

# Um problema inverso para obtenção de distribuição de Temperatura

Parte 1 - EP - MAP3121 - Prazo de entrega: 10 / 05

April 5, 2020

## Regras do Jogo

- Você deve implementar o exercício programa em C/C++ ou Python3.x
- Python:
  - Pode usar: Matplotlib, NumPy (apenas para trabalhar com aritmética de vetores, matrizes, leitura/escrita de dados), bibliotecas básicas auxiliares: sys, time, datetime, os, math.
  - **Não** pode usar: SciPy ou outras bibliotecas de algebra linear computacional
- C, C++:
  - **Não** pode usar recursos de versões além de C/C++14.
  - Pode usar qualquer biblioteca nativa do gcc/g++ (que não exija instalação adicional).
- Incluir, obrigatoriamente, um arquivo LEIAME.txt com instruções de compilação e execução, indicando versão de interpretador/compilador necessário.
- O exercício pode ser feito em duplas, não necessariamente da mesma turma.
- Apenas um aluno deve entregar o exercício, destacando no relatório e código o nome de ambos os alunos.
- A entrega deve conter o relatório (em .pdf), contendo a análise do problema estudado, e o código usado para as simulações computacionais (arquivos fonte). A entrega deve ser feita em um arquivo compactado único.
- O relatório deve apresentar resultados e análises de todas as tarefas descritas neste enunciado.
- O seu código deve estar bem documentado, de forma a facilitar a correção. Rodar os testes também deve ser fácil para o usuário do seu programa, sem que este tenha que editar seu código. Ou seja, você deve pedir como entrada qual teste o usuário quer rodar, qual método e os parâmetros para o teste.

## 1 Introdução

*Problemas inversos* são opostos aos chamados *problemas diretos*. Basicamente, em um problema direto determina-se o efeito gerado por uma causa, enquanto que no problema inverso, a partir do efeito observado procura-se recuperar a causa. A situação mais comum que origina um problema inverso é a necessidade de interpretar medidas físicas indiretas de um objeto de interesse desconhecido. Por exemplo, na tomografia de raios-X, o problema direto é determinar as imagens que obteríamos de um corpo físico cuja estrutura interna conhecemos precisamente, usando raios-X. O problema inverso correspondente é reconstruir a estrutura interna de um corpo físico desconhecido a partir do conhecimento de imagens de raios-X tiradas de diferentes direções. Na figura 1 encontra-se um exemplo bidimensional: a fatia através de uma noz (esquerda) é a causa e a coleta de dados de raios-X (direita) é o efeito.

*Problemas diretos* são em geral *bem-postos*. A noção de problema bem-posto foi introduzida por Jacques Hadamard (1865-1963). Um problema é bem-posto se ele satisfaz estas três condições:

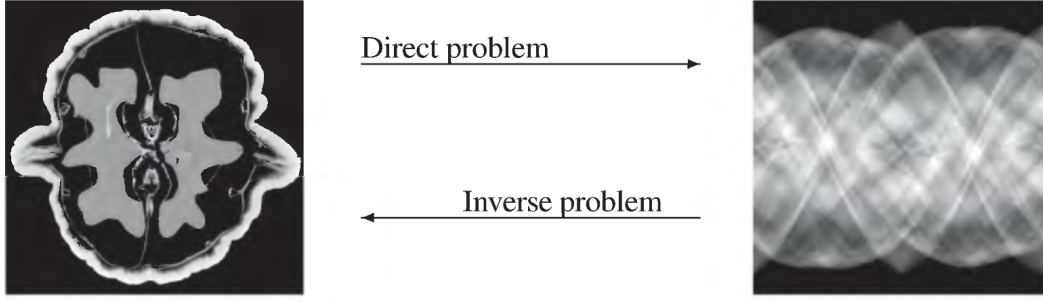


Figure 1: A imagem da fatia de uma noz à esquerda é cortesia de Keijo Hamalainen e Aki Kallonen da Universidade de Helsinque, Finlândia.

H1) Existência: existe pelo menos uma solução.

H2) Unicidade: se existir uma solução, ela é única.

H3) Estabilidade: a solução deve depender continuamente dos dados.

O exemplo típico de problema direto é uma equação diferencial parcial (EDP) da física, tais como a equação da onda ou a equação do calor. De fato, para estes problemas, conhecendo condições iniciais e de fronteiras, mais eventuais fontes, podemos calcular a solução única do problema.

Por outro lado, *problemas inversos* são frequentemente *mal-postos*, no sentido que eles não satisfazem pelo menos uma das hipóteses acima. Por exemplo, pode existir um grande número de soluções, e neste caso é difícil saber qual destas soluções é a mais relevante para a aplicação. A razão pela qual estes problemas geralmente são mal postos é porque não temos informações suficientes para encontrar a causa do efeito que estamos observando. As razões para tal são diversas, incluindo os custos de aquisição dos dados. Notemos ainda que medições em geral trazem imprecisões, o que mostra a importância da condição *H3* de um problema bem posto. Problemas inversos são muito relevantes em aplicações na física e engenharia, quando queremos determinar parâmetros que não podemos observar diretamente. Exemplos relevantes encontram-se, por exemplo, em detecção da origem de terremotos, mapeamento de camadas geológicas para determinação de onde se encontra petróleo e em aerodinâmica, na indústria automobilística e aeroespacial.

Nesse EP, dividido em duas partes, vamos resolver um problema relativamente simples, ligado à equação do calor. Esta primeira parte contemplada neste EP1, refere-se ao problema direto. Veremos como determinar a evolução da distribuição de temperatura em uma barra sujeita a fontes de calor, a partir de uma dada distribuição inicial.

## 2 Descrição do problema direto - equação do calor

A evolução da distribuição de temperatura em uma barra é dada pela seguinte equação diferencial parcial:

$$u_t(t, x) = u_{xx}(t, x) + f(t, x) \text{ em } [0, T] \times [0, 1], \quad (1)$$

$$u(0, x) = u_0(x) \text{ em } [0, 1] \quad (2)$$

$$u(t, 0) = g_1(t) \text{ em } [0, T] \quad (3)$$

$$u(t, 1) = g_2(t) \text{ em } [0, T]. \quad (4)$$

Aqui,  $t$  é a variável temporal e  $x$  a variável espacial. Estamos usando uma notação compacta para as derivadas parciais. Por exemplo,

$$u_{xx}(t, x) = \frac{\partial^2 u(t, x)}{\partial x^2}.$$

Normalizamos o comprimento da barra para 1 e vamos integrar a equação num intervalo de tempo de 0 a  $T$ . A variável  $u(t, x)$  descreve a temperatura no instante  $t$  na posição  $x$ , sendo a distribuição inicial  $u_0(x)$  dada. Na descrição acima as condições de fronteira (3)-(4) são do tipo Dirichlet, com as temperaturas nos extremos da barra prescritas. Alternativamente poderia ser prescrito o fluxo de calor nos extremos, com as derivadas de  $u$  dadas. A função  $f$  descreve as fontes de calor ao longo do tempo.

## 2.1 Discretizações da equação do calor

Queremos aproximar numericamente a solução de (1)-(4). Uma forma simples para se obter uma aproximação numérica das derivadas parciais é aproximá-las por *diferenças finitas*. Estas são baseadas em expansões de Taylor. Por exemplo, se  $g \in C^k(a, b)$  e  $x \in (a, b)$  temos que:

$$g(x+h) = g(x) + g'(x)h + g''(x)h^2/2 + \dots + g^{(k-1)}(x)h^{k-1}/(k-1)! + g^{(k)}(\bar{x})h^k/k!,$$

onde  $\bar{x}$  é um ponto entre  $x$  e  $x+h$ . Usando combinações desta expressão obtemos aproximações para os valores de uma função e suas derivadas. Por exemplo, temos:

$$g(x) = \frac{g(x-h) + g(x+h)}{2} + O(h^2) \quad (5)$$

$$g'(x) = \frac{g(x+h) - g(x-h)}{2h} + O(h^2) \quad (6)$$

$$g'(x) = \frac{g(x+h) - g(x-h)}{2h} + O(h^2) \quad (7)$$

$$g''(x) = \frac{g(x+h) - 2g(x) + g(x-h)}{h^2} + O(h^2) \quad (8)$$

onde a notação  $O(h^k)$  denota um erro proporcional a  $h^k$ . O termo exato depende de derivadas de ordem mais alta da função. Para a aproximação de derivadas parciais em relação a uma das variáveis usamos as mesmas fórmulas, com relação à variável em consideração. Por exemplo:

$$u_t(t, x) = \frac{u(t+\Delta t, x) - u(t, x)}{\Delta t} - \Delta t \frac{u_{tt}(\bar{t}, x)}{2} \quad (9)$$

$$u_{xx}(t, x) = \frac{u(t, x-\Delta x) - 2u(t, x) + u(t, x+\Delta x)}{\Delta x^2} - \Delta x^2 \frac{u_{xxxx}(t, \bar{x})}{4!} \quad (10)$$

onde  $\bar{t}$  é um valor entre  $t$  e  $t+\Delta t$  e  $\bar{x}$  é um valor entre  $x-\Delta x$  e  $x+\Delta x$ . Nas expressões para  $u_t(t, x)$  e  $u_{xx}(t, x)$  dadas acima os erros são proporcionais a  $\Delta t$  e  $\Delta x^2$  respectivamente. Se estes incrementos tenderem a zero teremos convergência das aproximações por diferenças finitas para as derivadas parciais correspondentes.

Para a discretização da equação do calor vamos introduzir uma malha espacial dada pelos pontos  $x_i = i\Delta x$ ,  $i = 0, \dots, N$ , com  $\Delta x = 1/N$ . Para a discretização temporal definimos  $\Delta t = T/M$ , e calculamos aproximações nos instantes  $t_k = k\Delta t$ ,  $k = 1, \dots, M$ . Denotamos a aproximação para a solução nos pontos de malha  $u(t_k, x_i)$  por  $u_i^k$ .

Desta forma teremos a condição inicial dada por:

$$u_i^0 = u_0(x_i), i = 0, \dots, N$$

Ao longo da evolução temporal as condições de fronteira são dadas por

$$u_0^k = g_1(t_k) \text{ e } u_N^k = g_2(t_k), k = 1, \dots, M.$$

Para os pontos interiores a evolução é aproximada pelas fórmulas de diferenças finitas:

$$u_i^{k+1} = u_i^k + \Delta t \left( \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + f(x_i, t_k) \right), \quad i = 1, \dots, N-1, \text{ e } k = 0, \dots, M-1. \quad (11)$$

Sabendo os valores iniciais da temperatura e seus valores na fronteira ao longo do tempo, a expressão (11) permite facilmente a determinação da solução aproximada em todos os instantes, computando sequencialmente desde  $t_0 = 0$  a  $t_M = T$ .

## 2.2 Um pouco de teoria

Seja  $u(t, x)$  a solução exata da equação do calor (1)-(4). Definimos o erro local de truncamento, que mede quão bem a solução exata  $u(t, x)$  satisfaz à equação discretizada (11) (na forma dividida por  $\Delta t$ ),

dado por

$$\tau_i^k(\Delta t, \Delta x) = \frac{u(t_{k+1}, x_i) - u(t_k, x_i)}{\Delta t} - \frac{u(t_k, x_{i-1}) - 2u(t_k, x_i) + u(t_k, x_{i+1}))}{\Delta x^2} - f(x_i, t_k) \quad (12)$$

$$= u_t(t_k, x_i) + \Delta t \frac{u_{tt}(\bar{t}_k, x_i)}{2} - u_{xx}(t_k, x_i) - \Delta x^2 \frac{u_{xxxx}(t_k, \bar{x}_i)}{4!} - f(x_i, t_k) \quad (13)$$

$$= \Delta t \frac{u_{tt}(\bar{t}_k, x_i)}{2} - \Delta x^2 \frac{u_{xxxx}(t_k, \bar{x}_i)}{4!}, \quad (14)$$

onde usamos as expansões de Taylor (9) e (10) para as aproximações das derivadas parciais e o fato de  $u(t, x)$  ser a solução exata da equação do calor na última passagem. Com a hipótese de  $u(t, x)$  ter 4 derivadas contínuas em  $x$  e duas em  $t$  no domínio de integração  $D = [0, T] \times [0, 1]$  podemos delimitar

$$\tau(\Delta t, \Delta x) = \max_{k,i} |\tau_i^k(\Delta t, \Delta x)| \leq C_1 \Delta t + C_2 \Delta x^2 \quad (15)$$

com

$$C_1 = \max_D \left| \frac{u_{tt}(t, x)}{2} \right| \text{ e } C_2 = \max_D \left| \frac{u_{xxxx}(t, \bar{x})}{4!} \right|.$$

Como consequência temos que

$$\lim_{\Delta t, \Delta x \rightarrow 0} \tau(\Delta t, \Delta x) = 0 \quad (16)$$

### Convergência da solução

Vimos que sob razoáveis hipóteses temos que o erro local de truncamento converge a zero. No entanto, o que realmente desejamos é que a solução aproximada nos pontos de malha convirja para a solução exata da equação do calor. Vamos agora estabelecer uma condição que garante que isto ocorre. Inicialmente definimos o erro entre a solução aproximada e a exata como:

$$e_i^k = u(t_k, x_i) - u_i^k. \quad (17)$$

Podemos agora combinar a equação para a determinação da solução aproximada (11) e a equação definindo o erro local de truncamento (12) para obter a seguinte equação para o erro:

$$e_i^{k+1} = e_i^k + \Delta t \left( \frac{e_{i-1}^k - 2e_i^k + e_{i+1}^k}{\Delta x^2} + \tau_i^k \right), \quad i = 1, \dots, N-1, \text{ e } k = 0, \dots, M-1. \quad (18)$$

Vamos ainda definir a norma do erro no instante  $t_k$  como

$$\|e^k\| = \max_i |e_i^k|. \quad (19)$$

Vamos agora estimar como o erro evolui em função do tempo. Vamos denominar  $\lambda = \Delta t / \Delta x^2$ . Então da equação do erro (18) obtemos que

$$|e_i^{k+1}| \leq |1 - 2\lambda| |e_i^k| + |\lambda| (|e_{i-1}^k| + |e_{i+1}^k|) + \Delta t |\tau_i^k| \quad (20)$$

$$\leq (|1 - 2\lambda| + 2|\lambda|) \|e^k\| + \Delta t \tau(\Delta t, \Delta x) \quad (21)$$

Temos que  $\lambda$  é sempre positivo. Por outro lado, se  $\lambda \leq 1/2$  então  $1 - 2\lambda \geq 0$ . Assim, para esta escolha de  $\lambda$ , obtemos

$$|e_i^{k+1}| \leq ((1 - 2\lambda) + 2\lambda) \|e^k\| + \Delta t \tau(\Delta t, \Delta x) \quad (22)$$

$$\leq \|e^k\| + \Delta t \tau(\Delta t, \Delta x), \text{ e segue que } \|e^{k+1}\| \leq \|e^k\| + \Delta t \tau(\Delta t, \Delta x). \quad (23)$$

Usando esta última estimativa recursivamente obtemos então que:

$$\|e^{k+1}\| \leq \|e^{k-1}\| + 2\Delta t \tau(\Delta t, \Delta x) \quad (24)$$

$$\leq \|e^0\| + (k+1)\Delta t \tau(\Delta t, \Delta x) = t_{k+1} \tau(\Delta t, \Delta x) \quad (25)$$

$$\leq T(C_1 \Delta t + C_2 \Delta x^2). \quad (26)$$

Nesta última estimativa usamos ainda que o erro inicial  $\|e^0\|$  é nulo, uma vez que  $u_0(x)$  é dado, e a delimitação do erro de truncamento (15). Esta estimativa mostra, que se  $\Delta t$  e  $\Delta x$  tenderem a zero o

erro também vai a zero e portanto a aproximação calculada converge para a solução exata da equação. Note no entanto, que nesta demonstração usamos fortemente a hipótese de que

$$\lambda = \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}. \quad (27)$$

Caso esta condição seja violada, você verificará com seu programa, o método fica instável, com o erro se amplificando enormemente. Este método é dito *condicionalmente convergente*. Para termos convergência,  $\Delta t$  deve ser da ordem de  $\Delta x^2$ . Portanto, se a condição (27) estiver satisfeita para  $\Delta t = \alpha \Delta x^2$ , com  $\alpha \leq 1/2$ , obtemos então que o erro fica menor que uma constante vezes  $\Delta x^2$ . Dizemos que o método é (condicionalmente) convergente de ordem 2 em  $\Delta x$ .

## 2.3 Primeira tarefa

Você deve implementar o método (11), deixando os valores de  $N$  e  $M$  (que determinam  $\Delta t$  e  $\Delta x$ ) como variáveis a serem escolhidas em tempo de execução. Teste o seu programa com os dados

a)  $T = 1$  com a fonte  $f(t, x) = 10x^2(x - 1) - 60xt + 20t$  a partir de  $u_0(x) = 0$  e condições de fronteira também nulas. Faça integrações com  $N = 10, 20, 40, 80, 160, 320$  e  $640$  para  $\lambda = 0.5$  e  $\lambda = 0.25$ . Experimente também com  $\lambda = 0.51$ . O que acontece? Verifique que a solução exata neste caso é igual a  $u(t, x) = 10tx^2(x - 1)$  e calcule o erro obtido em  $T = 1$  com as diversas resoluções para  $\lambda = 0.5$  e  $\lambda = 0.25$ . Verifique o comportamento do erro. Qual o fator de redução esperado a cada refinamento de malha. Qual o número de passos necessários ao se usar  $N = 640$ ? E se dobrarmos  $N$ ?

b) Determine quem deve ser  $u_0(x)$ ,  $g_1(t)$ ,  $g_2(t)$  e  $f(t, x)$  de forma que a solução exata seja dada por  $u(t, x) = e^{t-x} \cos(5tx)$  e repita os experimentos da parte a)

c)  $T = 1$  a partir da condição inicial nula, com uma fonte pontual localizada em um ponto  $p$  do domínio e intensidade variando ao longo do tempo dada por  $r(t) = 10000 * (1 - 2t^2)$ . Vamos agora ver como implementar uma fonte pontual, digamos de intensidade 1. Podemos ver esta força  $f$  localizada, como limite de forças  $g_h$  que atuam em uma pequena região (cujo tamanho vai a zero com  $h$ ) em torno do ponto em questão (ou seja, são não nulas apenas nesta pequena região), de forma que a integral de cada  $g_h$  seja constante igual a 1. Para obter tal efeito podemos adotar

$$g_h(x) = \frac{1}{h}, \text{ se } p - h/2 \leq x \leq p + h/2, \text{ e } g_h(x) = 0 \text{ caso contrário.}$$

Alternativamente  $g_h(x)$  poderia assumir o valor  $1/h$  em  $p$  e variar linearmente de 0 a  $1/h$  no intervalo  $[p - h, p]$  e de  $1/h$  a 0 no intervalo  $[p, p + h]$ , sendo nula no restante do domínio. Adote uma destas duas representações de  $g_h$  e defina a fonte pontual  $f(t, x) = r(t)g_h(x)$ , com  $h = \Delta x$ .

Trabalhe com a fonte em  $p = 0.25$  e use na fronteira  $g_1(t) = g_2(t) = 0$ . Neste caso não sabemos a solução exata. Determine-a numericamente.

Em todos os casos faça gráficos da solução obtida. Plotem os gráficos de 0.1 em 0.1, assim podem observar a evolução temporal.

## 2.4 Um método implícito

Você deve ter percebido através de seus experimentos com o método (11) que a quantidade de passos no tempo, necessários para se fazer integrações em malhas mais finas, cresce muito. Isto faz com que o esquema não seja muito eficiente. O que gostaríamos é ter um método convergente de ordem 2 em que pudéssemos usar  $\Delta t$  da mesma ordem que  $\Delta x$ . Para um tal método ser estável (você já deve ter notado o que ocorre quando há instabilidade ...), será no entanto necessário usar um método implícito. No esquema (11) a solução em cada ponto em um novo instante de tempo é obtida simplesmente avaliando uma combinação de valores vizinhos do passo anterior. Em um método implícito, a solução em um ponto de malha no novo instante depende também de outros valores no mesmo instante. Esta interdependência dos valores no novo instante leva à necessidade de resolver um sistema de equações a cada passo no tempo. Um primeiro exemplo de método implícito (também chamado de Euler implícito) é dado pelo esquema

$$u_i^{k+1} = u_i^k + \lambda(u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}) + \Delta t f(x_i, t_{k+1}), \quad i = 1, \dots, N-1, \text{ e } k = 0, \dots, M-1, \quad (28)$$

com as mesmas notações anteriores. Neste método, para a evolução temporal, necessitamos resolver a cada passo um sistema linear com uma matriz  $A$  tridiagonal simétrica, como segue

$$\begin{bmatrix} 1+2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\lambda & 1+2\lambda & -\lambda \\ 0 & \cdots & 0 & -\lambda & 1+2\lambda \end{bmatrix} \begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ \vdots \\ u_{N-2}^{k+1} \\ u_{N-1}^{k+1} \end{bmatrix} = \begin{bmatrix} u_1^k + \Delta t f_1^{k+1} + \lambda g_1(t^{k+1}) \\ u_2^k + \Delta t f_2^{k+1} \\ \vdots \\ u_{N-2}^k + \Delta t f_{N-2}^{k+1} \\ u_{N-1}^k + \Delta t f_{N-1}^{k+1} + \lambda g_2(t^{k+1}) \end{bmatrix} \quad (29)$$

Analogamente ao que fizemos para o método explícito (11) definimos o erro de truncamento

$$\tau_i^k(\Delta t, \Delta x) = \frac{u(t_{k+1}, x_i) - u(t_k, x_i)}{\Delta t} - \frac{u(t_{k+1}, x_{i-1}) - 2u(t_{k+1}, x_i) + u(t_{k+1}, x_{i+1}))}{\Delta x^2} - f(x_i, t_{k+1}) \quad (30)$$

e a delimitação

$$\tau(\Delta t, \Delta x) = \max_{k,i} |\tau_i^k(\Delta t, \Delta x)| \leq C_1 \Delta t + C_2 \Delta x^2. \quad (31)$$

Procedendo como anteriormente chegaremos à expressão (verifique!):

$$(1+2\lambda)|e_i^{k+1}| \leq |e_i^k| + \lambda(|e_{i-1}^{k+1}| + |e_{i+1}^{k+1}|) + \Delta t |\tau_i^k| \quad (32)$$

$$\leq \|e^k\| + 2\lambda \|e^{k+1}\| + \Delta t \tau(\Delta t, \Delta x), \quad (33)$$

de onde segue que

$$\|e^{k+1}\| = (1+2\lambda-2\lambda)\|e^{k+1}\| \leq \|e^k\| + \Delta t \tau(\Delta t, \Delta x). \quad (34)$$

Esta equação é análoga à (22), porém foi obtida sem que precisássemos fazer qualquer restrição na escolha de  $\Delta t$  e  $\Delta x$ . A demonstração da convergência segue daqui como feita para o método explícito. O método de Euler implícito é incondicionalmente estável e convergente de ordem 2 em  $\Delta x$  e ordem 1 em  $\Delta t$ . Assim, mesmo não sofrendo restrições de estabilidade, a precisão do esquema estará limitada pela escolha de  $\Delta t$ .

A seguir apresentamos o método de Crank-Nicolson, que também é incondicionalmente estável, mas tem convergência de ordem 2 em  $\Delta x$  e  $\Delta t$ .

O método é da forma:

$$u_i^{k+1} = u_i^k + \frac{\lambda}{2} ((u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}) + (u_{i-1}^k - 2u_i^k + u_{i+1}^k)) + \frac{\Delta t}{2} (f(x_i, t_k) + f(x_i, t_{k+1})) \quad (35)$$

com as mesmas notações anteriores.

Este é um esquema de segunda ordem, centrado no tempo intermediário  $t^k + 0.5\Delta t$ , combinando as fórmulas (5), (7) e (8). Procure analisar o erro de truncamento!

Analogamente ao método implícito temos que resolver um sistema linear. Este envolverá também uma matriz tridiagonal simétrica, como no método anterior, trocando  $\lambda$  por  $\lambda/2$ . O lado direito é distinto. Será sua tarefa elaborar os detalhes.

A análise da convergência deste esquema é no entanto mais complexa que a anterior, envolvendo os auto-valores da Matriz do sistema. Esta pode ser encontrada detalhada no livro **Analysis of Numerical Methods**, E. Isaacson and H. B. Keller e não será repetida aqui.

## 2.5 Segunda tarefa

a) Nos dois métodos implícitos apresentados, há a necessidade de resolução de um sistema tridiagonal simétrico. Para tanto você deve escrever uma rotina que calcula a decomposição  $A = LDL^t$  da matriz do sistema tridiagonal em questão, onde a matriz  $L$  é bidiagonal triangular unitária inferior e  $D$  diagonal, ou seja

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & l_{N-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{N-1} \end{bmatrix} \begin{bmatrix} 1 & l_2 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & l_{N-1} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \quad (36)$$

Note que a matriz  $A$ , sendo tridiagonal simétrica  $N - 1 \times N - 1$  pode ser armazenada em apenas dois vetores de comprimento  $N - 1$ , um para a diagonal de  $A$  e o outro para a subdiagonal. Já as matrizes  $L$  e  $D$  necessitam apenas de um vetor cada uma para seu armazenamento. Seu programa para o cálculo desta decomposição deve fazer uso deste tipo de armazenamento, tendo como entrada os dois vetores representando  $A$  e como saída os dois vetores representando  $L$  e  $D$ . Note que a matriz  $A$  não se altera em função do passo no tempo, permanecendo constante ao longo de toda a integração temporal. Assim, basta calcular sua decomposição uma única vez. Escreva então um procedimento que dada uma decomposição deste tipo e um lado direito do sistema  $Ax = LDL^t x = b$ , compute a solução  $x$ . Você irá usar estas rotinas tanto para o Método de Euler implícito como para o de Crank-Nicolson.

b) Repita os mesmos testes da primeira tarefa com o método de Euler implícito, utilizando  $\Delta t = \Delta x$ . Verifique a ordem de convergência.

c) Faça o mesmo que no item anterior para o método de Crank-Nicolson.

### 3 O que vem no EP2?

Na continuação desta tarefa vocês irão resolver um problema inverso relacionado à equação do calor, onde irão utilizar os desenvolvimentos feitos no EP1.