Lista 5

Introdução à Análise Numérica
Solução numérica de EDP

Lucas Emanuel Resck Domingues

23 de novembro de 2020

Esta é a solução da questão 7.

(a) O método CTCS para a equação da onda descrita foi implementado em Matlab e pode ser conferido no Apêndice A.

O algoritmo particiona $[-10, 10]$ com intervalo Δx desejado e $[0, T]$ com intervalo Δt desejado. Então calcula-se u_0 utilizando a função $f(x) = e^{-x^2}$, a matriz A utilizando $\sigma = c\Delta t/\Delta x = \Delta t/\Delta x$ (dado que $c = 1$) e, finalmente, u_1, utilizando u_0, $g(x) = 0$, $p(x) = 0$ e $q(x) = 0$. Dado que temos as condições iniciais, podemos partir para a iteração de u_i para obter o valor de u em todo o *grid*. Isso ocorre com a seguinte iteração (que é autoexplicativa):

```
aux = zeros(m, 1);
aux(1) = sigma^2*p(i*dt);
aux(end) = sigma^2*q(i*dt);
u(1:end, i+1) = A*u(1:end, i) - u(1:end, i-1) + aux;
```

A construção do gráfico ocorre utilizando a função **surf**. Antes da exibição do gráfico, é realizada uma diminuição da quantidade de pontos, amostrando apenas alguns deles, igualmente espaçados. Observe que isso não impacta o cálculo de u, afinal é apenas no momento da visualização que isso ocorre. Note que isso é necessário, pois são muitos pontos que impossibilitam a correta visualização da solução numérica u.

A Figura 1 mostra o resultado do método CTCS para $\Delta x = 0.04$ e $\Delta t = 0.02$ em $[-10, 10] \times [0, 120]$. Observamos que a onda inicial se divide em duas ondas menores, até que essas duas ondas "rebatem" na "parede" (condição de fronteira) e invertem. Depois elas se encontram novamente, dessa vez invertidas, logo se separaram, invertem e de juntam novamente, exatamente como a condição inicial.

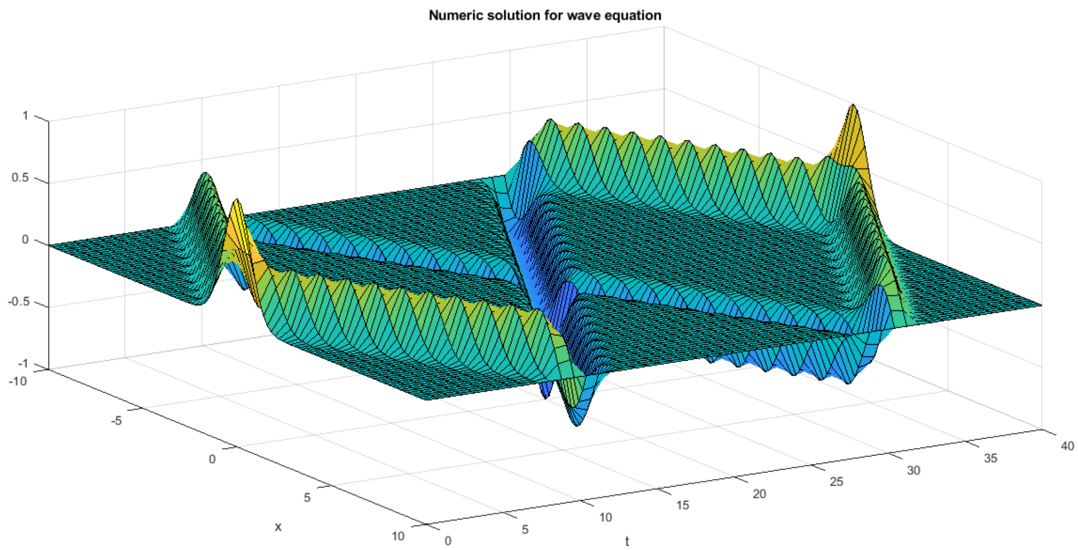

Figura 1: Solução numérica CTCS para a equação da onda, com $c = 1$, $\Delta x = 0.04$ e $\Delta t = 0.02$, na região $[-10, 10] \times [0, 40]$.

(b) As Figuras 2 e 3 mostram o resultado da solução numérica nas regiões $[-10, 10] \times [0, 80]$ e $[-10, 10] \times [0, 120]$, respectivamente.

Fica clara a repetição do padrão que é descrito na Figura 1. Esse padrão se repete 2 vezes na Figura 2 e 3 vezes na Figura 3.

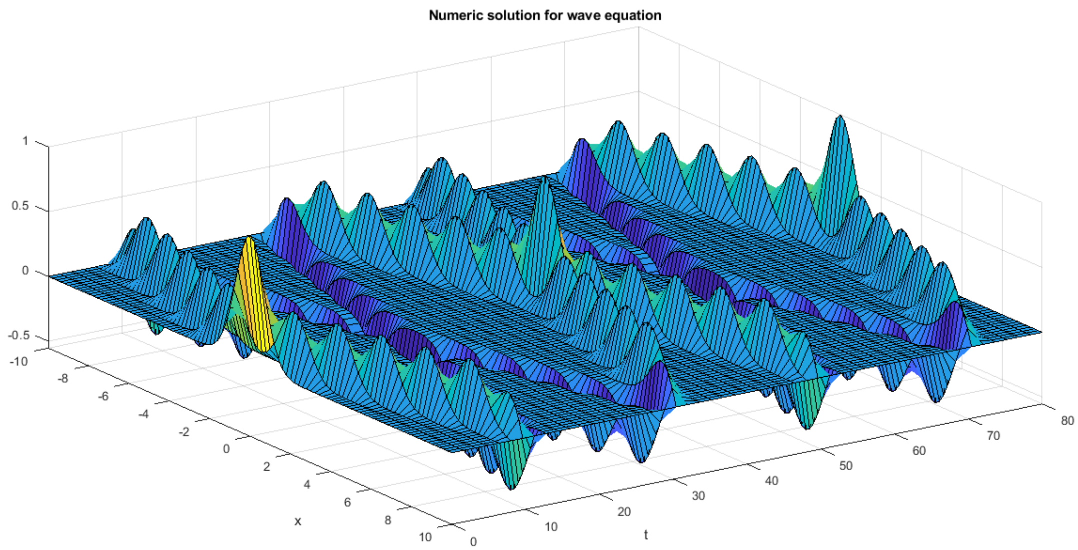

Figura 2: Solução numérica CTCS para a equação da onda, com $c = 1$, $\Delta x = 0.04$ e $\Delta t = 0.02$, na região $[-10, 10] \times [0, 80]$.

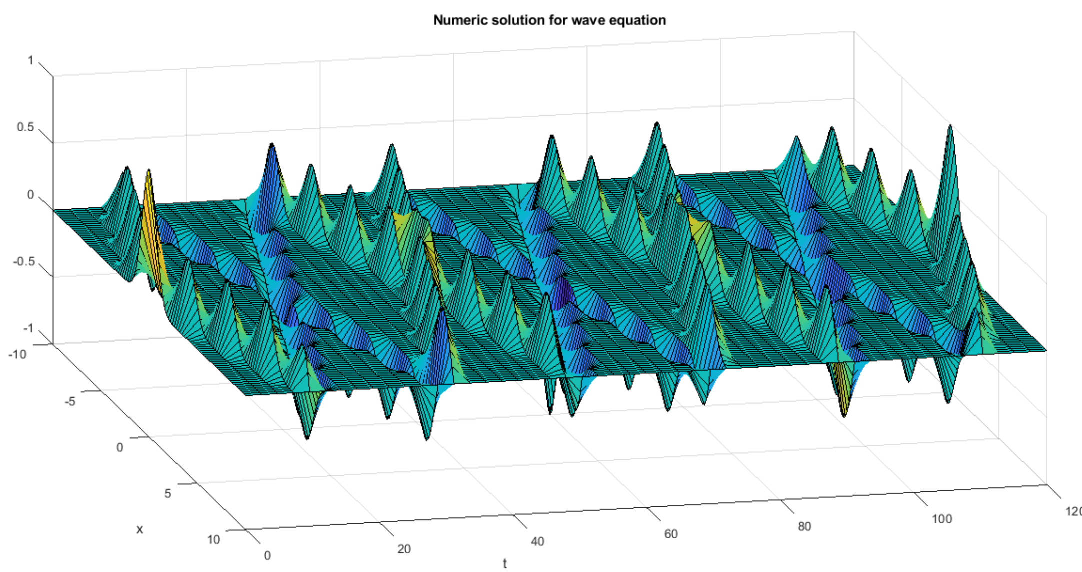

Figura 3: Solução numérica CTCS para a equação da onda, com $c = 1$, $\Delta x = 0.04$ e $\Delta t = 0.02$, na região $[-10, 10] \times [0, 120]$.

(c) A condição CFL foi obedecida em todos os exemplos anteriores. Vamos, agora, violar essa condição. Vamos escolher $\Delta x = 0.4$ e $\Delta t = 0.040008$. Ou seja,

$$\sigma = c\frac{\Delta t}{\Delta x} = \frac{0.040008}{0.4} > 1$$

Os valores foram escolhidos para que $\sigma > 1$.

A Figura 4 apresenta o resultado da solução numérica CTCS para $\Delta x = 0.04$ e $\Delta t = 0.040008$ em $[-10, 10] \times [0, 40]$. Observamos que a solução começa a "explodir". Note que esse valor curioso para Δt foi escolhido cuidadosamente de modo que fosse possível visualizar tanto a onda quanto a "explosão".

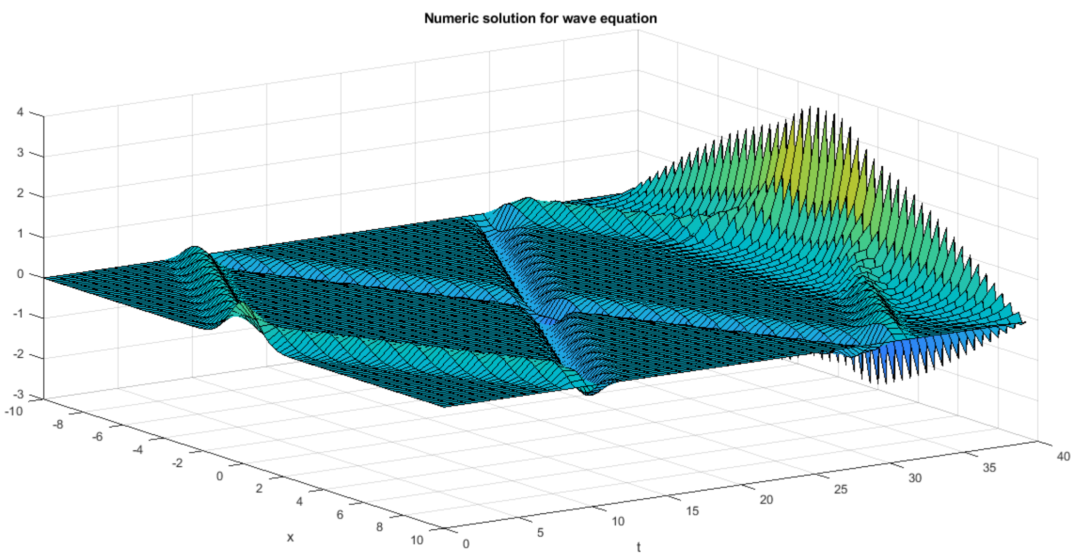

Figura 4: Solução numérica CTCS para a equação da onda, com $c = 1$, $\Delta x = 0.04$ e $\Delta t = 0.040008$, na região $[-10, 10] \times [0, 40]$.

A Figura 5 mostra o resultado para um valor de Δt, digamos, menos generoso: $\Delta t = 0.041$. A explosão chega à ordem de grandeza 10^{170}!

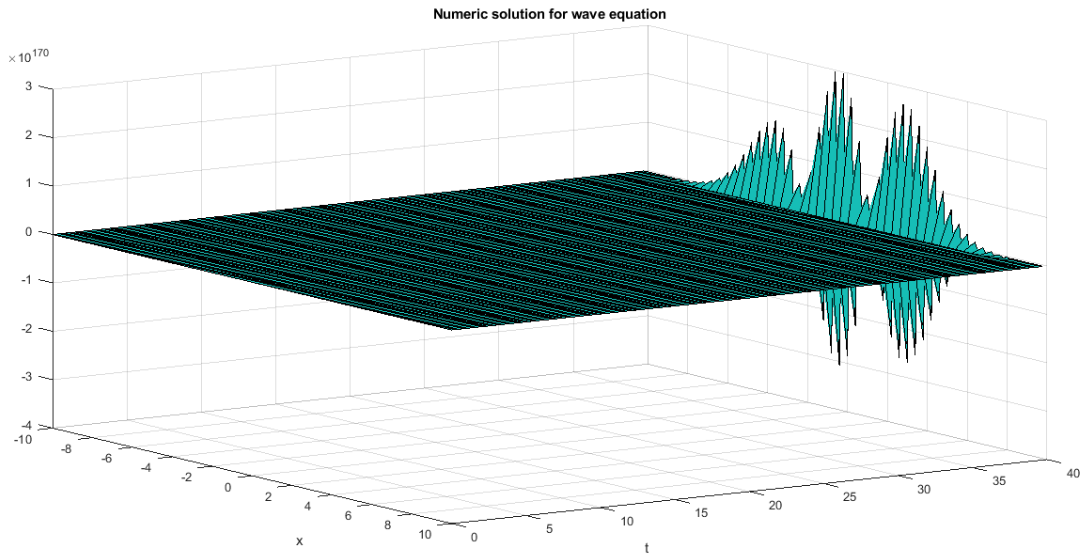

Figura 5: Solução numérica CTCS para a equação da onda, com $c = 1$, $\Delta x = 0.04$ e $\Delta t = 0.041$, na região $[-10, 10] \times [0, 40]$.

A Método CTCS para a equação da onda

Método CTCS para a equação da onda do exercício 7 implementado em Matlab. Também pode ser conferido neste link.

```
function [x, t, u] = ctcs_7(dx, dt, a, b, T)
    % Solve exercise 7 (wave equation and CTCS).
    % Examples:
        % [x, t, u] = ctcs_7(0.04, 0.02, -10, 10, 40);
        % [x, t, u] = ctcs_7(0.04, 0.040008, -10, 10, 40);
    [x, t, u] = ctcs(1, a, b, @f, @g, @p, @q, dx, dt, T);
    plot_u(x, t, u)
end

function [x, t, u] = ctcs(c, a, b, f, g, p, q, dx, dt, T)
    % Calculate the generic CTCS method.
    x = a:dx:b;
    x = x';
    [N_x, ~] = size(x);
    N_x = N_x - 1;
    u_0 = f(x(2:(end-1)));

    sigma = c*dt/dx;
    m = N_x - 1;
    A = A_matrix(sigma, m);
    aux = zeros(m, 1);
    aux(1) = p(0);
    aux(end) = q(0);
    u_1 = 1/2*A*u_0 + g(x(2:(end-1)))*dt + sigma^2/2*aux;

    t = 0:dt:T;
    t = t';
    [N_t, ~] = size(t);
    N_t = N_t - 1;
    u = zeros(m, N_t + 1);
    u(1:end, 1) = u_0;
    u(1:end, 2) = u_1;
    for i = 3:(N_t+1)
        i = i - 1;
        aux = zeros(m, 1);
        aux(1) = sigma^2*p(i*dt);
        aux(end) = sigma^2*q(i*dt);
        u(1:end, i+1) = A*u(1:end, i) - u(1:end, i-1) + aux;
    end
    u = [p(t)'; u; q(t)'];
end
```

```
function plot_u(x, t, u)
    % Plot the function through space and time.
    [m, n] = size(u);
    c = floor(m/100);
    d = floor(n/50);
    x = x(1:c:m);
    t = t(1:d:n);
    u = u(1:c:m, 1:d:n);
    [t, x] = meshgrid(t', x');
    surf(x, t, u);
    title('Numeric solution for wave equation');
    xlabel('x')
    ylabel('t')
end

function A = A_matrix(sigma, m)
    % Compute the A matrix of recurrence.
    A = eye(m)*(2-2*sigma^2);
    A(1:(end-1), 2:end) = A(1:(end-1), 2:end) + eye(m-1)*sigma^2;
    A(2:end, 1:(end-1)) = A(2:end, 1:(end-1)) + eye(m-1)*sigma^2;
end

function y = f(x)
    % Compute f function.
    y = exp(-x.^2);
end

function y = g(x)
    y = x - x;
end

function y = p(x)
    y = x - x;
end

function y = q(x)
    y = x - x;
end
```