ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ - VIỄN THÔNG

BÁO CÁO ĐỒ ÁN MÔN HỌC
THỰC HÀNH PHƯƠNG PHÁP TÍNH

Lớp 21DTV-CLC4- Nhóm:

Giảng viên hướng dẫn: Th.S HUỲNH QUỐC THỊNH
Thành viên nhóm
Nguyễn Phước Bảo- 21207123
Nguyễn Minh Quân - 21207204
Cao Nguyễn Hải Đăng - 21207012
Nguyễn Hoàng Bảo-212070121
Trương Hoàng Bách Hợp - 21207034
Lưu Hoàng Đông Anh - 21207007

THÀNH PHỐ HỒ CHÍ MINH 12/2023

Mục lục

THÀNH PHỐ HỒ CHÍ MINH 12/2023

I/ CÁC PHƯƠNG PHÁP TÍNH

1/ Bài toán tìm nghiệm

1.1/ Phương pháp chia đôi

Nội dung phương pháp chia đôi:
Giả sử (a, b) là một khoảng phân ly nghiệm của phương trình f(x) = 0. Nghiệm đúng của phương trình f(x) = 0 trong khoảng phân ly nghiệm (a, b) được lấy xấp xỉ bằng điểm giữa của khoảng phân ly.
Không làm mất tính tổng quát, bằng cách đổi dấu hàm f(x), luôn giả sử f(a) < 0 < f(b).

Thuật toán phương pháp chia đôi
Điều kiện: f(a).f(b)<0
B1 Tìm khoảng phân ly nghiệm
B2: Tính x = (a+b)/2
B3: Xác định khoảng phân ly nghiệm mới
B4: Lặp lại khi |b-a| < es

$$x_n = \frac{a_n + b_n}{2}$$

Tính chất của phương pháp chia đôi

Phương pháp chia đôi đảm bảo hội tụ, nghĩa là nếu phương pháp chia đôi được thực hiện với các bước tính hợp lệ, thì nó sẽ hội tụ về nghiệm x của phương trình f(x) = 0.

Phương pháp chia đôi đảm bảo độ chính xác tăng dần theo từng bước tính, nghĩa là sai số |f(x)| sẽ giảm dần khi số bước tính tăng dần.

Ưu điểm và nhược điểm của phương pháp chia đôi

Ưu điểm

- Phương pháp chia đôi đơn giản, dễ thực hiện.
- Phương pháp chia đôi có độ chính xác cao.

Nhược điểm

- Phương pháp chia đôi có thể tốn nhiều bước tính nếu khoảng phân ly nghiệm rộng.

1.2/ Phương pháp lặp

Phương pháp lặp là một phương pháp tìm nghiệm gần đúng của phương trình. Phương pháp này dựa trên tính chất lặp lại quy trình để cập nhật giá trị xấp xỉ của nghiệm

Thuật toán:

B1. Cho phương trình f(x) = 0.

B2. Ấn định sai số cho phép ϵ.

B3. Xác định khoảng phân ly nghiệm [a,b].

B4. Tìm hàm lặp hội tụ ϕ

B5 Chọn xấp xỉ đầu x0 Xác định dựa vào cách thu hẹp khoảng phân ly nghiệm như phương pháp chia đôi.

B6. Tính x_n = ϕ(x_n−1),n = 1,2,3,... Cho tới khi |$x_n - x_n$−1| < ϵ thì dừng

Tính chất của phương pháp lặp

- Phương pháp lặp đảm bảo hội tụ, nghĩa là nếu phương pháp lặp được thực hiện với các bước tính hợp lệ, thì nó sẽ hội tụ về nghiệm x của phương trình f(x) = 0.
- Phương pháp lặp đảm bảo độ chính xác tăng dần theo từng bước tính, nghĩa là sai số |f(x_n)| sẽ giảm dần khi số bước tính n tăng dần.

Ưu điểm và nhược điểm của phương pháp lặp

Ưu điểm

- Phương pháp lặp có thể hội tụ nhanh chóng nếu hàm φ(x) thỏa mãn các điều kiện hội tụ của phương pháp lặp.

Nhược điểm

- Phương pháp lặp có thể không hội tụ nếu hàm φ(x) không thỏa mãn các điều kiện hội tụ của phương pháp lặp

1.3/ Phương pháp tiếp tuyến

Phương pháp tiếp tuyến là phương pháp một phương pháp tìm nghiệm gần đúng của phương trình. Ở bước lặp thứ k ta thay hàm f(x) bởi tiếp tuyến với đồ thị tại điểm x_k .Nghiệm xấp xỉ tiếp theo là giao điểm của tiếp tuyến với trục hoành

Thuật toán

1. Cho phương trình f(x) = 0.
2. Ấn định sai số cho phép ϵ.
3. Xác định khoảng phân ly nghiệm [a,b].
4. Chọn xấp xỉ đầu x0 để $f(x0) * f''(x0) > 0$
5. Tính

$$x_n=\frac{xn-1-f(xn-1)}{f'x(xn-1)}$$

Cho tới khi $|x_n - x_{n-1}| < \epsilon$ thì dừng

Ưu điểm

- Hội tụ rất nhanh so với phương pháp lặp đơn và phương pháp chia đôi

Nhược điểm

- Phải biết đạo hàm của hàm f(x)
- Để x hội tụ về nghiệm, đoạn đồ thị trong khoảng a,b phải là đơn điệu

1.4/ Phương pháp dây cung

Phương pháp dây cung là phương pháp một phương pháp tìm nghiệm gần đúng của phương trình. Nội dung của phương pháp dây cung là ta thay đường cong y=f(x) bằng dây cung của nó trong (a; b)

Thuật toán

1. Cho phương trình f(x) = 0.
2. Ấn định sai số cho phép ϵ.
3. Xác định khoảng phân ly nghiệm [a,b].
4. Áp dụng thuật toán trong Hình 1 nhưng với

$$c = \frac{\mathrm{f(b)} - \mathrm{bf(a)}}{\mathrm{f(b)} - \mathrm{f(a)}}$$

và điều kiện dừng $| c_n - c_{n-1} | < \epsilon$

Ưu điểm:

- Có thể hội tụ rất nhanh nếu nghiệm x gần điểm khởi đầu

Dễ thực hiện

Nhược điểm

- Có thể không hội tụ nếu nghiệm x nằm xa điểm khởi đầu
- Có thể không hội tụ nếu f(x) có đạo hàm bằng 0 tại điểm x=x0

2/ Bài toán nội suy

2.1/ Nội suy Newton

Nội suy Newton là một phương pháp nội suy sử dụng các đa thức bậc thấp để suy ra giá trị của một hàm tại một điểm x không nằm trong tập dữ liệu cho trước

Tỉ hiệu cấp một của y tại x_i và x_j là: $y[x_i\ , x_j\ ] = \frac{(y_i - y_j)}{(x_i - x_j)}$.

Tỉ hiệu cấp hai của y tại x_i, x_j và $x_k$là: $y[x_i, x_j, x_k]$ = $\frac{y[x_i, x_j] - y[x_i, x_k]}{(x_i - x_k)}$

Tương ứng cho các tỉ hiệu cấp cao hơn.

Các tỉ hiệu có tính đối xứng: $y[x_i\ , x_j\ ]= y[x_j, x_i\ ]$

Đa thức Newton tiến xuất phát từ nút x_0:

$p_n\ (x)= y_0+ (x - x_0)$y$[x_0\ , x_1]$+ $(x - x_0)(x - x_1)$y$[x_0\ , x_1, x_2]$+…+ $(x - x_0)(x - x_1)\ldots(x - x_{n-1})y[x_0\ ,\ldots, x_n]$

Đa thức Newton lùi xuất phát từ nút x_n:

$p_n\ (x)= y_n+ (x - x_n)$y$[x_n\ , x_{n-1}]$+ $(x - x_n)(x - x_{n-1})$y$[x_n\ , x_{n-1}, x_{n-2}]$+…+ $(x - x_n)(x - x_{n-1})\ldots(x - x_1)y[x_n\ ,\ldots, x_0]$

Để thực hiện nội suy Newton chúng ra chia thành 2 bước

- Tính tỷ hiệu các cấp
- Thực hiện các số hạng dạng Newton

Ưu điểm và nhược điểm của nội suy Newton

Ưu điểm của nội suy Newton:

- Nội suy Newton là một phương pháp đơn giản và dễ thực hiện.
- Nội suy Newton có thể cho kết quả chính xác cao.

Nhược điểm của nội suy Newton:

- Nội suy Newton chỉ có thể được sử dụng cho các hàm có thể được biểu diễn bằng đa thức bậc thấp.
- Nội suy Newton có thể không chính xác nếu tập dữ liệu cho trước không đủ.

2.2/ Nội suy Lagrange

x	x_0	x_1	x_2	…	x_n
y	y_0	y_1	y_2	…	y_n

Đa thức nội suy $p(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) + \cdots + y_n l_n(x)$

$$l_0 = \frac{(x - x_1)(x - x_2) \ldots (x - x_{n-1})(x - x_n)}{(x_0 - x_1)(x_0 - x_2) \ldots (x_0 - x_{n-1})(x_0 - x_n)}$$

$$l_{n-1} = \frac{(x - x_0)(x - x_1)(x - x_2) \ldots (x - x_{n-2})(x - x_n)}{(x_{n-1} - x_0)(x_{n-1} - x_1(x_{n-1} - x_2) \ldots (x_{n-1} - x_{n-2})(x_{n-1} - x_n)}$$

Công thức thuật toán tổng quát $p_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$

Với

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

3/ Bài toán hồi quy

3.1/ Hồi quy tuyến tính

Hồi quy tuyến tính sử dụng mô hình đường thẳng y = a_0+ a_1x để làm khớp các dữ liệu dữ liệu có được. Phần sai khác (error) giữa mô hình và các giá trị quan sát được được minh họa bằng phương trình

$$e=y-a_0-a_1x$$

Mục tiêu của việc làm khớp sử dụng tiêu chuẩn bình phương tối thiểu là tổng bình phương các giá trị lỗi (S_r) này đạt giá trị tối thiểu

$$S_r=\sum_{i=1}^{n} e_i^2=\sum_{i=1}^{n}\left(y_{i(\text{đo được})}-y_{i(\text{mô hình})}\right)^2=\sum_{i=1}^{n}(y_i-a_0-a_1x_i)^2$$

Để tìm giá trị cực tiểu của S_r ta tìm các giá trị đạo hàm riêng và giải hệ phương trình các đạo hàm riêng

$$\begin{cases}\left(\frac{\partial s_n}{\partial s_0}\right)=0\\\left(\frac{\partial s_n}{\partial s_1}\right)=0\end{cases} \Leftrightarrow \begin{cases}-2\Sigma(y_i-a_0-a_1x_i)=0\\-2\Sigma[(y_i-a_0-a_1x_i)x_i]=0\end{cases}$$

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix}=\begin{bmatrix} \sum y_i \\ \sum y_i x_i \end{bmatrix}$$

Kết quả thu được hệ số của phương trình hồi quy tuyến tính như sau

$$a_1=\frac{n\sum x_iy_i-\sum x_i\sum y_i}{n\sum x_i^2-(\sum x_i)^2}$$

$$a_0=\bar{y}-a_1\bar{x}$$

Trong đó: $\bar{x}$, $\bar{y}$ là giá trị trung bình của mảng x và mảng y tương ứng

Hệ số tương quan r^2

Gọi S_r là tổng bình phương các giá trị lỗi (error) giữa các giá trị đo được và giá trị của mô hình

$$S_r=\sum_{i=1}^{n} e_i^2=\sum_{i=1}^{n}\left(y_{i(\text{đo được})}-y_{i(\text{mô hình})}\right)^2=\sum_{i=1}^{n}(y_i-a_0-a_1x_i)^2$$

Gọi S_t là tổng bình phương của các giá trị lỗi giữa các giá trị đo được và giá trị trung bình

$$S_t=\sum_{i=1}^{n}(y_i-\bar{y})^2$$

Khi đó hệ số tương quan r^2 được tính theo công thức

$$r^2=\frac{S_t-S_r}{S_t}$$

r^2 thể hiện kết quả của mô hình hồi quy khớp hay chưa chưa khớp. r càng gần 1 thì kết quả hồi quy càng khớp

3.2/ Hồi quy phi tuyến

Để hồi quy không tuyến tính, tùy theo dạng dữ liệu ta tuyến tính hóa sau đó thực hiện hồi quy tuyến tính. Cuối cùng suy ra được các hệ số của phương trình phi tuyến

Dạng $y = ae^{bx}$

Lấy Logarit cơ số e 2 vế: $lny = lna + bx$

Đặt $Y = lny$; $A_0 = lna$; $A_1 = b$; $X = x$;

Đưa về dạng $Y = A0 + A1X$

Giải hệ phương trình tìm được A_0, A_1=> $a = e^{A_0}$; $b = A_1$

Dạng $y = ax^b$

Lấy Logarit cơ số 10 2 vế: $lgy = lga + blgx$

Đặt $Y = lgy$; $A_0 = lga$; $A_1 = b$; $X = lgx$;

Đưa về dạng $Y = A_0 + A_1$X

Giải hệ phương trình tìm được A_0, A_1=> $a = 10^{A_0}$; $b = A_1$

Dạng $y = a\frac{x}{b+x}$

Nghịch đảo 2 vế:

$\frac{1}{y} = \frac{b}{a}, \frac{1}{x} + + \frac{1}{a}$

Đặt $Y = \frac{1}{y} =$; $A_1 = \frac{b}{a}$; $A_0 = \frac{1}{a}$; $X = \frac{1}{x}$;

Đưa về dạng $Y = A_0 + A_1$X

Giải hệ phương trình tìm được A, B => $a = \frac{1}{A_0}$; $b = A_1$.a

4/ Bài toán tích phân

Tính tích phân xác định bình thường như sau:

$$S=\int_a^b f(x)dx$$

Nhưng nếu việc tìm nguyên hàm của f(x) quá khó khăn thì việc tính tích phân phải làm gần đúng.

4.1/ Hình thang

Tính gần đúng tích phân với công thức hình thang:

Ý tưởng: chia nhỏ khoảng lấy tích phân [a,b]. Hình cong được thay thế gần đúng bởi hình thang. Như vậy, tích phân gần đúng là tổng các diện tích hình thang nhỏ. Cách này tương đương với việc lấy tích phân của hàm nội suy bậc 1 của phương pháp nội suy Newton tiến với khoảng cách đều.

Tính gần đúng tích phân với công thức Simpson

Ý tưởng: do việc tính gần đúng tích phân là việc tích phân của hàm nội suy nên hàm nội suy chính xác hơn cho kết quả gần đúng có sai số nhỏ hơn. Trong công thức Simpson, việc tính gần đúng tích phân là việc tích phân của hàm nội suy bậc 2 của phương pháp nội suy Newton tiến với khoảng cách đều.

4.2/ Simpson ⅓

$$I=\int_a^b f(x)dx \approx \frac{h}{3}[\{f(a)+f(b)\}+4\sum_{i=1}^{N-1} f(a+ih)]+2\sum_{i=2}^{N-1} f(a+ih)]$$

Công thức Simpson chia đoạn [a,b] thành 2N đoạn cong bằng nhau: $h=\frac{b-a}{N}$

4.3/ Simpson ⅜

Công thức Simpson chia đoạn [a,b] thành 3N đoạn cong bằng nhau:

$$I=\int_a^b f(x)dx \approx \frac{3h}{8}[\{f(a)+f(b)\}+3\sum_{i=1}^{N-2} f(a+ih)]+3\sum_{i=2}^{N-1} f(a+ih)+2\sum_{i=3}^{N-3} f(a+ih)]$$

Công thức Simpson chia đoạn [a,b] thành 3N đoạn cong bằng nhau:

$$h=\frac{b-a}{N}$$

5/ Bài toán đạo hàm

5.1/ Đạo hàm tiến

Đạo hàm tiến là một phương pháp tính đạo hàm gần đúng của hàm số tại điểm $x=a$ bằng cách sử dụng giá trị của hàm số tại điểm $x=a$ và điểm $x=a+h$, với h là một số thực dương nhỏ.

Công thức đạo hàm tiến

$$\frac{f(a+h)-f(a)}{h}$$

là công thức đạo hàm tiến. Nó được sử dụng để tính đạo hàm gần đúng của hàm số f tại điểm $x=a$ bằng cách sử dụng giá trị của hàm số tại $x=a$ và $x=a+h$, với h là một số thực dương nhỏ.

Công thức này cho thấy đạo hàm $f'(a)$ xấp xỉ bằng tỉ số giữa sự thay đổi của hàm số $f(a+h)-f(a)$ và sự thay đổi của biến độc lập h. Nói cách khác, nó xấp xỉ độ dốc của tiếp tuyến của đồ thị hàm số tại điểm $x=a$.

Ưu điểm và nhược điểm của đạo hàm tiến

Ưu điểm:
- Đơn giản và dễ tính toán
- Có thể áp dụng cho mọi hàm số

Nhược điểm:
- Độ chính xác không cao, nếu h quá lớn

5.2/ Đạo hàm lùi

Công thức: $\frac{f(a)-f(a-h)}{h}$

Đạo hàm lùi là một phương pháp tính đạo hàm gần đúng của hàm số tại điểm $x=a$ bằng cách sử dụng giá trị của hàm số tại điểm $x=a-h$ và điểm $x=a$, với h là một số thực dương nhỏ.

Công thức này cho thấy đạo hàm $f'(a)$ xấp xỉ bằng tỉ số giữa sự thay đổi của hàm số $f(a)-f(a-h)$ và sự thay đổi của biến độc lập h. Nói cách khác, nó xấp xỉ độ dốc của tiếp tuyến của đồ thị hàm số tại điểm $x=a$.

Ưu điểm và nhược điểm của đạo hàm lùi

Ưu điểm:

- Độ chính xác cao hơn đạo hàm tiến, đặc biệt khi h nhỏ
- Dễ dàng tính toán

Nhược điểm:

- Khó áp dụng cho hàm số có nhiều biến

5.3/ Đạo hàm trung tâm

Đạo hàm trung tâm là một phương pháp tính đạo hàm gần đúng của hàm số f tại điểm $x=a$ bằng cách sử dụng giá trị của hàm số tại điểm $x=a-h$, điểm $x=a$ và điểm $x=a+h$, với h là một số thực dương nhỏ

Công thức : $\frac{f(a-h)+f(a+h)-2f(a)}{2h}$

Công thức này cho thấy đạo hàm f''a) xấp xỉ bằng tỉ số giữa sự thay đổi của hàm số $f(a-h)+f(a+h)-2f(a)$ và sự thay đổi của biến độc lập h. Nói cách khác, nó xấp xỉ độ dốc của tiếp tuyến của đồ thị hàm số tại điểm $x=a$.

Ưu điểm

- Độ chính xác cao nhất trong ba phương pháp tính đạo hàm gần đúng: Đạo hàm trung tâm sử dụng giá trị của hàm số tại ba điểm $x=a-h$ $x=a$ và $x=a+h$. Do đó, đạo hàm trung tâm có thể phản ánh tốt hơn sự thay đổi của hàm số tại điểm $x=a$.

Sự khác biệt giữa đạo hàm tiến, đạo hàm lùi và đạo hàm trung tâm

Đạo hàm tiến và đạo hàm lùi đều sử dụng hai giá trị của hàm số tại hai điểm $x=a$ và $x=a\pm h$ để tính đạo hàm gần đúng của hàm số tại điểm $x=a$. Tuy nhiên, đạo hàm tiến sử dụng điểm $x=a+h$, trong khi đạo hàm lùi sử dụng điểm $x=a-h$.Đạo hàm trung tâm sử dụng ba giá trị của hàm số tại ba điểm $x=a-h$, $x=a$ và $x=a+h$ để tính đạo hàm gần đúng của hàm số tại điểm $x=a$.

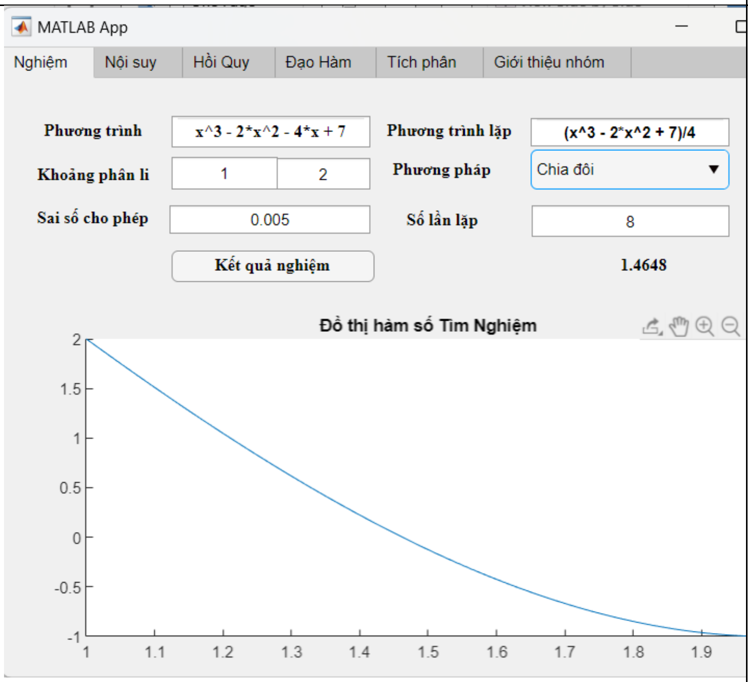
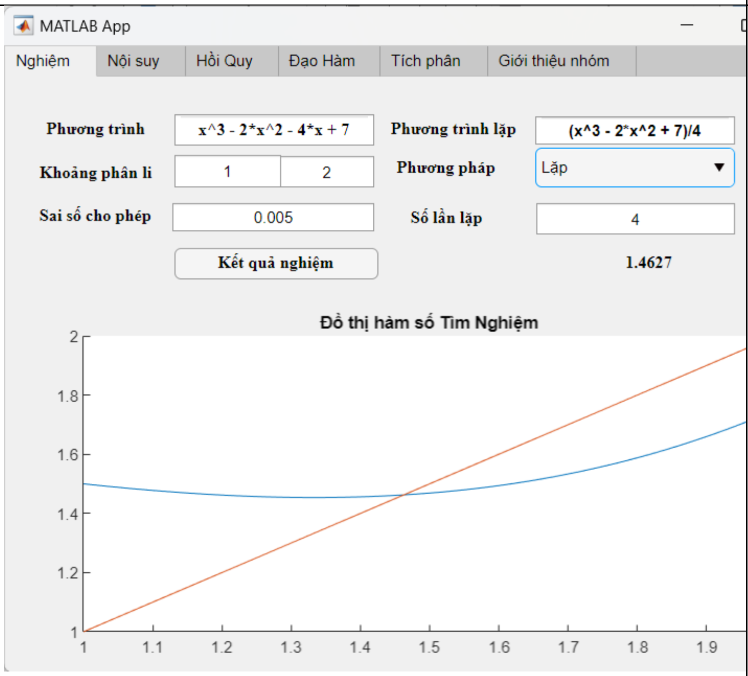
Ưu điểm và nhược điểm của đạo hàm tiến, đạo hàm lùi và đạo hàm trung tâm

- Đạo hàm tiến và đạo hàm lùi có ưu điểm là đơn giản và dễ tính toán. Tuy nhiên, chúng có nhược điểm là độ chính xác không cao, đặc biệt khi
- h lớn.
- Đạo hàm trung tâm có ưu điểm là độ chính xác cao hơn đạo hàm tiến và đạo hàm lùi

Phương pháp	Ưu điểm	Nhược điểm
Đạo hàm tiến	Dễ tính toán	Độ chính xác thấp
Đạo hàm lùi	Độ chính xác cao hơn đạo hàm tiến	Khó tính toán
Đạo hàm trung tâm	Độ chính xác cao nhất	Khó áp dụng cho hàm số có nhiều biến

Tùy theo yêu cầu của bài toán mà ta có thể lựa chọn phương pháp tính đạo hàm gần đúng phù hợp

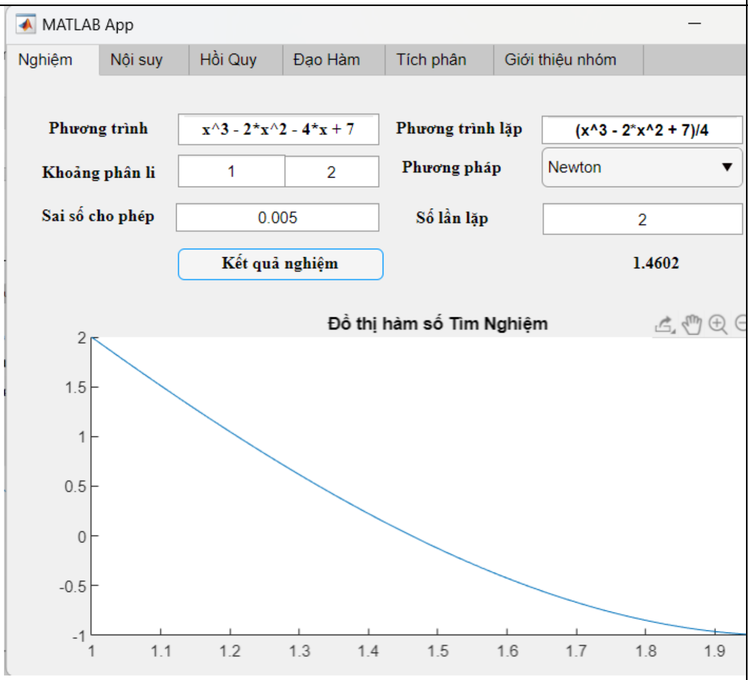
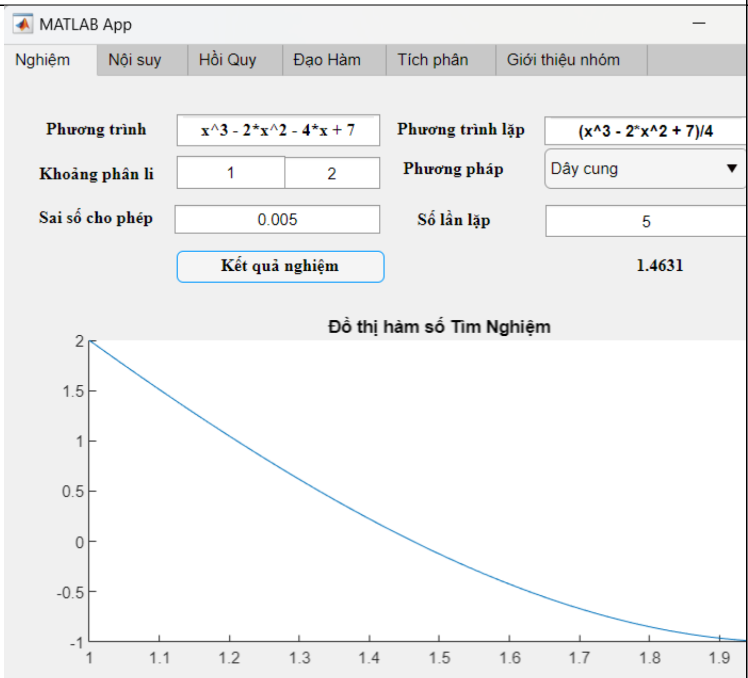
II/ Tổng hợp code và giao diện

<table>
<tr><th>PP</th><th>Hình</th><th>Code (hàm function)</th></tr>
<tr><td>Tìm nghiệm (chia đôi)</td><td>
</td><td>function [c, n] = PPChiaDoi(fx, a, b, saiso)
%gia su fx la string
fx = str2func(['@(x)',fx]);
n = 0 ;
while(1)
c = (a+b)/2;
if(fx(a)*fx(c)<0)
b = c;
else
a = c;
end
%dem so lan lap
n = n+1;
%dieu kien de thoat vong lap
e = abs(b-a);
if(e< saiso)
break;
end
end
end</td></tr>
<tr><td>Tìm nghiệm (lặp)</td><td>
</td><td>function [c1, n1]=PPLapDon(fx, fp, a, b, saiso)
fx = str2func(['@(x)',fx]);
fp = str2func(['@(x)',fp]);

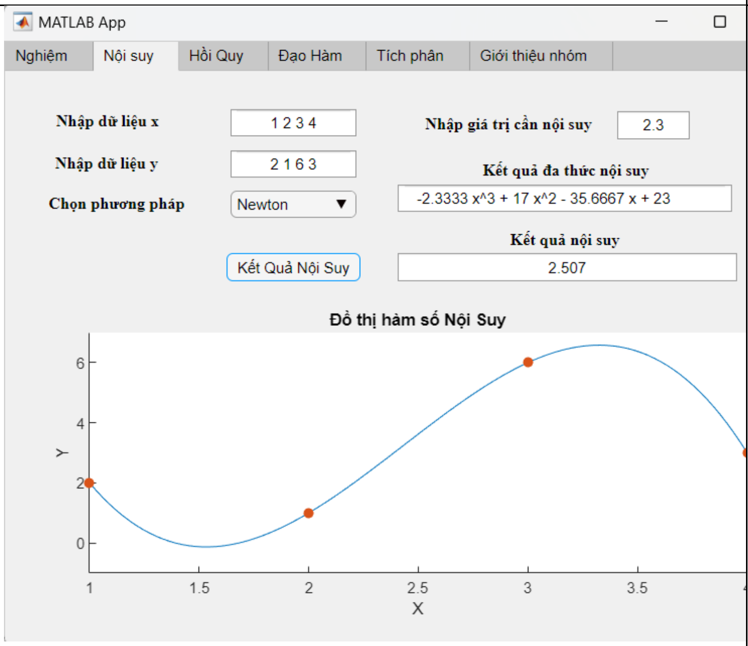
n1 = 0;
c1 = (a+b)/2;

if(fx(a)*fx(c1)<0)
c0 = a;
else
c0 = b;
end
while(1)
c1 = fp(c0);
n1 = n1+1;
e = abs(c1-c0);
%ktra dieu kien
if(e<saiso)
break;
end
%cap nhat c0 moi
c0=c1;</td></tr>
</table>

<table>
<tr><td></td><td></td><td>end

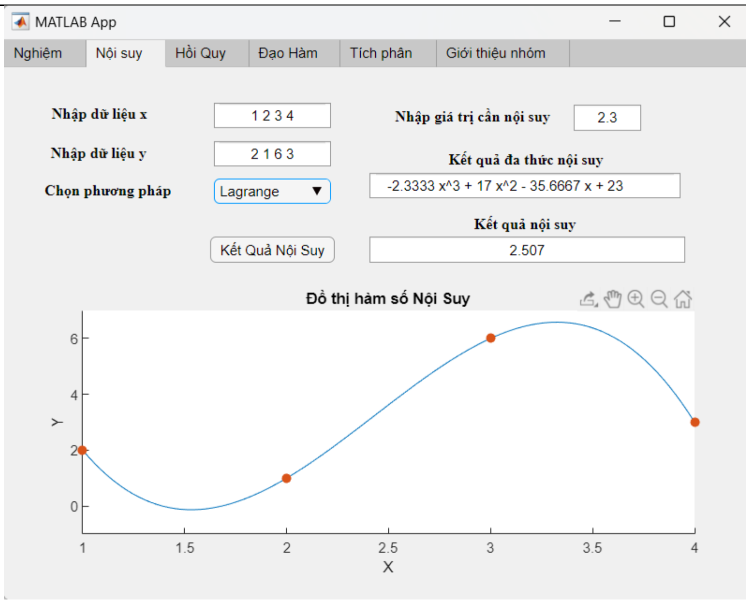
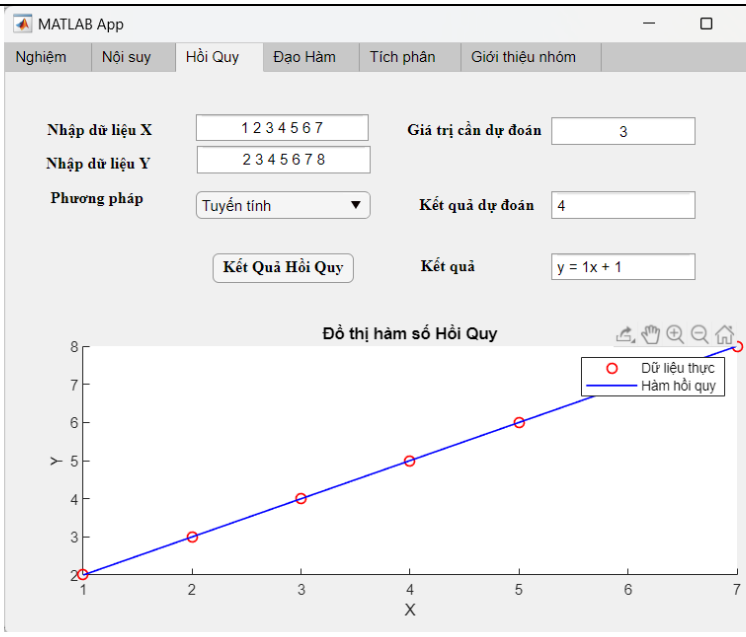
end</td></tr>
<tr><td>Tìm nghiệm (Newton)</td><td>
</td><td>function [c2,n2]=PPNewton(fx, a, b, saiso)
    fx = str2func(['@(x)',fx]);
    syms c2_sym;
    n2 = 0;
    c2 = (a+b)/2;
    if(fx(a)*fx(c2) < 0)
        c2 = a;
    else
        c2 = b;
    end
    while(1)
        df = diff(fx(c2_sym));
        df_func = matlabFunction(df, 'Vars', c2_sym);
        z = fx(c2) / df_func(c2);
        y = c2-z;
        if abs(y - c2) < saiso || n2 > 500
            break;
        end
        c2 = y;
        n2 = n2 + 1;
    end
 end</td></tr>
<tr><td>Tìm nghiệm(dây cung)</td><td>
</td><td>function[c,n]= PPDaycung(fx, a, b, ss)
%fx là phýõng tr?nh ban ð?u
%a, b là kho?ng phân li nghi?m
%ss là sai s?
fx= str2func(['@(x)', fx]);
n=0; %ð?m s? l?n l?p
while(1)
    c=(a*fx(b) - b*fx(a))/(fx(b)-fx(a));
    n= n+1;

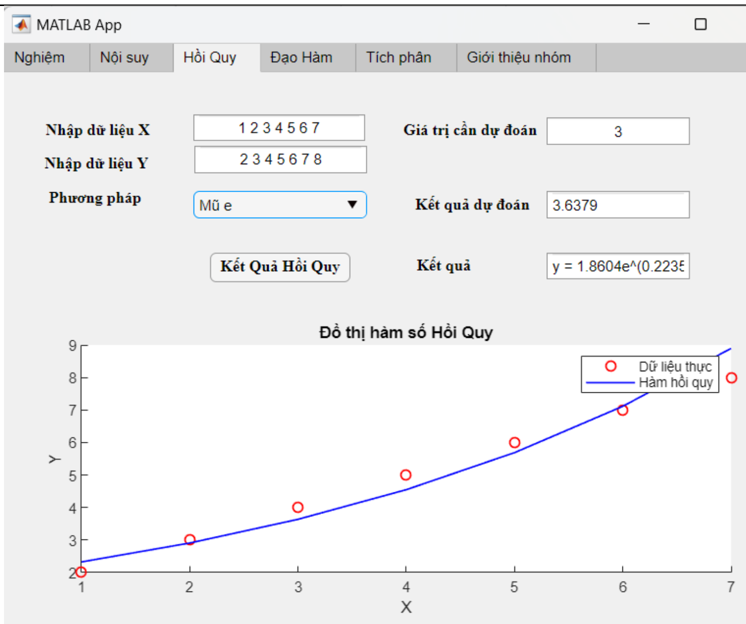
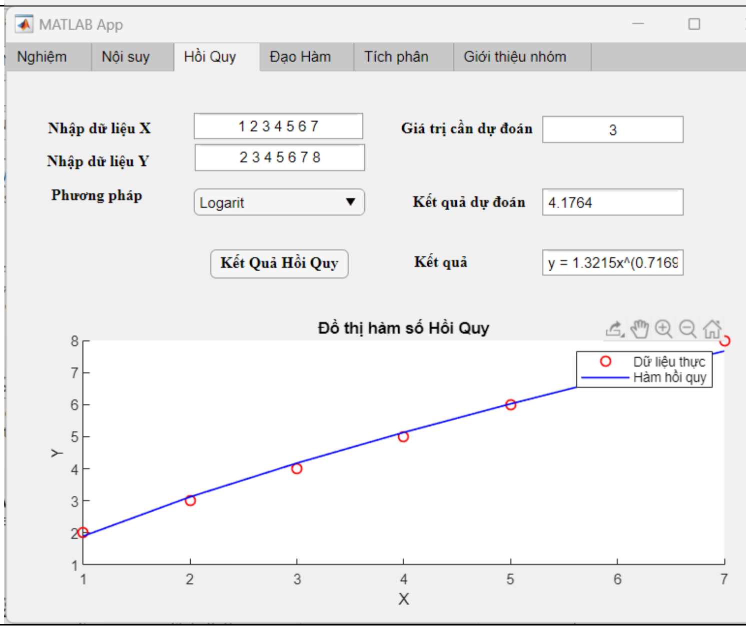
    if abs(fx(c)) < ss
        a=c;
        break;
    end
    if (fx(c) * fx(a) < 0 && fx(b) ~= fx(a))
        b=c;
    else
        a = c;</td></tr>
</table>

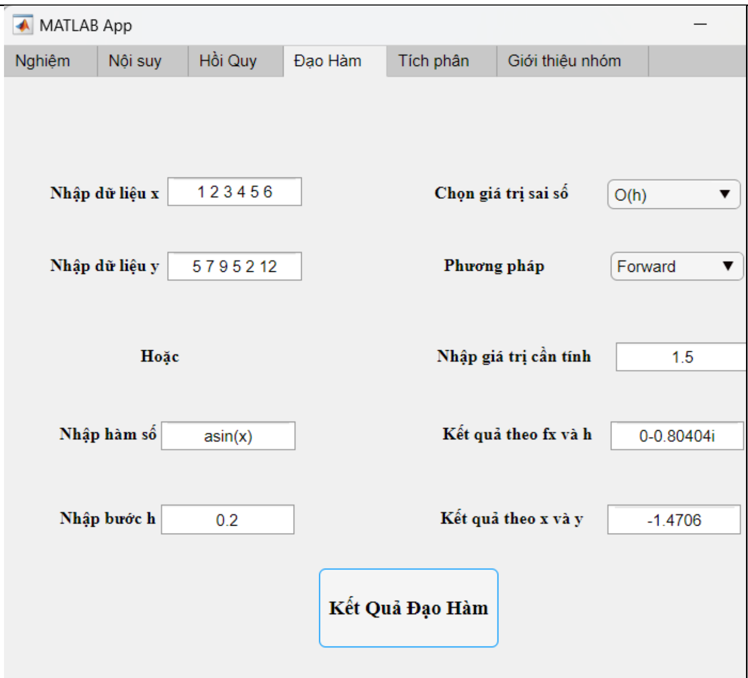
<table>
<tr>
<td></td>
<td></td>
<td>end
e= abs(b - a); %e là sai s?
if(e < ss)
break;
end
end
end</td>
</tr>
<tr>
<td>Nội suy Newton</td>
<td></td>
<td>Gồm 3 thuật toán: thuật toán tính tỉ hiệu,thuật toán các số hạng dạng newton, thuật toán kết hợp thành 1 hàm để tìm giá trị nội suy Newton
function result = NS_NEWTON(xa, ya, x)
d = DividedDifference(xa, ya);
result = NewtonForm(xa, d, x);
end

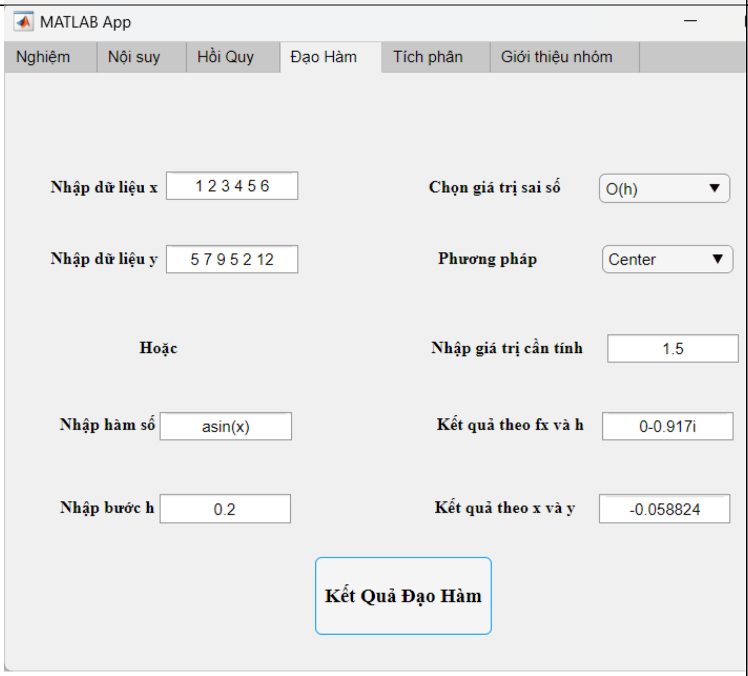
function result = NewtonForm(xa, da, x)
n = length(da);
result = da(n);
for i = n-1:-1:1
result = result .* (x - xa(i)) + da(i);
end
end

function d = DividedDifference(xa, ya)
n = length(xa);
d = ya;
for i = 2:n
for j = n:-1:i
d(j) = (d(j) - d(j-1)) / (xa(j) - xa(j-i+1));
end
end
end</td>
</tr>
</table>

<table>
<tr>
<td>Nội suy Lagrange</td>
<td>
</td>
<td><pre>
function result = NS_LAGRANGE(xa, ya, x)
    n = length(xa) - 1;
    totalSum = 0;
    for i = 1:n+1
        product = 1;
        for j = 1:n+1
            if j ~= i
product = product .* (x - xa(j)) ./ (xa(i) - xa(j));
            end
        end
        totalSum = totalSum + ya(i) * product;
    end
    result = totalSum;
end
</pre></td>
</tr>
<tr>
<td>Hồi quy phi tuyến</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hồi quy tuyến tính dạng $y = ax^b$</td>
<td>
</td>
<td><pre>
function [a0, a1, y_predict] = PPHoiQuyTuyenTinh(x, y, x_predict)
    n = length(x);
    sumxy = 0;
    sumx = 0;
    sumy = 0;
    sumx2 = 0;
    st = 0;
    sr = 0;
    r2 = 0;
    for i = 1:1:n
        sumxy = sumxy + x(i)*y(i);
        sumx = sumx + x(i);
        sumy = sumy + y(i);
        sumx2 = sumx2 + x(i)^2;
    end
    a1 = (n * sumxy - sumx * sumy)/(n * sumx2 - (sumx)^2);
    a0 = mean(y) - a1 * mean(x);
    y_predict = a1 * x_predict + a0;
end
</pre></td>
</tr>
</table>

<table>
<tr>
<td>Hồi quy tuyến tính dạng $y = ae^{bx}$</td>
<td>
</td>
<td>function [A0, A1, y_predict] = PPHoiQuyMuE(x, y, x_predict)
Y = log(y);
X = [ones(length(x), 1), x']
coefficients = X\Y'
A0 = coefficients(1);
A1 = coefficients(2);
y_predict = exp(A0 + A1 * x_predict);
end</td>
</tr>
<tr>
<td>Hồi quy tuyến tính dạng $y = a\frac{x}{b+x}$</td>
<td>
</td>
<td>function [A0,A1,y_predict] = PPHoiQuyLogarit(x,y,x_predict)
Y = log10(y);
X = [ones(length(x), 1), log10(x)'];
coefficients = X\Y';
A0 = coefficients(1);
A1 = coefficients(2);
y_predict = 10^(A0)*x_predict^(A1);
end</td>
</tr>
</table>

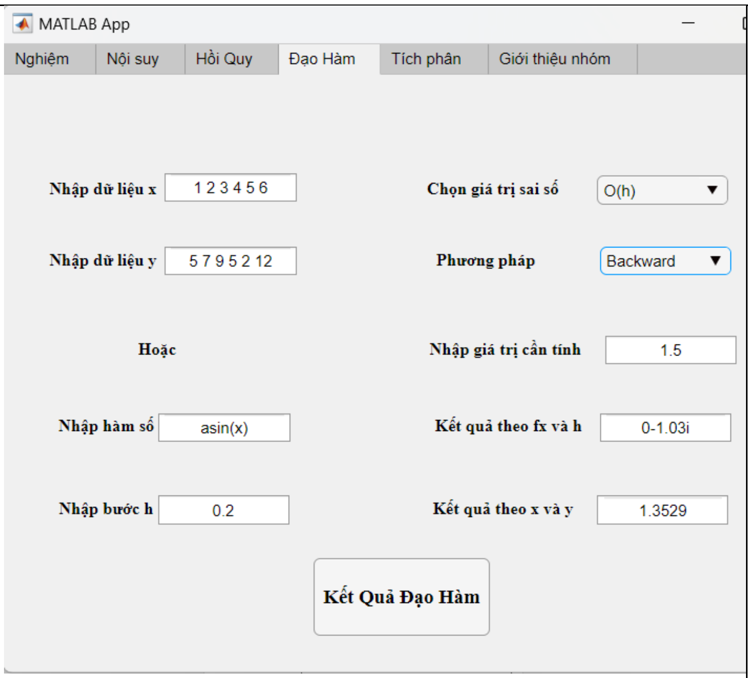
<table>
<tr>
<td>Đạo hàm tiến</td>
<td>


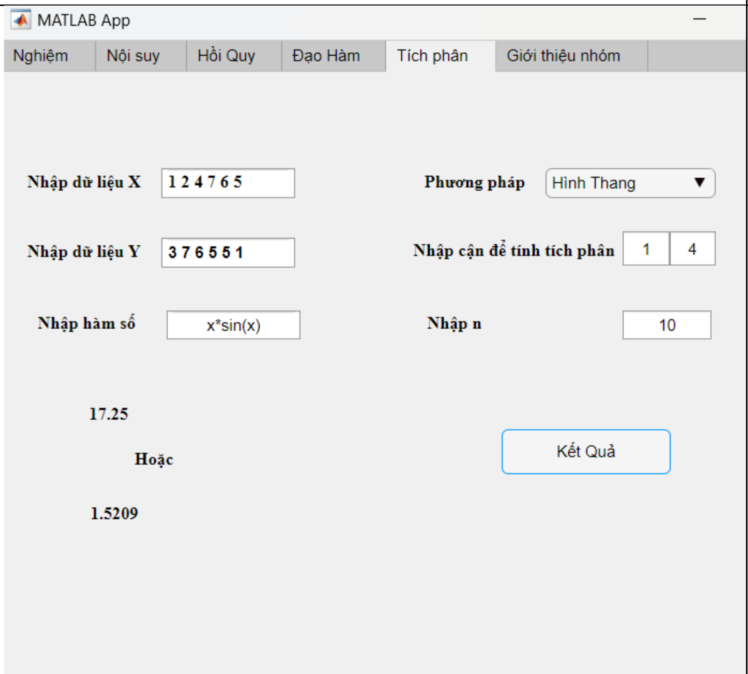
</td>
<td></td>
</tr>
<tr>
<td>Đạo hàm trung tâm</td>
<td>


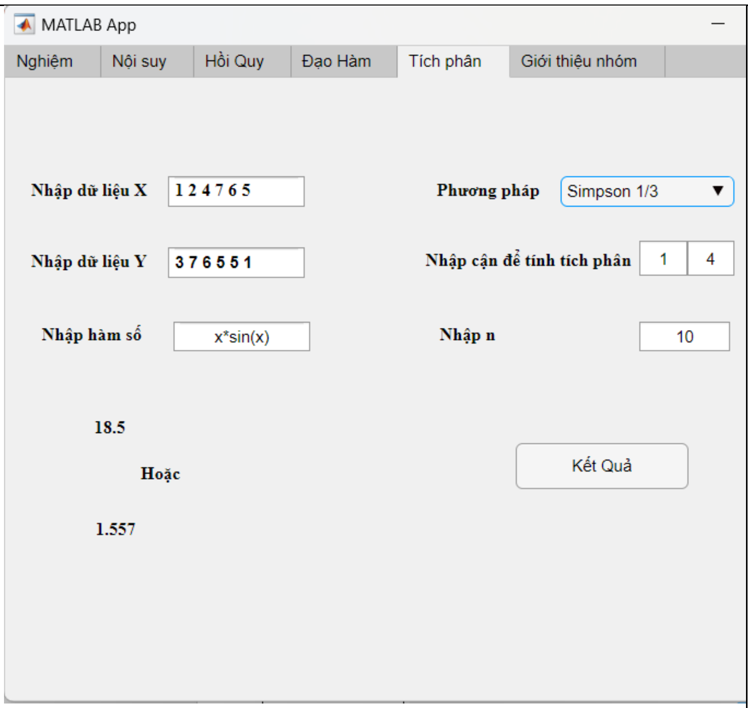
</td>
<td>

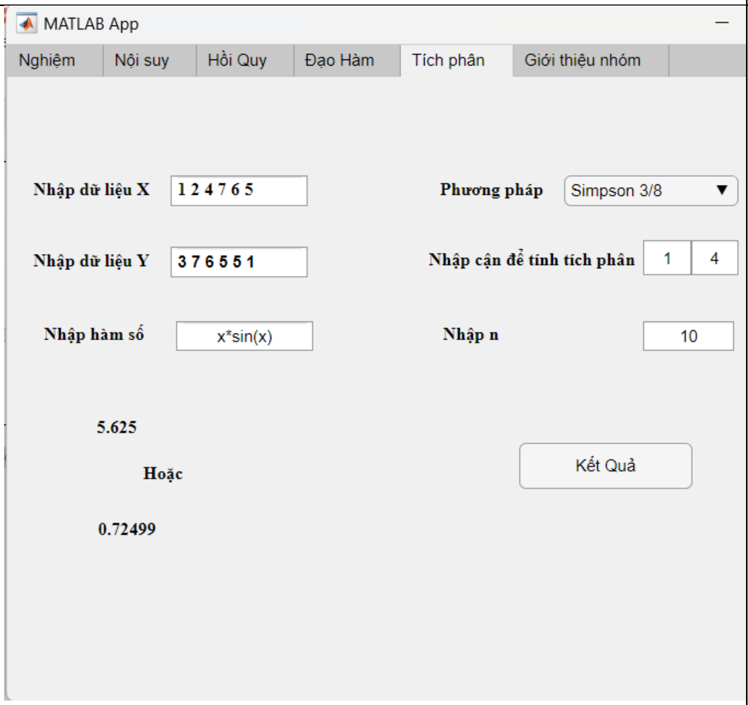
```
function dx = DaoHam(fx, x, h, pp, ss)
    fx = str2func(['@(x)', fx]);
    if pp == "Forward"
        if ss == "O(h)"
            dx = (fx(x + h) - fx(x)) / h;
        elseif ss == "O(h^2)"
            dx = (4 * fx(x + h) - 3 * fx(x) - fx(x + 2 * h)) / (2 * h);
        end
    elseif pp == "Center"
        dx = (fx(x + h) - fx(x - h)) / (2 * h);
    elseif pp == "Backward"
        if ss == "O(h)"
            dx = (fx(x) - fx(x - h)) / h;
        elseif ss == "O(h^2)"
            dx = (3 * fx(x) - 4 * fx(x - h) + fx(x - 2 * h)) / (2 * h);
        end
    end
end
```

</td>
</tr>
</table>

Đạo hàm lùi		function [dxy] = DaoHamXY(xi, yi, x, pp, ss)
		  h = xi(2) - xi(1);
		  a = (x - xi(1)) / h + 1;
		  a = cast(a, 'uint16');
		  disp(a);
		  if pp == "Center"
		      dxy = (yi(a + 1) - yi(a - 1)) / (2 * h);
		  elseif pp == "Forward"
		    if ss == "O(h^2)"
		      dxy = (4 * yi(a + 1) - yi(a + 2) - 3 * yi(a)) / (2 * h);
		    elseif ss == "O(h)"
		      dxy = (yi(a + 1) - yi(a))/h;
		    end
		  else
		    if ss == "O(h^2)"
		      dxy = (-4 * yi(a - 1) + yi(a - 2) + 3 * yi(a)) / (2 * h);
		    elseif ss == "O(h)"
		      dxy = (yi(a) - yi(a - 1))/h;
		    end
		  end
		end

Tích phân hình thang		function result = TichPhanSimpson13XY(x, y, a, b)
		  index_a = find(x >= a, 1);
		  index_b = find(x <= b, 1, 'last');
		  x_selected = x(index_a:index_b);
		  y_selected = y(index_a:index_b);
		
		  N_selected = length(x_selected) - 1;
		  h = (x_selected(end) - x_selected(1)) / N_selected;
		
		  fa = y_selected(1);
		  fb = y_selected(end);
		
		  sum_odd = sum(y_selected(2:2:end-1));
		  sum_even = sum(y_selected(3:2:end-2));
		
		  result = (h / 3) * (fa + 4 * sum_odd + 2 * sum_even + fb);
		end

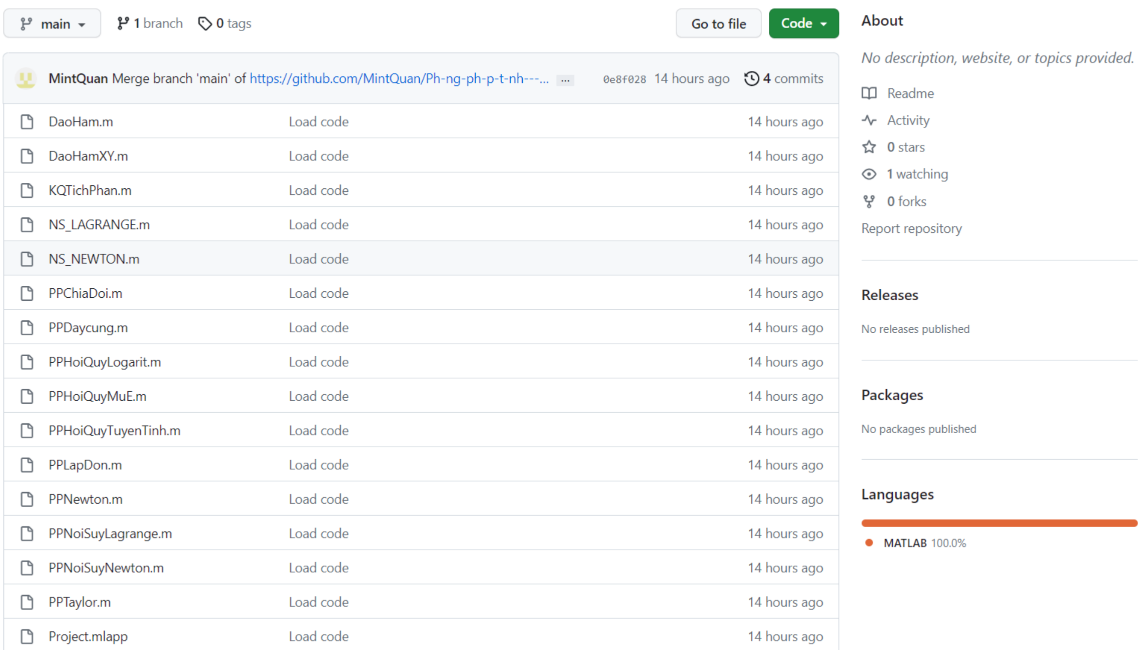
Tích phân Simpson 1/3	
	
		function result = TichPhanSimpson13XY(x, y, a, b)
		    index_a = find(x >= a, 1);
		    index_b = find(x <= b, 1, 'last');
		    x_selected = x(index_a:index_b);
		    y_selected = y(index_a:index_b);
		
		    N_selected = length(x_selected) - 1;
		    h = (x_selected(end) - x_selected(1)) / N_selected;
		
		    fa = y_selected(1);
		    fb = y_selected(end);
		
		    sum_odd = sum(y_selected(2:2:end-1));
		    sum_even = sum(y_selected(3:2:end-2));
		
		    result = (h / 3) * (fa + 4 * sum_odd + 2 * sum_even + fb);
		end

Tích phân Simpson 3/8	
	
		function result = TichPhanSimpson83XY(x, y, a, b)
		    index_a = find(x >= a, 1);
		    index_b = find(x <= b, 1, 'last');
		    x_selected = x(index_a:index_b);
		    y_selected = y(index_a:index_b);
		
		    N_selected = length(x_selected) - 1;
		    h = (x_selected(end) - x_selected(1)) / N_selected;
		
		    fa = y_selected(1);
		    fb = y_selected(end);
		
		    sum_odd = sum(y_selected(2:2:end-1));
		    sum_even = sum(y_selected(3:2:end-2));
		    sum_third = sum(y_selected(4:3:end-2));
		
		    result = (h / 8) * (fa + 3 * sum_odd + 3 * sum_even + 2 * sum_third + fb);
		end

RIÊNG PHẦN ĐẠO HÀM CẢ 3 PHƯƠNG PHÁP ĐỀU CHUNG CODE

III/ Quản Lý Mã Nguồn Github

Nhóm sử dụng công cụ Github để quản lý mã nguồn giúp cho việc quản lý phiên bản mã nguồn trong quá trình xây dựng ứng dụng trở nên dễ dàng và hiệu quả hơn. • Đường dẫn đến Respository của dự án trên GitHub: https://github.com/MintQuan/Project-App-Designer?fbclid=IwAR0-IkCj6HF6b7j-WKiutP3MYnMB4VJ-qj588rIW_0rBIxG1z_anusvrExs



IV/ Hạn chế - Giải pháp

<u>Hạn chế</u>

Chưa nắm rõ được tích phân Simpson 3/8.

Phần đạo hàm khi chọn sai số O(h).

Khi chạy app còn một số vấn đề nhỏ.

<u>Khắc phục</u>

Vì vậy mà giải pháp đưa ra là nhóm cần nắm rõ hơn về lí thuyết cũng như nội dung của phần mà nhóm chưa rõ.

Đã khắc phục được lỗi khi chạy app.

V/ Đánh giá thành viên nhóm

STT	Họ và tên	Phân công	Hoàn thành (*)
1	Nguyễn Phước Bảo (nhóm trưởng)	Viết word, trình bày nội dung, thiết kế giao diện, việt thuật toán	100%
2	Nguyễn Minh Quân	Viết thuật toán, thiết kế giao diện, quản lý github	100%
3	Cao Nguyễn Hải Đăng	Viết thuật toán, thiết kế app	100%
4	Nguyễn Hoàng Bảo	Viết thuật toán, thiết kế app	100%
5	Trương Hoàng Bách Hợp	Viết thuật toán , thiết kế app	100%
6	Lưu Hoàng Đông Anh	Viết thuật toán, viết word, trình bày nội dung	100%

- ✓ Phân công hợp lí và phù hợp với từng thành viên của nhóm.
- ✓ Thành viên trong nhóm hoàn thành tốt nhiệm vụ được giao và đúng thời hạn.
- ✓ Các thành viên tranh luận và đưa ra những giải pháp tốt nhất cho đồ án nhóm.
- ✓ Giúp hiểu rõ hơn về nội dung và tính chất của môn học.

TÀI LIỆU THAM KHẢO

- Giáo trình Phương pháp tính và Matlab – Thầy Nguyễn Xuân Vinh.
- Giáo trình thực hành Phương pháp tính - Khoa Điện Tử - Viễn Thông - Trường Đại học Khoa học tự nhiên - Đại học quốc gia Thành phố Hồ Chí Minh.