

Métodos numéricos para EDPs – segunda lista de exercícios

Leonardo de Oliveira Carvalho

13 de abril de 2010

Exercício 1

Consideremos a equação de difusão $u_t = \kappa u_{xx}$ com condições de contorno iguais a zero no intervalo $[0, 1]$. Neste exercício analisaremos o método de Crank-Nicolson para encontrar a solução discreta da equação de difusão. Vamos aproximar a solução em m pontos igualmente espaçados no interior do intervalo $[0, 1]$ (solução zero nos extremos conforme as condições de contorno). Em cada instante de tempo n a solução discreta será dada pelo vetor $U^n \in \mathbb{R}^m$, sendo tratado como uma matriz coluna $m \times 1$. Seja $\Delta x = 1/(m+1)$, $x_j = j\Delta x$ e $t_n = n\Delta t$, temos U_j^n como aproximação de $u(x_j, t_n)$. U^0 indica a condição inicial. Para cada $n \geq 0$, é considerado $U_0^n = U_{m+1}^n = 0$, indicando assim as condições de contorno.

1.A

Vamos escrever o método de Crank-Nicolson na forma matricial.

Inicialmente consideremos a aplicação $C : \mathbb{R}^m \rightarrow \mathbb{R}^m$ dada por

$$(CV)_j = V_{j-1} - 2V_j + V_{j+1}, \quad j = 1, \dots, m, \quad \text{onde } V_0 = V_{m+1} = 0.$$

C pode ser escrita como uma matriz $m \times m$, em que cada linha j possui o valor -2 na posição j , e o valor 1 nas posições $j-1$ e $j+1$. Assim C é uma matriz tridiagonal da seguinte forma:

$$C = \begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -2 \end{bmatrix}.$$

Onde os termos omitidos são iguais a zero.

O método de Crank-Nicolson consiste em usar as aproximações

$$u_t(x_j, t_{n+1}) \approx \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} = \frac{1}{\Delta t} (U_j^{n+1} - U_j^n),$$

$$\begin{aligned} u_{xx}(x_j, t_{n+1}) &\approx \frac{0,5}{\Delta x^2} (u(x_{j-1}, t_n) - 2u(x_j, t_n) + u(x_{j+1}, t_n)) \\ &\quad + \frac{0,5}{\Delta x^2} (u(x_{j-1}, t_{n+1}) - 2u(x_j, t_{n+1}) + u(x_{j+1}, t_{n+1})) \\ &= \frac{0,5}{\Delta x^2} \left((U_{j-1}^n - 2U_j^n + U_{j+1}^n) + (U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1}) \right) \\ &= \frac{0,5}{\Delta x^2} \left((CU^n)_j + (CU^{n+1})_j \right). \end{aligned}$$

Onde a aplicação C pode ser usada já que as condições de contorno são iguais a zero. A equação de difusão torna-se então a cada instante $n \geq 0$, para $j = 1, \dots, m$:

$$\frac{1}{\Delta t}(U_j^{n+1} - U_j^n) = \kappa \frac{0,5}{\Delta x^2} \left((CU^n)_j + (CU^{n+1})_j \right),$$

$$U_j^{n+1} - U_j^n = 0,5\alpha \left((CU^n)_j + (CU^{n+1})_j \right), \quad \text{onde } \alpha = \kappa \frac{\Delta t}{\Delta x^2}.$$

Separando os termos n de n + 1 teremos:

$$\begin{aligned} 0,5\alpha (CU^{n+1})_j - U_j^{n+1} &= -0,5\alpha (CU^n)_j - U_j^n \\ (CU^{n+1})_j - \frac{2}{\alpha} U_j^{n+1} &= -(CU^n)_j - \frac{2}{\alpha} U_j^n \\ \left(\left(C - \frac{2}{\alpha} I \right) U^{n+1} \right)_j &= \left(\left(-C - \frac{2}{\alpha} I \right) U^n \right)_j \\ (AU^{n+1})_j &= (BU^n)_j \\ AU^{n+1} &= BU^n. \end{aligned}$$

Onde $A = C - \frac{2}{\alpha} I$ e $B = -C - \frac{2}{\alpha} I$. Portanto teremos:

$$\begin{bmatrix} \beta & 1 & & & & & & & \\ 1 & \beta & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & \beta & 1 & & \\ & & & & & & & 1 & \beta \\ & & & & & & & & \beta \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ \vdots \\ U_{m-1}^{n+1} \\ U_m^{n+1} \end{bmatrix} = \begin{bmatrix} \gamma & -1 & & & & & & & \\ -1 & \gamma & -1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & & & & -1 & \gamma & -1 \\ & & & & & & & -1 & \gamma \end{bmatrix} \begin{bmatrix} U_1^n \\ U_2^n \\ \vdots \\ U_{m-1}^n \\ U_m^n \end{bmatrix},$$

onde $\beta = -2 - 2/\alpha$ e $\gamma = 2 - 2/\alpha$.

1.B

O sistema pode ser resolvido fazendo

$$U^{n+1} = A^{-1}BU^n.$$

Teremos então

$$\begin{aligned} U^1 &= A^{-1}BU^0, \\ U^2 &= A^{-1}BU^1 = (A^{-1}B)^2U^0, \\ U^3 &= A^{-1}BU^2 = (A^{-1}B)^3U^0, \\ &\vdots \\ U^n &= (A^{-1}B)^nU^0. \end{aligned}$$

Sabemos que a solução da equação de difusão tende a zero, então espera-se que U^n tenda a zero para qualquer dado inicial. Para isto sabemos que é necessário que $|\lambda| < 1$ para todo autovalor λ de $A^{-1}B$.

Seja v um autovetor de $A^{-1}B$, associado ao autovalor λ . Pode-se observar que

$$A^{-1}Bv = \lambda v \Rightarrow Bv = \lambda Av \Rightarrow (B - \lambda A)v = 0.$$

Assim, estudar a solução desta equação é útil para analisarmos a estabilidade do método de Crank-Nicolson.

Seja $\mu_k \in \sigma(C)$. Teremos então $(\mu_k - \frac{2}{\alpha})^{-1} \in \sigma(A^{-1})$, e $(-\mu_k - \frac{2}{\alpha}) \in \sigma(B)$. Logo

$$\left(\mu_k - \frac{2}{\alpha}\right)^{-1} \left(-\mu_k - \frac{2}{\alpha}\right) \in \sigma(A^{-1}B).$$

Podemos então fazer

$$\lambda_k = \left(\mu_k - \frac{2}{\alpha}\right)^{-1} \left(-\mu_k - \frac{2}{\alpha}\right) = \frac{-\mu_k - \frac{2}{\alpha}}{\mu_k - \frac{2}{\alpha}} = \frac{2 + \alpha\mu_k}{2 - \alpha\mu_k}.$$

Portanto, se tivermos os autovalores de C podemos facilmente calcular os autovalores de $A^{-1}B$.

Vamos analisar como o método se comporta para o modo de Fourier:

$$u_k(x, t) = e^{\lambda t} e^{i(2k\pi x)}, \quad \text{para } k = 1, 2, \dots$$

Seja

$$U_j^n = u_k(j\Delta x, n\Delta t) = e^{\lambda n\Delta t} e^{i(2k\pi j\Delta x)} = e^{n\lambda_k} e^{ijk_0},$$

onde $k_0 = 2k\pi\Delta x$ e $\lambda_k = \lambda\Delta t$. Queremos que a amplitude $e^{n\lambda_k}$ diminua conforme n cresce, portanto precisamos que $|e^{\lambda_k}| < 1$.

Utilizando esta expressão de U_j^n na fórmula de recorrência do método teremos:

$$\begin{aligned} U_j^{n+1} - U_j^n &= 0,5\alpha \left(U_{j-1}^n - 2U_j^n + U_{j+1}^n + U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1} \right) \\ U_j^n e^{\lambda_k} - U_j^n &= 0,5\alpha \left(U_j^n e^{-ik_0} - 2U_j^n + U_j^n e^{ik_0} + U_j^n e^{-ik_0} e^{\lambda_k} - 2U_j^n e^{\lambda_k} + U_j^n e^{ik_0} e^{\lambda_k} \right) \\ e^{\lambda_k} - 1 &= 0,5\alpha \left(e^{-ik_0} - 2 + e^{ik_0} + e^{-ik_0} e^{\lambda_k} - 2e^{\lambda_k} + e^{ik_0} e^{\lambda_k} \right) \\ &= 0,5\alpha \left(e^{-ik_0} - 2 + e^{ik_0} + e^{\lambda_k} (e^{-ik_0} - 2 + e^{ik_0}) \right) \\ &= 0,5\alpha \left(e^{-ik_0} - 2 + e^{ik_0} \right) (e^{\lambda_k} + 1) \\ &= 0,5\alpha (2 \cos(k_0) - 2) (e^{\lambda_k} + 1) \\ &= \alpha (\cos(k_0) - 1) (e^{\lambda_k} + 1) \\ e^{\lambda_k} &= \frac{1 + \alpha(\cos(k_0) - 1)}{1 - \alpha(\cos(k_0) - 1)} \\ e^{\lambda_k} &= \frac{2 + \alpha\mu_{k_0}}{2 - \alpha\mu_{k_0}}, \end{aligned}$$

onde $\mu_{k_0} = 2(\cos k_0 - 1) = -4 \text{sen}^2(k_0/2) = -4 \text{sen}^2(k\pi\Delta x)$. Observe que a expressão de e^{λ_k} e λ_k são semelhantes, e ambos precisam possuir módulo menor do que 1 para garantir a convergência por este método. Isto sugere que e^{λ_k} sejam os autovalores de $A^{-1}B$, e μ_{k_0} sejam os autovalores de C . Isto é verdade, pois podemos verificar que o vetor $v = (\text{sen } \theta, \text{sen } 2\theta, \dots, \text{sen } m\theta)$, é um autovetor associado ao autovalor $\mu_k = -4 \text{sen}^2(\theta/2)$, onde $\theta = 2k\pi\Delta x$, para a matriz C . De fato temos

$$\begin{aligned} (Cv)_j &= v_{j-1} - 2v_j + v_{j+1} \\ &= \text{sen}((j-1)\theta) - 2\text{sen}(j\theta) + \text{sen}((j+1)\theta) \\ &= \text{sen}(j\theta - \theta) - 2\text{sen}(j\theta) + \text{sen}(j\theta + \theta) \\ &= 2\text{sen}(j\theta) \cos(\theta) - 2\text{sen}(j\theta) \\ &= 2\text{sen}(j\theta)(\cos(\theta) - 1) \\ &= 2(\cos(\theta) - 1)v_j \\ &= 2(-2 \text{sen}^2(\theta/2))v_j \\ &= \mu_k v_j. \end{aligned}$$

Onde foram utilizadas as relações trigonométricas $\sin(a + b) + \sin(a - b) = 2 \sin a \cos b$ e $1 - \cos(\theta) = 2 \sin^2(\theta/2)$.

Observe que isto também vale para os casos $j = 1$ e $j = m$, pois se fizermos $v_j = \sin(j\theta)$, teremos $v_0 = \sin(0) = 0$, e $v_{m+1} = \sin((m + 1)(2k\pi\Delta x)) = \sin(2k\pi) = 0$ (pois $(m + 1)\Delta x = 1$).

Temos então $Cv = \lambda v$.

1.C

Para o método ser estável vimos pelo item anterior que é preciso

$$\left| \frac{2 + \alpha\mu_k}{2 - \alpha\mu_k} \right| < 1, \quad \text{onde } \mu_k = -4 \sin^2(k\pi\Delta x), \quad k = 1, \dots, m.$$

Porém isto sempre ocorre, já que

$$\sin^2(k\pi\Delta x) \geq 0 \Rightarrow \mu_k \leq 0 \Rightarrow 2 + \alpha\mu_k \leq 2 \leq 2 - \alpha\mu_k \Rightarrow \frac{2 + \alpha\mu_k}{2 - \alpha\mu_k} < 1.$$

Portanto concluímos que o método de Crank-Nicolson é incondicionalmente estável.

Exercício 2

Neste exercício vamos trabalhar com a equação da onda $u_t + cu_x = 0$ com o método *upwind* e o método *Lax-Wendroff*, ambos implementados em MATLAB (código em anexo).

Pelo método *upwind*, a solução discreta da equação da onda é dado por

$$u_j^{n+1} = u_j^n - \sigma (u_j^n - u_{j-1}^n),$$

onde $\sigma = c \frac{\Delta t}{\Delta x}$ é o número de Courant.

Já o método *Lax-Wendroff* utiliza a relação

$$u_j^{n+1} = u_j^n - \frac{\sigma}{2} (u_{j+1}^n - u_{j-1}^n) + \frac{\sigma^2}{2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n).$$

Para fazer testes, utilizamos uma Gaussiana normalizada centrada na origem com desvio padrão igual a 9 (na maioria dos casos), e uma função de Heaviside. Foi utilizado $x \in [-20, 20]$, com $\Delta x = 0, 1$. Para a velocidade da onda foi utilizado o valor $c = 2$, em todos os casos. Desta forma, o valor de σ foi controlado apenas através da variação do tempo Δt .

Uma análise útil é através da EDP modificada em cada caso. No método *upwind* sabemos que a EDP modificada é $u_t + cu_x = \epsilon u_{xx}$, onde $\epsilon = 0, 5c\Delta x(1 - \sigma)$, ou seja é uma equação do calor com coeficiente de difusão ϵ . Já no método *Lax-Wendroff* pode-se mostrar que a EDP modificada é $u_t + cu_x = \epsilon c_{xxx}$, com $\epsilon = \frac{c\Delta x^2}{6}(\sigma^2 - 1)$, portanto é uma equação dispersiva.

Primeiro caso: $\sigma < 1$

Neste caso a função muda de forma ao longo do tempo. O resultado é influenciado pelo termo difusivo no *upwind*, e pelo termo dispersivo no *Lax-Wendroff*.

A Figura 1 mostra o resultado obtido com 400 passos no tempo com os dois métodos aplicados a uma Gaussiana com desvio padrão 0, 16, utilizando $\Delta t = 0, 01$ ($\sigma = 0, 2$), a solução sem distorções pode ser vista tracejada nos gráficos.

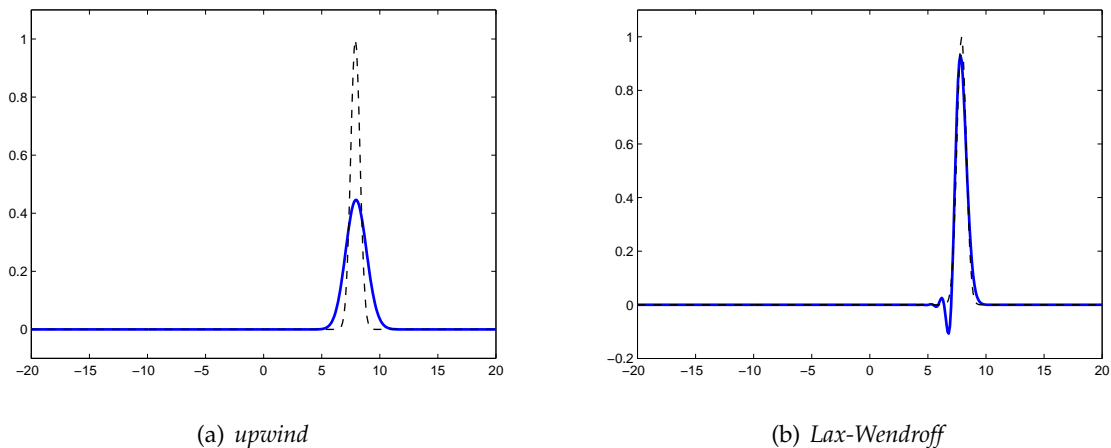


Figura 1: Resultados $\sigma < 1$.

A Figura 2 mostra o resultado obtido após 200 passos no tempo com os dois métodos aplicados à função de Heaviside, utilizando $\Delta t = 0,01$ ($\sigma = 0,2$), a solução sem distorções pode ser vista tracejada nos gráficos.

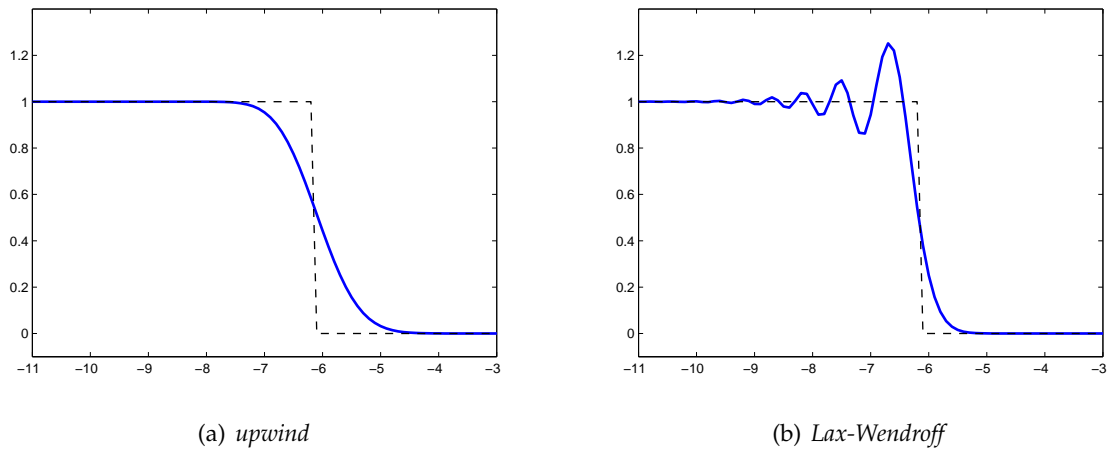


Figura 2: Resultados $\sigma < 1$.

Estes efeitos de difusão e dissipação diminuem conforme nos aproximamos de $\sigma = 1$. A Figura 3 nos mostra o resultado aproximadamente na mesma posição dos resultados da Figura 2, obtido após 44 passos de tempo, com $\Delta t = 0,045$ ($\sigma = 0,9$). É possível ver que estes efeitos diminuíram, em relação aos resultados da figura anterior.

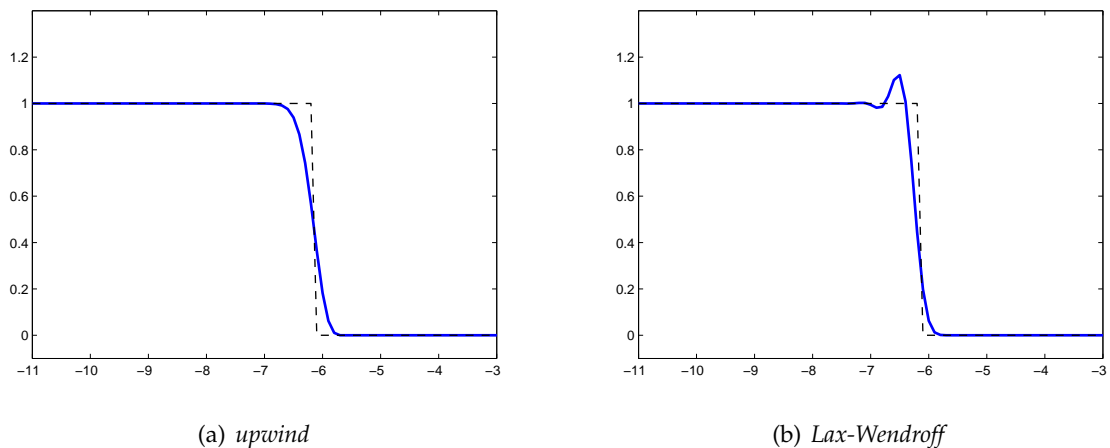


Figura 3: Resultados $\sigma < 1$.

Segundo caso: $\sigma = 1$

Este é o caso ideal, quando a velocidade da onda é igual à velocidade numérica de cada método. O termo difusivo e o dispersivo se anulam neste caso, portanto a solução não apresenta alterações numéricas no formato da função. É fácil ver que em ambos os métodos teremos

$$U_j^{n+1} = U_{j-1}^n,$$

e portanto a solução inicial “viaja” da esquerda para a direita sem alteração no seu formato, conforme o esperado.

Nos testes foi utilizado um passo de tempo $\Delta t = 0,05$ para que $\sigma = 1$. A Figura 4 ilustra o resultado do método *upwind* aplicado à Gaussiana (com um passo de tempo maior para facilitar a visualização), o mesmo resultado é obtido através do método de *Lax-Wendroff*.

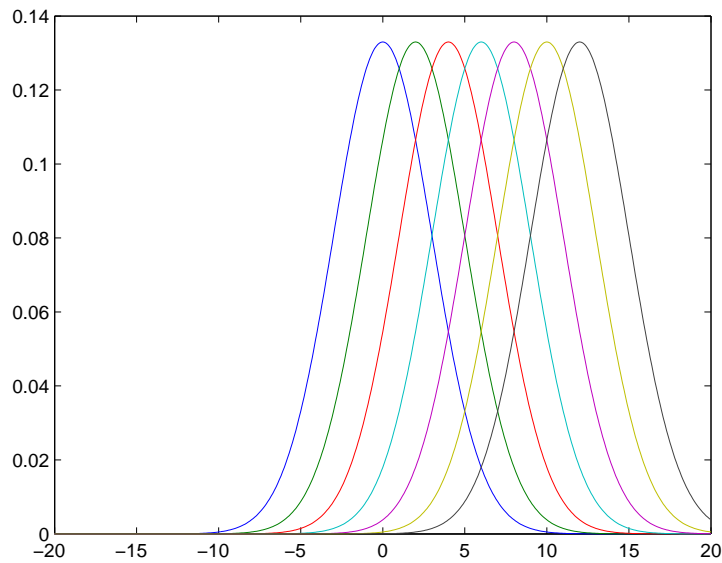


Figura 4: Resultado do método *upwind* para $\sigma = 1$.

A Figura 5 ilustra o resultado do método *Lax-Wendroff* aplicado à Heaviside, no tempo inicial, e após 200 iterações ($t = 10$). O mesmo resultado é obtido através do método de *upwind*.

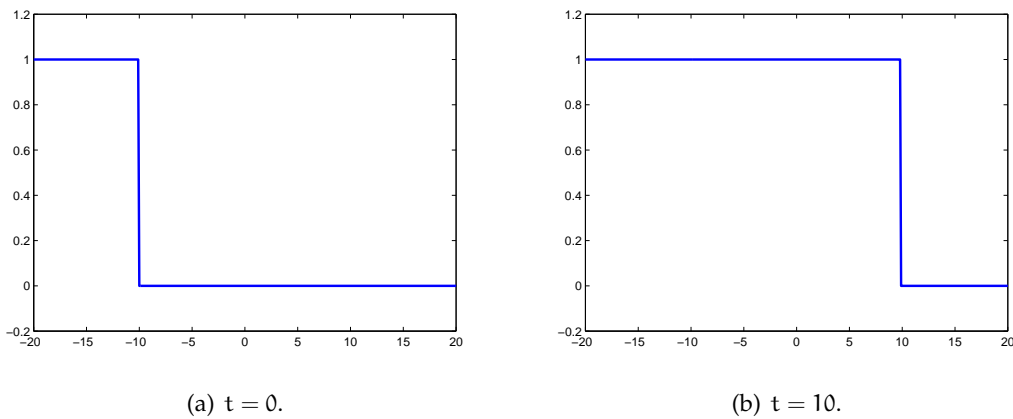


Figura 5: Resultado do método *Lax-Wendroff* para $\sigma = 1$.

Terceiro caso: $\sigma > 1$

Neste caso estaremos fora da região de estabilidade dos dois métodos, portanto as soluções divergem. A Figura 6 ilustra o resultado obtido após 50 passos no tempo com os métodos *upwind* e *Lax-Wendroff* aplicados à Gaussiana utilizando $\Delta t = 0,075$ ($\sigma = 1,5$).

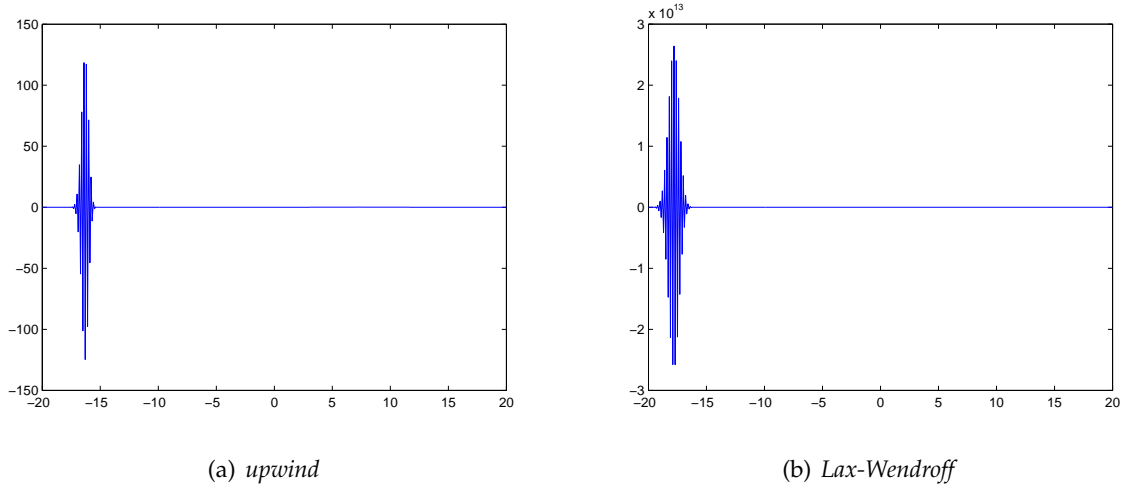


Figura 6: Resultados $\sigma > 1$.

A Figura 7 ilustra o resultado obtido após 50 passos no tempo com os métodos *upwind* e *Lax-Wendroff* aplicados à Heaviside utilizando $\Delta t = 0,075$ ($\sigma = 1,5$).

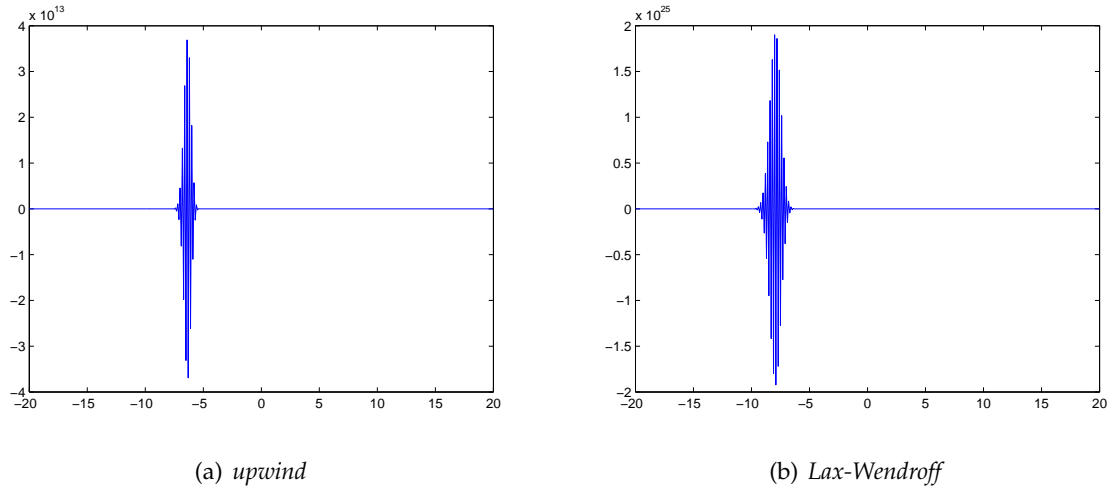


Figura 7: Resultados $\sigma > 1$.

Exercício 3

Neste exercício estudamos a equação de Burgers:

$$u_t + uu_x = 0.$$

Esta equação também pode ser escrita na forma

$$u_t + 0,5(u^2)_x = 0.$$

A partir desta forma podemos encontrar um método iterativo para encontrar a solução discreta através de:

$$u_j^{n+1} = u_j^n + 0,5 \frac{\Delta t}{\Delta x} \left((u_j^n)^2 - (u_{j-1}^n)^2 \right).$$

Este método será analisado através de uma implementação em MATLAB (código em anexo). Utilizamos $x \in [-20, 20]$, com $\Delta x = 0,1$, as condições de contorno são nulas, e como dado inicial foi considerada uma Gaussiana normalizada com desvio padrão 9.

A Figura 8 mostra o resultado obtido. No cálculo foi utilizado um passo de tempo $\Delta t = 0,75$ (na figura o passo de tempo é maior, para facilitar a visualização).

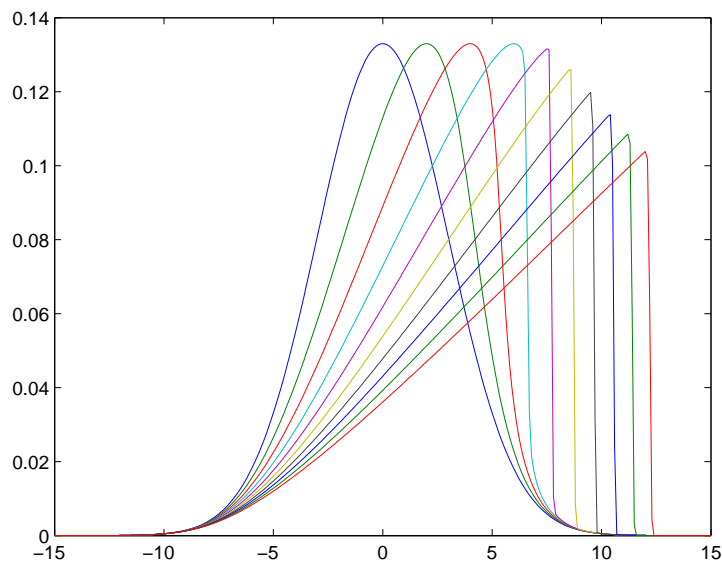


Figura 8: Resultado do método.

Pode-se perceber a formação de uma descontinuidade, isto pode ser explicado ao analisarmos as curvas características da equação de Burgers, dadas por:

$$\begin{cases} \frac{dx}{dt} = u, \\ \frac{du}{dt} = 0. \end{cases}$$

Vemos que u é constante ao longo das características, que são retas no plano xt com inclinação igual ao valor de u em cada uma destas retas. Por esta variação da inclinação, em algum instante as características se cruzam, a partir deste instante a solução será descontínua, e a equação de Burgers deixa de fazer sentido. Mesmo assim, ao continuarmos a execução do método vemos que a descontinuidade é transportada, formando o que é conhecido como “salto hidráulico”.

Ao longo do tempo a solução fica cada vez menos suave, isto pode ser observado ao se analisar a representação no domínio da frequência da função em cada instante. Utilizando a função `fft` do MATLAB obtemos os gráficos da Figura 9. Estes gráficos ilustram a transformada de Fourier discreta (em módulo) da solução no instante inicial (linha tracejada) e após 50 iterações do método (linha contínua). Estas soluções são exibidas na Figura 10 (com o mesmo padrão de linhas). Analisando a Figura 9 podemos ver que frequências mais altas começam a surgir ao longo do tempo.

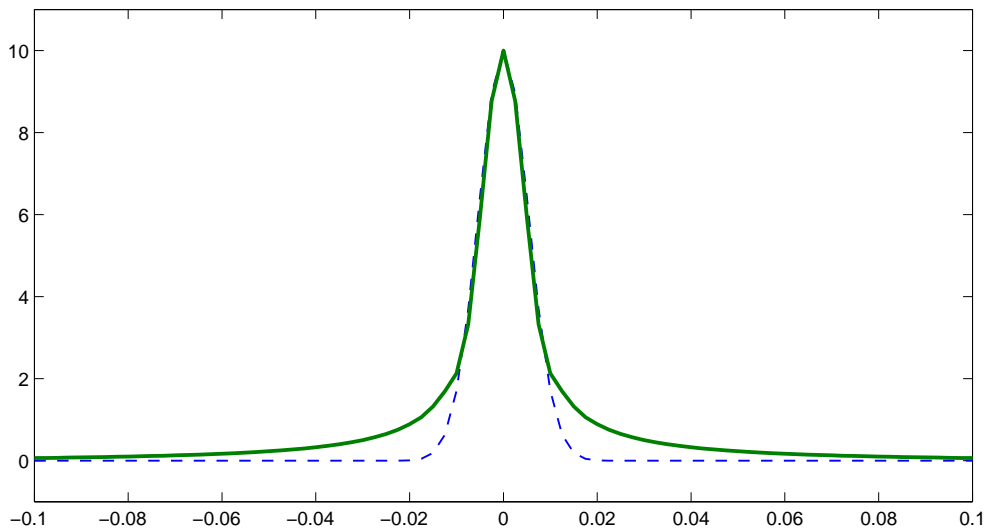


Figura 9: Transformada de Fourier da solução em dois instantes.

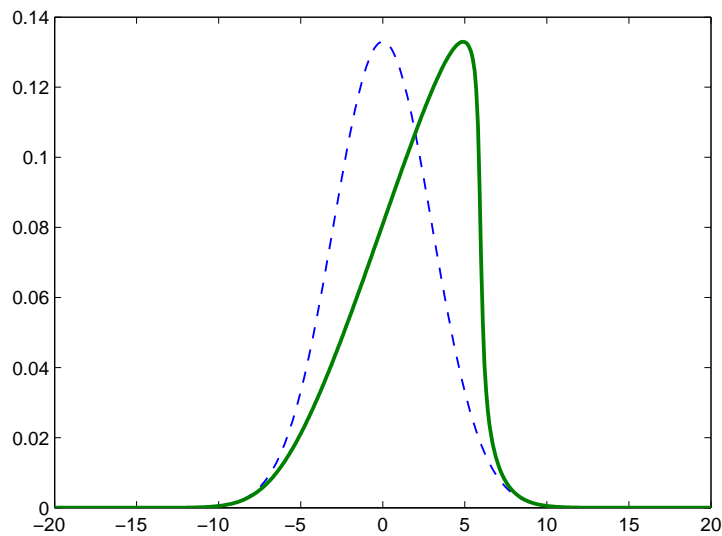


Figura 10: Solução nos dois instantes usados na Figura 9.

Esta mudança no espectro é causada pela não linearidade na equação de Burgers. No caso linear $u_t + cu_x = 0$, onde c é uma constante, a solução mantém a forma ao longo do tempo, e consequentemente o espectro de Fourier não muda.

Da mesma forma que acontece com a equação linear, este método apresenta difusão numérica, mas neste caso o termo $\sigma = u\Delta t/\Delta x$ não é constante, portanto a difusão sempre irá ocorrer, podendo apenas ser minimizada, como fizemos na Figura 8. Na Figura 11 utilizamos $\Delta t = 0,1$ com a mesma função da Figura 8, com este passo de tempo menor, o efeito de difusão pode ser melhor observado, perceba que o topo das curvas está diminuindo mesmo antes da formação do “salto hidráulico”.

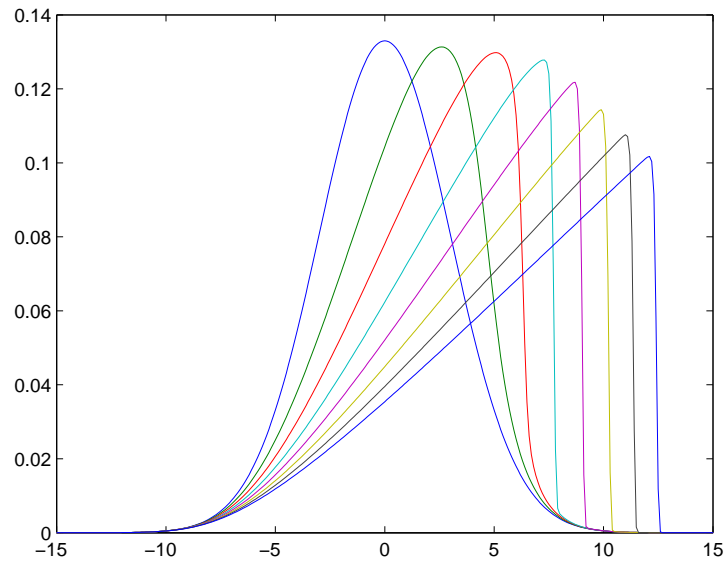


Figura 11: Resultado do método.

Anexo - códigos em MATLAB

Os códigos aqui presentes resolvem as EDPs dos exercícios 2 e 3. Os parâmetros de cada função são:

c: velocidade do vento (para a equação da onda);

u0: matriz linha $1 \times N$ representando o perfil inicial dos problemas;

x: matriz linha $1 \times N$ com as posições no espaço x ;

niter: número de iterações que o esquema deve executar;

deltat: tamanho dos passos no tempo.

Exercício 2

Esquema *upwind*

```
% Solves the wave equation  $u_t + c*u_x = 0$ .
function u = wave_upwind(c, u0, x, niter, deltat)
    N = size(x,2);
    deltax = (x(N) - x(1))/(N-1);
    sigma = c*deltat/deltax;

    u = zeros(niter, N);
    u(1,:) = u0;

    j = 2:N;
    for n = 2:niter
        u(n,1) = u0(1); % Boundary condition.

        u(n,j) = u(n-1,j) - sigma*(u(n-1,j) - u(n-1,j-1));
    end
end
```

Esquema *Lax-Wendroff*

```
% Solves the wave equation  $u_t + c*u_x = 0$ .
function u = wave_lax_wendroff(c, u0, x, n_iter, deltat)
    N = size(x,2);
    deltax = (x(N) - x(1))/(N-1);
    sigma = c*deltat/deltax;
    hsigma = 0.5*sigma;
    hsigma2 = 0.5*sigma*sigma;

    u = zeros(n_iter, N);
    u(1,:) = u0;

    j = 2:N-1;
    for n = 2:n_iter
        u(n,1) = u0(1); % Boundary condition.
        uo = u(n-1, :);
        u(n,j) = uo(j) - hsigma*(uo(j+1) - uo(j-1)) + ...
            hsigma2*(uo(j+1) - 2*uo(j) + uo(j-1));
        u(n,N) = u0(N); % Boundary condition.
    end
end
```

Exercício 3

```
% Solves the Burgers equation  $u_t + uu_x = 0$ .
function u = burgers(u0, x, niter, deltat)
    N = size(x,2);
    deltax = (x(N) - x(1))/(N-1);
    s = 0.5*deltat/deltax;

    u = zeros(niter, N);
    u(1,:) = u0;

    j = 2:N;
    for n = 2:niter
        u2 = u(n-1,:).^2;
        u(n,j) = u(n-1,j) - s*(u2(j) - u2(j-1));
    end
end
```