

Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна

Розрахунково-графічне завдання #2
Чисельне розв'язання крайових задач для
звичайних диференціальних рівнянь

Виконав:
Захаров Дмитро Олегович
Група МП-41

Харків – 2025

Зміст

1	Постановка задачі	2
2	Опис методів	3
2.1	Метод Гальоркіна	3
2.2	Метод найменших квадратів	4
3	Імплементация	5
3.1	Підбір параметрів	5
3.2	Код на Wolfram Mathematica	6

1 Постановка задачі

Знайти розв'язок диференціального рівняння другого порядку з заданими граничними умовами:

$$y'' + x^3 y' - (2 + \ln^2(1 + x))y = 3, \quad x \in (0, 2)$$

Граничні умови:

$$-y'(0) + 3y(0) = 1, \quad y(2) = 2.$$

Розв'язок слід знайти з точністю $\varepsilon = 10^{-3}$ та побудувати графік розв'язку. Знайти розв'язок потрібно за допомогою хоча б одного з проєкційних методів (метод Гальоркіна, метод скінченних елементів, метод колокацій, метод найменших квадратів).

2 Опис методів

Нехай маємо рівняння другого порядку

$$\begin{cases} \mathcal{L}[y] = y'' + p(x)y' + q(x)y = f(x), & x \in (a, b), \\ e_0[y] := r_0y(a) + m_0y'(a) = \nu_0, \\ e_1[y] := r_1y(b) + m_1y'(b) = \nu_1, \end{cases}$$

Будемо шукати розв'язок у вигляді

$$\widehat{f}(x|\mathbf{c}) = \varphi_0(x) + \sum_{i=1}^n c_i \varphi_i(x),$$

де $\{\varphi_j\}_{j \in \{0, \dots, n\}}$ — лінійно незалежні базисні функції, причому $\varphi_0(x)$ має задовільняти початковим умовам, а решта функції задовільняють однорідним граничним умовам:

$$e_0[\varphi_0] = \nu_0, \quad e_1[\varphi_0] = \nu_1, \quad e_0[\varphi_j] = e_1[\varphi_j] = 0, \quad j \in \{1, \dots, n\}.$$

Наша задача — підібрати так набір коефіцієнтів $\mathbf{c} = (c_0, \dots, c_n) \in \mathbb{R}^{n+1}$, щоб зробити нев'язку $R(x|\mathbf{c})$ якомога меншою:

$$R(x|\mathbf{c}) = \mathcal{L}[\widehat{f}(x|\mathbf{c})] - f(x) = \mathcal{L}[\varphi_0(x)] + \sum_{i=1}^n c_i (\mathcal{L}[\varphi_i](x) - f(x)).$$

Далі, кожен з методів по суті задає критерій “малості” нев'язки $R(x|\mathbf{c})$. Розглянемо два з них, що будемо використовувати в нашій роботі.

2.1 Метод Гальоркіна

Головна ідея в тому, що якщо б ми могли гарантувати, що нев'язка $R(x)$ є ортогональною до всіх базисних функцій $\{\varphi_j(x)\}_{j \in \{0, \dots, n\}}$, то тоді звідси б випливало, що $R \equiv 0$. Якщо б в нас був би нескінченний набір базисних функцій, то ми б могли задовольнити цю умову точно.

В нашому ж випадку, ми підбираємо так коефіцієнти, щоб це виконувалось для перших n базисних функцій:

$$\langle R(x|\mathbf{c}), \varphi_i(x) \rangle = \int_a^b R(x|\mathbf{c}) \varphi_i(x) dx = 0, \quad i \in \{1, \dots, n\}.$$

Якщо розписати, то це зводиться до системи лінійних рівнянь:

$$\sum_{j=1}^n \alpha_{i,j} c_j = \beta_i, \quad \alpha_{i,j} = \int_a^b \varphi_i(x) \mathcal{L}[\varphi_j] dx, \quad \beta_i = \int_a^b \varphi_i(x) (f(x) - \mathcal{L}[\varphi_0]) dx.$$

2.2 Метод найменших квадратів

Тут ми кажемо, що нев'язка $R(x|\mathbf{c})$ є малим числом в середньому, тобто $\int_a^b R^2(x|\mathbf{c}) dx$ є достатньо малим. Формально:

$$\mathcal{I}(\mathbf{c}) = \int_a^b R^2(x|\mathbf{c}) dx \rightarrow \min$$

Ця задача оптимізації розв'язується стандартно: беремо градієнт $\nabla_{\mathbf{c}} \mathcal{I}(\mathbf{c})$ і прирівнюємо до нуля:

$$\nabla_{\mathbf{c}} \mathcal{I}(\mathbf{c}) = 2 \int_a^b R(x|\mathbf{c}) \nabla_{\mathbf{c}} R(x|\mathbf{c}) dx = \mathbf{0}_n \implies \int_a^b R(x|\mathbf{c}) \frac{\partial R(x|\mathbf{c})}{\partial c_i} dx = 0.$$

3 Імплементация

3.1 Підбір параметрів

В нашому випадку маємо підставити наступні значення та функції:

$$\mathcal{L}[y] = y'' + x^3 y' - (2 + \ln^2(1 + x))y, \quad f(x) \equiv 3, \quad a = 0, \quad b = 2,$$

Далі потрібно обрати базисні функції $\varphi_i(x)$, які задовольняють граничним умовам:

$$\begin{aligned} -\varphi'_0(0) + 3\varphi_0(0) &= 1, \quad \varphi_0(2) = 2, \\ -\varphi'_i(0) + 3\varphi_i(0) &= \varphi_i(2) = 0, \quad i \in \{1, \dots, n\}. \end{aligned}$$

Функцією $\varphi_0(x)$ можна шукати як лінійну: $\varphi_0(x) = ax + b$. Нескладно перевірити, що підійде $\varphi_0(x) = \frac{5}{7}x + \frac{4}{7}$.

З функціями $\varphi_n(x)$, $n \geq 1$ все складніше. Зокрема, з теорії диференціальних рівнянь, для такої умови можна взяти наступний ортогональний набір функцій:

$$\varphi_n(x) = \cos(\omega_n x) + \frac{3}{\omega_n} \sin(\omega_n x), \quad \tan(2\omega_n) = -\frac{\omega_n}{3}$$

Проте працювати з такими функціями не дуже зручно, тому спробуємо прибрати умову на ортогональність і візьмемо поліноміальний набір функцій:

$$\varphi_n(x) = (x - 2)^n (a_n x + b_n)$$

Умова $\varphi_n(2) = 0$ виконується автоматично, лише підберемо коефіцієнти a_n та b_n . Маємо отримати $-\varphi'_n(0) + 3\varphi_n(0) = 0$ або:

$$\left[-n(x - 2)^{n-1}(a_n x + b_n) - a_n(x - 2)^n + 3(x - 2)^n(a_n x + b_n) \right]_{x=0} = 0$$

Це спрощується до умови $-nb_n + 2a_n - 6b_n = 0$, звідки $a_n = \frac{n+6}{2}b_n$. Візьмемо $b_n := 2$ для зручності. Тоді:

$$\varphi_0(x) = \frac{5}{7}x + \frac{4}{7}, \quad \varphi_n(x) = (x - 2)^n ((6 + n)x + 2)$$

3.2 Код на Wolfram Mathematica

Код програми наведений на мові *Wolfram Mathematica* у файлі, що прикріплений далі.

Перегорніть на наступну сторінку →

Section 2. Projective Methods


■ Section 2.1. Galerkin Method

Section 2.1.1. Solving the initial problem.

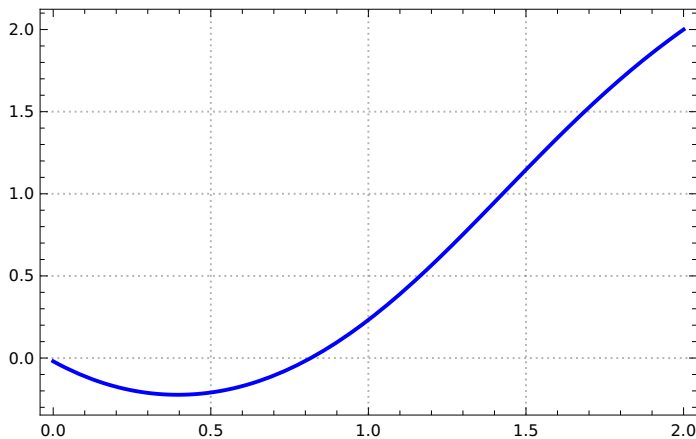
First, let us solve the initial problem


```
In[29]:= Clear[y, x]
s = NDSolve[{y'[x] + x^3 y'[x] - (2 + Log[x + 1]^2) y[x] == 3, -y'[0] + 3 y[0] == 1, y[2] == 2}, y, {x, 0, 2}]
Plot[Evaluate[y[x] /. s], {x, 0, 2},
PlotRange -> All,
PlotTheme -> "Detailed",
AxesLabel -> {"x", "y"},
GridLines -> Automatic,
PlotStyle -> Blue]
```

Out[30]=

```
{{y -> InterpolatingFunction[ Domain: {{0., 2.}} Output: scalar ]}}
```

Out[31]=



— InterpolatingFunction[ Domain: (0. Output: scalar]

Section 2.1.2. Choosing the basis functions

Now, choose basis functions. Choosing the 0th basis function is trivial:

In[175]:=

```

 $\phi_0[x_] = 5 x / 7 + 4 / 7;$ 
 $(-\phi_0'[x] + 3 \phi_0[x]) /. \{x \rightarrow 0\} (* \text{ Must be 1 } *)$ 
 $\phi_0[2] (* \text{ Must be 2 } *)$ 

```

Out[176]=

1

Out[177]=

2

Next functions will be chosen as the system of linearly independent polynomial of the special form:

In[178]:=

```

Clear[ $\phi_n$ , x, an, bn, n]
 $\phi_n[x_] = (x - 2)^n (an x + bn);$ 
boundaryn[x_] =  $-\phi_n'[x] + 3 \phi_n[x]$ 
Solve[boundaryn[0] == 0, {an, bn}]

```

Out[180]=

$$-an(-2+x)^n - n(-2+x)^{-1+n}(bn+anx) + 3(-2+x)^n(bn+anx)$$

 **Solve:** Equations may not give solutions for all "solve" variables.

Out[181]=

$$\left\{ \left\{ bn \rightarrow \frac{2 an}{6 + n} \right\} \right\}$$

Thus, we are ready

$n = 5; (* \text{ Number of basis functions. For } n > 5 \text{ I reach the limit of Wolfram Cloud } *)$

```

 $\phi_0[x_] = 5 x / 7 + 4 / 7;$ 
 $\phi = \text{Table}[(x - 2)^k ((6 + k) x + 2), \{k, 1, n\}];$ 
Print["Basis Functions: ",  $\phi_0[x]$ , " and ",  $\phi$ ]

```

Basis Functions: $\frac{4}{7} + \frac{5x}{7}$ and

$$\{(-2+x)(2+7x), (-2+x)^2(2+8x), (-2+x)^3(2+9x), (-2+x)^4(2+10x), (-2+x)^5(2+11x)\}$$

Verify that these basis functions satisfy boundary conditions as needed

In[206]:=

```

Table[ $(-D[\phi[k], x] + 3 \phi[k]) /. \{x \rightarrow 0\}, \{k, 1, n\} (* \text{ Must be all 0 } *)$ 
Table[ $\phi[k] /. \{x \rightarrow 2\}, \{k, 1, n\} (* \text{ Must be all 0 as well } *)$ 

```

Out[206]=

{0, 0, 0, 0, 0}

Out[207]=

{0, 0, 0, 0, 0}

Setting the approximation:

In[228]:=

```
Clear[c]
coeffs = Array[Subscript[c, #] &, {n}];
u[x_] = 0[x] + Sum[coeffs[[k]] * 0[k], {k, 1, n}]
```

Out[230]=

$$\frac{4}{7} + \frac{5x}{7} + (-2+x)(2+7x)c_1 + (-2+x)^2(2+8x)c_2 + (-2+x)^3(2+9x)c_3 + (-2+x)^4(2+10x)c_4 + (-2+x)^5(2+11x)c_5$$

Section 2.1.3. Optimizing w.r.t. coefficients

In[232]:=

```
L[y_] = D[y[x], {x, 2}] + x^3 D[y[x], x] - (2 + Log[1 + x]^2) y[x];
R[x_] = L[u] - 3;
eqs = Table[Integrate[R[x] * 0[k], {x, 0.0, 2.0}] == 0, {k, 1, n}];
```

In[235]:=

```
Print[eqs]
{67.9364 - 569.638 c_1 + 697.672 c_2 - 921.527 c_3 + 1311.69 c_4 - 1979.89 c_5 == 0,
 -80.3527 + 573.329 c_1 - 939.299 c_2 + 1487.47 c_3 - 2402.02 c_4 + 3978.31 c_5 == 0,
 110.273 - 731.356 c_1 + 1399.7 c_2 - 2486.13 c_3 + 4369.65 c_4 - 7717.86 c_5 == 0,
 -164.923 + 1053.25 c_1 - 2222.04 c_2 + 4267.69 c_3 - 7968.23 c_4 + 14756.3 c_5 == 0,
 260.807 - 1628.81 c_1 + 3676.88 c_2 - 7473.44 c_3 + 14600.3 c_4 - 28041.7 c_5 == 0}
```

In[237]:=

```
optimalCoeffs = Solve[eqs, coeffs]
```

Out[237]=

```
{{c_1 -> 0.0401299, c_2 -> -0.0516019, c_3 -> 0.0587259, c_4 -> 0.0514798, c_5 -> 0.0113561}}
```

In[243]:=

```
optimalCoeffs = Table[optimalCoeffs[[1]][[k]][[2]], {k, 1, n}]
```

Out[243]=

```
{0.0401299, -0.0516019, 0.0587259, 0.0514798, 0.0113561}
```

In[244]:=

```
uOptimal[x_] = 0[x] + Sum[optimalCoeffs[[k]] * 0[k], {k, 1, n}]
```

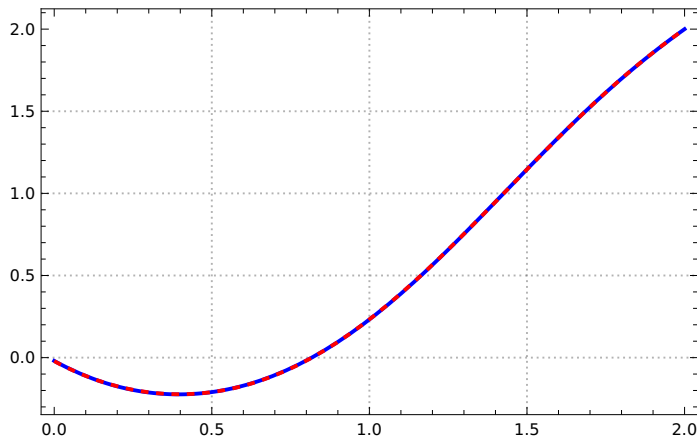
Out[244]=

$$\frac{4}{7} + \frac{5x}{7} + 0.0401299(-2+x)(2+7x) - 0.0516019(-2+x)^2(2+8x) + 0.0587259(-2+x)^3(2+9x) + 0.0514798(-2+x)^4(2+10x) + 0.0113561(-2+x)^5(2+11x)$$

In[246]:=

```
Plot[{Evaluate[y[x] /. s], uOptimal[x]}, {x, 0, 2},
PlotRange -> All,
PlotTheme -> "Detailed",
AxesLabel -> {"x", "y"},
GridLines -> Automatic,
PlotStyle -> {Blue, Directive[Dashed, Red]}]
```

Out[246]=



— InterpolatingFunction[
 - - - uOptimal(x)

Print absolute differences in the specified set of points:

In[299]:=

```
m = 20;
xs = Table[2 k / m, {k, 1, m}];
t = Table[Abs[uOptimal[xs[[k]]] - y[xs[[k]] /. s], {k, 1, m}]
Max[t]
```

Out[301]=

```
{{0.000381356}, {0.000788737}, {0.00121535}, {0.000774551},
{0.000114845}, {0.000905342}, {0.00119906}, {0.0008864}, {0.000151172},
{0.000635991}, {0.00109295}, {0.00100677}, {0.000445847}, {0.000258035},
{0.000675533}, {0.000539131}, {0.0000356648}, {0.000543849}, {0.00045365}, {0.}}
```

Out[302]=

```
0.00121535
```

Section 2.2. Least Squares Method

We simply change the system of equations for finding coefficients. The rest is the same.

In[304]:=

```
eqs = Table[Integrate[R[x] * D[R[x], coeffs[[k]]], {x, 0.0, 2.0}] == 0, {k, 1, n}];
Print[eqs]
{-495.681 + 6912.42 c1 - 3371.77 c2 + 3773.29 c3 - 5309.95 c4 + 8429.41 c5 == 0,
 466.082 - 3371.77 c1 + 5498.65 c2 - 8663.26 c3 + 14 891.7 c4 - 26 879.8 c5 == 0,
 -615.633 + 3773.29 c1 - 8663.26 c2 + 17 601.5 c3 - 35 070.8 c4 + 69 560.3 c5 == 0,
 928.554 - 5309.95 c1 + 14 891.7 c2 - 35 070.8 c3 + 76 975.6 c4 - 163 204. c5 == 0,
 -1525.51 + 8429.41 c1 - 26 879.8 c2 + 69 560.3 c3 - 163 204. c4 + 363 413. c5 == 0}
```

In[306]:=

```
optimalCoeffs = Solve[eqs, coeffs]
```

Out[306]=

```
{{c1 → 0.040226, c2 → -0.0523426, c3 → 0.0557251, c4 → 0.0486133, c5 → 0.0105585}}
```

In[307]:=

```
optimalCoeffs = Table[optimalCoeffs[[1]][[k]][[2]], {k, 1, n}]
uOptimal[x_] = φ[x] + Sum[optimalCoeffs[[k]] * φ[k], {k, 1, n}]
```

Out[307]=

```
{0.040226, -0.0523426, 0.0557251, 0.0486133, 0.0105585}
```

Out[308]=

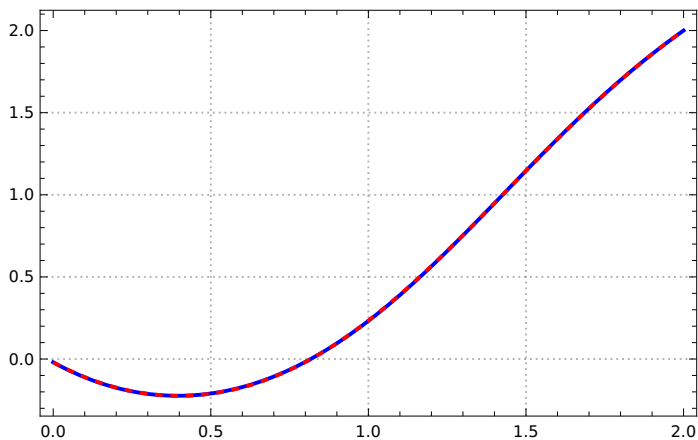
$$\frac{4}{7} + \frac{5x}{7} + 0.040226(-2+x)(2+7x) - 0.0523426(-2+x)^2(2+8x) +$$



$$0.0557251(-2+x)^3(2+9x) + 0.0486133(-2+x)^4(2+10x) + 0.0105585(-2+x)^5(2+11x)$$

In[309]:=

```
Plot[{Evaluate[y[x] /. s], uOptimal[x]}, {x, 0, 2},
 PlotRange → All,
 PlotTheme → "Detailed",
 AxesLabel → {"x", "y"},
 GridLines → Automatic,
 PlotStyle → {Blue, Directive[Dashed, Red]}]
```

Out[309]=



— InterpolatingFunction[  Domain:
Output: :

- - - uOptimal(x)

In[310]:=

```
m = 20;
xs = Table[2 k / m, {k, 1, m}];
t = Table[Abs[uOptimal[xs[[k]] - y[xs[[k]] /. s], {k, 1, m}]]
Max[t]
```

Out[312]=

```
{{0.000476117}, {0.000991769}, {0.000728391}, {0.000186296},
 {0.0012365}, {0.00188564}, {0.00181404}, {0.00102341}, {0.000185937},
 {0.00134327}, {0.00199801}, {0.00190942}, {0.00116135}, {0.000140637},
 {0.000638744}, {0.000811352}, {0.000404109}, {0.000128277}, {0.000230791}, {0.}}
```

Out[313]=

```
0.00199801
```