Софийски университет "Св. Климент Охридски"

Факултет по математика и информатика

УЧЕБЕН ПРОЕКТ

по

Диференциални уравнения и приложения

спец. Софтуерно инженерство, 2$^{\text{ри}}$ курс, летен семестър,
учебна година 2021/2022

Тема №17

5 юли 2022 г.

София

Изготвил: *Даниел Халачев*

Ф. Номер: *62547*

Група: *II*

Оценка:...................................

Съдържание

1 Тема (задача) на проекта

Учебен проект по ДУПрил
спец. СИ, 2 курс, летен семесътр, уч. год. 2021/2022

Тема СИ2022-Г2-17. Движението на струна се моделира със следната задача

$$\left|\begin{array}{l} u_{tt} = 5u_{xx}, \quad t > 0, \ 0 < x < 7, \\ u|_{t=0} = \begin{cases} \arctan[(x^2 - 11x + 30)^2], & x \in [5,6] \\ 0, & x \in [0,5) \cup (6,7], \end{cases} \\ u_t|_{t=0} = 0, \quad 0 \leq x \leq 7, \\ u_x|_{x=0} = 0, \quad u_x|_{x=7} = 0, \quad t \geq 0. \end{array}\right.$$

1. (10 т.) Опишете как се получава решението на дадената задача с помощта на формулата на Даламбер и метода на продълженията.

2. (10 т.) Направете (с MATLAB) анимация на трептенето на струната за $t \in [0, 20]$. Начертайте с червен цвят в един прозорец една под друга графиките от направената анимация в моментите $t_1 = 0$, $t_2 = 5$, $t_3 = 15$ и означете коя графика за кое t се отнася (или тримерна графика на решението).

Срок за предаване 30.06.2022 г.

Фигура 1.1: Условие на задачата

2 Решение на задачата

2.1 Теоретична част

Трябва да решим следната задачата на Коши:

$$\left|\begin{array}{l} u_{tt} = 5u_{xx}, \quad t > 0, \quad 0 < x < 7 \\ u|_{t=0} = \begin{cases} \arctan\left[\left(x^2 - 11x + 30\right)^2\right], & x \in [5, 6] \\ 0, & x \in [0, 5) \cup (6, 7] \end{cases} \\ u_t|_{t=0} = 0, \quad 0 \le x \le 7 \\ u_x|_{x=0} = 0, \quad u_x|_{x=7} = 0, \quad t \ge 0 \end{array}\right. \tag{1}$$

От условието на задачата можем да заключим, че системата представя движение на ограничена струна със свободни краища. Левият край на струната е 0, а десният - $L = 7$. В началния момент струната е в положение

$$u|_{t=0} = \varphi(x) = \begin{cases} \arctan\left[\left(x^2 - 11x + 30\right)^2\right], & x \in [5, 6] \\ 0, & x \in [0, 5) \cup (6, 7] \end{cases}$$

Началната ѝ скорост е

$$u_t|_{t=0} = \psi(x) = 0$$

Преди да предприемем решение на задачата, е необходимо да проверим следните условия:

$$\left|\begin{array}{l} \varphi(x) \in C^2[0, L] \\ \psi(X) \in C^1[0, L] \\ \varphi'(0) = \psi'(0) = 0 \\ \varphi'(L) = \psi'(L) = 0 \end{array}\right.$$

В действителност всички условията са изпълнени. Задачата ще решим по следния начин:

- Ще направим четни продължения на функциите φ и ψ в интервала $[-L, L]$:

$$\widetilde{\varphi}(x) := \begin{cases} \varphi(x), & x \in [0, L] \\ \varphi(-x), & x \in [-L, 0) \end{cases}$$

$$\widetilde{\psi}(x) := \begin{cases} \psi(x), & x \in [0, L] \\ \psi(-x), & x \in [-L, 0) \end{cases}$$

- Ще направим периодични продължения с период $2L$ на новополучените функции $\widetilde{\varphi}$ и $\widetilde{\psi}$. Тези функции, точно като началните, са от същите класове - $\hat{\varphi}(x) \in C^2(\mathrm{R}), \hat{\psi}(x) \in C^1(\mathrm{R})$

$$\hat{\varphi}(x) := \begin{cases} \widetilde{\varphi}(x), & x \in [-L, L] \\ \hat{\varphi}(x - 2L), & x \in (L, +\infty) \\ \hat{\varphi}(x + 2L), & x \in (-\infty, L), \end{cases}$$

$$\hat{\psi}(x) := \begin{cases} \widetilde{\psi}(x), & x \in [-L, L] \\ \hat{\psi}(x - 2L), & x \in (L, +\infty) \\ \hat{\psi}(x + 2L), & x \in (-\infty, L) \end{cases}$$

- Ще формулираме нова задача на Коши за неограничена струна с новополучените функции $\hat{\varphi}$ и $\hat{\psi}$, но ще разгледаме решенията само за $-L < x < L, \quad t > 0$. По този начин ще получим решение и на изходната задача (1)

- Новополучената задача ще решим с формулата на Даламбер, която дава единственото решение $u \in C^2(\mathbb{R} \times [0, +\infty))$:

$$u(x,t) = \frac{1}{2}\left[\varphi(x - at) + \varphi(x + at)\right] + \frac{1}{2a} \int\limits_{x-at}^{x+at} \psi(s) \mathrm{d}s$$

- В MatLab ще изобразяваме само частта от графиката за $x \geq 0$

2.2 MatLab код и получени в командния прозорец резултати при изпълнението му

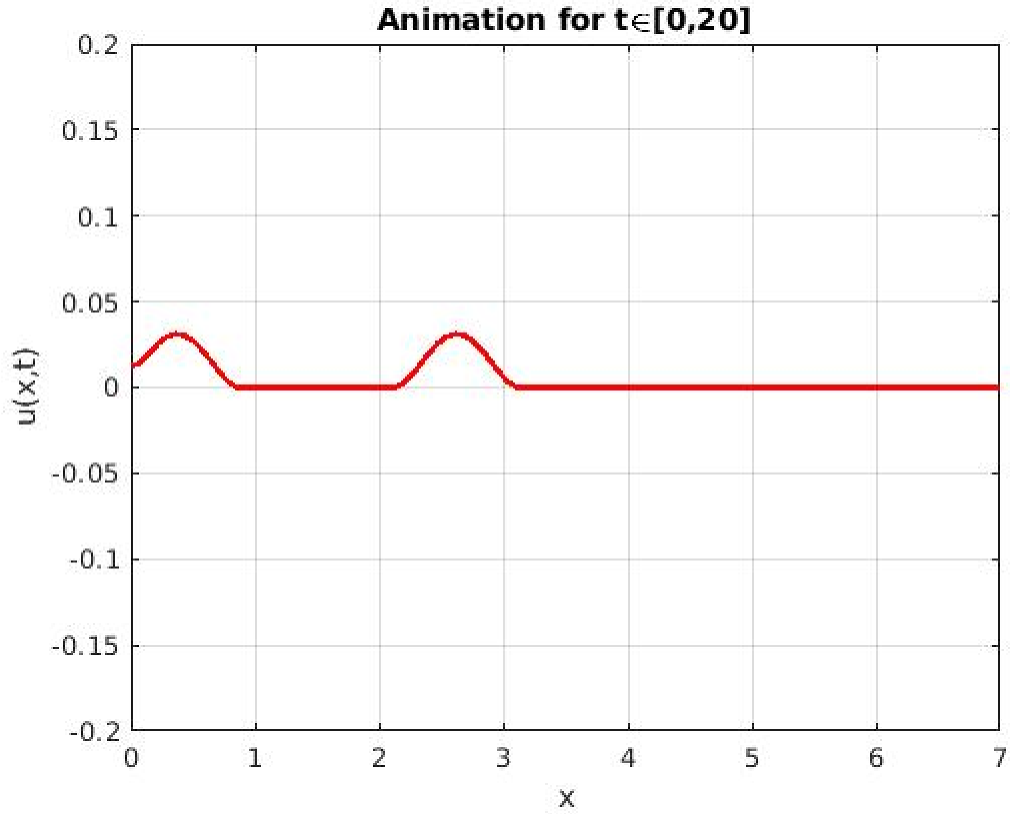
Забележка: кодът се изпълнява без предупреждения и компилационни грешки

```
function DalambertIsBoss
    clf
    T_MIN = 0;        % set t >= 0
    T_MAX = 20;       % set t <= 20
    t = linspace(T_MIN,T_MAX);
    X_MIN = 0;        % set x >= 0
    L = 7;            % set x <= L = 7
    x = linspace(X_MIN,L);
    A = sqrt(5);      % set a

    % varphi(x)
    function y = phi(x)
        y = zeros(1, length(x));
        for phiCounter = 1:length(x)
            if x(phiCounter) >= 5 && x(phiCounter) <= 6
                y(phiCounter) = atan((x^2-11*x+30)^2);
            else
                y(phiCounter) = 0;
            end
        end
    end

    % psi(x)
    function y = psi(~)
        y = 0;
    end

    % wildetilde{phi}(x)
    function y = phiEven(x)
        if x > 0
            y = phi(x);
        else
            y = phi(-x);
        end
    end

    % wildetilde{psi}(x)
    function y = psiEven(x)
        y = zeros(1, length(x));
        for psi2Counter = 1:length(x)
            if x(psi2Counter) > 0
                y(psi2Counter) = psi(x(psi2Counter));
            else
                y(psi2Counter) = psi(-x(psi2Counter));
```

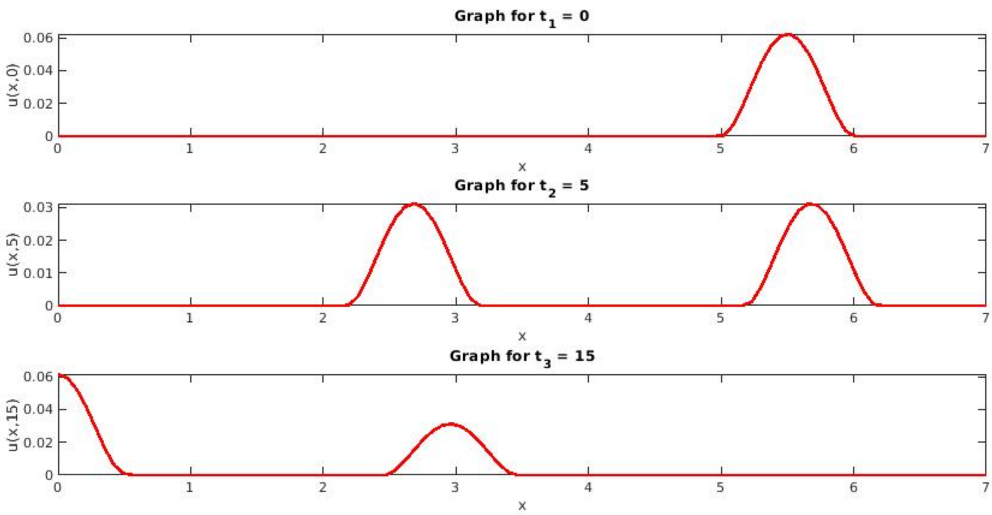
```
            end
        end
    end

    % hat{varphi}(x)
    function y = phiCont(x)
        if x > L
            y = phiCont(x-2*L);
        elseif x < -L
            y = phiCont(x+2*L);
        else
            y = phiEven(x);
        end
    end

    % hat{psi}(x)
    function y = psiCont(x)
        y = zeros(1, length(x));
        for psi3Counter = 1:length(x)
            if x(psi3Counter) > L
                y(psi3Counter) = psiCont(x(psi3Counter)-2*L);
            elseif x(psi3Counter) < -L
                y(psi3Counter) = psiCont(x(psi3Counter)+2*L);
            else
                y(psi3Counter) = psiEven(x(psi3Counter));
            end
        end
    end

    % D'Alambert's formula
    function y = Dalambert(x,t)
        a = A;
        y = zeros(1, length(x));
        for dalambertCounter = 1:length(x)
            if t == 0
                integral = 0;
            else
                s = linspace(x(dalambertCounter)-a*t,x(
                   dalambertCounter)+a*t);
                integral = trapz(s,psiCont(s));
            end
            y(dalambertCounter) = (phiCont(x(dalambertCounter)-a
               *t)+phiCont(x(dalambertCounter)+a*t))/2 +
               integral/(2*a);
        end
    end

    % Animation
```

```
    for i = 1:length(t)
        plot(x,Dalambert(x,t(i)),'r','Linewidth',2);
        title(sprintf('Animation for t\\in[0,20], t=%i', round(i/5)));
        axis ([X_MIN,L,-0.2,0.2]);
        grid on;
        xlabel('x');
        ylabel('u(x,t)');
        M = getframe;
    end

    % Graph for t_1 = 0
    subplot(3,1,1)
    plot(x,Dalambert(x,t(1)),'r','Linewidth',2)
    title('Graph for t_1 = 0')
    xlabel('x')
    ylabel('u(x,0)')
    % Graph for t_2 = 5
    subplot(3,1,2)
    plot(x,Dalambert(x,5),'r','Linewidth',2)
    title('Graph for t_2 = 5')
    xlabel('x')
    ylabel('u(x,5)')
    % Graph for t_3 = 15
    subplot(3,1,3)
    plot(x,Dalambert(x,15),'r','Linewidth',2)
    title('Graph for t_3 = 15')
    xlabel('x')
    ylabel('u(x,15)')
end
```

2.3 Графики (включително от анимация)



Фигура 2.1: Графиката по време на изпълнение на анимацията



Фигура 2.2: Графики в моментите $t_1 = 0, t_2 = 5$ и $t_3 = 15$

2.4 Коментари към получените с MatLab резултати

Кодът на Matlab се компилира без предупреждения и грешки. Форматиран е в удобен и лесен за четене стил. Приложени са оптимизации като предварителното алокиране на масива със стойности за итеративни функции като φ (в кода `phi`), без които на всяка итерация масивът се преоразмерява с 1 клетка, за да побере новото съдържание. В резултат кодът се изпълнява по-бързо. Резултатите за $u(x,t)$ под формата на анимация за t и на статични изображения за трите избрани момента $t_1 = 0, t_2 = 5$ и $t_3 = 15$ отговарят на очакванията - изобразяват единствено интервала $0 \leq x \leq 7$ и показват логични стойности за $u(x,t)$. Статичните изображения могат да бъдат разгледани след изпълнението на анимацията. И анимацията, и изображенията са генерирани с помощта на формулата на Даламбер (в кода `Dalambert`), имплементирана като функция на функциите $\hat{\varphi}$ и $\hat{\psi}$ (в кода `phiCont` и `psiCont`), и решаване на интеграла `integral` чрез функцията `trapz`.