duas operações são realizadas, o 2 executa o tempo 2 e o 1 executa o tempo 4. Este não necessariamente precisa ser um conjunto computacional. Acreditamos que um tipo de operação pode ser feita em duas GPU's, por exemplo, dada a possibilidade de calcular os termos de uma matriz por paralelismo.

As possibilidades com este método são variadas e possibilitam a redução do tempo de execução, se corretamente implementadas. No entanto, ressaltamos que o recurso computacional é caro e o implementador deve tomar cuidado para evitar gargalos.

5 - Conclusões

Notamos que o Número de Fourier é critério fundamental na escolha do método. Um número superior a 0,5 compromete a convergência do método explícito. De forma geral, o método implícito é mais robusto, porém mais complicado de ser implementado, além de ter sua precisão afetada pelo número de pontos temporais. O método explícito é mais fácil de ser pensado e não sofre prejuízo em sua precisão com uma alteração do número de pontos temporais, onde a resolução do problema dentro de um $Fo < 0.5$ é facilmente implementada. Não obstante, o método explícito permite a paralelização, como mostrado no desenvolvimento da forma geral para o tempo ζ em forma vetorial.

A discretização das malhas implica diretamente no Número de Fourier. Implicamos que as condições do problema, como tempo de análise, material, são dependentes das causas naturais ou artificiais de produção do ser humano. A escolha de um ou outro método depende de se é possível ou não gerar o método. Portanto, o usuário deve estar atento a estas condições que regem o problema e em especial o número de Fourier. Deve, o usuário, analisar também, a urgência em que se necessita da resposta e recurso computacional disponível.

Ambos os métodos se comprovaram muito realísticos com relação aos resultados se comparados ao método analítico. O método explícito apresenta melhor capacidade de implementação na sua forma de edição de array e é muito mais intuitivo do que o implícito, além da possibilidade que encontramos de sua implementação em paralelo e independência dos resultados em iteração. O método implícito, em seu núcleo, é mais difícil de ser implementado e desenvolvido, porém sua implementação para diversos casos, como executamos nesta metodologia modular de desenvolvimento do código, se mostrou muito mais eficiente para aderir a outras implementações. Em outras palavras, uma vez que o núcleo do código foi feito, a implementação de diferentes condições de contorno se tornou mais fácil. Mesmo com uma distribuição não uniforme, os algoritmos se mostraram sólidos em sua resolução.

Desta forma concluímos que a melhor abordagem depende do problema. Não há um método ou outro que dê um melhor ou pior resultado. Tudo dependerá das condições gerais do problema, inclusive as de recurso e tempo que o responsável tem para resolver. Caso precise de um código mais rápido para ser implementado e o Fo seja menor que 0,5, o implementador pode escolher o explícito. Caso precise testar diversas condições de contorno, o mais adequado é o implícito. Portanto, fica a cargo do usuário utilizar-se de seu senso crítico, analisando todos os fatores correlacionados ao seu problema e objetivando a resolução do mesmo nos prazos e condições possíveis.