BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

KHOA ĐIỆN TỬ - VIỄN THÔNG

BÁO CÁO ĐỒ ÁN CUỐI KHÓA

MÔN PHƯƠNG PHÁP TÍNH VÀ MATLAB

Học kỳ I – Năm học: 2022-2023

Nguyễn Dũng   MSSV: 20200121.

GVHD: Nguyễn Xuân Vinh.

Thành phố Hồ Chí Minh, ngày 25 tháng 12 năm 2022

MỤC LỤC

I. PHẦN LÝ THUYẾT CÁC PHƯƠNG PHÁP SỬ DỤNG TRONG ĐỒ ÁN MÔN HỌC:

1. Chương 8: Giải gần đúng phương trình:

1.1. Các bước giải gần đúng phương trình $f(x) = 0$:

Để tìm nghiệm gần đúng (thực) của phương trình $f(x) = 0$, ta thường tiến hành theo các bước sau:

1. Khẳng định phương trình có nghiệm trong một khoảng nào đó: **khoảng phân ly nghiệm $[\boldsymbol{a}, \boldsymbol{b}]$.**
2. Chọn một xấp xỉ đầu x_o, xem đó là nghiệm gần đúng đầu tiên thuộc khoảng đang xét $[a, b]$.
3. Điều chỉnh dần x_o sao cho càng gần tới nghiệm đúng α càng tốt.
4. Đánh giá sai số của nghiệm gần đúng tìm được so với nghiệm đúng.

1.2. Khoảng phân ly nghiệm:

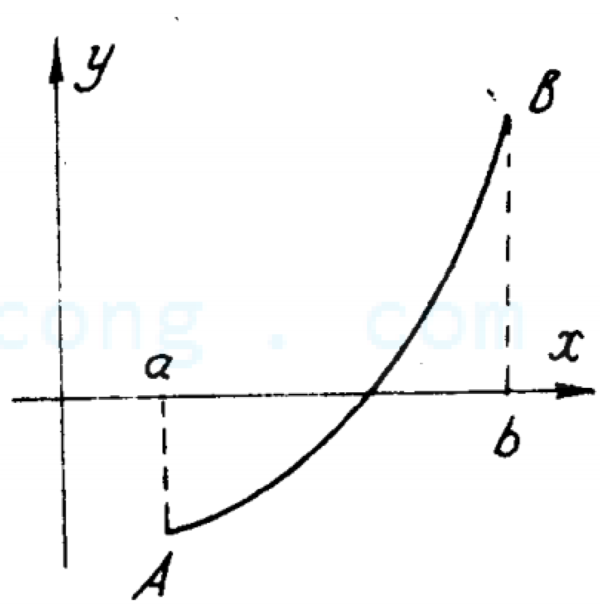
Xét phương trình $f(x) = 0$ $(*)$

Nếu $f(x)$ liên tục và đơn điệu trên khoảng $[a, b]$, đồng thời $f(a)$ và $f(b)$ trái dấu

Định nghĩa: Giả sử α là một nghiệm của phương trình (3.1). Ta nói [a, b] là khoảng phân ly nghiệm α của phương trình $(*)$ nếu trong khoảng đó phương trình $(*)$ chỉ có một nghiệm thực α duy nhất.

Định lý 1: Giả sử $f(x)$ là hàm số liên tục trên tập $D \subset \mathbb{R}$. Khi đó $(a, b) \subset D$ là một khoảng phân ly nghiệm của phương trình (*) nếu $f(a).f(b) < 0$ và $f(x)$ đơn điệu trên $[a, b]$.

Định lý 2: Với giả thiết của định lý 1 thì (a, b) là khoảng phàn ly nghiêm α của phương trình $(*)$ nếu $f(a) * f(b) < 0$ và tồn tại $f'(x)$ đồng thời $f'(x)$ giữ nguyên một dấu $\forall x \in [a, b]$.



Từ định lý trên ta suy ra, nếu (a, b) là khoảng phân ly nghiệm α của phương trình $(*)$ thì mọi khoảng $(c, d) \subset (a, b)$, sao cho $f(c).f(d) < 0$ cũng là khoảng phân ly nghiệm của phương trình trên (thu hep khoảng chứa nghiệm).

1.3. Phương pháp lặp đơn:

Giả sử $[a, b]$ là khoảng phân ly nghiệm α của phương trình $f(a) = 0, \alpha \in [a, b]$.

Tiến hành viết lại phương trình dưới dạng: $x = g(x)$ trong đó $g(x)$ liên tục trong $[a, b]$.

Chọn x_o là điểm bất kỳ trong $[a, b]$. Tính dãy lặp theo công thức sau (cho đến khi sai số thỏa mãn)

$$x_n = g(x_{n-1}) \;\; n = 1,2, \dots, k$$

Định lý: Giả thuyết tồn tại số $q > 0$ sao cho $|g'(x)| \leq q < 1$. Khi đó quá trình lặp hội tụ.

Minh họa hình học của phương pháp:

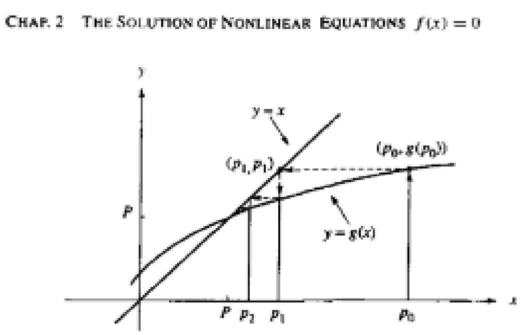


Figure 2.4 (a) Monotone convergence when $0 < g'(P) < 1$.

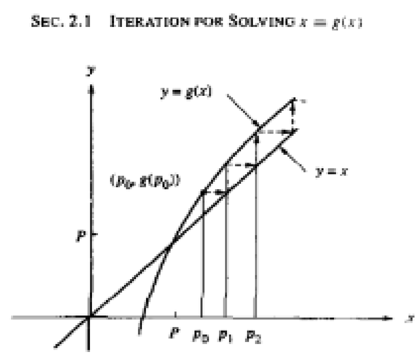


Figure 2.5 (a) Monotone divergence when $1 < g'(P)$.

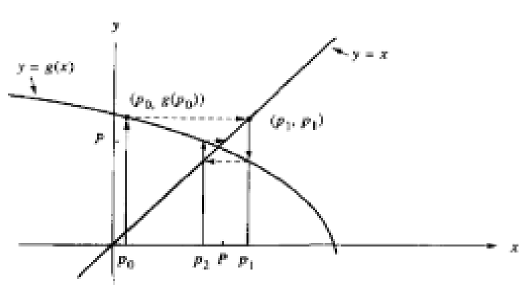


Figure 2.4 (b) Oscillating convergence when $-1 < g'(P) < 0$.

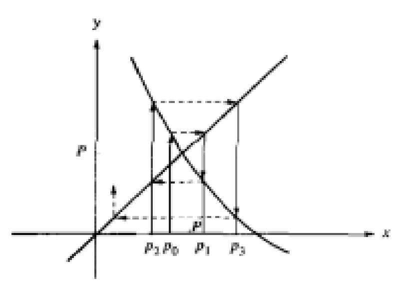


Figure 2.5 (b) Divergent oscillation when $g'(P) < -1$.

Sai số ước lượng:

$$|x_n - \alpha| \leq \frac{q}{1-q}|x_n - x_{n-1}|$$

Đánh giá:

- Ưu điểm: xấp xỉ đầu x_o không nhất thiết phải gần α, có khả năng sửa sai, dễ lập trình.
- Nhược điểm: Khi q gần 1 thì hội tụ rất chậm.

Thuật toán:

Từ phương trình $f(x) = 0$ đề cho:

Bước 1: Xác định sai số cho phép.

Bước 2: Xác định khoảng phân li nghiệm $[a, b]$.

Bước 3: Tìm hàm lặp hội tụ $g(x)$ (tức biến đổi x về vế trái, phần còn lại về vế phải chính là $g(x)$.

Bước 4: Chọn xấp xỉ ban đầu là $x_o \in [a, b]$.

Bước 5: Tính (lặp) giá trị $x_n = g(x_{n-1})$,với $n = 1, 2, 3. ..$đến khi $|\, x_n – x_{n-1}| < \ sai\ số$ thì dừng.

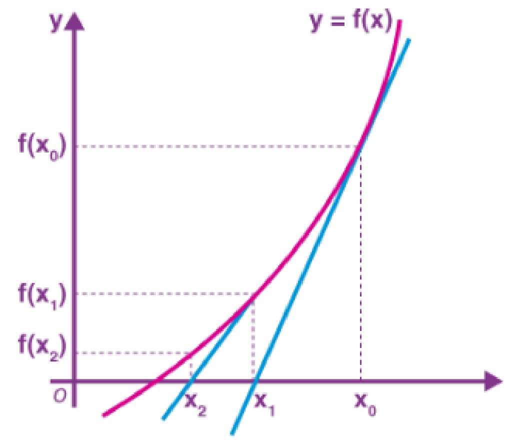
Bước 6: Kết luận nghiệm = x_n mới nhất và số lần lặp.

Sau k lần lặp, được $x_n \approx \alpha$ là nghiệm gần đúng của phương trình. Sai số được đánh giá theo công thức (hoặc khi thỏa điều kiện sai số tính toán < sai số cho phép ε.

1.4. Phương pháp tiếp tuyến (PP Newton-Raphson):

Ý tưởng chủ đạo của phương pháp này là tìm cách thay phương trình $f(x) = 0$, phi tuyến đối với x, bằng một phương trình gần đúng, tuyến tính đối với x.

Giả sử $[a, b]$ là khoảng phân ly nghiệm α của phương trình $f(x) = 0$ liên tục với $f'(x)$ và $f''(x)$ không đổi dấu trên $[a, b]$. Nghiệm đúng của phương trình là hoành độ giao điểm của đường cong $y = f(x)$ với trục hoành Ox.



Nội dung phương pháp:

Thay đường cong $y = f(x)$ bằng tiếp tuyến kẻ từ $A\big(a, f(a)\big)$ hay $B\big(b, f(b)\big)$ coi hoành độ giao điểm của tiếp tuyến và Ox là nghiệm gần đúng.

Đặt $x_o = a$ nếu tiếp tuyến kẻ từ A, $x_o = b$ nếu tiếp tuyến kẻ từ B. Phương trình tiếp tuyến tại $\big(x_o, f(x_o)\big)$ có dạng như sau:

$$y = f'(x_o)(x - x_o) + f(x_o) = 0$$

$$\Rightarrow x_1 = x_o - \frac{f(x_o)}{f'(x_o)}$$

Ta lại có tiếp tuyến vẽ từ điểm $\big(x_1, f(x_1)\big)$, tương tự xấp xỉ tuyến tính tiếp theo có dạng:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Ta đó, ta có công thức tổng quát như sau:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Điều kiện để thực hiện được phương pháp tiếp tuyến:

Điều kiện để hội tụ là $f'(x)$ và $f''(x)$ không đổi dấu trên $[a, b]$.

Xấp xỉ đầu x_o được chọn là a hay b sao cho $f(x_o).f''(x_o) > 0\ \forall x \in [a, b]$, nói cách khác $f(x_0)$ cùng dấu với $f''(x_o)$ khi đó x_o sẽ hội tụ.

Đánh giá sai số:

$$|x_n - \alpha| \leq \frac{|f(x_n)|}{m_1}$$

trong đó $0 < m_1 \leq |f'(x)|\ \forall x \in [a, b]$, hoặc

$$|x_n - \alpha| \leq \frac{M_2}{2m_1}|x_n - x_{n-1}|^2$$

với $M_2 \geq |f''(x)|\ \forall x \in [a, b]$.

Trên thực tế thường dừng quá trình tính toán khi

$$|x_n - x_{n-1}| < \varepsilon \text{ (sai số cho phép)}$$

Nhận xét:

Phương pháp tiếp tuyến (PP Newton-Raphson) còn gọi là phương pháp tuyến tính hóa, hội tụ nhanh hơn so với phương pháp chia đôi và phương pháp lặp đơn. Tuy nhiên việc kiểm tra điều kiện để áp dụng phương pháp này phức tạp hơn.

Tóm tắt thuật toán:

Bước 1: Tính $f'(x)$, $f''(x)$ và xét dấu của chúng.

Bước 2: Chọn x_o được chọn là a hay b sao cho $f(x_o)$ cùng dấu với $f''(x)$

$$f(x_o).f''(x) > 0 \quad \forall x \in [a, b]$$

Bước 3: Từ xấp xỉ đầu x_o, tính

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Sau k lần lặp ta thu được $x_k \approx \alpha$ là nghiệm gần đúng của phương trình.

1.5. Phương pháp dây cung:

Ý tưởng: Trong phương pháp tiếp tuyến, ta đã thay cung đồ thị AB của hàm $y = f(x)$ bởi tiếp tuyễn vẽ tại A hay B. Bây giờ ta thay cung AB thành dây cung vẽ từ A đến B.

Giả sử $[a, b]$ là khoảng phân ly nghiệm α của phương trình $f(\alpha) = 0$ liên tục với $f'(x)$ và $f''(x)$ không đổi dấu trên $[a, b]$.

Trong phương pháp tiếp tuyến ta thay đường cong $y = f(x)$ bằng tiếp tuyến của đường cong. Trong phương pháp dây cung, ta thay bằng dây cung nối 2 điểm $A\big(a, f(a)\big)$ và $B\big(b, f(b)\big)$. Nghiệm đúng của phương trình là hoành độ giao điểm của dây cung AB với trục hoành Ox.

Phương trình dây cung:

$$y = \frac{f(b) - f(a)}{b - a} = \frac{x - a}{b - a} \Rightarrow x_1 = a - \frac{b - a}{f(b) - f(a)} f(a)$$

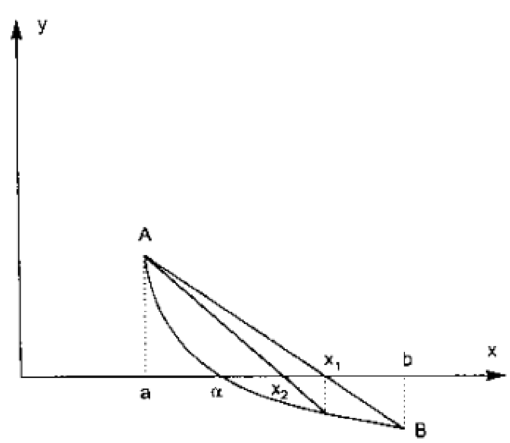
$$\Rightarrow x_1 = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

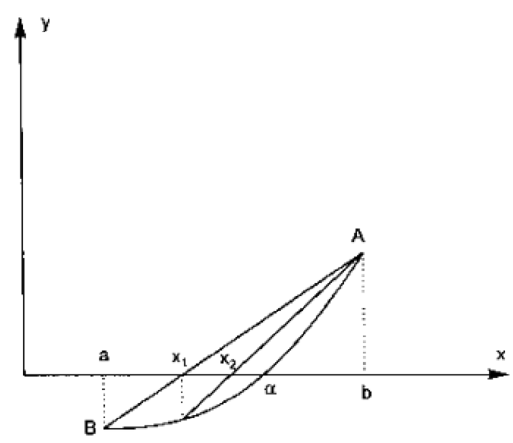
Chọn $x_o = a$, đặt $d = b$. Lặp lại quá trình trên, ta có:

$$x_n = x_{n-1} - \frac{d - x_{n-1}}{f(d) - f(x_{n-1})} f(x_{n-1})$$

$$x_n = \frac{x_{n-1} f(d) - d f(x_{n-1})}{f(d) - f(x_{n-1})}$$

Minh họa hình học:





Điều kiện hội tụ:

Giả sử $[a, b]$ là khoảng phân ly nghiệm α của phương trình $f(x) = 0$. Hàm số $f(x)$ liên tục và có đạo hàm $f'(x), f''(x)$; đồng thời các đạo hàm đó giữ nguyên dấu trên đoạn $[a, b]$. Xấp xỉ ban đầu x_o được chọn sao cho

$$\begin{cases} f(x_o).f''(x) < 0 \\ f(x_o).f(d) < 0 \end{cases}$$

Khi đó dãy $\{x_n\}$ tính theo công thức ở trên sẽ đơn điệu hội tụ tới α khi $n \to \infty$.

Đánh giá sai số:

Được đánh giá theo công thức sau:

$$|x_n - \alpha| \le \frac{M_1 - m_1}{m_1}|x_n - x_{n-1}|$$

trong đó,

$$0 < m_1 \le |f'(x)| \le M_1 \quad \forall x \in [a, b]$$

Để $|x_n - \alpha| < \varepsilon$ thì

$$|x_n - x_{n-1}| < \frac{m_1\varepsilon}{M_1 - m_1}.$$

Nhận xét: Phương pháp dây cung, hội tụ khá nhanh so với phương pháp chia đôi và phương pháp lặp đơn nhưng chậm hơn phương pháp tiếp tuyến (PP Newton-Raphson). Tuy nhiên việc kiểm tra điều kiện để áp dụng phương pháp này khá phức tạp.

Tóm tắt thuật toán:

Bước 1: Tính $f'(x)$ và $f''(x)$, xét dấu của chúng.

Hội tụ khi $f'(x)$ và $f''(x)$ không đổi dấu trong $[a, b]$

Bước 2: Kiểm tra điều kiện:

$$\begin{cases} f(x_o).f''(x) < 0 \\ f(x_o).f(d) < 0 \end{cases}$$

Để chọn xấp xỉ ban đầu x_o là a hay b.

Bước 3: Từ xấp xỉ ban đầu x_o, tính

$$x_n = x_{n-1} - \frac{d - x_{n-1}}{f(d) - f(x_{n-1})}f(x_{n-1})$$

hoặc,

$$x_n = \frac{x_{n-1}.f(d) - d.f(x_{n-1})}{f(d) - f(x_{n-1})}$$

Sau k lần lặp như phương pháp chia đôi ta sẽ thu được $x_k \approx \alpha$ là ngiệm gần đúng của phương trình.

2. Chương 9: Giải hệ phương trình:

2.1. Dạng tổng quát của một hệ phương trình tuyến tính:

Một hệ phương trình tuyến tính có thể có m phương trình n ẩn.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

Pham vị môn học chỉ xét hệ phương trình có n phương trình n ẩn như sau:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Trong đó a_{ij} là hệ số của ẩn x_j ở phương trình thứ i, f_i là vế phải của phương trình thứ i. Giả sử, ta đã biết a_{ij} và f_i, ta cần tìm các giá trị ẩn x_j

Từ hệ phương trình, ta có được các ma trận:

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

được gọi là ma trận hệ số, và A là ma trận vuông.

$$b = \begin{vmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{vmatrix} \quad x = \begin{vmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{vmatrix}$$

là các vector hệ số và vector của ẩn.

Từ đó, ta chuyển hệ phương trình trên thành dạng ma trận $Ax = b$ như sau:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

2.2. Sự tồn tại và duy nhất của nghiệm:

Sự dụng phương pháp Cramer:

Xét một hệ phương trình $Ax = b$, trong đó A là ma trận vuông cấp n. Để A khả nghịch thì $\det(A) \neq 0$.

Khi đó, nghiệm duy nhất của hệ phương trình được tính theo công thức $X = A^{-1}b$ hoặc sử dụng công thức Cramer:

$$x_i = \frac{\Delta_i}{\Delta} = \frac{\det(A_i)}{\det(A)} \quad (i = 1,2, \dots n)$$

Trong đó $\Delta = \ det(A)$ và $\Delta_i = \ det(A_i)$ là định thức khi A thay bởi cột thứ i bằng cột B.

Nhược điểm của phương pháp này là khi n càng lớn thì thực hiện càng nhiều phép tính.

(Nguồn: Slide/74)

n	Số phép nhân	Số phép cộng
4	360	115
5	2880	716
10	359,251,810	39,916,791

Có rất nhiều phương pháp khác để tìm nghiệm gần đúng của hệ phương trình. Một trong số là **phương pháp Gauss-Jordan**.

2.3. Phương pháp Gauss-Jordan:

Phương pháp khử Gauss-Jordan dùng cách khử dẫn các ẩn đưa hệ phương trình đã cho về một dạng ma trận đường chéo rồi giải hệ phương trình này, chúng ta không cần tính một định thức nào cả. Phương pháp này là loại trừ ẩn nên thực chất là biến đổi hệ số của hệ phương trình để được hệ tương đương sao cho trong một phương trình của hệ mới chỉ còn lại một ẩn số x_k. Rồi sau đó tiến hành giải hệ phương trình.

Xét về khối lượng tính toán, đối với hệ n phương trình n ẩn, người ta dựa vào các công thức tính để đếm các phép nhân, chia. Số lượng các phép tính cụ thể như sau:

- Nhân và chia:

$$N = \frac{n}{3}(n^2 + 3n - 1)$$

- Cộng và trừ:

$$C = \frac{n}{6}(2n^2 + 3n - 5)$$

Ưu điểm: giảm được số phép tính toán khi giải hệ phương trình.

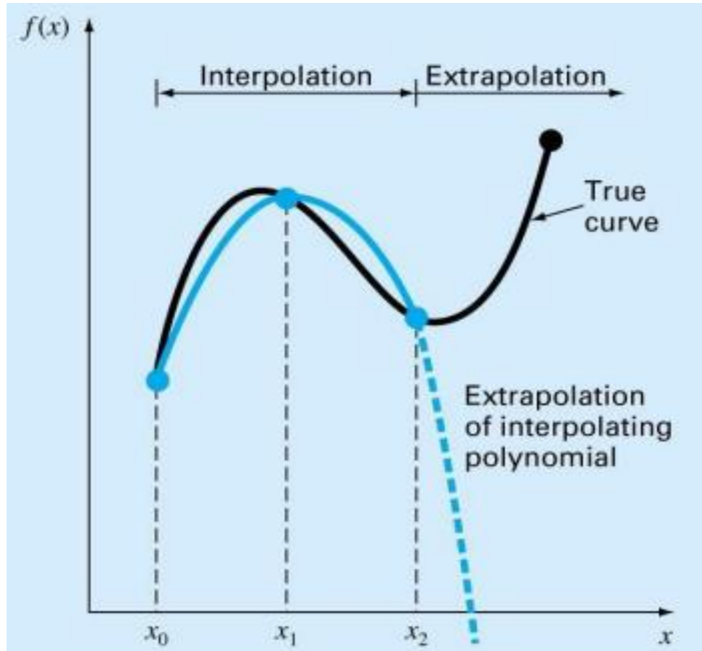
n	N	C
3	17	11
4	36	26
5	65	50
10	430	375

Chi tiết các bước tiến hành sẽ được trình bày rỏ ở mục II (giải bài tập).

3. Chương 10: Nội suy, hồi quy:

3.1. Các khái niệm:

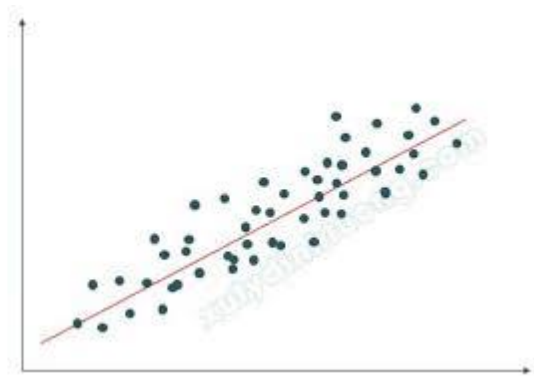
Nội suy: là phương pháp ước tính giá trị của các điểm dữ liệu chưa biết trong phạm vi của một tập hợp rời rạc chứa một số điểm dữ liệu đã biết, tức dự báo giá trị của biến phụ thuộc từ giá trị của biến độc lập nằm trong khoảng biến thiên của kết quả quan sát cho nó, kết quả dự báo đáng tin cậy hơn.



Ngoại suy: là phương pháp ước tính giá trị (tương lai) chưa biết dựa vào các giá trị (quá khứ) đã biết

.

Hồi quy: Tìm một đường biểu diễn thể hiện khuynh hướng thay đổi của dữ liệu.



3.2. Nội suy Newton (có mốc cách đều):

Xét bài toán nội suy có $a = x_o < x_1 < x_2 \dots < x_n = b$, các nút x_i cách đều nhau với

$$h = x_i - x_{i-1} = \frac{b-a}{n} \quad (i = 1,2,3, \dots n)$$

Bảng sai phân hưu hạn:

Sai phân tiến:

$$\Delta y_i = y_{i+1} - y_i \quad (i = 0{:}\overline{n-1})$$

được gọi là sai phân tiến cấp một.

Sai phân tiến cấp 2 là sai phân tiến của sai phân tiến cấp một và được ký hiệu như sau:

$$\Delta^2 y_i = \Delta(\Delta y_i) = \Delta(y_{i+1} - y_i) = \Delta y_{i+1} - \Delta y_i = y_{i+2} - 2y_{i+1} + y_i$$

$$i = 0:\overline{n-2}$$

Tổng quát, sai phân tiến của sai phân tiến cấp $n - 1$ là sai phân tiến cấp n, được ký hiệu:

$$\Delta^n y_o = \Delta(\Delta^{n-1} y_o) = \Delta^{n-1} y_1 - \Delta^{n-1} y_o$$

Sai phân lùi:

$$\nabla y_i = y_i - y_{i-1} \quad (i = 1:\overline{n})$$

được gọi là sai phân lùi cấp một.

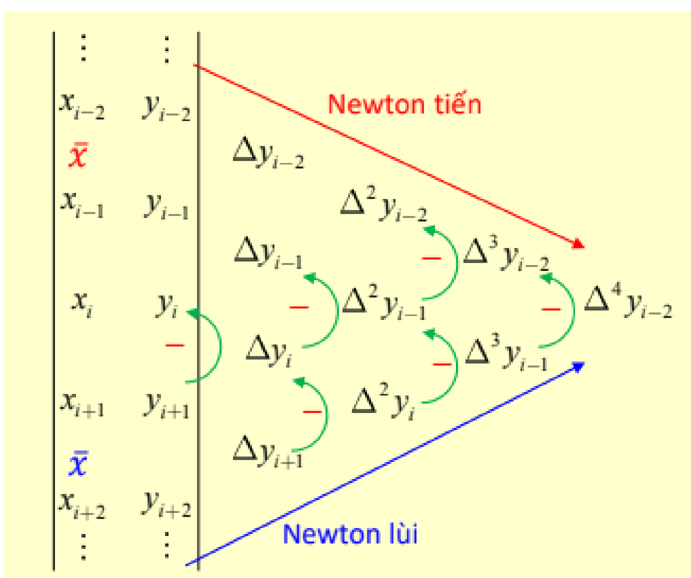
Sai phân lùi cấp n là sai phân lùi của sai phân lùi cấp $n - 1$ và được ký hiệu như sau:

$$\nabla^n y_1 = \nabla(\nabla^{n-1} y_1) = \nabla^{n-1} y_1 - \nabla^{n-1} y_o$$

Mối quan hệ giữa sai phân tiến và sai phân lùi

$$\Delta^k y_i = \nabla^k y_{i+k}$$

Từ đó, ta có bảng sai phân như sau:



Nguồn: Slide/134.

Các công thức:

$$t_{tiến} = \frac{x - x_o}{h}$$

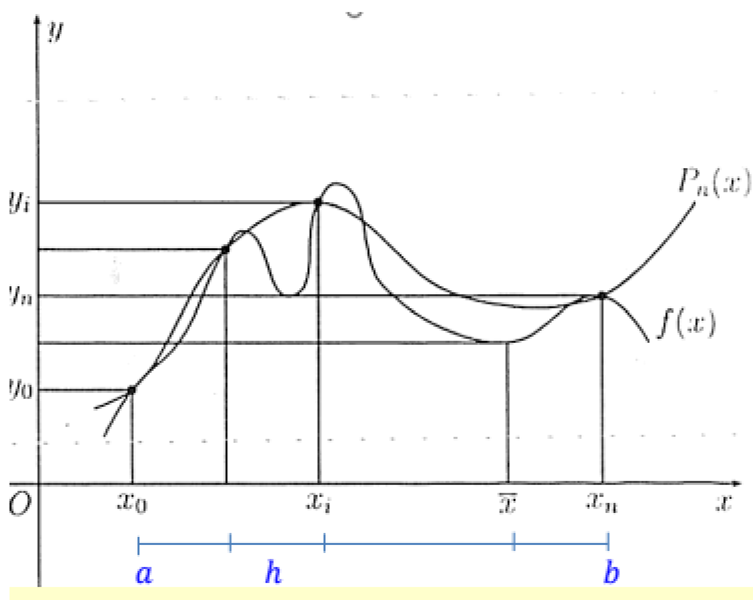
$$t_{lùi} = \frac{x - x_n}{h}$$

Đa thức nội suy Newton tiến có mốc nội suy cách đều:

$$P_n(x) = P_n(x_o + ht) = y_o + \frac{t}{1!}\Delta y_o + \frac{t(t-1)}{2!}\Delta^2 y_o + \cdots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_o$$

Đa thức nội suy Newton lùi có mốc nội suy cách đều:

$$P_n(x) = P_n(x_n + ht) = y_n + \frac{t}{1!}\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \cdots + \frac{t(t+1)\ldots t(t+n-1)}{n!}\Delta^n y_o$$



Đánh giá sai số:

$$R_n(x) \approx \frac{\Delta^{n+1} y_o}{(n+1)!}\prod_{k=0}^{n}(t-k)$$

Chú ý: Nếu cần tìm giá trị gần đúng $\bar{y}$ tại điểm $\bar{x} \notin [x_o, x_n]$ (ngoài bảng), trong trường hợp này

Nếu $\bar{x} < x_o$ ta sử dụng công thức tiến.
Nếu $\bar{x} > x_o$ ta sử dụng công thức lùi.

4. Chương 11 + 12: Đạo hàm, tích phân, Phương trình vi phân:

4.1. Bài toán tích phân:

Già sử $f(x)$ là hàm số liên tuc trên đoan $[a,b]$ có nguyên hàm là $F(x)$, khi đó giá trị của tích phân xác đinh trên đoan $[a,b]$ đươc tính theo công thức Newton-Leibniz:

$$I = \int_a^b f(x)dx \ = F(x)|_a^b = F(b) - F(a)$$

Tuy nhiên trong nhiều trường hợp có $F(x)$ là nguyên hàm của hàm $f(x)$ không phải là hàm số sơ cấp hoặc $F(x)$ có biểu thức quá phức tạp. Khi đó giá trị của tích phân I chỉ tìm được là số gần đúng. Trong trường hợp hàm số cho trong dạng bảng số thì khái niệm nguyên hàm không còn ý nghĩa gì nữa.

Vậy nên chúng ta có một số cách giải gần đúng như sau:

Các phương pháp Newton - Cotes (khoảng lấy tích phân được chia đều: phương pháp hình thang, Simpson).

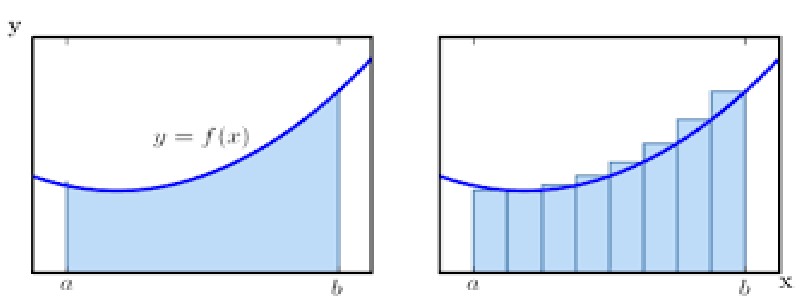
Phương pháp Gauss: các điểm chia được chọn để đạt độ chính xác cao nhất.

Ý tưởng của phương pháp Newton – Cotes: Chia $[a,b]$ thành $(n\text{-} 1)$ đoạn bằng nhau có chiều dài $h \ = \ (b \text{ - } a)/(n \text{ - } 1)$ và kí hiệu $x_1, x_2, .., x_n$. Sau đó ta xấp xỉ hàm $f(x)$ bằng đa thức bậc $(n \text{ - } 1)$ đi qua các nút. Thay các biểu thức hàm phức tạp hay các bảng dữ liệu bằng các hàm xấp xỉ dễ lấy tích phân.

$$I = \int_a^b f(x)dx \approx \int_a^b f_n(x)dx$$

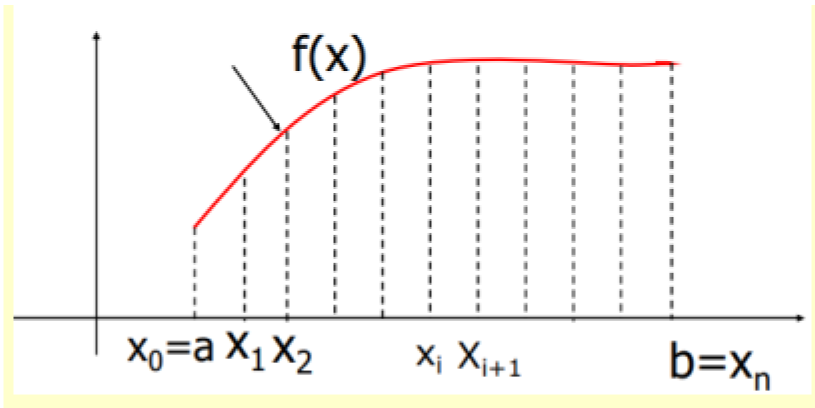
trong đó,

$$f_n(x) = a_o + a_1x + a_2x^2 + \cdots a_nx^n$$



4.2. *Phương pháp hình thang:*

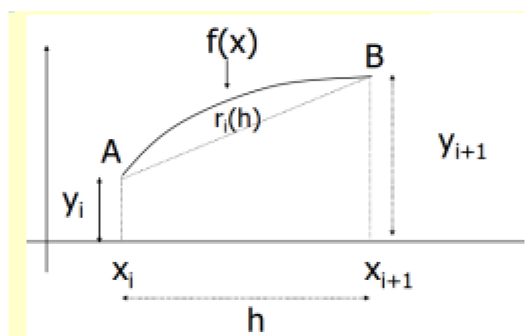
Phân đoạn $[a,b]$ thành n đoạn con bằng nhau: $a = x_0 < x_1 <\ldots< x_n = b$ với $h = \frac{b-a}{n}$. Nối các điểm y_i và y_{i+1} bằng các đường thẳng



Khi đó: Giá trị tích phân là tổng diện tích các tích phân từng phần hình thang

$$I = \int_a^b f(x)dx = \int_{x_o}^{x_1} f(x)dx = \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx.$$

Tiến hành xét trên từng hình thang:



Công thức tính phân hình thang:

$$I_i = \int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{1}{2}.h.(y_i + y_{i+1})$$

Sai số địa phương (Sai số từng phần):

$$|r_i(x)| \leq \frac{Mh^3}{12}, \quad M = \max|f''(x)| \text{ với } x \in [x_i, x_{i+1}]$$

Công thức hình thang tổng quát:

$$I = \int_a^b f(x)dx \approx \frac{b-a}{2n}(y_o + 2y_1 + 2y_2 + \cdots + 2y_{n-1} + y_n)$$

Sai số toàn phần:

$$|r(h)| \leq n.\frac{Mh^3}{12} = \frac{M(b-a)}{12}.h^2$$

Với $M = \max|f''(x)| \quad x \in [a,b]$

Dựa trên công thức sai số toàn phần, để tăng độ chính xác, ta tiến hành chia bước nhảy h càng nhỏ, tương ứng với tăng số đoạn n lên.

II. PHẦN GIẢI CÁC BÀI TẬP THEO LÝ THUYẾT:

Chương 8: Giải gần đúng phương trình

Bài tập 3.2 (bài số 1.) trang 103: Bằng phương pháp lặp đơn, phương pháp Newton, và phương pháp dây cung tìm nghiệm gần đúng của các phương trình sau sao cho đạt sai số tuyệt đối < 0.01

1. $f(x) = x^3 - 12x - 5 = 0$

Phương pháp lặp đơn:

- Xác định khoảng phân ly nghiệm:
  Ta có: $f(x)$ liên tục trên $\mathbb{R}$.
  $f(3) = -14 < 0$
  $f(4) = 11 > 0$
  $\Rightarrow f(3).f(4) < 0$
  Nên suy ra khoảng phân ly nghiệm của $f(x)$ là $[a,b] = [3,4]$
- Xác định hàm lặp:
  Chọn $x = g(x) = \sqrt[3]{(5+12x)} = (5+12x)^{\frac{1}{3}}$
  Kiểm tra điều kiện hội tụ:
  Ta có: $g'(x) = 4(5+12x)^{-\frac{2}{3}}$
  $|g'(x)| = \left|4(5+12x)^{-\frac{2}{3}}\right| \le |g'(3)| \approx 0.3364 < 1$
  $\Rightarrow$Thõa mãn điều kiện hàm lặp hội tụ.
- Xấp xỉ giá trị ban đầu $x_o = 3$
- Tìm nghiệm dựa vào công thức hàm lặp:

$$x_n = g(x_{n-1}) = (5+12x)^{\frac{1}{3}}$$

n	x_{n-1}	x_n
1	3	3.4482
2	3.4482	3.5928
3	3.5928	3.6371

4	3.6371	3.6505
5	3.6505	3.6545

Theo đề bài, ta có sai số tuyệt đối $e < 0.01$ mà $|3.6545 - 3.6505| = 0.004 < 0.01$

⇒Dừng quá trình lặp.

Vậy nghiệm gần đúng của phương trình $f(x) = 0$ là $x \approx 3.6545$

<u>Phương pháp Newton – Raphson:</u>

- Xác định khoảng phân ly nghiệm:
  Ta có: $f(x)$ liên tục trên $\mathbb{R}$.
  $f(3) = -14 < 0$
  $f(4) = 11 > 0$
  $\Rightarrow f(3).f(4) < 0$
  Nên suy ra khoảng phân ly nghiệm của $f(x)$ là $[a, b] = [3,4]$
- Kiểm tra điều kiện hội tụ:
  Ta có:
  $$f'(x) = 3x^2 - 12 > 0\ \forall x \in [3,4]$$
  $$f''(x) = 6x > 0 \forall x \in [3,4]$$
  ⇒Thõa mãn điều kiện hội tụ.
- Xấp xỉ giá trị ban đầu $x_o = 4$. Vì thõa mãn điều kiện
  $$f(4).f''(x) > 0, \forall x \in [3,4]$$
- Tìm nghiệm dựa trên công thức tiếp tuyến:
  $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

n	x_n	x_{n+1}
0	4	3.6944
1	3.6944	3.6567
2	3.6567	3.6562

Theo đề bài, ta có sai số tuyệt đối $e < 0.01$ mà $|3.6562 - 3.6567| = 0.0005 < 0.01$

⇒Dừng quá trình lặp.

Vậy nghiệm gần đúng của phương trình $f(x) = 0$ là $x \approx 3.6562$

Phương pháp dây cung:

- Xác định khoảng phân ly nghiệm:
  Ta có: $f(x)$ liên tục trên $\mathbb{R}$.
  $f(3) = -14 < 0$
  $f(4) = 11 > 0$
  $\Rightarrow f(3).f(4) < 0$
  Nên suy ra khoảng phân ly nghiệm của $f(x)$ là $[a, b] = [3,4]$
- Kiểm tra điều kiện hội tụ:
  Ta có:
  $$f'(x) = 3x^2 - 12 > 0 \ \forall x \in [3,4]$$
  $$f''(x) = 6x > 0 \forall x \in [3,4]$$
  ⇒Thõa mãn điều kiện hội tụ.
- Xấp xỉ giá trị ban đầu $x_o = 3, d = 4$. Vì thõa mãn điều kiện
  $$\begin{cases} f(3).f''(x) < 0, \forall x \in [3,4] \\ f(3).f(4) < 0 \end{cases}$$
- Tìm nghiệm theo công thức dây cung:
  $$x_n = \frac{x_{n-1}f(d) - d.f(x_{n-1})}{f(d) - f(x_{n-1})}$$

Ta có bảng sau:

Số lần lặp n	x_{n-1}	d	x_n
1	3	4	3.56
2	4	3.56	3.6442
3	3.56	3.6442	3.6566
4	3.6442	3.6566	3.6562

Theo đề bài, ta có sai số tuyệt đối $e < 0.01$ mà $|3.6566 - 3.6562| = 0.0004 < 0.01$

⇒Dừng quá trình lặp.

Vậy nghiệm gần đúng của phương trình $f(x) = 0$ là $x \approx 3.6562$.

Chương 9: Giải hệ phương trình

Bài tập 4.1 (bài số 4,5,6) trang 158: Giải hê phương trình $Ax = b$, bằng phương pháp Gauss-Jordan, trong đó:

Các được toán được làm tròn tới 4 chữ số ở phần thập phân.

4.

$$A = \begin{bmatrix} 1.7 & 0.2 & 0.1 & 0.3 \\ 0.2 & 1.9 & -0.1 & 0.2 \\ 0.1 & -0.1 & 1.3 & -0.4 \\ 0.3 & 0.2 & -0.4 & 4.1 \end{bmatrix} \qquad b = \begin{bmatrix} 1.6 \\ 0.3 \\ -1.2 \\ 0.7 \end{bmatrix}$$

Ghép hai ma trận:

$$Ab = \begin{bmatrix} 1.7 & 0.2 & 0.1 & 0.3 & & 1.6 \\ 0.2 & 1.9 & -0.1 & 0.2 & & 0.3 \\ 0.1 & -0.1 & 1.3 & -0.4 & & -1.2 \\ 0.3 & 0.2 & -0.4 & 4.1 & & 0.7 \end{bmatrix}$$

Tuyến tính hóa hàng 1: Chia hàng đầu tiên cho hệ số đầu tiên = 1.7, ta có:

$$Ab = \begin{bmatrix} 1 & 0.1176 & 0.0588 & 0.1765 & & 0.9412 \\ 0.2 & 1.9 & -0.1 & 0.2 & & 0.3 \\ 0.1 & -0.1 & 1.3 & -0.4 & & -1.2 \\ 0.3 & 0.2 & -0.4 & 4.1 & & 0.7 \end{bmatrix}$$

Triệt tiêu hệ số x_1 ở hàng 2,3,4 bằng cách:

Lấy hàng 2 – (hàng 1)*0.2

Lấy hàng 3 – (hàng 1)*0.1

Lấy hàng 4 – (hàng 1)*0.3

$$Ab = \begin{bmatrix} 1 & 0.1176 & 0.0588 & 0.1765 & 0.9412 \\ 0 & 1.8765 & -0.1118 & 0.1647 & 0.1118 \\ 0 & -0.1118 & 1.2941 & -0.4177 & -1.2941 \\ 0 & 0.1647 & -0.4176 & 4.0471 & 0.4176 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 2 bằng cách chia hàng 2 cho 1.8765

$$Ab = \begin{bmatrix} 1 & 0.1176 & 0.0588 & 0.1765 & 0.9412 \\ 0 & 1 & -0.0596 & 0.0878 & 0.0596 \\ 0 & -0.1118 & 1.2941 & -0.4177 & -1.2941 \\ 0 & 0.1647 & -0.4176 & 4.0471 & 0.4176 \end{bmatrix}$$

Triệt tiêu x_2 ở hàng 1,3,4 bằng cách:

Lấy hàng 1 – (Hàng 2)*0.1176

Lấy hàng 3 – (Hàng 2)*(-0.1118)

Lấy hàng 4 – (Hàng 2)*(0.1647)

$$Ab = \begin{bmatrix} 1 & 0 & 0.0658 & 0.1662 & 0.9342 \\ 0 & 1 & -0.0596 & 0.0878 & 0.0596 \\ 0 & 0 & 1.2874 & -0.4079 & -1.2874 \\ 0 & 0 & -0.4078 & 4.0326 & 0.4078 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 3 bằng cách chia hàng 3 cho 1.2874

$$Ab = \begin{bmatrix} 1 & 0 & 0.0658 & 0.1662 & 0.9342 \\ 0 & 1 & -0.0596 & 0.0878 & 0.0596 \\ 0 & 0 & 1 & -0.3168 & -1 \\ 0 & 0 & -0.4078 & 4.0326 & 0.4078 \end{bmatrix}$$

Triệt tiêu x_3 ở hàng 1, 2, 4 bằng cách:

Lấy hàng 1 – (Hàng 3)*0.0658

Lấy hàng 2 – (Hàng 3)*(-0.0596)

Lấy hàng 4 – (Hàng 3)*(-0.4078)

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0.1870 & 1 \\ 0 & 1 & 0 & 0.0689 & 0 \\ 0 & 0 & 1 & -0.3168 & -1 \\ 0 & 0 & 0 & 3.9034 & 0 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 4 bằng cách chia hàng 4 cho 3.9034

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0.1870 & 1 \\ 0 & 1 & 0 & 0.0689 & 0 \\ 0 & 0 & 1 & -0.3168 & -1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Triệt tiêu x_4 ở hàng 1, 2, 3 bằng cách:

Lấy hàng 1 – (Hàng 4)*0.1870

Lấy hàng 2 – (Hàng 4)*0.0689

Lấy hàng 3 – (Hàng 4)*(-0.3168)

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Kết luận nghiệm của $Ax = b$ là $x = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$

5.

$$A = \begin{bmatrix} 6.1 & 1.7 & 1.0 & 9.1 \\ 1.7 & 1.6 & 1.1 & 8.1 \\ 1.0 & 1.1 & 1.0 & 6.1 \\ 9.1 & 8.1 & 0.8 & 7.1 \end{bmatrix} \qquad b = \begin{bmatrix} 18.27 \\ 12.88 \\ 9.51 \\ 26.07 \end{bmatrix}$$

Ghép hai ma trận:

$$Ab = \begin{bmatrix} 6.1 & 1.7 & 1.0 & 9.1 & 18.27 \\ 1.7 & 1.6 & 1.1 & 8.1 & 12.88 \\ 1.0 & 1.1 & 1.0 & 6.1 & 9.51 \\ 9.1 & 8.1 & 0.8 & 7.1 & 26.07 \end{bmatrix}$$

Tuyến tính hóa hàng 1: Chia hàng đầu tiên cho hệ số đầu tiên = 6.1, ta có:

$$Ab = \begin{bmatrix} 1 & 0.2787 & 0.1639 & 1.4918 & 2.9951 \\ 1.7 & 1.6 & 1.1 & 8.1 & 12.88 \\ 1.0 & 1.1 & 1.0 & 6.1 & 9.51 \\ 9.1 & 8.1 & 0.8 & 7.1 & 26.07 \end{bmatrix}$$

Triệt tiêu hệ số x_1 ở hàng 2,3,4 bằng cách:

Lấy hàng 2 – (hàng 1)*1.7

Lấy hàng 3 – (hàng 1)*1.0

Lấy hàng 4 – (hàng 1)*9.1

$$Ab = \begin{bmatrix} 1 & 0.2787 & 0.1639 & 1.4918 & 2.9951 \\ 0 & 1.1262 & 0.8214 & 5.5639 & 7.7883 \\ 0 & 0.8213 & 0.8361 & 4.6082 & 6.5149 \\ 0 & 5.5638 & -0.6915 & -6.4754 & -1.1854 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 2 bằng cách chia hàng 2 cho 1.1262

$$Ab = \begin{bmatrix} 1 & 0.2787 & 0.1639 & 1.4918 & 2.9951 \\ 0 & 1 & 0.7294 & 4.9404 & 6.9156 \\ 0 & 0.8213 & 0.8361 & 4.6082 & 6.5149 \\ 0 & 5.5638 & -0.6915 & -6.4754 & -1.1854 \end{bmatrix}$$

Triệt tiêu x_2 ở hàng 1,3,4 bằng cách:

Lấy hàng 1 – (Hàng 2)*0.2787

Lấy hàng 3 – (Hàng 2)*0.8213

Lấy hàng 4 – (Hàng 2)*5.5638

$$Ab = \begin{bmatrix} 1 & 0 & -0.0394 & 0.1149 & 1.0677 \\ 0 & 1 & 0.7294 & 4.9404 & 6.9156 \\ 0 & 0 & 0.2370 & 0.5506 & 0.8351 \\ 0 & 0 & -4.7497 & -33.9628 & -39.6624 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 3 bằng cách chia hàng 3 cho 0.2370

$$Ab = \begin{bmatrix} 1 & 0 & -0.0394 & 0.1149 & 1.0677 \\ 0 & 1 & 0.7294 & 4.9404 & 6.9156 \\ 0 & 0 & 1 & 2.3232 & 3.5236 \\ 0 & 0 & -4.7497 & -33.9628 & -39.6624 \end{bmatrix}$$

Triệt tiêu x_3 ở hàng 1, 2, 4 bằng cách:

Lấy hàng 1 – (Hàng 3)*(-0.0394)

Lấy hàng 2 – (Hàng 3)*0.7294

Lấy hàng 4 – (Hàng 3)*(-4.7497)

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0.2064 & 1.2065 \\ 0 & 1 & 0 & 3.2459 & 4.3455 \\ 0 & 0 & 1 & 2.3232 & 3.5236 \\ 0 & 0 & 0 & -22.9283 & -22.9264 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 4 bằng cách chia hàng 4 cho -22.9283

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0.2064 & 1.2065 \\ 0 & 1 & 0 & 3.2459 & 4.3455 \\ 0 & 0 & 1 & 2.3232 & 3.5236 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Triệt tiêu x_4 ở hàng 1, 2, 3 bằng cách:

Lấy hàng 1 – (Hàng 4)*0.2064

Lấy hàng 2 – (Hàng 4)*3.2459

Lấy hàng 3 – (Hàng 4)*2.3232

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1.1 \\ 0 & 0 & 1 & 0 & 1.2 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Kết luận nghiệm của $Ax = b$ là $x = \begin{bmatrix} 1 \\ 1.1 \\ 1.2 \\ 1 \end{bmatrix}$

6.

$$A = \begin{bmatrix} 3.2 & 7.7 & 0.6 & 9.1 \\ 1.7 & 1.6 & 1.1 & 8.1 \\ 1.0 & 1.1 & 1.0 & 6.1 \\ 9.1 & 8.1 & 0.8 & 7.1 \end{bmatrix} \qquad b = \begin{bmatrix} 8.26 \\ 1.89 \\ 1.15 \\ 2.40 \end{bmatrix}$$

Ghép hai ma trận:

$$Ab = \begin{bmatrix} 3.2 & 7.7 & 0.6 & 9.1 & 8.26 \\ 1.7 & 1.6 & 1.1 & 8.1 & 1.89 \\ 1.0 & 1.1 & 1.0 & 6.1 & 1.15 \\ 9.1 & 8.1 & 0.8 & 7.1 & 2.40 \end{bmatrix}$$

Tuyến tính hóa hàng 1: Chia hàng đầu tiên cho hệ số đầu tiên = 3.2, ta có:

$$Ab = \begin{bmatrix} 1 & 2.40625 & 0.1875 & 2.84375 & 2.58125 \\ 1.7 & 1.6 & 1.1 & 8.1 & 1.89 \\ 1.0 & 1.1 & 1.0 & 6.1 & 1.15 \\ 9.1 & 8.1 & 0.8 & 7.1 & 2.40 \end{bmatrix}$$

Triệt tiêu hệ số x_1 ở hàng 2,3,4 bằng cách:

Lấy hàng 2 – (hàng 1)*1.7

Lấy hàng 3 – (hàng 1)*1.0

Lấy hàng 4 – (hàng 1)*9.1

$$Ab = \begin{bmatrix} 1 & 2.40625 & 0.1875 & 2.84375 & 2.58125 \\ 0 & -2.490625 & 0.78125 & 3.265625 & -2.498125 \\ 0 & -1.30625 & 0.8125 & 3.25625 & -1.43125 \\ 0 & -13.796875 & -0.90625 & -18.778125 & -21.089375 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 2 bằng cách chia hàng 2 cho -2.490625

$$Ab = \begin{bmatrix} 1 & 2.40625 & 0.1875 & 2.84375 & 2.58125 \\ 0 & 1 & -0.31368 & -1.311167 & 1.003011 \\ 0 & -1.30625 & 0.8125 & 3.25625 & -1.43125 \\ 0 & -13.796875 & -0.90625 & -18.778125 & -21.089375 \end{bmatrix}$$

Triệt tiêu x_2 ở hàng 1,3,4 bằng cách:

Lấy hàng 1 – (Hàng 2)*2.40625

Lấy hàng 3 – (Hàng 2)*(-1.30625)

Lấy hàng 4 – (Hàng 2)*(-13.796875)

$$Ab = \begin{bmatrix} 1 & 0 & 0.94229 & 5.99875 & 0.16775 \\ 0 & 1 & -0.31368 & -1.311167 & 1.003011 \\ 0 & 0 & 0.40276 & 1.54354 & -0.12107 \\ 0 & 0 & -5.23405 & -36.86813 & -7.25096 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 3 bằng cách chia hàng 3 cho 0.40276

$$Ab = \begin{bmatrix} 1 & 0 & 0.94229 & 5.99875 & 0.16775 \\ 0 & 1 & -0.31368 & -1.311167 & 1.003011 \\ 0 & 0 & 1 & 3.83241 & -0.30060 \\ 0 & 0 & -5.23405 & -36.86813 & -7.25096 \end{bmatrix}$$

Triệt tiêu x_3 ở hàng 1, 2, 4 bằng cách:

Lấy hàng 1 – (Hàng 3)* 0.94229

Lấy hàng 2 – (Hàng 3)* (-0.31368)

Lấy hàng 4 – (Hàng 3)*(-5.23405)

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 2.38751 & 0.45100 \\ 0 & 1 & 0 & -0.10902 & 0.90872 \\ 0 & 0 & 1 & 3.83241 & -0.30060 \\ 0 & 0 & 0 & -16.80910 & -8.82432 \end{bmatrix}$$

Tiếp theo, ta tuyến tính hóa hàng 4 bằng cách chia hàng 4 cho -16.80910

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 2.38751 & 0.45100 \\ 0 & 1 & 0 & -0.10902 & 0.90872 \\ 0 & 0 & 1 & 3.83241 & -0.30060 \\ 0 & 0 & 0 & 1 & 0.52497 \end{bmatrix}$$

Triệt tiêu x_4 ở hàng 1, 2, 3 bằng cách:

Lấy hàng 1 – (Hàng 4)*2.38751

Lấy hàng 2 – (Hàng 4)*(-0.10902)

Lấy hàng 3 – (Hàng 4)*(3.83241)

$$Ab = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.80237 \\ 0 & 1 & 0 & 0 & 0.96595 \\ 0 & 0 & 1 & 0 & -2.31250 \\ 0 & 0 & 0 & 1 & 0.52497 \end{bmatrix}$$

Kết luận nghiệm của $Ax = b$ là $x = \begin{bmatrix} -0.80237 \\ 0.96595 \\ -2.31250 \\ 0.52497 \end{bmatrix}$

Chương 10: Nội suy, hồi quy

Bài tập 5.7 trang 190: Cho giá trị của hàm số

$$y = arctan\frac{3x - x^3}{1 - 3x^2} - 3arctanx + \frac{x^2}{4}(2lnx - 3)$$

x	58	58.34	58.68	59.02	59.36	59.7
y	4303.52	4364.11	4425.17	4486.69	4548.69	4611.16

Dùng nội suy Newton tiến, tính gần đúng giá trị của y tại $x = 58.17$

Đầu tiên, ta có bảng sai phân:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
58	4303.52	…	…	…	…	…
…	…	60.59	…	…	…	…
58.34	4364.11	…	0.47	…	…	…
…	…	61.06	…	-0.01	…	…
58.68	4425.17	…	0.46	…	0.03	…
…	…	61.52	…	0.02	…	-0.06
59.02	4486.69	…	0.48	…	-0.03	…
…	…	62	…	-0.01	…	…
59.36	4548.69	…	0.47	…	…	…
…	…	62.47	…	…	…	…
59.7	4611.16	…	…	…	…	…

Theo công thức nội suy Newton tiến:

$$P_n(x) = P_n(x_o + ht) = y_o + \frac{t}{1!}\Delta y_o + \frac{t(t-1)}{2!}\Delta^2 y_o + \cdots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_o$$

$$P_5(x) = P_5(x_o + ht)$$

$$= 4303.52 + 60.59t + \frac{0.47}{2}t(t-1) - \frac{0.01}{3!}t(t-1)(t-2)$$

$$+ \frac{0.03}{4!}t(t-1)(t-2)(t-3) - \frac{0.06}{5!}t(t-1)(t-2)(t-3)(t-4)$$

Với $h = 0.34$

Với $x = 58.17 \Rightarrow 58 + 0.34 * t = 58.17 \Rightarrow t = 0.5$

$$P_5(58.17) = P_5(58 + 0.34 * 0.5)$$
$$= 4303.52 + 60.59 * 0.5 + \frac{0.47}{2}0.5(0.5-1)$$
$$- \frac{0.01}{3!}0.5(0.5-1)(0.5-2) + \frac{0.03}{4!}0.5(0.5-1)(0.5-2)(0.5-3)$$
$$- \frac{0.06}{5!}0.5(0.5-1)(0.5-2)(0.5-3)(0.5-4) \approx 4333.75$$

Chương 11 + 12: Đạo hàm, tích phân, Phương trình vi phân

Bài tập 6.3 trang 221: Tính gần đúng tích phân xác định

$$I = \int_1^2 \sqrt{x}dx$$

Bằng công thức hình thang tổng quát với $n = 10$. Đánh giá sai số.

Ta có $f(x) = \sqrt{x}$, ta tiến hành phân hoạch đoạn [1,2] thành $n = 10$ đoạn bằng nhau với $h = \frac{2-1}{10} = \frac{1}{10}$, ta được bảng sau:

x	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
$f(x) = \sqrt{x}$	1	$\frac{\sqrt{110}}{10}$	$\frac{\sqrt{30}}{5}$	$\frac{\sqrt{130}}{10}$	$\frac{\sqrt{35}}{5}$	$\frac{\sqrt{6}}{2}$	$\frac{2\sqrt{10}}{5}$	$\frac{\sqrt{170}}{10}$	$\frac{3\sqrt{5}}{5}$	$\frac{\sqrt{190}}{10}$	$\sqrt{2}$

Áp dụng công thức hình thang tổng quát:

$$\int_a^b f(x)dx \approx \frac{b-a}{2n}(y_o + 2y_1 + 2y_2 + \cdots + 2y_{n-1} + y_n)$$

Từ đó,

$$I = \int_1^2 \sqrt{x}dx = \frac{2-1}{2*10}(1 + 2*\frac{\sqrt{110}}{10} + 2*\frac{\sqrt{30}}{5} + 2*\frac{\sqrt{130}}{10} + 2*\frac{\sqrt{35}}{5} + 2*\frac{\sqrt{6}}{2} + 2$$
$$* \frac{2\sqrt{10}}{5} + 2*\frac{\sqrt{170}}{10} + 2*\frac{3\sqrt{5}}{5} + 2*\frac{\sqrt{190}}{10} + \sqrt{2}) \approx 1.21883$$

Sai số toàn phần

$$f'(x) = \frac{1}{2\sqrt{x}}$$

$$f''(x) = -\frac{1}{4}.\frac{1}{\sqrt{x^3}}$$

$$\Rightarrow M = \max_{1\le x\le 2}|f''(x)| = \max_{1\le x\le 2}\left|-\frac{1}{4}.\frac{1}{\sqrt{x^3}}\right| = 0.25$$

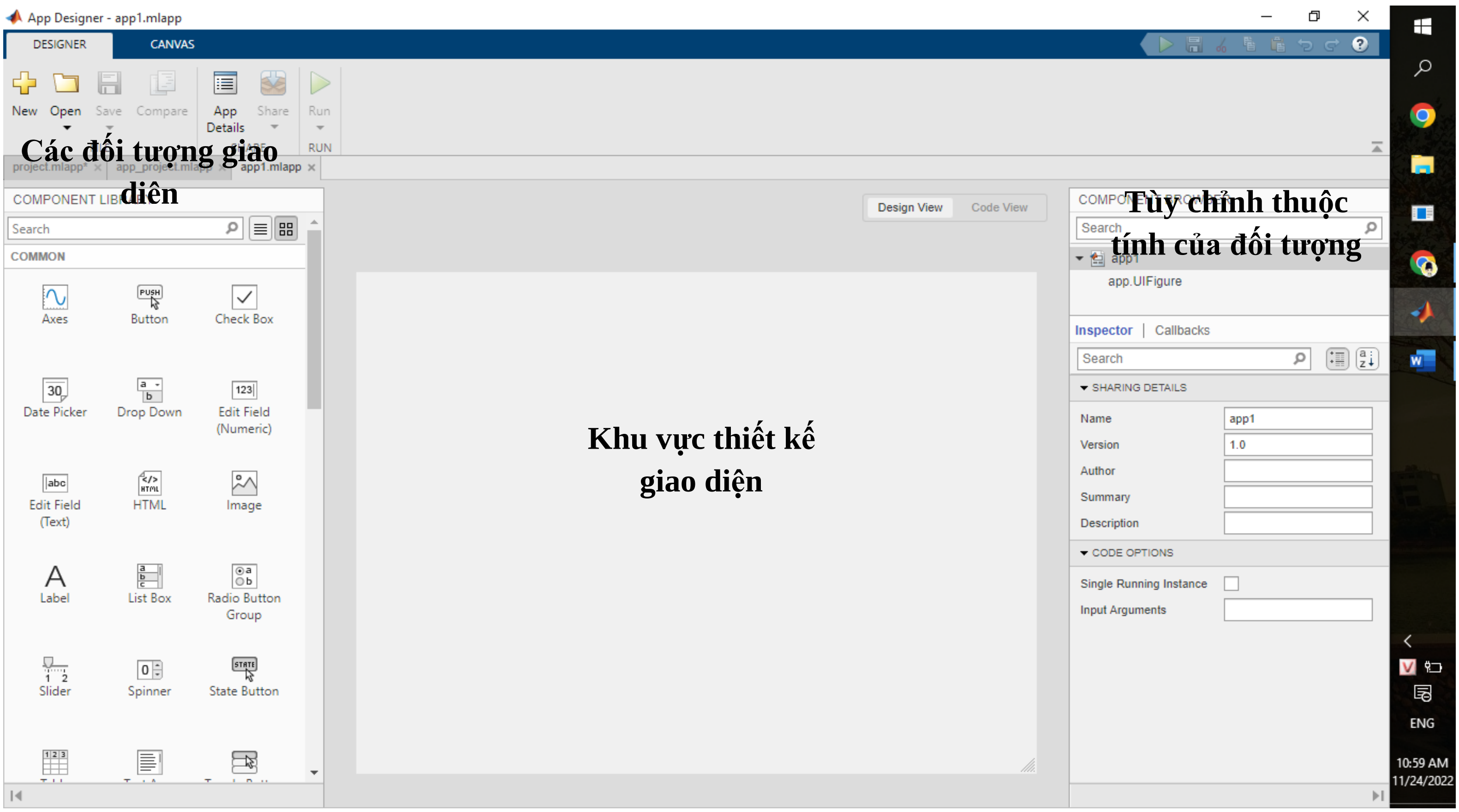
$$R = \frac{M(b-a)}{12}h^2 = \frac{0.25*(2-1)}{12}.(0.1)^2 = \frac{1}{4800} \approx 2.0833*10^{-4}$$

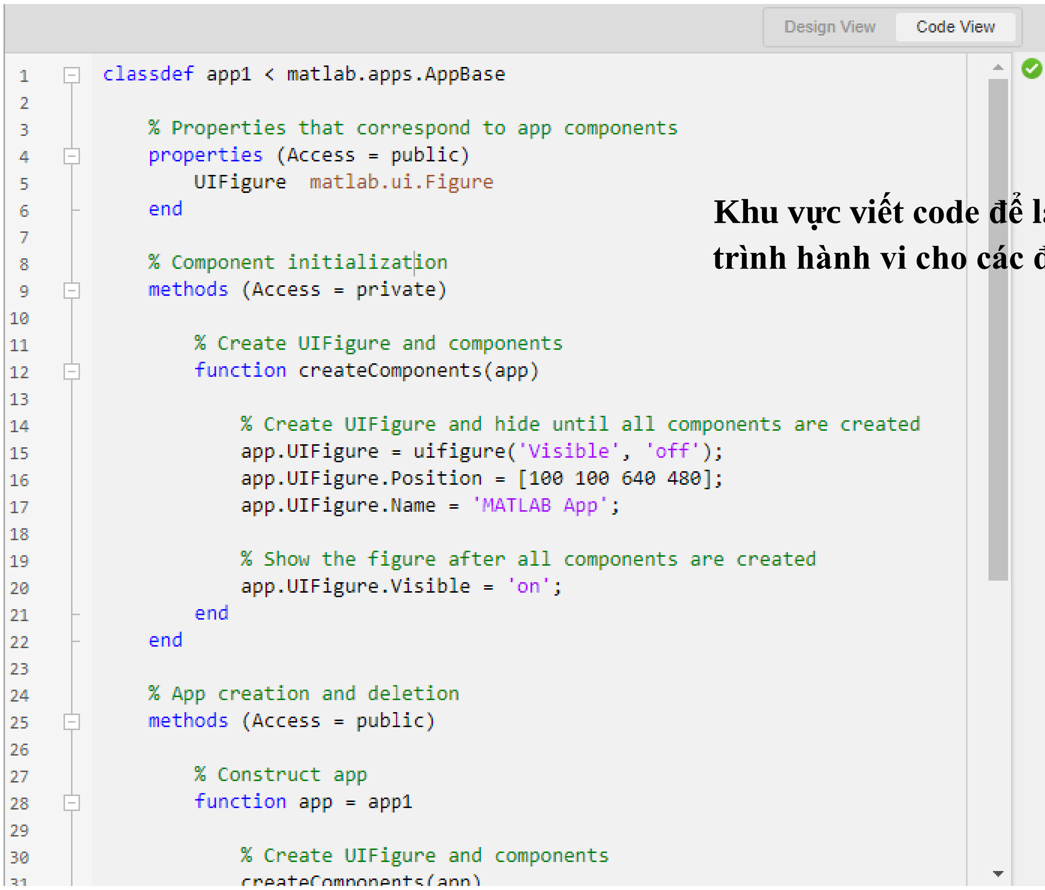
III. PHẦN THIẾT KẾ GIAO DIỆN:

1. Giới thiệu về App Designer trong Matlab:

App Designer cho phép tạo ra các ứng dụng chuyên nghiệp mà không cần phải là nhà phát triển phần mềm chuyên nghiệp. Kéo và thả các thành phần trực quan để bố trí thiết kế giao diện người dùng đồ họa (GUI) của bạn và sử dụng trình chỉnh sửa tích hợp để nhanh chóng lập trình hành vi cho nó.

Giao diện chính:





Khu vực viết code để lập trình hành vi cho các đối

Một số cú pháp nhập xuất dữ liệu, vẽ đồ thị:

```
Variable = app.ComponentName.Properties %Cú pháp lấy dữ liệu.
app.ComponentName.Properties = %Variable Cú pháp xuất dữ liệu ra
plot(app.UIAxes,x,y) %Vẽ đồ thị.
```

2. Thiết kế giao diện:

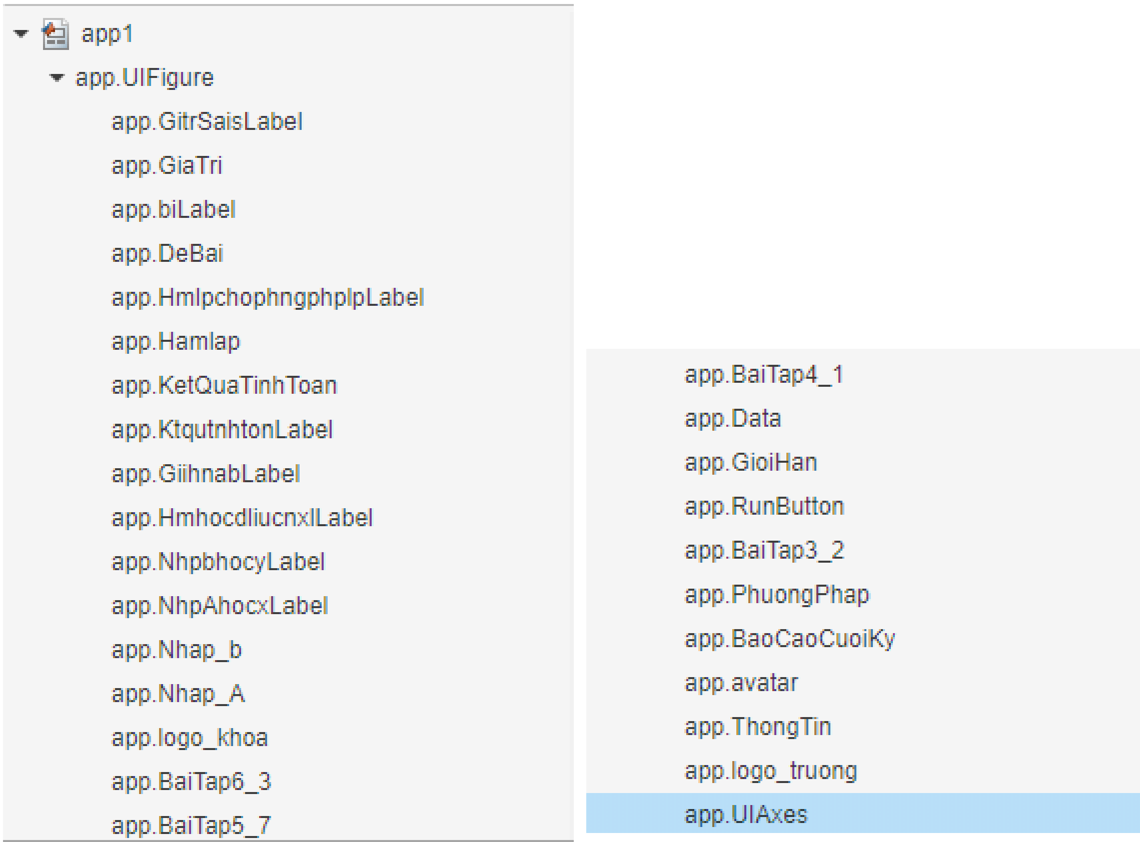
<u>**Yêu cầu:**</u> Thiết kế giao diện để tính toán và hiện thị kết quả ra app các bài tập đã làm ở phần II.

2.1. *Giao diện chính của giao diện:*

Giao diện bao gồm một số thành phần chính sau:

- Bên trái giao diện là các ô Edit Field Text để nhập dữ liệu đầu vào, và chọn phương pháp để tính toán tương ứng.
- Ở giữa là các nút ấn để import dữ liệu các bài tập đã làm ở trên mục II.
- Bên trái là ô Text Area để hiện thị đề bài và một vùng để vẽ đồ thị.
- Bên dưới cùng là nút Run để chạy app, tính toán và hiển thị kết quả.
- Ngoài ra trên giao diện còn có Logo Trường và Khoa, Thông tin và ảnh người làm.

Một số Component Browser (tham chiếu tới những đối tượng trong giao diện để viết code hành vi cho nó).



2.2. Phần code thuật toán các phương pháp đã dùng để giải các bài tập ở phần II:

2.2.1 Giải phương trình:

Phương pháp lặp đơn: lap.m

```
function [nghiem solanlap]=lap(fp,a,b,saiso)
    syms x
    fd1 = str2func(['@(x)' char(diff((fp(x))))]);
    if (abs(fd1(a))>=1)||(abs(fd1(b))>=1)
        disp('Không hội tụ')
        return
    end
    x0 = a;
    x1 = fp(x0);
```

```
    solanlap=1;
    while abs(x1-x0)>= saiso
        x0 = x1;
        x1 = fp(x0);
        solanlap=solanlap+1;
    end
    nghiem = x1;
end
```

Phương pháp tiếp tuyến (Newton – Raphson): tieptuyen.m

```
function [nghiem solanlap]=tieptuyen(f,a,b,saiso)
    syms x;
    fd1 = str2func(['@(x)' char(diff(f(x)))]);
    fd2 = str2func(['@(x)' char(diff(f(x),2))]);
    nostop = 1;
    temp1 = double(solve(diff(f(x))));
    temp2 = double(solve(diff(fd1(x))));
    %Kiem tra f'(x) co doi dau trong khoang phan li
nghiem hay khong
    if ~isempty(temp1)
        for i=1:length(temp1)
            if (temp1(i)<=b) && (temp1(i)>=a)
                disp('Ham f'' doi dau trong khoang
phan li nghiem');
                nostop=0;
                nghiem = 'khong xac dinh';
                solanlap = 'khong xac dinh';
                break;
            end
        end
     end
     %Kiem tra f"(x) co doi dau trong khoang phan li
nghiem hay ko
     if ~isempty(temp2)
        for i=1:length(temp2)
            if (temp2(i)<=b) && (temp2(i)>=a)
```

```
                 disp('Ham f" doi dau trong khoang phan
li nghiem');
                 nostop=0;
                 nghiem = 'khong xac dinh';
                 solanlap = 'khong xac dinh';
                 break;
             end
         end
     end
     if (nostop) %thuat toan phuong phap tiep tuyen.
        x0 = a;
        while f(x0)*fd2(x0)<=0
            if f(x0)==0
                nghiem = x0;
                solanlap = 0;
                return
            else
                x0 = b;
            end
        end
        nghiem=x0-(f(x0)/fd1(x0));
        solanlap=0;
        while abs(nghiem-x0)>=saiso
            x0=nghiem;
            nghiem = x0-(f(x0)/fd1(x0));
            solanlap=solanlap+1;
        end
        nghiem = double(nghiem);
    end
end
```

Phương pháp dây cung: secant.m

```
function [X,n]=secant(fun,xO,x1,err)
if nargin< 4
    err=le-5;
end
n=0;
```

```
while abs(x1-x0)>=err
    x2 = x0;
    x0 = x1;
    x1 = x1+(x1-x2)/(feval (fun,x2)/feval(fun,x1)-1);
    n=n+1;
end
    X=x1;
end
```

2.2.2 Giải hệ phương trình:

Phương pháp Gauss – Jordan: gauss_jordan.m

```
function nghiem =  gauss_jordan (A,B)
    AB = [A B'];
    for k = 1:length(B)
        AB(k,:) = AB(k,:)./AB(k,k);
        for m = 1:length(B)
            if m~=k
                AB(m,:) = AB(m,:) - AB(k,:).*AB(m,k);
            end
        end
    end
    c = size(AB);
    nghiem = round(AB(:,c(2)),4);
end
```

2.2.3 Nội suy, Hồi quy:

Nội suy Newton: noisuy_newton.m

```
function [dathuc Yn] =
noisuy_newton(x_array,y_array,No)
   syms t  d
   mid = mean(x_array);
   h = abs(x_array(2) - x_array(1));
   d = 1;
   n = 0;
   if No < mid     %Newton tiến
       P(t) = y_array(1) + 0*t;
       for m = 1:length(x_array)-1
            for k = 1:length(y_array)-1
```

```
                y_array(k) = y_array(k+1) - y_array(k);
            end
            d = (t - m + 1)*d;
            P = P + (d/factorial(m))*y_array(1);
            y_array(length(y_array)  - n) = 0;
            n = n + 1;
        end
        a = (No - x_array(1))/h;
        dathuc = simplify(P);
        dathuc = str2func(['@(t)' char(dathuc)]);
        Yn = dathuc(a);
    else   %Newton lùi
        P(t) = y_array(length(y_array)) + 0*t;
        for m = 1:length(x_array)-1
            for k = 1:length(y_array)-1
                y_array(k) = y_array(k+1) - y_array(k);
            end
            d = (t + m - 1)*d;
            y_array(length(y_array) - m);
            P = P +
(d/factorial(m))*y_array(length(y_array) - m);
            y_array(length(y_array)  - n) = 0;
            n = n + 1;
        end
        a = (No - x_array(length(x_array)))/h;
        dathuc = simplify(P);
        dathuc = str2func(['@(t)' char(dathuc)]);
        Yn = dathuc(a);
    end
end
```

2.2.4 Đạo hàm, tích phân, Phương trình vi phân:

Tính tích phân theo phương pháp hình thang: tichphanhinhthang.m

```
function y = tichphanhinhthang(fx,a,b,N)
    h = (b - a)/N;
    k = 1:N-1;
    y = (h/2).*(fx(a)+fx(b)+2.*sum(fx(a+k.*h)));
end
```

2.3. Phần code giao diện:

```
    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: RunButton
        function RunButtonPushed(app, event)
        try
            app.KetQuaTinhToan.Value = '';
            switch app.PhuongPhap.Value
                case 'Lặp đơn'
                    fx = app.Data.Value;
                    fx = str2func(['@(x)' fx]);
                    fLap = app.Hamlap.Value;
                    fLap = str2func(['@(x)' fLap]);
                    Seg = app.GioiHan.Value;
                    if (Seg == "")
                        msgbox({'Không để trống!',' Vui
lòng nhập khoảng giới hạn'},'Error','error');
                    else
                    Seg =str2num(Seg);
                    saiso = app.GiaTri.Value;
                    if (saiso == "")
                        msgbox({'Không để trống!',' Vui
lòng nhập giá trị sai số'},'Error','error');
```

```
                    else
                    saiso = str2num(saiso);
                    [nghiem solanlap] =
lap(fLap,Seg(1),Seg(2),saiso);
                    x = 0:0.01:10;
                    y = fx(x);
                    plot(app.UIAxes,x,y);
                    text(app.UIAxes,nghiem,0,'x');
                    app.KetQuaTinhToan.Value = ['Nghiệm
gần đúng của phương trình là:' num2str(nghiem) ' và Số
lần lặp là: ' num2str(solanlap)];
                    end
                    end
                    case 'Tiếp tuyến'
                    fx = app.Data.Value;
                    fx = str2func(['@(x)' fx]);
                    Seg = app.GioiHan.Value;
                    if (Seg == "")
                        msgbox({'Không để trống!',' Vui
lòng nhập khoảng giới hạn'},'Error','error');
                    else
                    Seg =str2num(Seg);
                    saiso = app.GiaTri.Value;
                    if (saiso == "")
```

```
                    msgbox({'Không để trống!',' Vui
lòng nhập giá trị sai số'},'Error','error');
                else
                saiso = str2num(saiso);
                [nghiem solanlap] =
tieptuyen(fx,Seg(1),Seg(2),saiso);
                x = 0:0.01:10;
                y = fx(x);
                plot(app.UIAxes,x,y);
                text(app.UIAxes,nghiem,0,'x');
                app.KetQuaTinhToan.Value = ['Nghiệm
gần đúng của phương trình là:' num2str(nghiem) ' và Số
lần lặp là: ' num2str(solanlap)];
                end
                end
            case 'Dây cung'
                fx = app.Data.Value;
                fx = str2func(['@(x)' fx]);
                Seg = app.GioiHan.Value;
                if (Seg == "")
                    msgbox({'Không để trống!',' Vui
lòng nhập khoảng giới hạn'},'Error','error');
                else
                Seg =str2num(Seg);
                saiso = app.GiaTri.Value;
```

```
                    if (saiso == "")
                        msgbox({'Không để trống!',' Vui lòng nhập giá trị sai số'},'Error','error');
                    else
                    saiso = str2num(saiso);
                    [nghiem solanlap] = secant(fx,Seg(1),Seg(2),saiso);
                    x = 0:0.01:10;
                    y = fx(x);
                    plot(app.UIAxes,x,y);
                    text(app.UIAxes,nghiem,0,'x');
                    app.KetQuaTinhToan.Value = ['Nghiệm gần đúng của phương trình là:' num2str(nghiem) ' và Số lần lặp là: ' num2str(solanlap)];
                    end
                    end
                 case 'Gauss-Jordan'
                    A = app.Nhap_A.Value;
                    b = app.Nhap_b.Value;
                    if (A == "") || (b == "")
                        msgbox({'Không để trống!','Vùi lòng nhập dữ liệu để giải hệ phương trình.'},'Error','error');
                    else
                    A = str2num(A);
```

```
                    b = str2num(b);
                    nghiem = (gauss_jordan(A,b))';
                    nghiem = num2str(nghiem);
                    nghiem = "{ " + nghiem + " }";
                    app.KetQuaTinhToan.Value = ['Nghiệm của hệ phương trình là: ' num2str(nghiem)];
                    end
                    case 'Newton'
                    x = app.Nhap_A.Value;
                    y = app.Nhap_b.Value;
                    if (x == "") || (y == "")
                        msgbox({'Không để trống!','Vùi lòng nhập dữ liệu để tìm đa thức nội suy.'},'Error','error');
                    else
                    x = str2num(x);
                    y = str2num(y);
                    No = app.Data.Value;
                    if (No == "")
                        msgbox("Vùi lòng nhập giá trị nội suy",'Error','error');
                    else
                    No = str2num(No);
                    [dathuc Yn] = noisuy_newton(x,y,No);
```

```
                    j = 1;
                    a = 0:0.01:(x(length(x)) -
x(1))/(x(2) - x(1));
                    for i = 0:0.01:(x(length(x)) -
x(1))/(x(2) - x(1))
                        b(j) = dathuc(i);
                        j = j + 1;
                    end
                    plot(app.UIAxes,a,b);
                    dathuc = func2str(dathuc);
                    Yn = num2str(Yn);
                    if contains(dathuc,'@(t)')
                        dathuc =
strrep(dathuc,'@(t)','');
                    end
                    app.KetQuaTinhToan.Value = ['Đa
thức nội suy Newton là: ' dathuc ' và Giá trị nội suy
là ' Yn];
                    end
                    end
                case 'Hình thang'
                    fx = app.Data.Value;
                    fx = str2func(['@(x)' fx]);
                    Seg = app.GioiHan.Value;
                    if (Seg == "")
```

```
                        msgbox({'Không để trống!',' Vui
lòng nhập khoảng giới hạn'},'Error','error');
                    else
                    Seg = str2num(Seg);
                    N = app.GiaTri.Value;
                    if (N == "")
                        msgbox({'Không để trống!',' Vui
lòng nhập giá trị số đoạn con.'},'Error','error');
                    else
                    N = str2num(N);
                    if (N < 1)
                        msgbox({'Dùng sai phương
pháp','Vui lòng chọn phương pháp phù hợp để
giải!'},'Error','error');
                    else
                    I =
tichphanhinhthang(fx,Seg(1),Seg(2),N);
                    I = num2str(I);
                    x = 0:0.01:10;
                    y = fx(x);
                    plot(app.UIAxes,x,y);
                    line(app.UIAxes,[Seg(1) Seg(1)], [0
fx(Seg(1))],'Color','black','linewidth',2);
                    line(app.UIAxes,[Seg(2) Seg(2)], [0
fx(Seg(2))],'Color','black','linewidth',2);
```

```
                    app.KetQuaTinhToan.Value = ['Kết quả tích phân là: ' I];
                    end
                    end
                    end
                otherwise
            end
        catch error
            switch (error.identifier)
                case {'MATLAB:badsubscript','MATLAB:m_illegal_character','MATLAB:dimagree','MATLAB:catenate:dimensionMismatch'}
                    msgbox({'Dùng sai phương pháp','Vui lòng chọn phương pháp phù hợp để giải!'},'Error','error');
                case 'MATLAB:m_incomplete_statement'
                    msgbox({'Không để trống','Vùi lòng nhập hàm/hàm lặp hoặc dữ liệu cần xử lý'},'Error','error');
                otherwise
                    msgbox({error.identifier,error.message},'Error','error');
            end

        end
```

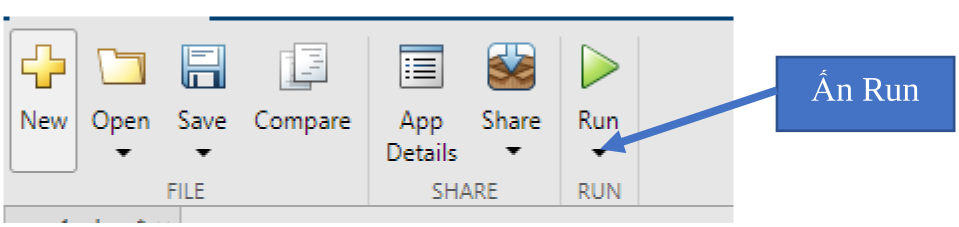
```
        end


        % Button pushed function: BaiTap3_2
        function BaiTap3_2ButtonPushed(app, event)
            app.Data.Value = 'x.^3 - 12*x - 5';
            app.Hamlap.Value = '(5+12.*x).^(1/3)';
            app.GioiHan.Value = '[3 4]';
            app.Nhap_A.Value = 'Không nhập';
            app.Nhap_b.Value = 'Không nhập';
            app.GiaTri.Value = '0.01';
            app.KetQuaTinhToan.Value = '';
            app.DeBai.Value = 'Bằng phương pháp lăp
đơn, phương pháp Newton, và phương pháp dây cung tìm
nghiệm gần đúng của phương trình x.^3 -12.*x - 5 = 0
sao cho đạt sai số tuyệt đối < 0.01';
            plot(app.UIAxes,0,0);


        end


        % Button pushed function: BaiTap4_1
        function BaiTap4_1ButtonPushed(app, event)
            app.Nhap_A.Value = '[1.7 0.2 0.1 0.3;0.2
1.9 -0.1 0.2;0.1 -0.1 1.3 -0.4;0.3 0.2 -0.4 4.1]';
            app.Nhap_b.Value = '[1.6 0.3 -1.2 0.7]';
```

```
            app.Data.Value = 'Không nhập';

            app.Hamlap.Value = 'Không nhập';

            app.GioiHan.Value = 'Không nhập';

            app.GiaTri.Value = 'Không nhập';

            app.KetQuaTinhToan.Value = '';

            app.DeBai.Value = 'Giải hê phương trình Ax
= b bằng phương pháp Gauss-Jordan vơi A, b đã được nhập
sẳn.'

            plot(app.UIAxes,0,0);

        end


        % Button pushed function: BaiTap5_7

        function BaiTap5_7ButtonPushed(app, event)

            app.Nhap_A.Value = '[58    58.34    58.68
    59.02    59.36    59.7]';

            app.Nhap_b.Value = '[4303.52   4364.11
    4425.17  4486.69  4548.69  4611.16]';

            app.Data.Value = '58.17';

            app.Hamlap.Value = 'Không nhập';

            app.GioiHan.Value = 'Không nhập';

            app.GiaTri.Value = 'Không nhập';

            app.KetQuaTinhToan.Value = '';

            app.DeBai.Value = 'Xây dưng công thức nội
suy Newton với dãy x, y đã được nhập sẳn, tại x =
58.17';
```

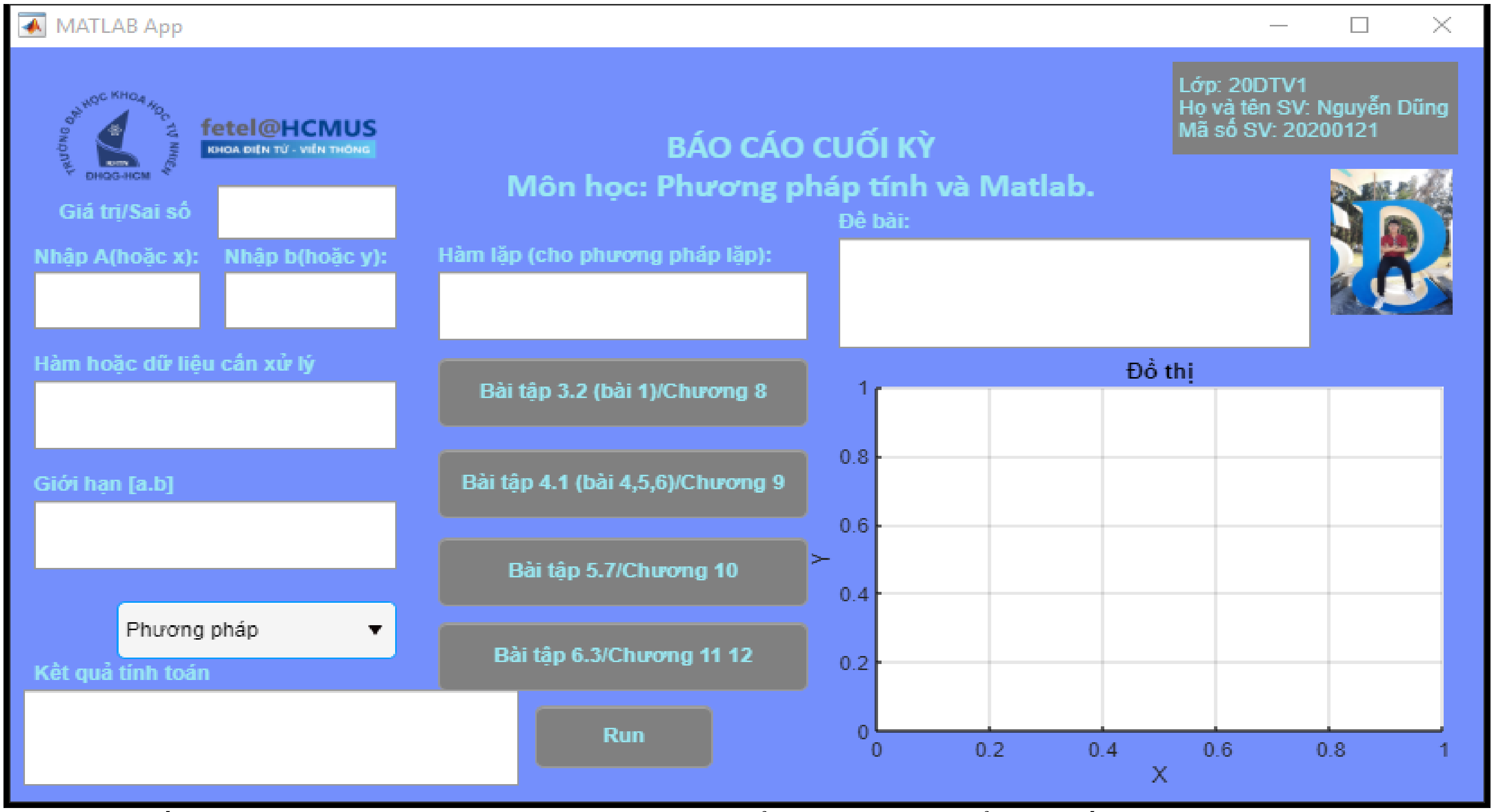
```
            plot(app.UIAxes,0,0);
        end

        % Button pushed function: BaiTap6_3
        function BaiTap6_3ButtonPushed(app, event)
            app.Data.Value = 'sqrt(x)';
            app.GioiHan.Value = '[1,2]';
            app.GiaTri.Value = '10';
            app.Hamlap.Value = 'Không nhập';
            app.Nhap_A.Value = 'Không nhập';
            app.Nhap_b.Value = 'Không nhập';
            app.KetQuaTinhToan.Value = '';
            app.DeBai.Value = 'Tính gần đúng tích phân xác định: fx = sqrt(x), a = 1, b = 2, N =10';
            plot(app.UIAxes,0,0);
        end
    end
```

3. Kết quả:

Tiền hành chạy app: Ấn nút Run (trong tab DESIGNER).



sau đó, sẽ xuất hiện giao diện chính như sau:

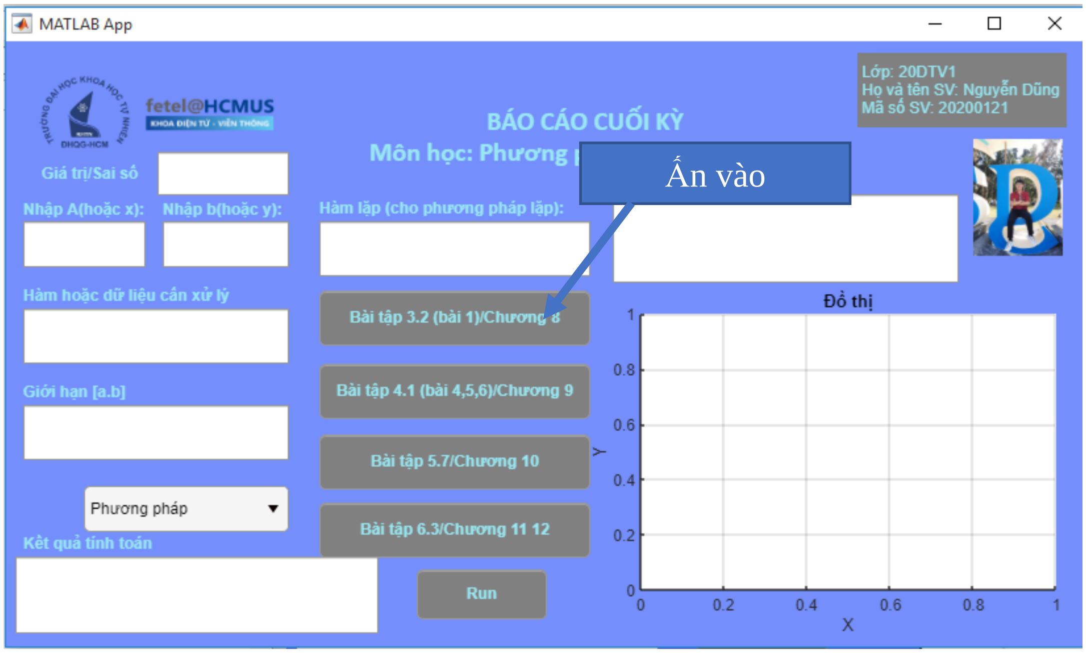


Tiến hành thực hiện các bước như bên dưới để tính toán và hiển thị kết quả trên App

3.1. Giải phương trình:

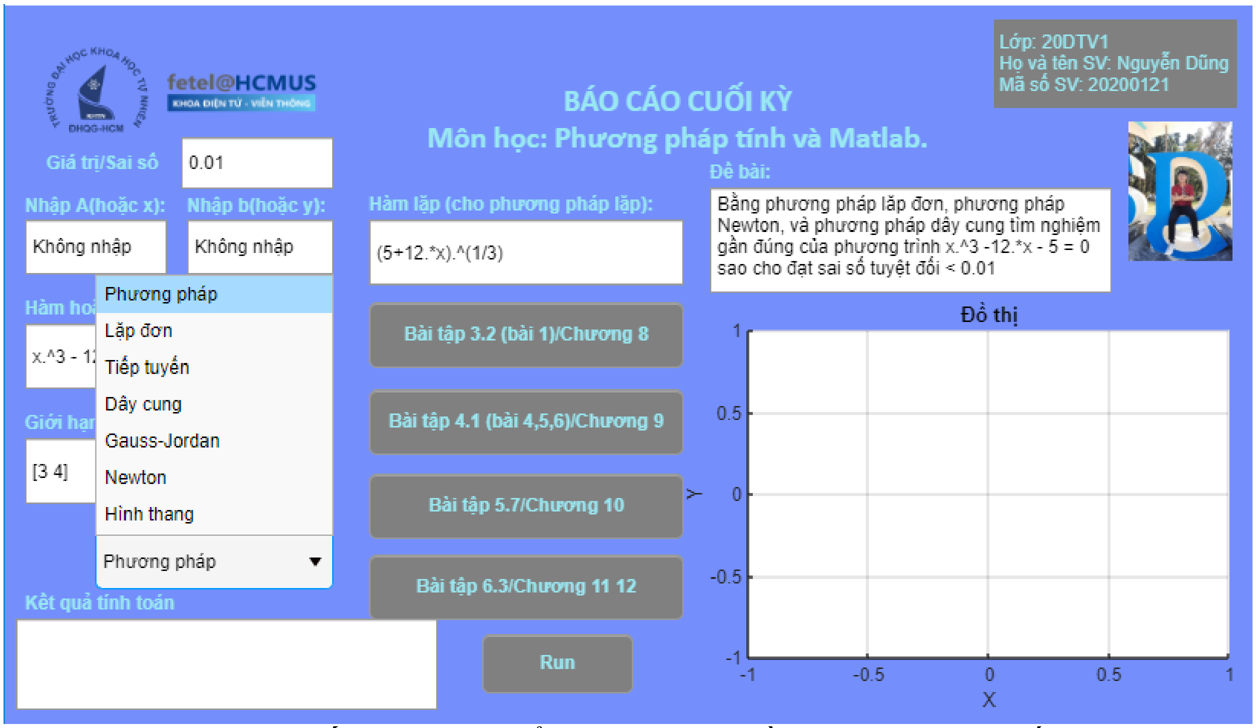
- Bước 1: Nhấn vào nút **"Bài tập 3.2 (bài 1)/Chương 8"** để load dữ liệu bài tập lên. Các dữ liệu cần cho tính toán như:
  - Giá trị/Sai số:
  - Hàm lặp (cho phương pháp lặp).
  - Hàm hoặc dữ liệu cần xử lý (Hàm cần giải nghiệm).
  - Khoảng giới hạn $[a, b]$

  Ngoài ra còn hiện đề bài ở khung "Đề bài" để tiện quan sát.

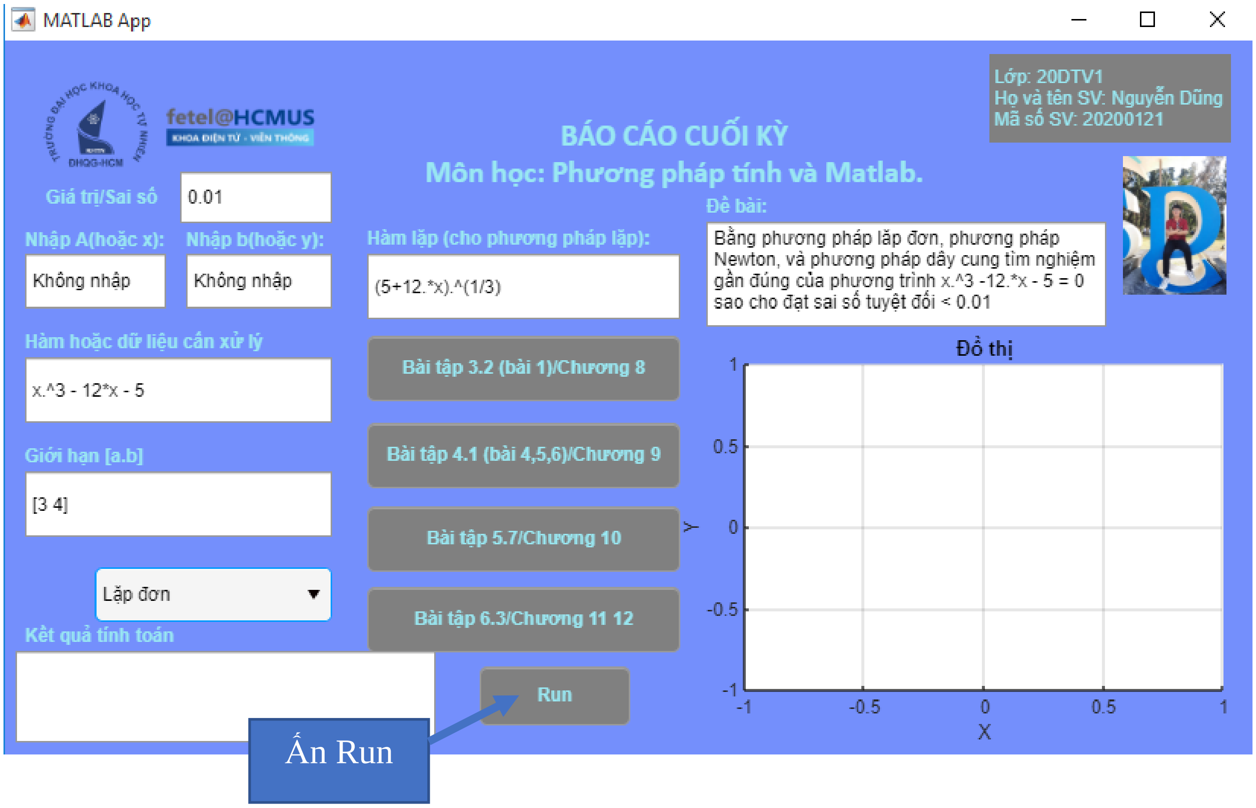


Ta thu được hình như sau:

➢ Bước 2: **Chọn phương pháp phù hợp**: trong phần này chỉ xét 3 phương pháp:
- Phương pháp lặp đơn.
- Phương pháp tiếp tuyến (Newton – Raphson).
- Phương pháp dây cung.



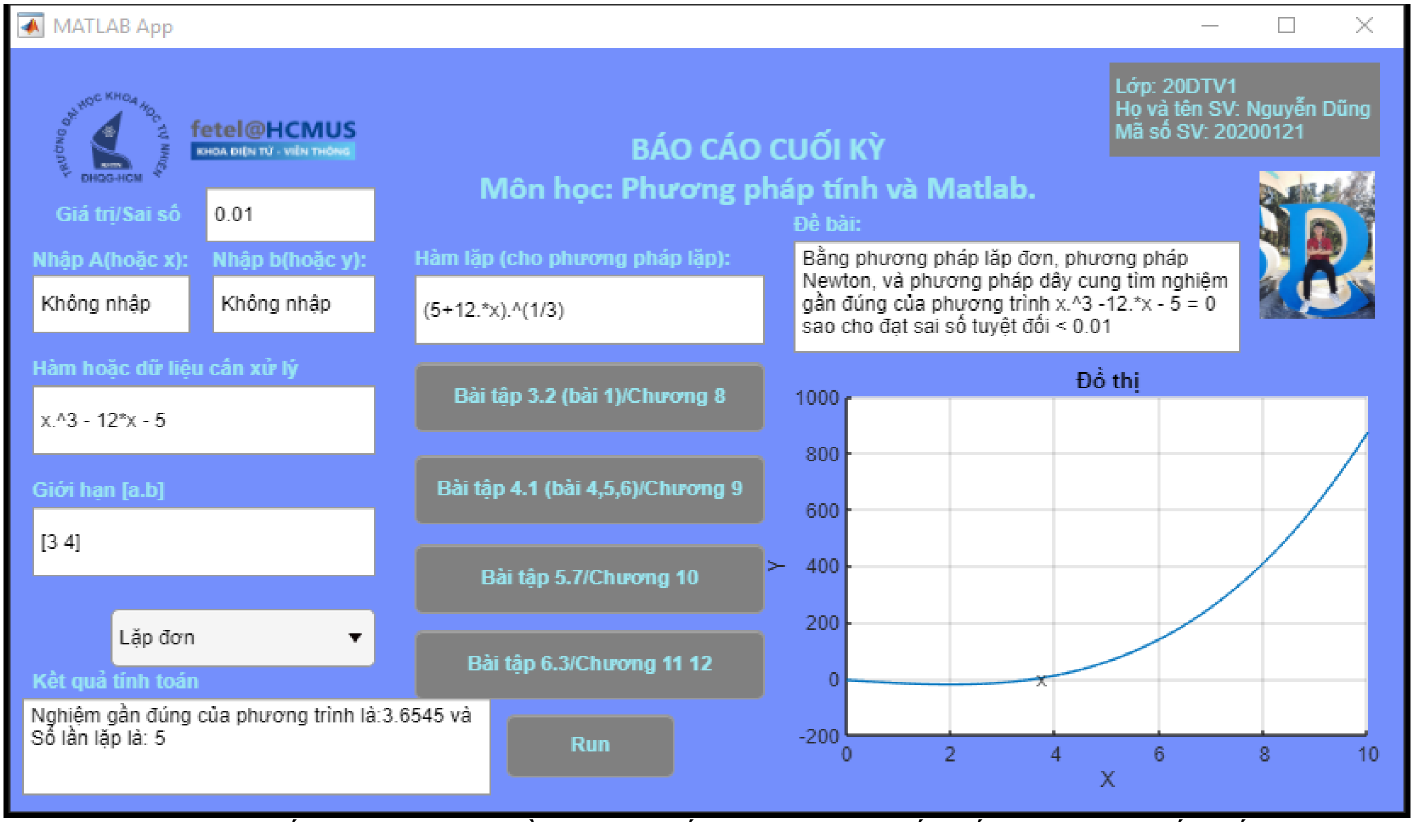
➢ Bước 3: **Ấn nút Run** để tính toán, vẽ đồ thị, và hiện thị kết quả trên giao diện.



Chú ý: Khi load dữ liệu có những Field hiển thị "không nhập", tức có nghĩa trường đó không ảnh hưởng đến kết quả tính toán.

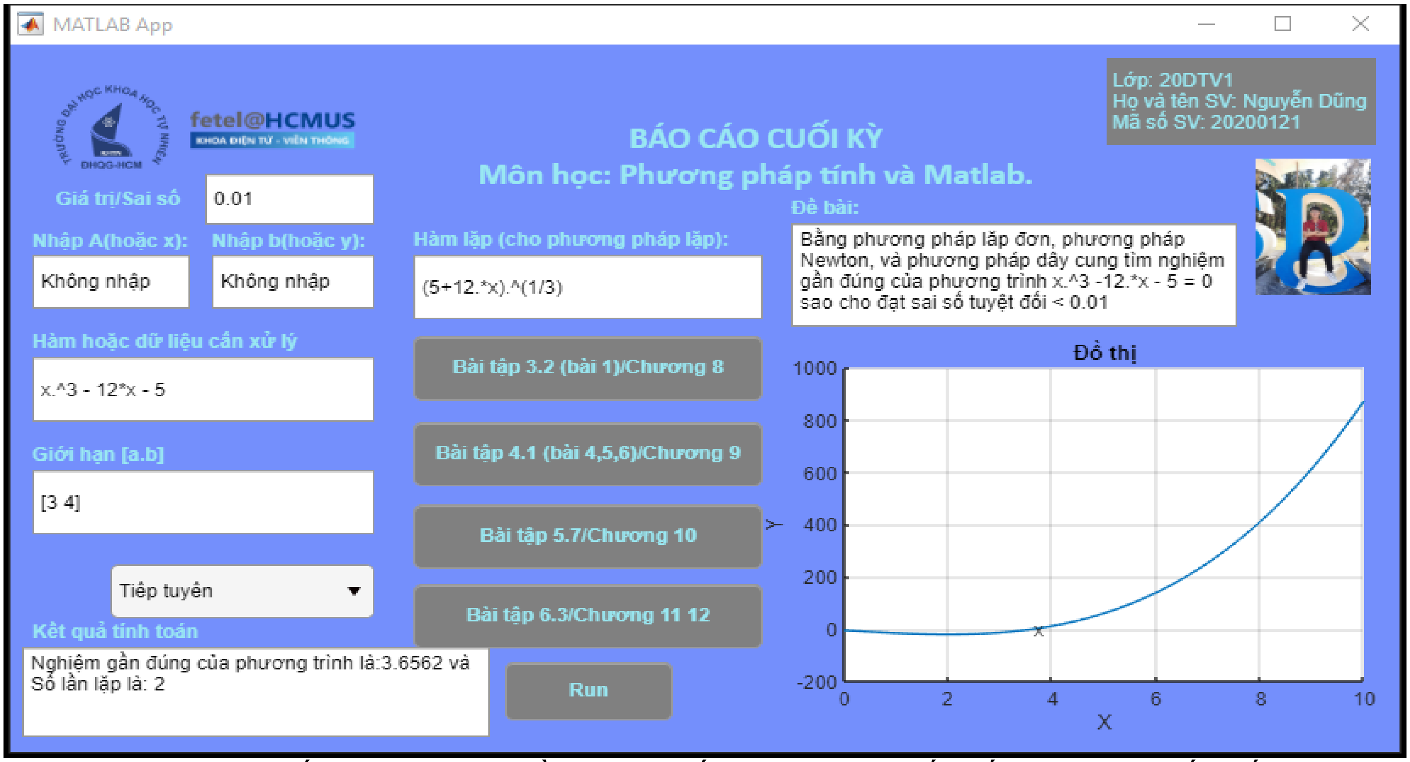
Kết quả các hình tương ứng với từng phương pháp.

3.1.1 Phương pháp lặp đơn:



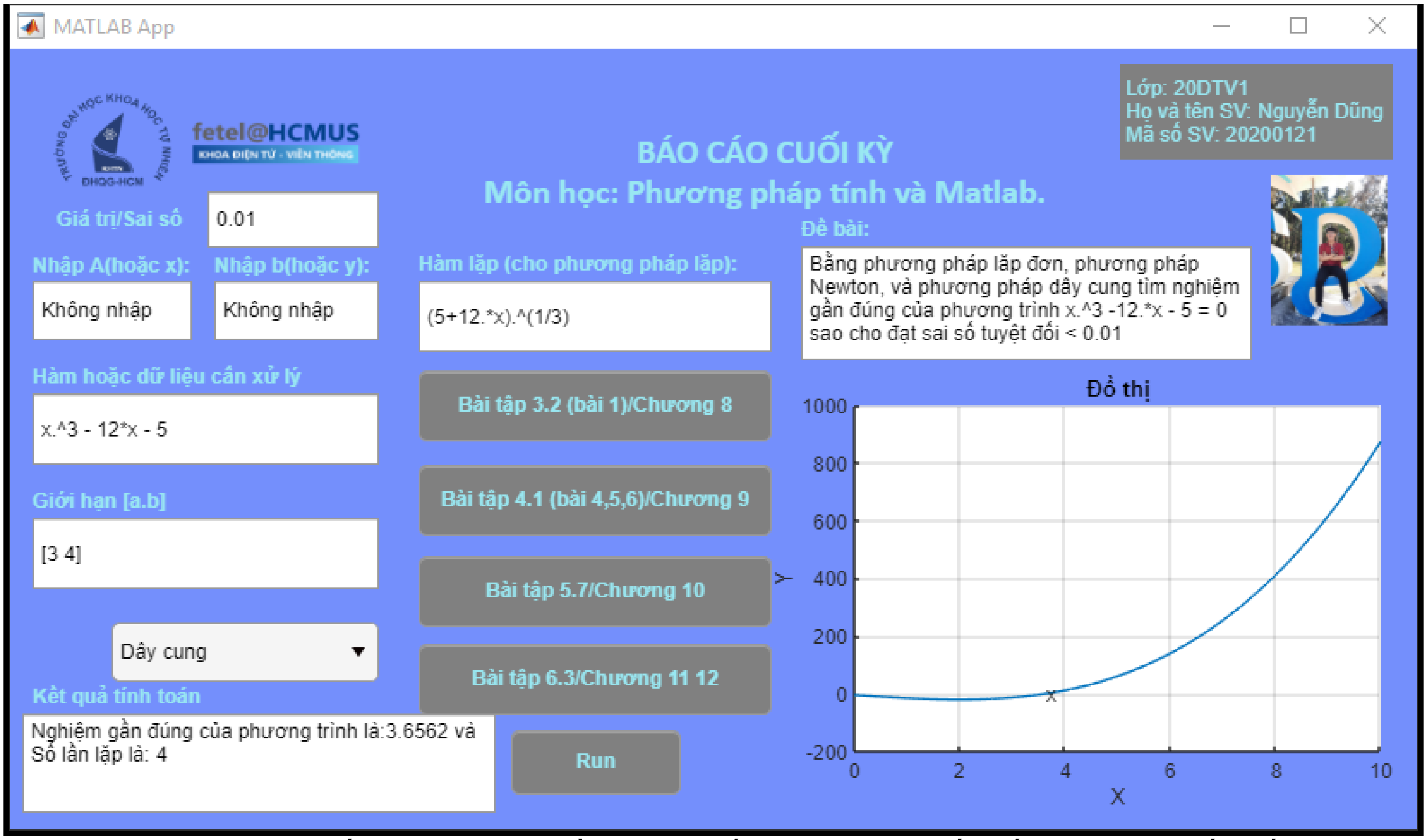
So sánh với kết quả tính toán bằng lý thuyết ở mục II, ta thấy kết quả tương đối giống nhau, mặc dù cũng có sai số nhưng không đáng kể.

3.1.2 Phương pháp tiếp tuyến (Newton – Raphson):



So sánh với kết quả tính toán bằng lý thuyết ở mục II, ta thấy kết quả tương đối giống nhau, mặc dù cũng có sai số nhưng không đáng kể.

3.1.3 Phương pháp dây cung:

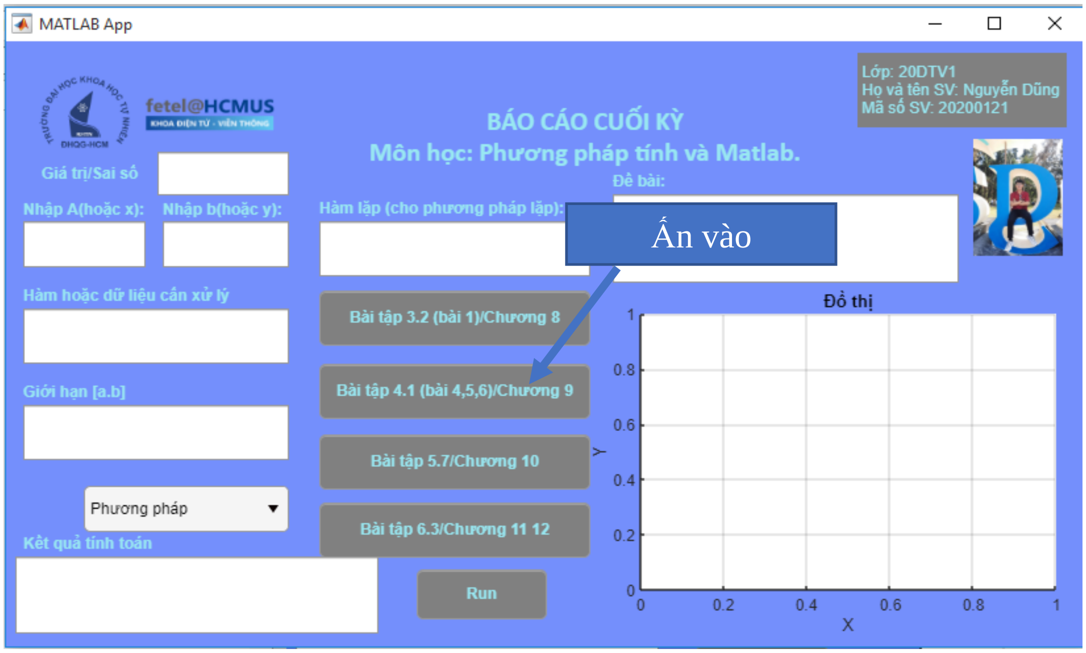


So sánh với kết quả tính toán bằng lý thuyết ở mục II, ta thấy kết quả tương đối giống nhau, mặc dù cũng có sai số nhưng không đáng kể.

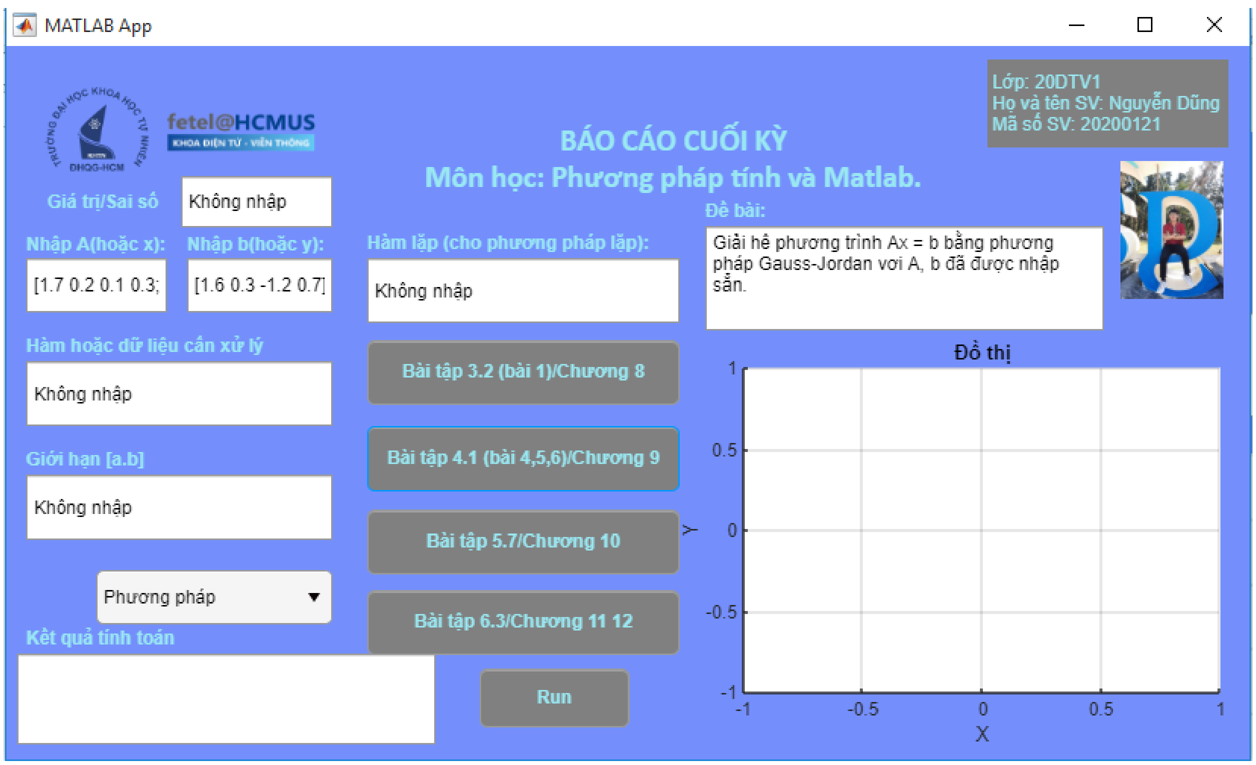
3.2. Giải hệ phương trình:

➢ Bước 1: Nhấn vào nút **"Bài tập 4.1 (bài 4,5,6)/Chương 9"** để load dữ liệu bài tập lên. Các dữ liệu cần cho tính toán như:
- Ma trận A.
- Ma trận b.

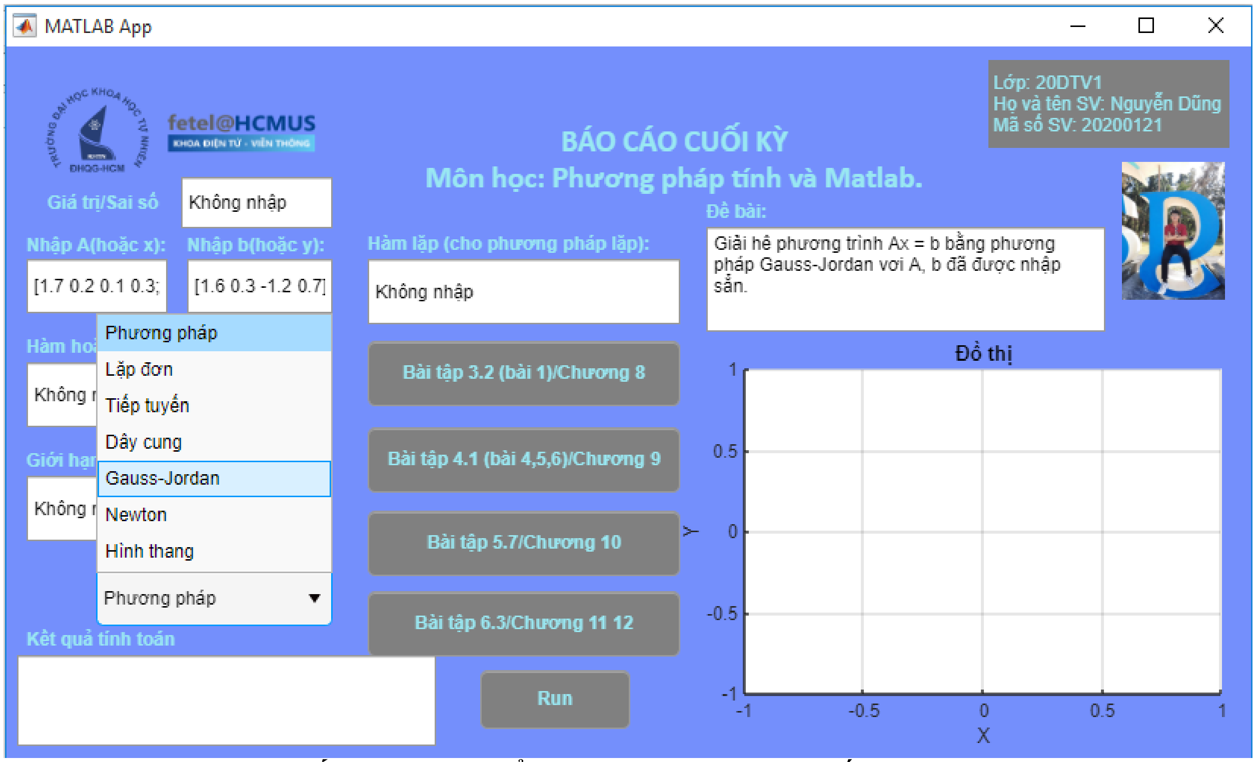
Ngoài ra còn hiện đề bài ở khung "Đề bài" để tiện quan sát.



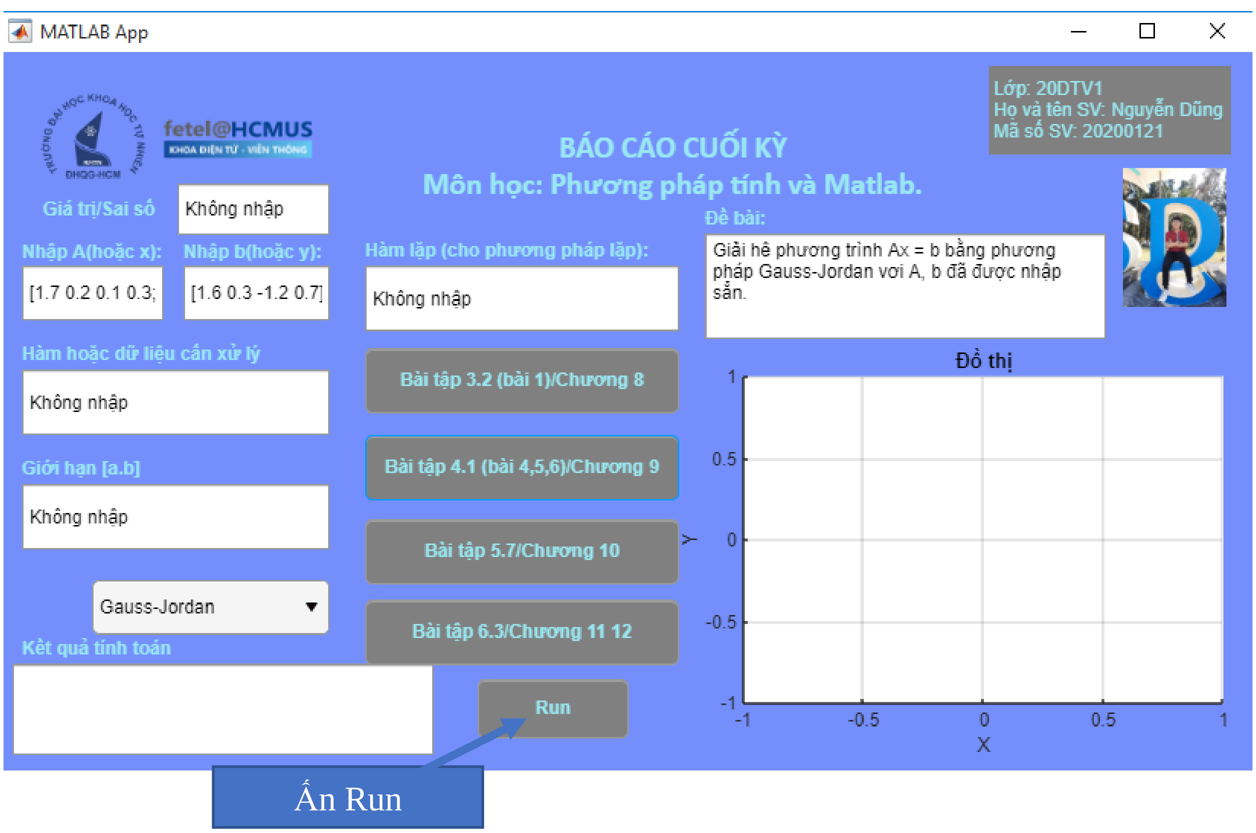
Ta thu được hình như sau:



➢ Bước 2: **Chọn phương pháp phù hợp**: trong phần này chỉ xét một phương pháp: Phương pháp Gauss-Jordan.



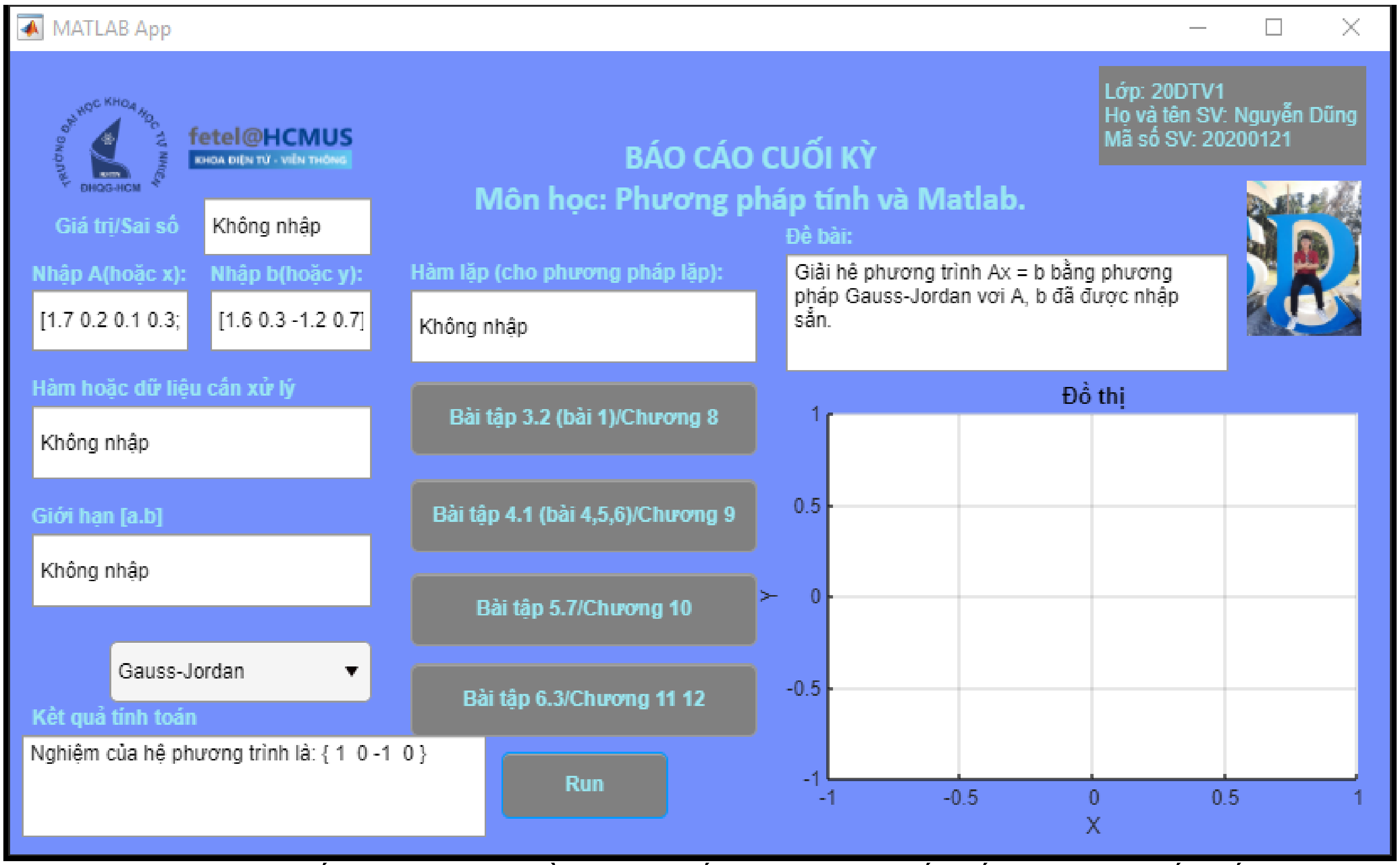
➢ Bước 3: **Ấn nút Run** để tính toán và hiện thị kết quả trên giao diện.



Chú ý: Khi load dữ liệu có những Field hiển thị "không nhập", tức có nghĩa trường đó không ảnh hưởng đến kết quả tính toán.

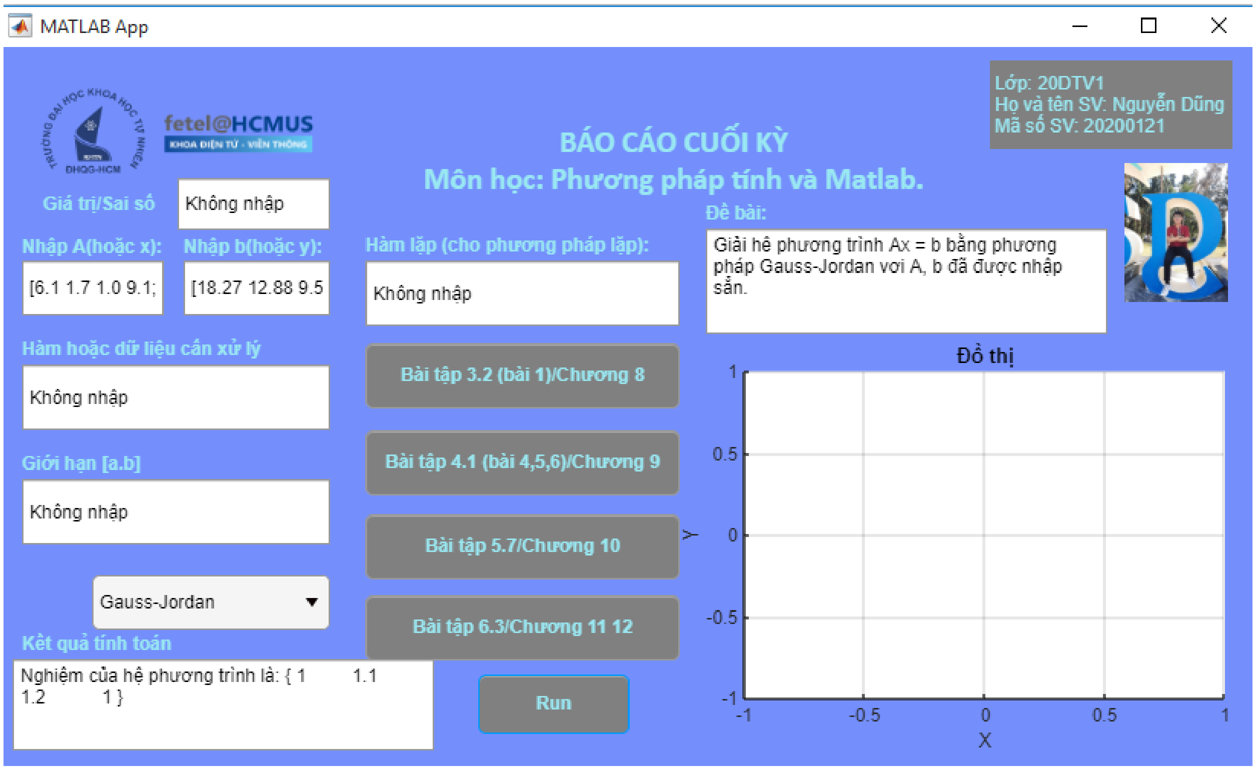
Kết quả hình tương ứng với phương pháp.

3.2.1 Phương pháp Gauss-Jordan:



So sánh với kết quả tính toán bằng lý thuyết ở mục II, ta thấy kết quả tương đối giống nhau, mặc dù cũng có sai số nhưng không đáng kể.

Trong phần giải hệ phương trình, chúng ta còn hai bài nữa, ta cần nhập tương ứng ma trận A, ma trận b sau đó nhấn ấn Run để hiển thị ra kết quả như sau:



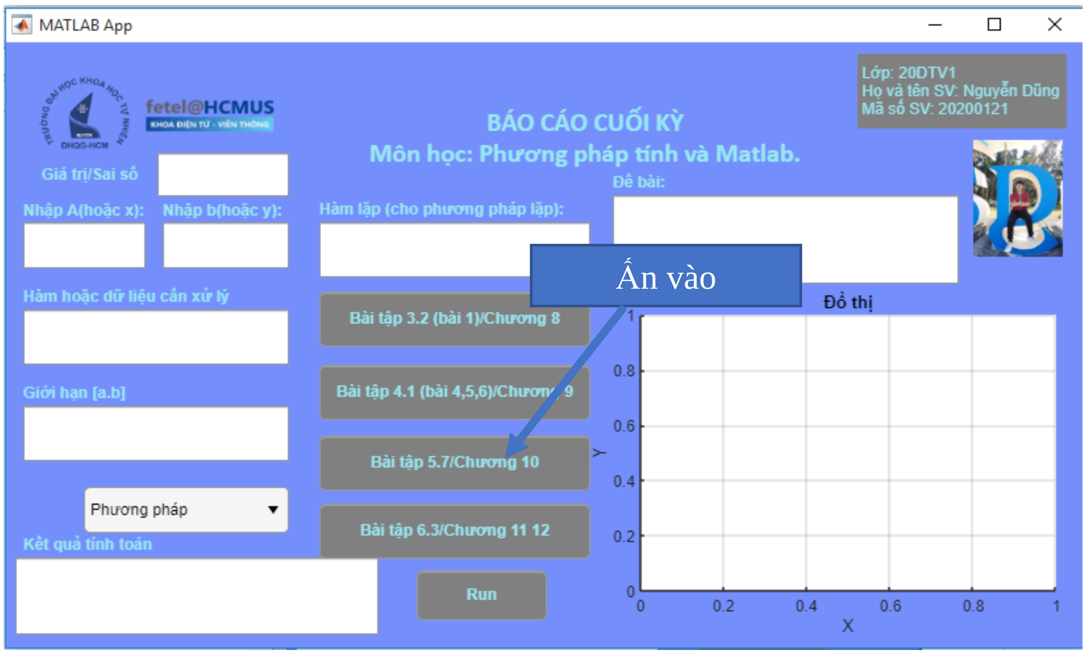
So sánh với kết quả lý thuyết, ta thấy nó tương đối giống nhau, mặc dù có sai số nhưng không đáng kể.

3.3. Nội suy, Hồi quy:

➢ Bước 1: Nhấn vào nút **"Bài tập 5.7 /Chương 10"** để load dữ liệu bài tập lên. Các dữ liệu cần cho tính toán như:

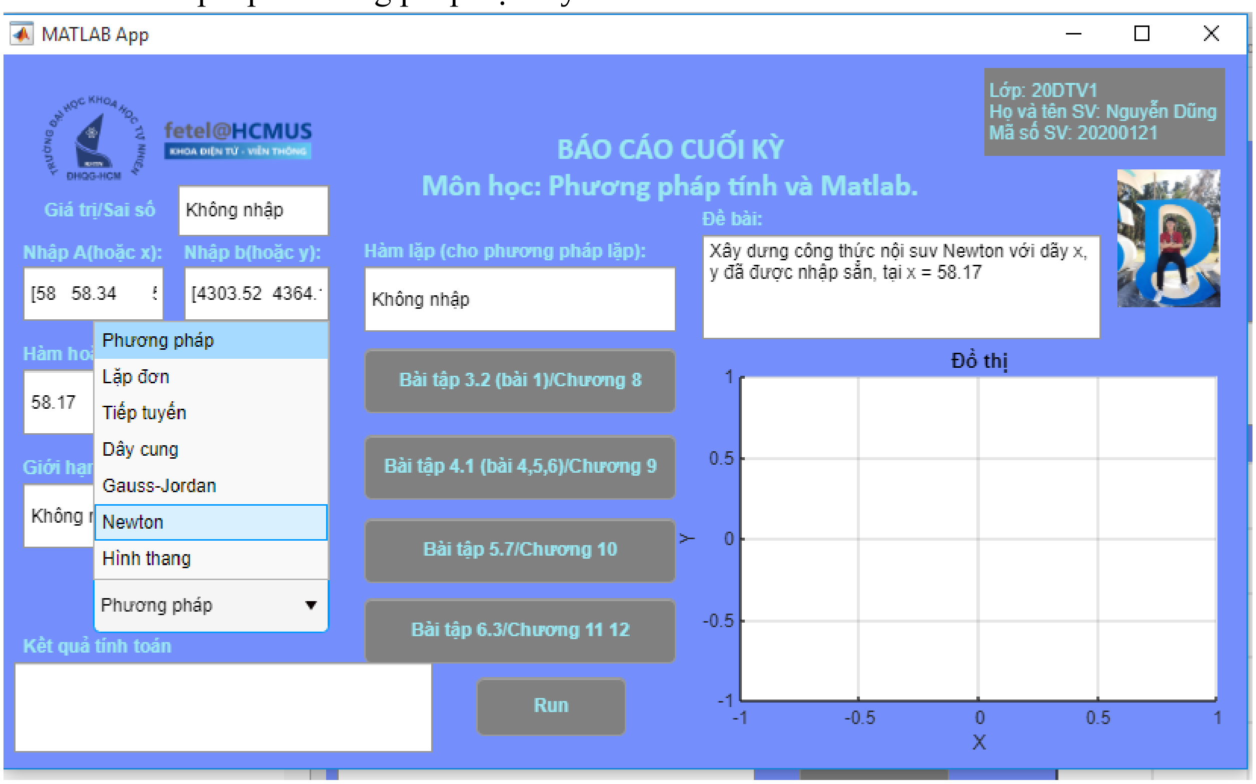
- Hàm hoặc dữ liệu cần xử lý (Điểm cần nội suy).
- Ma trận x, Ma trận y (để tìm đa thức nội suy).

Ngoài ra còn hiện đề bài ở khung "Đề bài" để tiện quan sát.

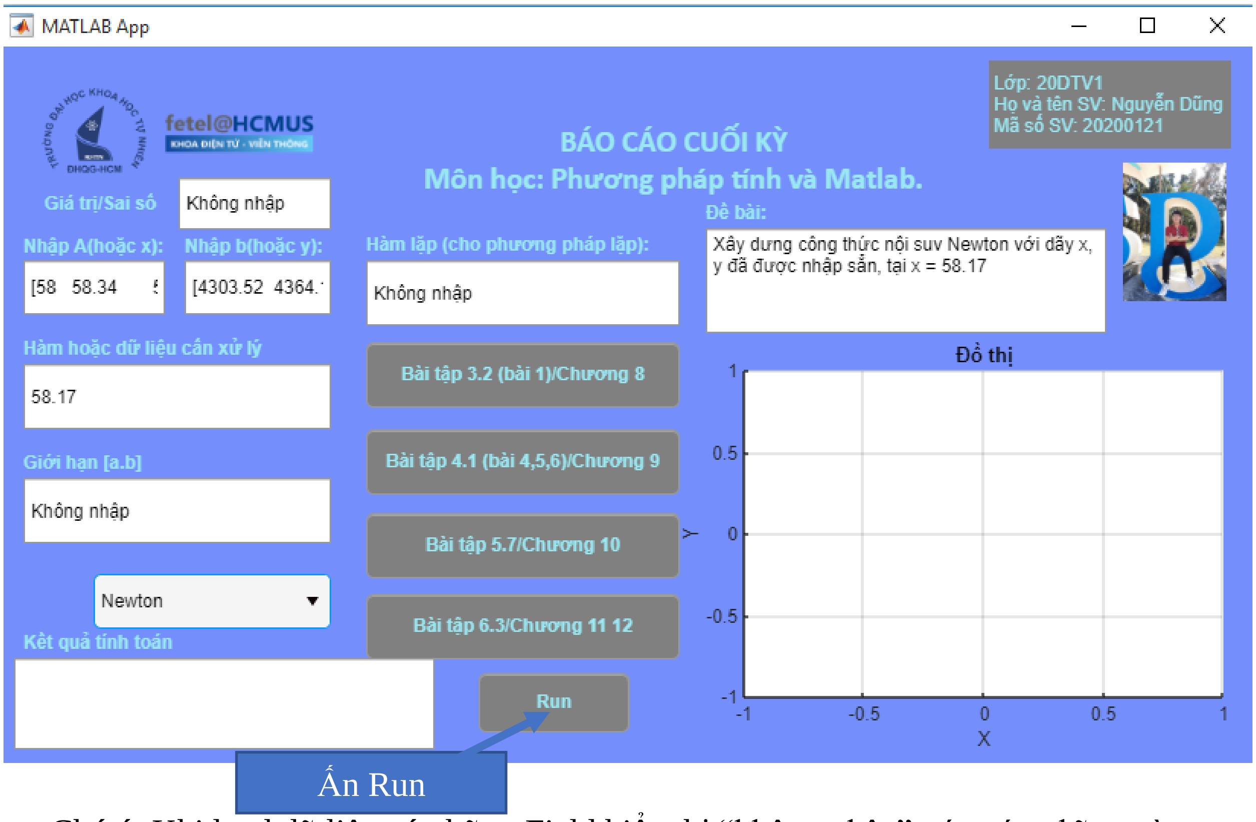


Ta thu được hình như sau:

➢ Bước 2: **Chọn phương pháp phù hợp**: trong phần này chỉ xét 1 phương pháp: Phương pháp nội suy Newton



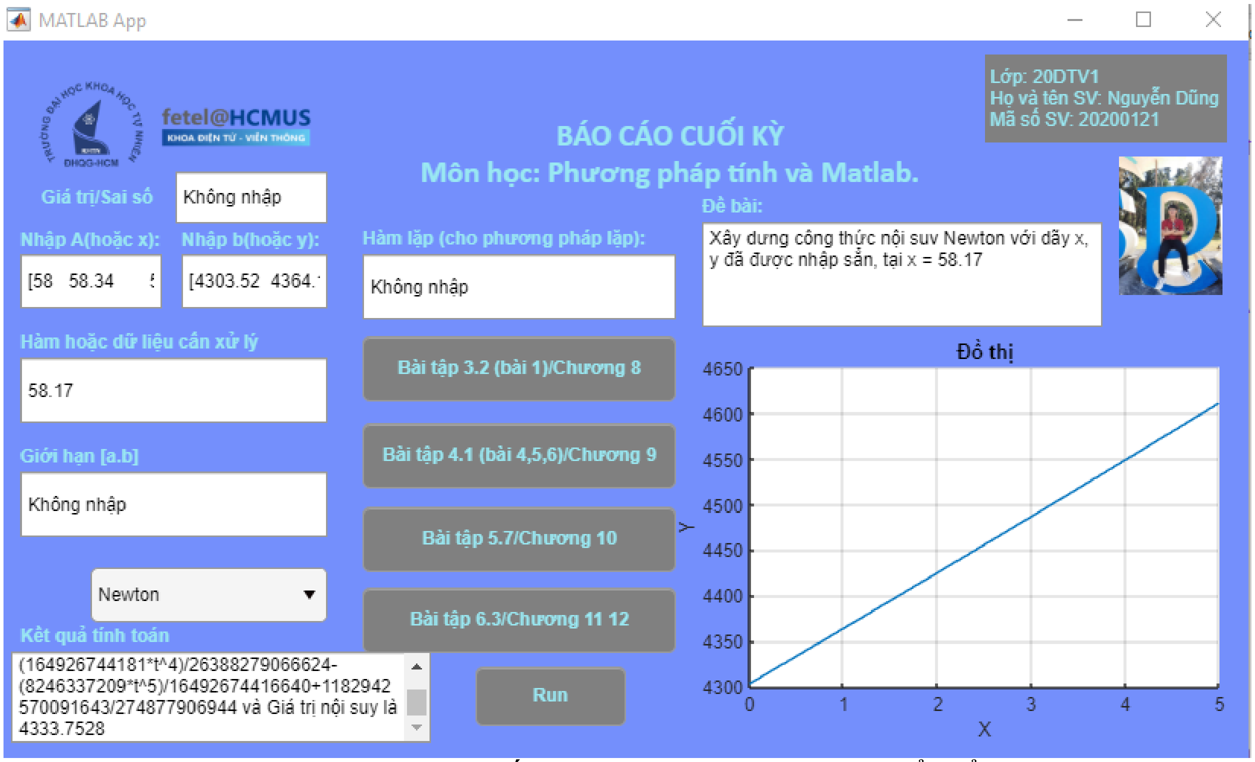
➢ Bước 3: **Ấn nút Run** để tính toán, vẽ đồ thị và hiện thị kết quả trên giao diện.



Chú ý: Khi load dữ liệu có những Field hiển thị "không nhập", tức có nghĩa trường đó không ảnh hưởng đến kết quả tính toán.

Kết quả hình tương ứng với phương pháp.

3.3.1 Nội suy Newton:



Do kích thước của khung "Kết quả tính toán" không đủ để hiển thị ra đa thức nội suy Newton, nên em sẽ ghi lại như sau (chú ý đây là đa thức theo biến t, theo công thức

$$x = x_o + ht \Rightarrow t = \frac{x - x_o}{h}$$

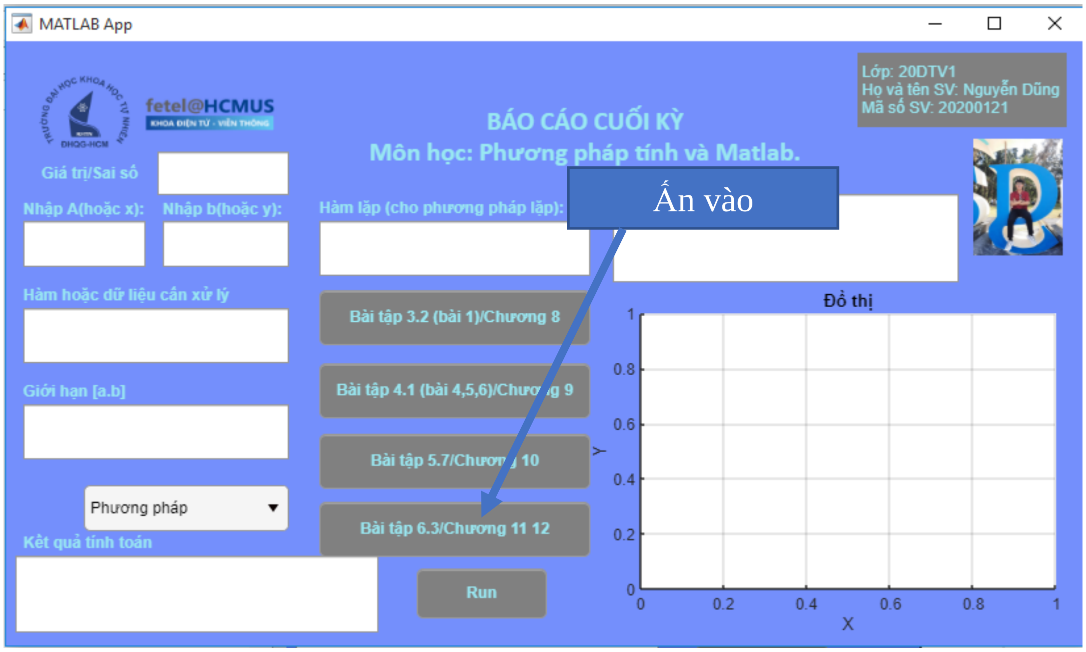
$$P_n = \frac{6633591877891619*t}{109951162777600} + \frac{7355732789959*t^2}{26388279066624} - \frac{351843720923*t^3}{13194139533312} + \frac{164926744181*t^4}{26388279066624} - \frac{8246337209*t^5}{16492674416640} + \frac{1182942570091643}{274877906944}$$

So sánh với kết quả tính toán bằng lý thuyết ở mục II, ta thấy kết quả tương đối giống nhau, mặc dù cũng có sai số nhưng không đáng kể.

3.4. Đạo hàm, tích phân, Phương trình vi phân:

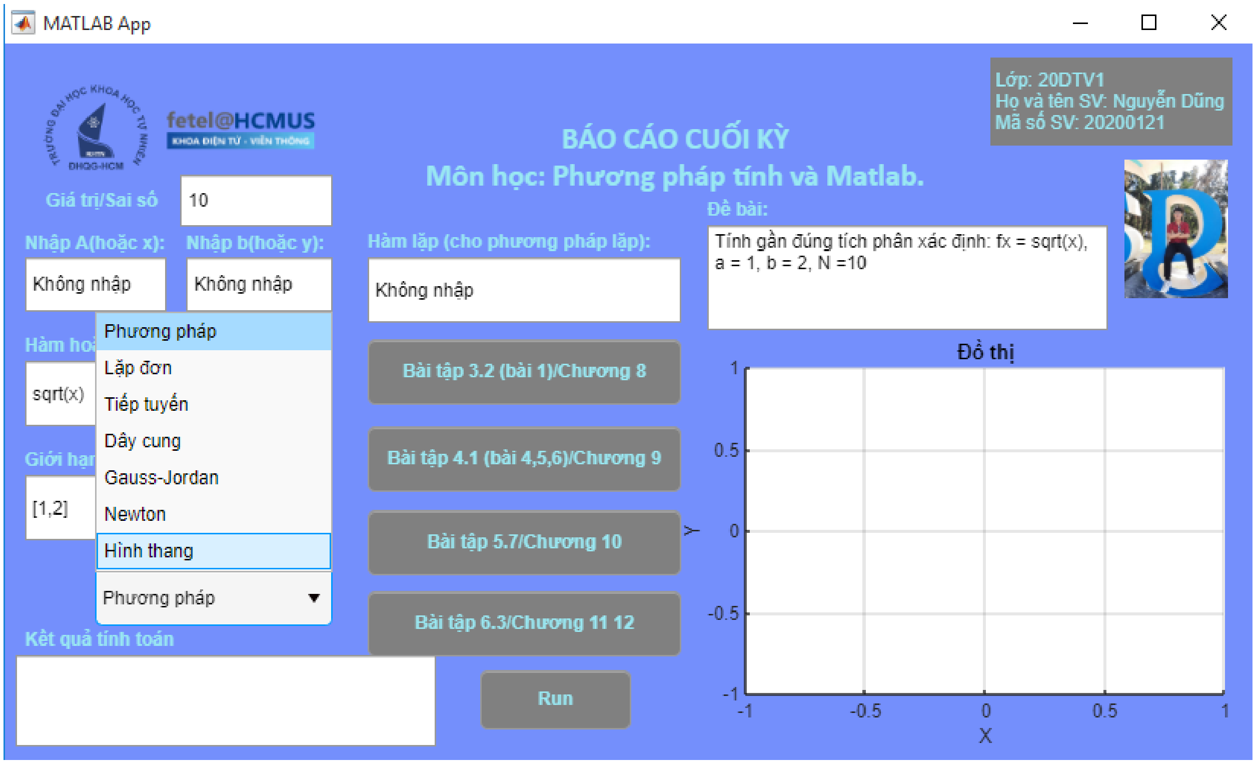
- Bước 1: Nhấn vào nút **"Bài tập 6.3 /Chương 11 12"** để load dữ liệu bài tập lên. Các dữ liệu cần cho tính toán như:
  - Giá trị, sai số (Số N đoạn con).
  - Hàm hoặc dữ liệu cần xử lý (Hàm cần tính tích phân).
  - Hai cận a, b để tính tích phân (Khoảng giới hạn $[a, b]$)

Ngoài ra còn hiện đề bài ở khung "Đề bài" để tiện quan sát.



Ta thu được hình như sau:

➢ Bước 2: **Chọn phương pháp phù hợp**: trong phần này chỉ xét 1 phương pháp: Phương pháp hình thang.

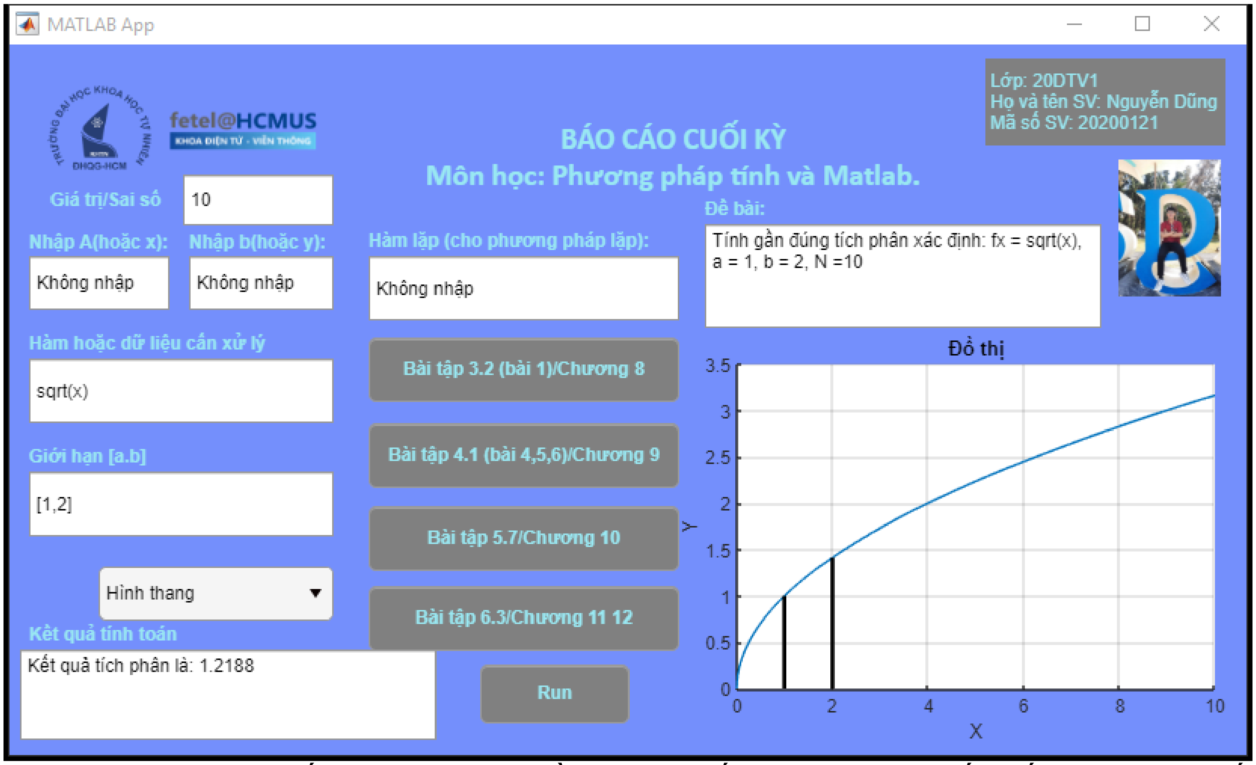


➢ Bước 3: **Ấn nút Run** để tính toán, vẽ đồ thị và hiện thị kết quả trên giao diện.

Chú ý: Khi load dữ liệu có những Field hiển thị "không nhập", tức có nghĩa trường đó không ảnh hưởng đến kết quả tính toán.

Kết quả hình tương ứng với phương pháp.

3.4.1 Phương pháp hình thang:



So sánh với kết quả tính toán bằng lý thuyết ở mục II, ta thấy kết quả tương đối giống nhau, mặc dù cũng có sai số nhưng không đáng kể.

4. Thiết lập một số lỗi trong giao diện:

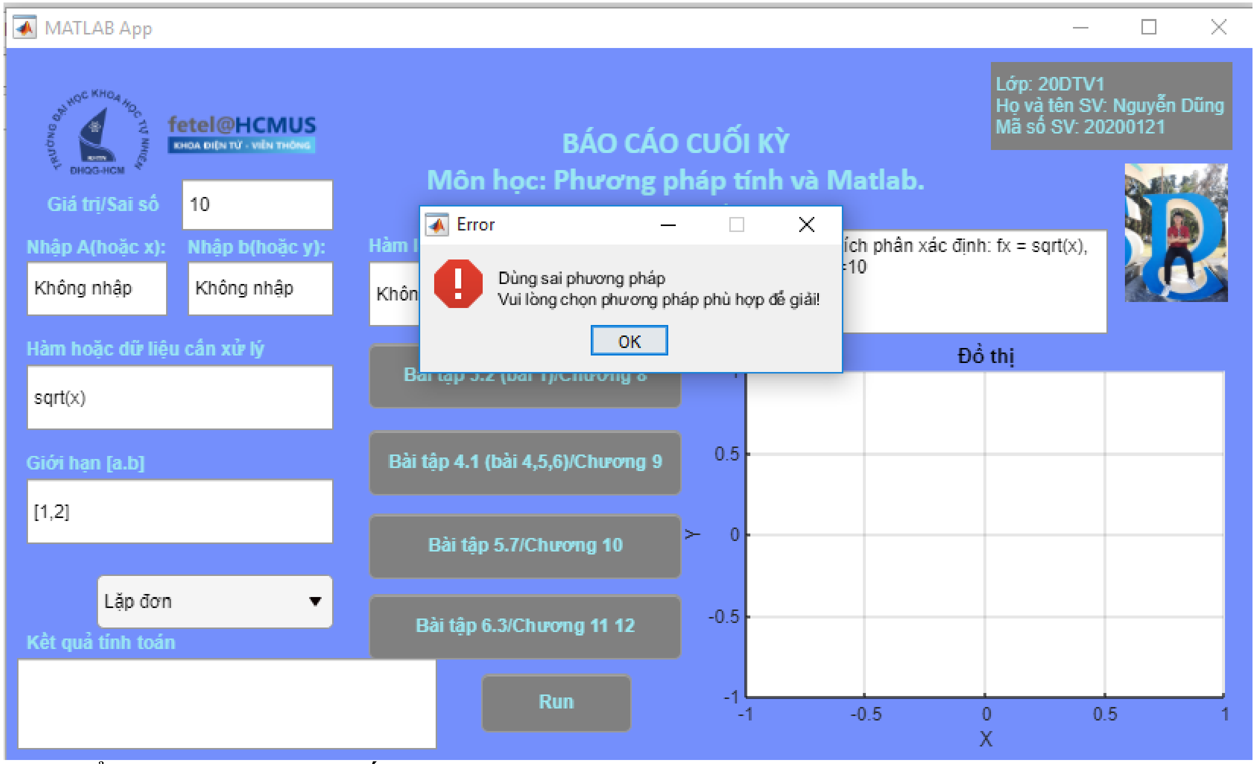
Trong phần code giao diện, chúng ta sử dụng khối lệnh

```
try
statements
catch exception
statements
end
```

thực thi các câu lệnh trong khối `try` và bắt những lỗi trong khối `catch`. Cách tiếp cận này cho phép chúng ta ghi đè lỗi mặc định cho một tập hợp các câu lệnh chương trình (xử lý lỗi). Nếu bất kỳ câu lệnh nào trong khối `try` tạo ra lỗi, chương trình sẽ ngay lập tức chuyển sang khối `catch`, khối này chứa các câu lệnh xử lý lỗi. Vậy nên chúng ta sử dụng khối lệnh này để ra các hộp thoại báo lỗi khi xuất hiện lỗi ở chương trình.

Khảo sát một số lỗi đã thiết lập:

Lỗi dùng sai phương pháp:



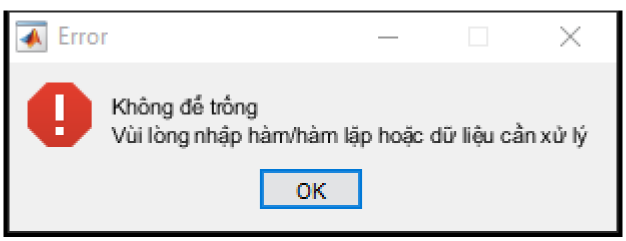
Ở đây, ta đang muốn tính tích phân nhưng lại chọn phương pháp tìm nghiệm (phương pháp lặp đơn)⇒ Lỗi, xuất hiện hộp thoại như trên hình.

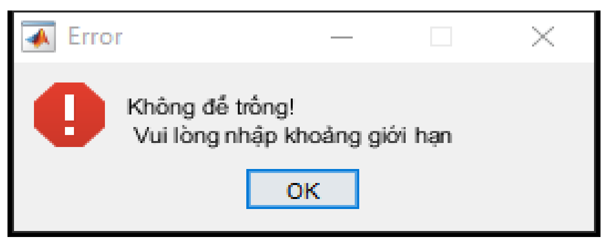
Đối với các phần khác cũng vậy, khi chọn sai phương pháp thì sẽ xuất hiện hộp thoại thông báo lỗi.

Nhập thiếu dữ liệu đầu vào:

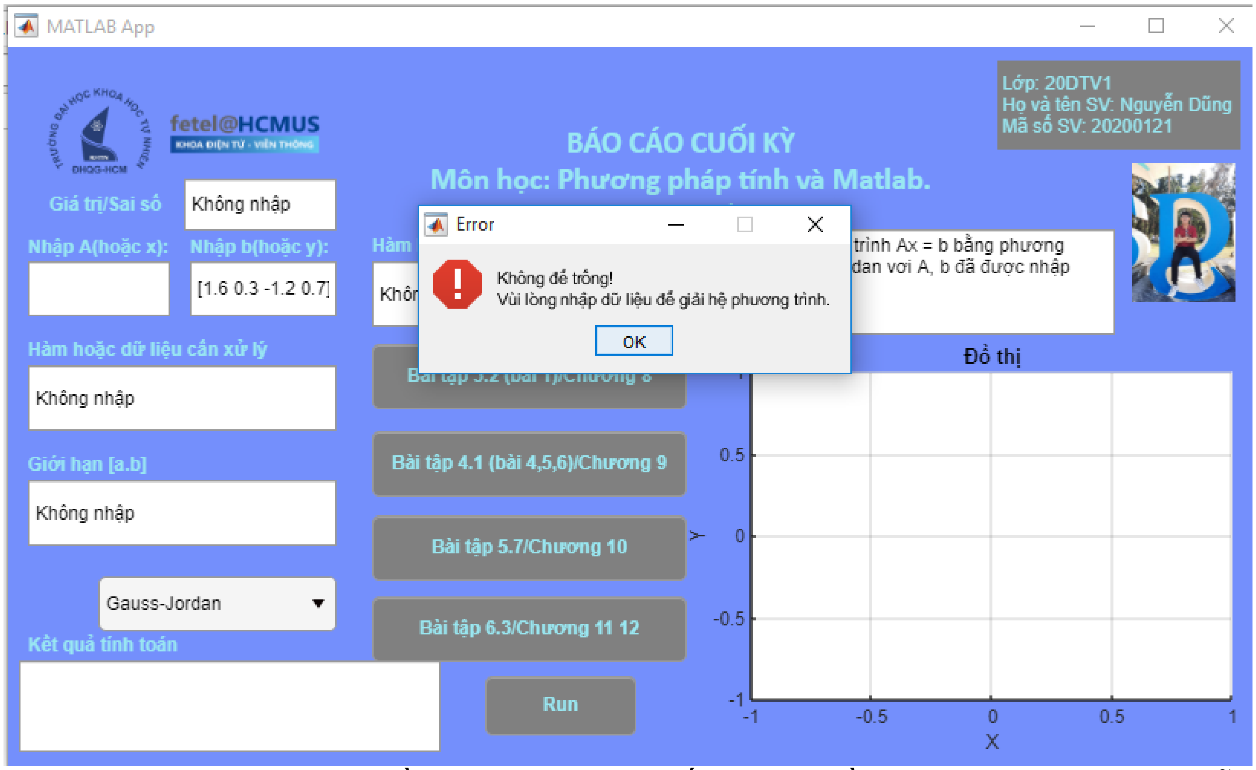
Giả sử trong quá trình tìm nghiệm, đã chọn đúng phương pháp, ta để trống giá trị sai số thì xuất hiện hộp thoại báo lỗi như hình.

Tương tự với các trường hợp khác như để trống "Hàm hoặc dữ liệu cần xử lý", "Giới hạn [a, b]" và "Hàm lặp" lần lượt sẽ xuất hiện các hộp thoại như sau:





Trong phần giải hệ phương trình, khi chọn đúng phương pháp, nếu nhập thiếu dữ liệu đầu vào cũng sẽ thông báo lỗi.



Tương tự như các phần khác khi nhập thiếu dữ liệu đầu vào cũng sẽ thông báo lỗi.

5. Đánh giá sản phẩm:

5.1. Ưu điểm:

- Giao diện cho ra kết quả như mong đợi (giống với kết quả lý thuyết đã làm ở mục 2).
- Áp dụng được các thuật toán cơ bản trong Phương pháp tính để giải nghiệm gần đúng, giải hệ phương trình, nội suy Newton và tính tích phân.
- Giao diện đầy đủ các thành phần cơ bản như yêu cầu…

5.2. Nhược điểm:

- Trong quá trình tính toán, xử lý và đưa ra kết quả còn chậm (do trong code thuật toán có nhiều vòng lặp).
- Hiển thị kết quả chưa được đẹp.

6. Kết luận:

Qua quá trình nghiên cứu, em đã hoàn thành được sản phẩm theo yêu cầu đề ra, mặc dù cũng còn nhiều thiếu sót trong báo cáo cũng như trong phần thiết kế giao diện, em mong quý thầy cô có thể bỏ qua. Lời cuối cùng, em xin cảm ơn thầy Nguyễn Xuân Vinh đã truyền tải cho em rất nhiều kiến thức để em có thể hoàn thành được đồ án này. Một lần nữa, em xin chân thành cảm ơn thầy.