ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

----------*----------

KHOA ĐIỆN TỬ VIỄN THÔNG

Chương trình chất lượng cao

Báo cáo thực hành Phương pháp tính

Giảng viên: ThS. Huỳnh Quốc Thịnh

Nhóm: 3 Brothers

STT	MSSV	Họ và tên
1	22207123	Phạm Nguyễn Cao Triều
2	22207094	Huỳnh Phạm Minh Tú
3	22207113	Trương Quang Hưng

TP.HCM, ngày 23 tháng 12 năm 2024

MỤC LỤC

GIỚI THIỆU ĐỀ TÀI

MATLAB là một phần mềm mạnh mẽ trong lĩnh vực toán học, kỹ thuật và khoa học, được sử dụng rộng rãi trong các ứng dụng học thuật và công nghiệp. Một trong những công cụ nổi bật của MATLAB là App Designer, giúp người dùng thiết kế các ứng dụng tương tác một cách trực quan và hiệu quả.

Trong dự án này, nhóm chúng tôi đã phát triển một ứng dụng tích hợp 6 chức năng quan trọng gồm: Tìm nghiệm, Nội suy, Hồi quy, Đạo hàm, Tích phân, và Giới thiệu thành viên. Chức năng Tìm nghiệm hỗ trợ giải phương trình bằng các phương pháp như phương pháp chia đôi, phương pháp lặp và phương pháp Newton (tiếp tuyến) với kết quả hiển thị dưới dạng số và đồ thị trực quan. Còn chức năng Nội suy cho phép người dùng sử dụng hai phương pháp Lagrange và Newton để dự đoán giá trị dựa trên dữ liệu đã có, kết hợp biểu đồ hiển thị rõ ràng. Đối với Hồi quy, ứng dụng hỗ trợ phân tích mối quan hệ giữa các biến thông qua hồi quy tuyến tính và phi tuyến, giúp xây dựng các mô hình dự đoán với kết quả dưới dạng công thức hoặc đồ thị. Ngoài ra chức năng Đạo hàm được thiết kế để tính đạo hàm của hàm số bằng các phương pháp số học hoặc ký hiệu, hỗ trợ tính toán bậc đạo hàm và giá trị tại điểm cụ thể. Trong khi đó, chức năng Tích phân hỗ trợ giải các bài toán tích phân xác định và bất định với các phương pháp như hình thang và Simpson's 1/3, Simpson's 3/8. Ngoài các chức năng toán học, ứng dụng còn tích hợp mục Giới thiệu thành viên, cung cấp thông tin về các thành viên phát triển dự án như tên, vai trò và đóng góp nhằm thể hiện sự cống hiến của các cá nhân trong nhóm.

Ứng dụng này không chỉ hỗ trợ người dùng giải quyết các bài toán phức tạp một cách nhanh chóng mà còn có thể sử dụng trong giảng dạy để minh họa các khái niệm toán học, giúp tiết kiệm thời gian và nâng cao hiệu suất làm việc.

Nhóm phát triển gồm các thành viên:

- Huỳnh Phạm Minh Tú 22207094 (Tổng hợp, Chức năng "Hồi quy" và "Giới thiệu")
- Phạm Nguyễn Cao Triều 22207123 (Chức năng "Tích phân" và "Đạo hàm")
- Trương Quang Hưng 22207113 (Chức năng "Tìm nghiệm" và "Nội suy")

Với sự tích hợp 6 chức năng hữu ích trên, ứng dụng này được kỳ vọng sẽ trở thành công cụ hỗ trợ đắc lực trong học tập và nghiên cứu, đồng thời rất mong nhận được sự góp ý từ mọi người để tiếp tục hoàn thiện trong tương lai.

I. Giải gần đúng phương trình

1. Lý thuyết

Trong kỹ thuật, ta thường gặp các bài toán tìm nghiệm của phương trình f(x) = 0. Thông thường kết quả nghiệm tìm được thường khó có thể ở dạng nghiệm đúng hoàn toàn vì trong máy tính giá trị tìm được luôn bị quy tròn số. Để tìm nghiệm gần đúng của phương trình f(x) = 0 ta tiến hành như sau:

1. Khẳng định phương trình có nghiệm trong một khoảng nào đó
2. Chọn xấp xỉ x_0 – nghiệm gần đúng đầu tiên thuộc khoảng đang xét
3. Xây dựng thuật toán điều chỉnh dần x_0 sao cho càng gần tới nghiệm đúng càng tốt
4. Đánh giá sai số của nghiệm gần đúng tìm được so với nghiệm đúng

Hiện nay có rất nhiều phương pháp được sử dụng để tối ưu khả năng xử lí thuật toán nhưng trong đồ án lần này, nhóm xin được phép thực hiện 3 phương pháp sau:

- Phương pháp chia đôi
- Phương pháp lặp đơn
- Phương pháp Newton (tiếp tuyến)

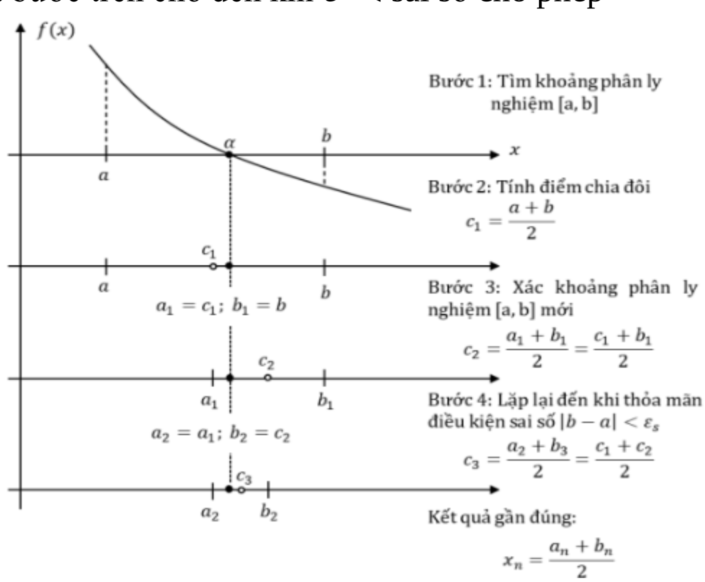
2. Minh họa thuật toán

a. Phương pháp chia đôi

Phương pháp chia đôi là phương pháp xấp xỉ để tìm kiếm phương pháp đã được chọn bằng cách chia khoảng liên tục. Phương pháp này sẽ chia khoảng cho đến khi tìm được kết quả khoảng, khoảng này cực kỳ nhỏ.

Đối với bất kỳ hàm liên tục $f(x)$:

- Xác định khoảng phân ly, tìm hai điểm a và b sao cho $a < b$ và $f(a) * f(b) < 0$
- Tìm trung điểm của a và b là c_1
- Xác định khoảng phân ly mới $[a\ b]$ – Nếu $f(x) * f(a) < 0$ thì tồn tại một nghiệm giữa x và a, ngược lại nếu $f(x) * f(b) < 0$ thì tồn tại một trải nghiệm giữa x và b
- Lặp lại ba bước trên cho đến khi $\varepsilon <$ sai số cho phép



b. Phương pháp lặp đơn

Phương pháp lặp là phương pháp thực hiện lặp lại cách tính sao cho tồn tại:

$$\lim_{n\to\infty} x_n = \overline{x}$$

Phương pháp này có giá trị xấp xỉ khi quá trình tính toán dừng ở bước thứ n (không thể kéo dài vô hạn), lúc này ta xem $x_n \approx \overline{x}$ là nghiệm gần đúng của phương trình (tức là có sai số).

Giả sử $[a, b]$ là khoảng phân ly nghiệm α của phương trình

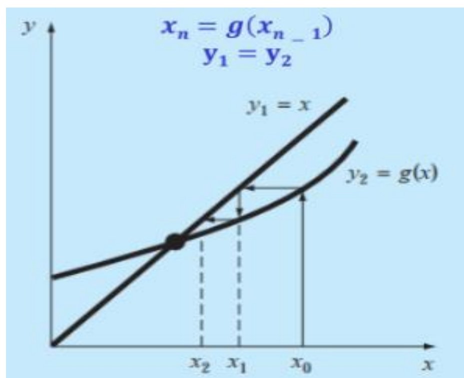
$$f\ (\alpha)\ =\ 0, \qquad \alpha\ \in\ [a, b]$$

Viết phương trình lại dưới dạng: $x\ =\ g(x)$ trong đó $g(x)$ liên tục trong $[a, b]$

Chọn x0 là điểm bất kỳ trong $[a, b]$. Tính dãy lặp theo công thức (cho đến khi sai số thỏa mãn)

$$x_n\ =\ g(x_n - 1), \qquad n\ =\ 1,2, \ldots, k$$

Giả thuyết tồn tại số $q\ >\ 0$ sao cho $g\ '(x)\ \leq\ q\ <\ 1\ \forall\ x\ \in\ [a, b]$, khi đó quá trình lặp hội tụ.

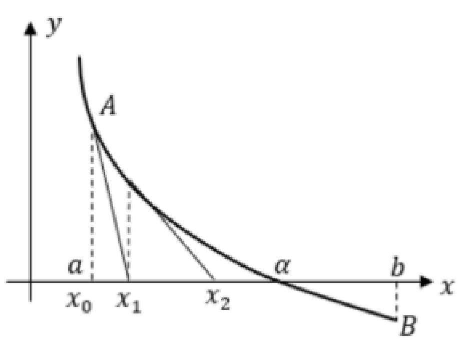


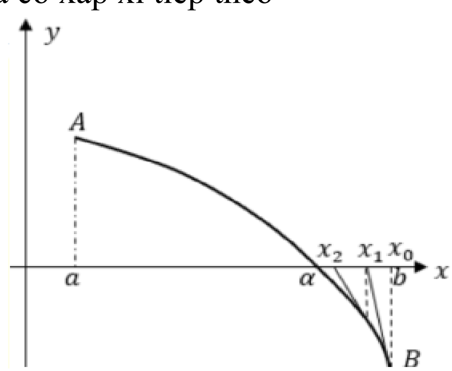
c. Phương pháp Newton (tiếp tuyến)

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. Thay đường cong $y\ =\ f(x)$ bằng tiếp tuyến kẻ từ A$(a, f(a)$) hay $B(b, f(b))$ coi hoành độ giao điểm của tiếp tuyến và Ox là nghiệm gần đúng. Đặt $x_0 = a$ nếu tiếp tuyến kẻ từ A, $x_0 = b$ nếu kẻ từ B. Phương trình tiếp tuyến tại $(x_0, f\ (x_0)$) có dạng:

$$y = f(x_0) + f'(x_0)(x1 - x0) = 0 => x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Tiếp tục vẽ tiếp tuyến tại $(x1, f(x1)$), tương tự ta có xấp xỉ tiếp theo





3. Mô tả hoạt động
   a. Phương pháp chia đôi

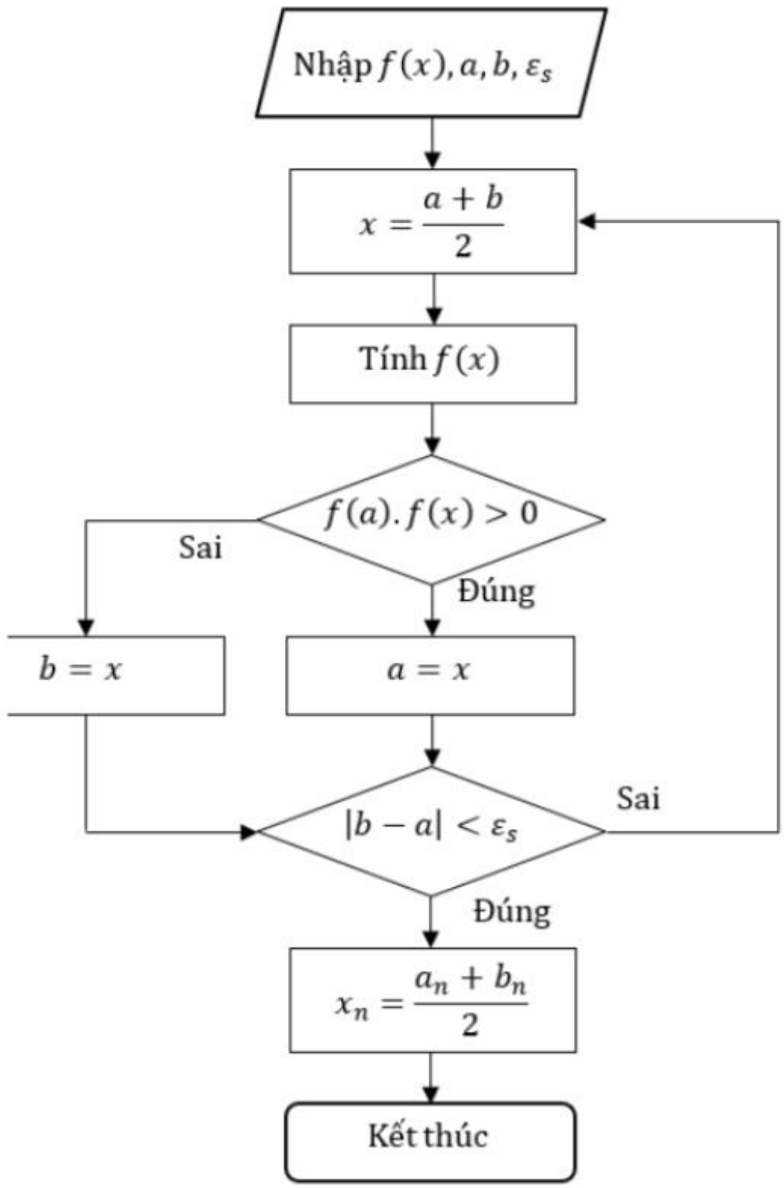
Bước 1 : Nhập phương trình cần tìm , Cận biên trái (a) và biên phải (b) , sai số cho phép của phương pháp.

Bước 2 : Hệ thống sẽ bắt đầu xét trung điểm giữa a và b để bắt đầu tìm điểm chia đôi là x

Bước 3 : Tính f(x) và xét dấu của f(x). Nếu f(x) và f(a) cùng dấu thì a=x , Ngược lại nếu f(x) và f(b) cùng dấu thì b=x.

Bước 4 : Lặp lại đến khi thỏa mãn điều kiện sai số cho phép của phương pháp : $|b-a|<\varepsilon$

Bước 5 : Nếu đã thỏa sai số cho phép , xuất ra x(n) tại vị trí a(n) và b(n)



b. Phương pháp lặp

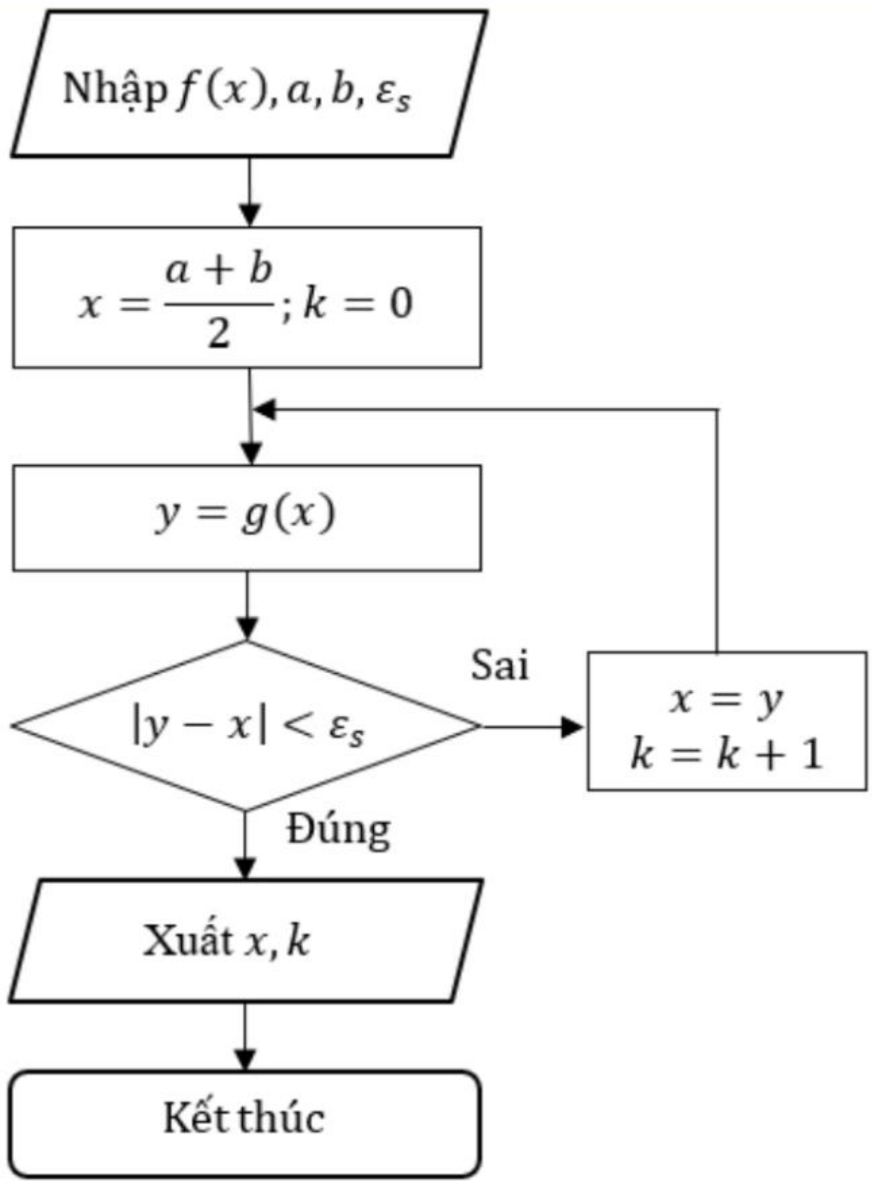
Bước 1 : Chuyển $f(x) = 0$ về dạng $x = g(x)$;

Bước 2 : Kiểm tra điều kiện $|g'(x)| \le q < 1$

Bước 3 : Chọn xấp xỉ ban đầu $x_0 \in [a, b]$

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

Sau k lần lặp , được $x_n \approx a$ là nghiệm gần đúng phương trình



c. Phương pháp tiếp tuyến

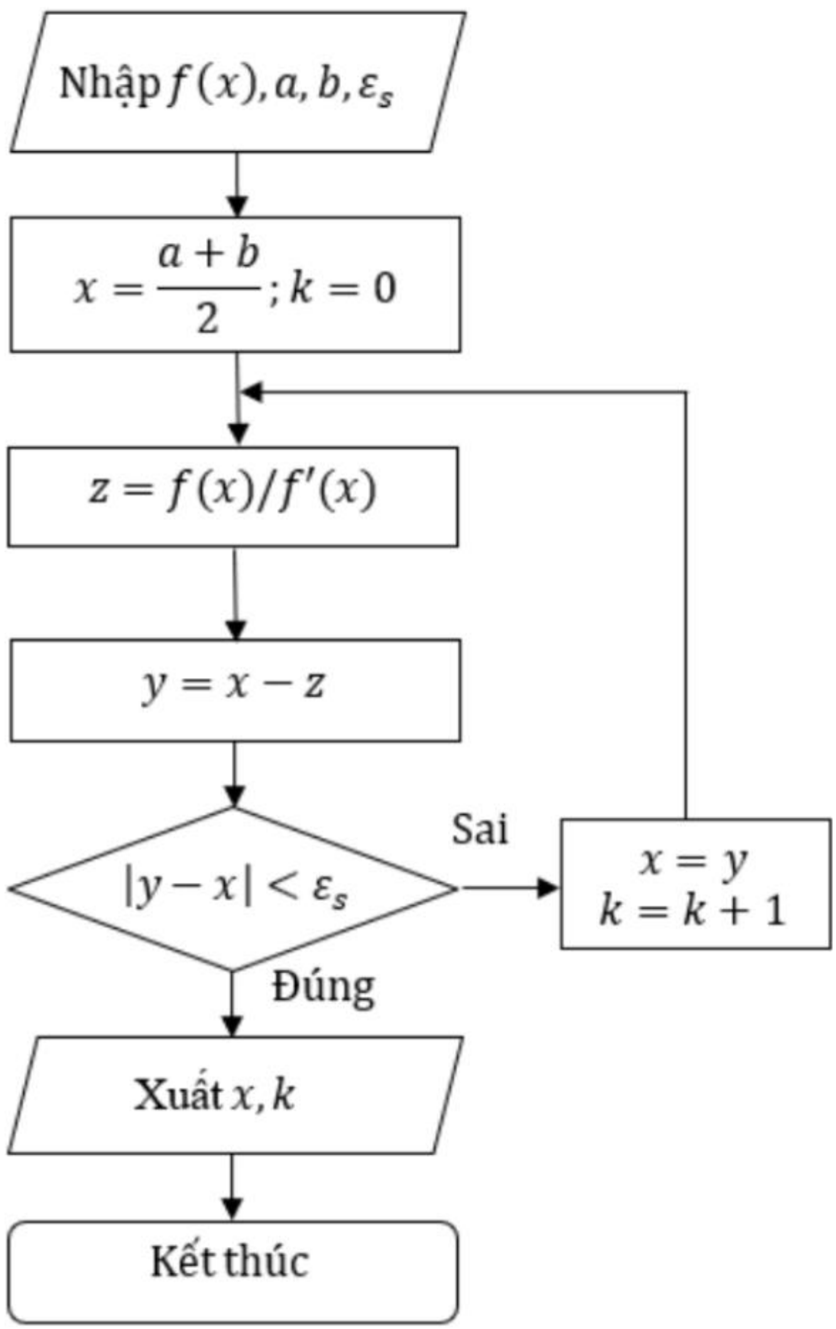
Bước 1 : Tính $f'(x), f''(x)$ và xét dấu . Điều kiện để hội tụ $f'(x)$
Và $f''(x)$ không đổi dấu trên $[a, b]$

Bước 2 : Chọn x_0 và a hay b sap cho $f(x_0)$ cùng dấu với $f''(x)$

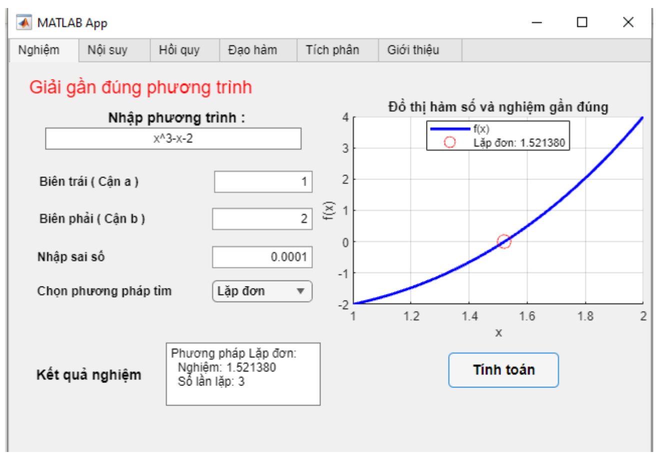
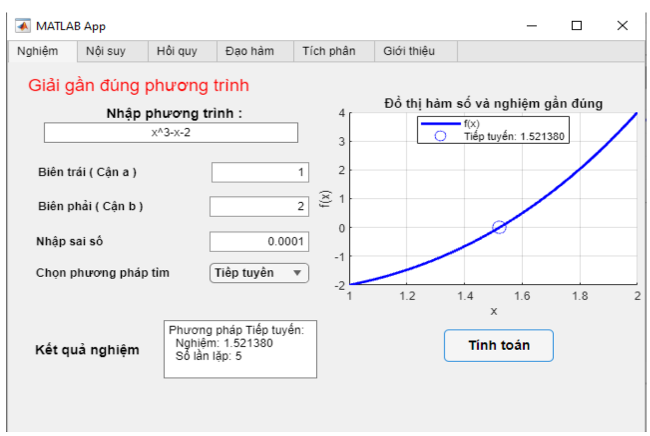
Bước 3 : Từ xấp xỉ ban đầu x_0 , Tính

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

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



4. Minh họa và kiểm tra kết quả cho từng phương pháp

Phương pháp	Kiểm nghiệm
Chia đôi	
Lặp	
Tiếp tuyến	

5. Vấn đề hạn chế và hướng giải quyết
   - Thuật toán còn chưa tối ưu , ở phương pháp lặp và chia đôi đôi lúc chương trình tính mất thời gian.
   - Với các hàm phức tạp , đồ thị hàm số chỉ vẽ được 1 phần khoảng phân ly đúng như hàm nhập vào. Không thể hiện trực quan cả một biểu đồ của hàm.
   - Hướng giải quyết : Tối ưu thuật toán và hạn chế các bước khi chuyển các function của hàm. Sử dụng thêm các Call back để giảm tải công việc để tính toán nghiệm cuối cùng của phương trình.

II. Nội suy

1. Lý thuyết

   Nội suy Lagrange là một phương pháp trong phân tích số dùng để tìm một đa thức đi qua một tập hợp các điểm cho trước. Phương pháp này đặc biệt hữu ích khi cần xây dựng một đa thức xấp xỉ giá trị của một hàm số tại các điểm trung gian.

   Cho n+1 điểm dữ liệu $(x_0 y_0)$, $(x_1 y_1)$, ... , $(x_n y_n)$. Trong đó $y_i = f(x_i)$ với i =0,1,...,n. Đa thức nội suy Lagrange được biểu diễn như sau

$$P(x) = \sum_{i=0}^{n} y_i \,.\, L_i(x)$$

   Trong đó $L_i(x)$ là các đa thức cơ sở Lagrange , được định nghĩa là

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

   Đa thức nội suy Newton được biểu diễn qua công thức sau:

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0) + .. + a_n(x - x_{n-1})$$

   Trong đó là các hệ số a_0, a_1, a_2, ... tính bằng phương pháp sai phân tiến hoặc lùi dựa trên giá trị tại các nút dữ liệu.

2. Minh họa thuật toán

   Cho một bài toán nội suy có a =x_0 < x_1< … <x_n = b

   Đa thức nội suy Largrange $P_n(x)$ có dạng:

$$P_n(x) = \sum_{j=0}^{n} y_j \, L_j(x)$$

   Với

$$L_j(x) = \frac{(x - x_0)(x - x_1)..(x - x_{j-1})(x - x_{j+1})..(x - x_n)}{(x_j - x_0)(x_j - x_1)..(x_j - x_{j-1})(x_j - x_{j-1})..(x_j - x_n)}$$

   Đa thức nội suy Largrange có tính chất $L_j(x) = \begin{cases} 1 \; khi \; i = j \\ 0 \; khi \; i \neq j \end{cases}$ để $P_n(x_i) = y_j$

Lúc này đa thức nội suy Newton tiến có dạng:

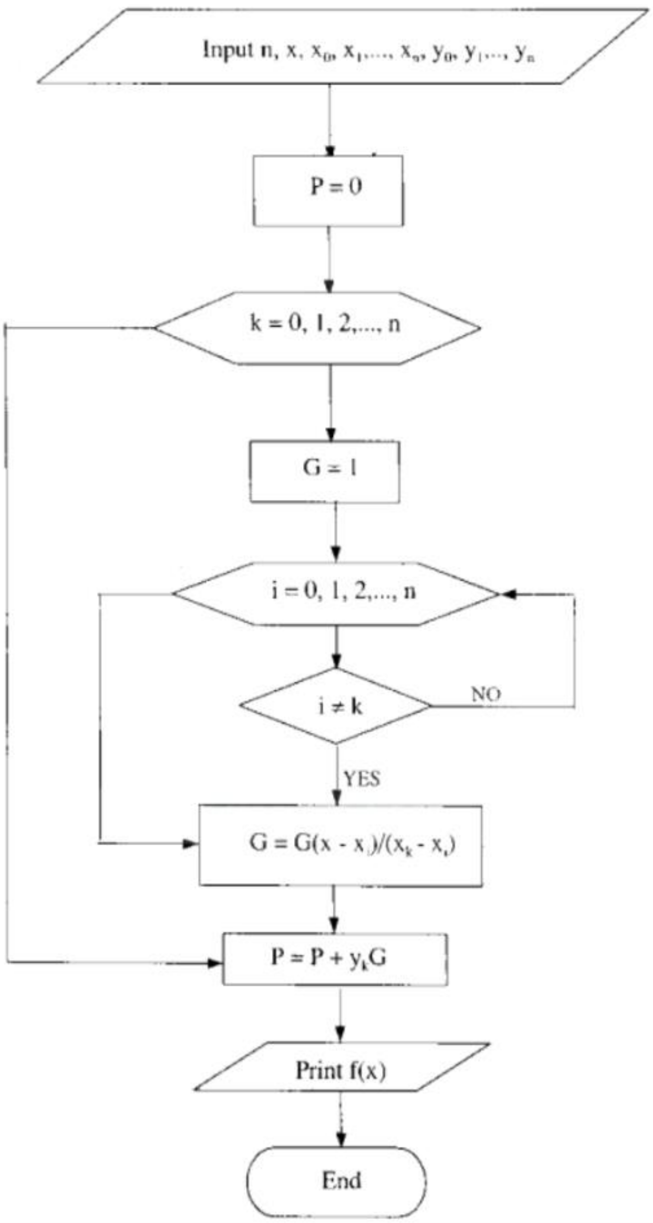
$$P_n(x) = P_n(x_0 + ht) = y_0 + \frac{t}{1!}\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + .. + \frac{t(t-1)(t-n+1)}{n!}\Delta^n y_0$$

Còn đa thức nội suy Newton lùi có dạng:

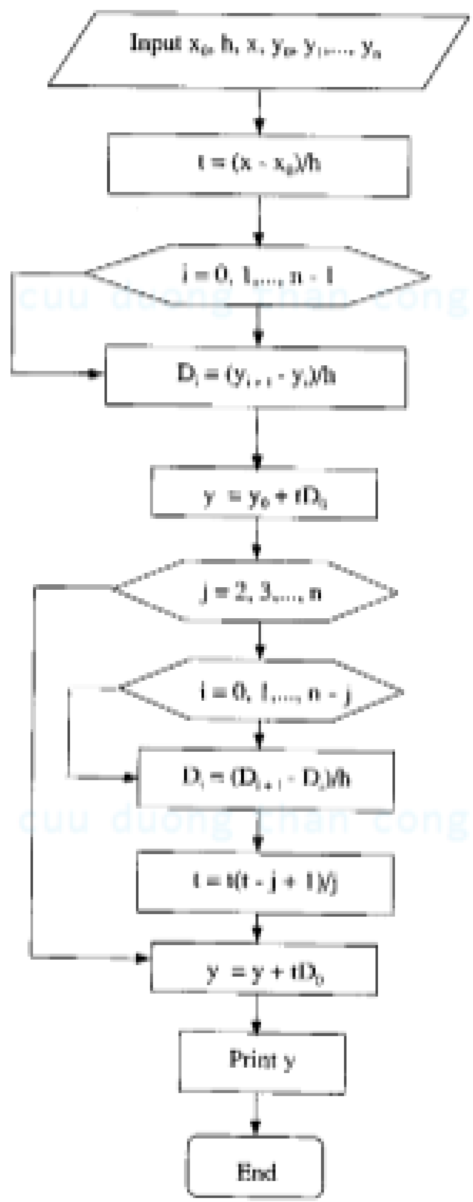
$$P_n(x) = P_n(x_0 + ht) = y_n + \frac{t}{1!}\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + .. + \frac{t(t+1)(t+n-1)}{n!}\Delta^n y_0$$

Với $t = \frac{\overline{x} - x_0}{h}$ ($\overline{x}$ là điểm cần xác định $f(\overline{x})$)

3. Mô tả hoạt động



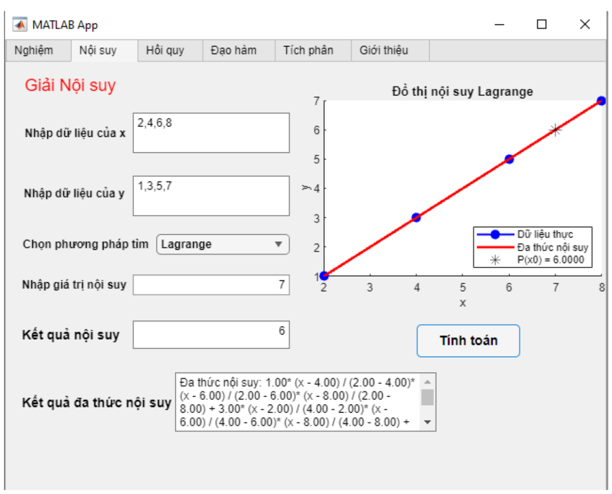
Nội suy Largrange

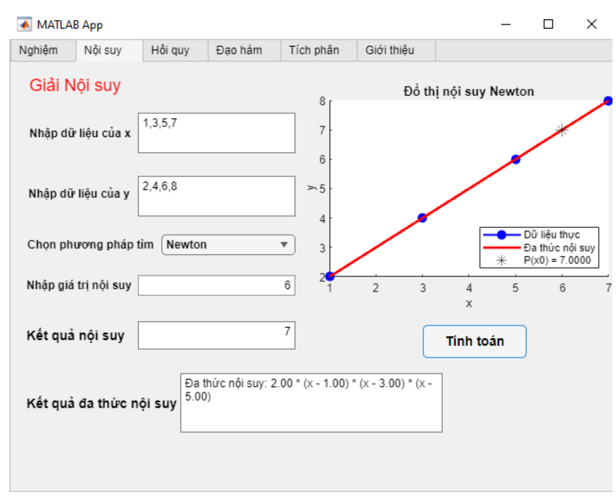


Nội suy Newton

- Khi người dùng nhập dữ liệu vào các ô , chương trình sẽ tiến hành kiểm tra dữ liệu . Nếu hợp lệ sẽ tiếp tục chạy chương trình . Nếu không sẽ xuất hiện dòng : Hàm không hợp lệ
- Nếu người dùng xác định số x,y không khớp thì chương trình sẽ xuất ra Lỗi: Không lấy được dữ liệu từ giao diện, để nhập lại
- Nếu nhập giá trị nội suy không nằm trong khoảng x có sẵn thì chương trình sẽ báo Lỗi: x0 (...) nằm ngoài phạm vi [x, y] của x
- Nếu dữ liệu gốc nhập vào đã đúng theo mẫu , chương trình sẽ chuyển các hàm thành các hàm ẩn danh trong MATLAB
- Tùy thuộc vào phương pháp người dùng chọn : Hệ thống sẽ tự lọc theo Switch case và đưa đúng dạng function tính của phương pháp đó .
- Lưu giá trị kết quả và xuất ra đa thức của phương pháp. Dùng chính giá trị kết quả này để vẽ biểu đồ theo đúng khoảng phân [x,y] từ người nhập

4. Minh họa và kiểm tra kết quả cho từng phương pháp

<table>
<tr><th>Nội suy</th><th>Largrange</th><th>Newton</th></tr>
<tr><td>Kiểm nghiệm</td><td>


</td><td>


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

5. Vấn đề hạn chế và hướng giải quyết
   - Thuật toán còn chưa tối ưu , ở phương pháp Newton đôi lúc chương trình tính mất thời gian. Chưa xác định rõ dạng sử dụng là Newton tiến, lùi hay trung tâm
   - Hướng giải quyết : Tối ưu thuật toán và hạn chế các bước khi chuyển các function của hàm. Sử dụng thêm các Call back để giảm tải công việc để tính toán nghiệm cuối cùng của phương trình.

III. Hồi quy

1. Lý thuyết

a. Hồi quy tuyến tính

Hồi quy tuyến tính sử dụng mô hình đường thẳng $y = a_0 + a_1 \times x$ 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: $e = y - a_0 - a_1 \times x$

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

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_{i(\text{đo được})} - y_{i(\text{mô hình})})^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 \times x_i)^2$$

Tìm giá trị cực tiểu của S_r phải tìm các giá trị đạo hàm riêng và giải hệ phương trình các đạo hàm riêng:

$$\begin{cases} \dfrac{\partial S_r}{\partial a_0} = 0 \\ \dfrac{\partial S_r}{\partial a_1} = 0 \end{cases} \rightarrow \begin{cases} -2\sum(y_i - a_0 - a_1 \times x_i) = 0 \\ -2\sum[(y_i - a_0 - a_1 \times x_i)x_i] = 0 \end{cases}$$

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

Kết quả thu được hệ số của phương trình như sau:

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

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

Hệ số tương quan r^2, gọi S_r là tổng bình phương các giá trị lỗi giữa các giá trị đo được và giá trị của mô hình, 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:

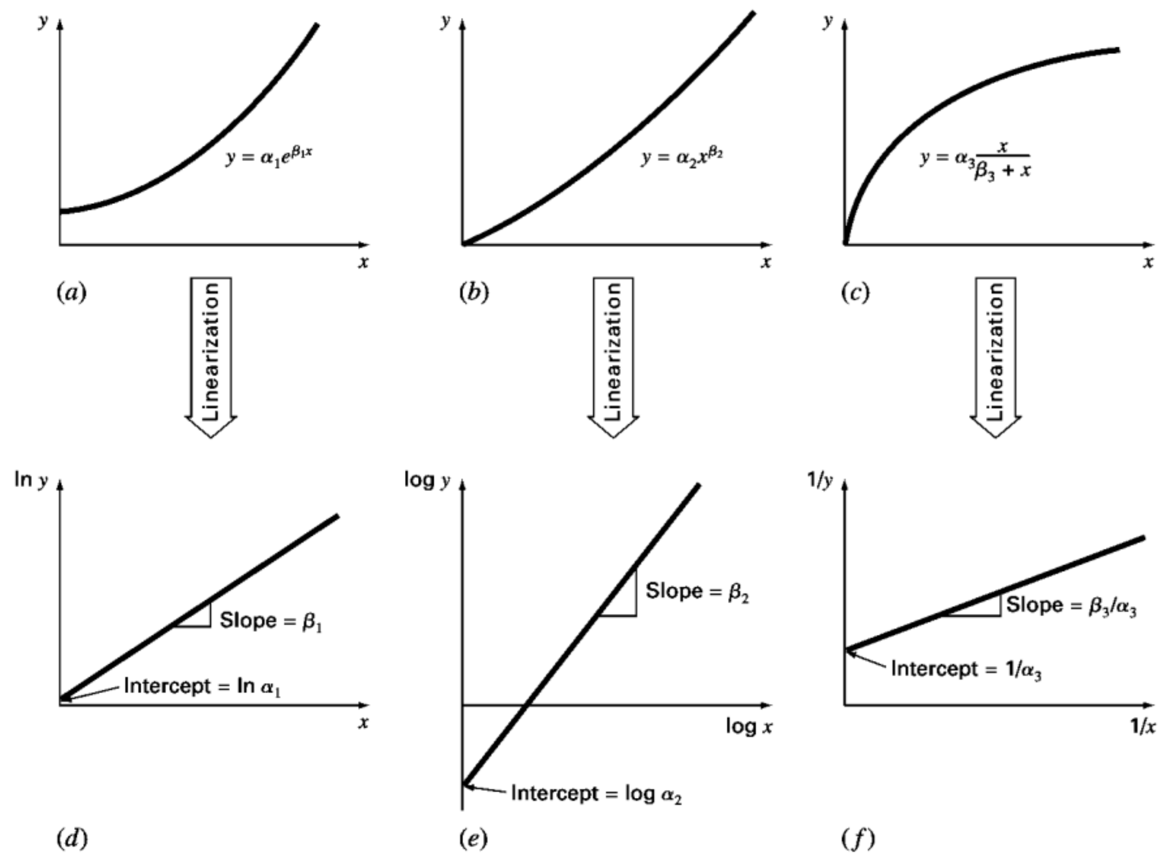
$$r^2 = \frac{S_t - S_r}{S_t}$$

Hệ số tương quan r^2 thể hiện kết quả mô hình hồi quy khớp hay chưa khớp, giá trị càng gần 1 thì kết quả hồi quy càng khớp.

b. 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.

Một số ví dụ:



Trong đồ án áp dụng dạng hàm số mũ và logarit cho phương pháp tuyến tính.

b.1) Hàm mũ $y = ae^{bx}$:

Để hồi quy hàm mũ dạng này, cần thực hiện:

B1: lấy logarit cơ số e 2 vế: $lny = lna + bx$

B2: Đặt Y = lny, $A_o = lna$, $X = x$, $A_1 = b$

B3: đưa về dạng: $Y = A_0 + A_1x$

B4: áp dụng phương pháp hồi quy phi tuyến tìm $A_0, A_1 \rightarrow a = e^{Ao}; b = A_1$

→ Do đó ở dạng này tất cả các giá trị của chuỗi dữ y phải lớn hơn 0.

b.2) Hàm Logarit $y = ax^b$:

B1: lấy logarit cơ số 10 2 vế: $lgy = lga + blgx$

B2: Đặt Y = lgy, $A_o = lga$, $X = lgx$, $A_1 = b$

B3: đưa về dạng: $Y = A_0 + A_1x$

B4: áp dụng phương pháp hồi quy phi tuyến tìm $A_0, A_1 \rightarrow a = 10^{Ao}; b = A_1$

→ Do đó ở dạng này tất cả các giá trị của chuỗi dữ x và y phải lớn hơn 0.

2. Minh họa thuật toán

Mô hình áp dụng hàm Function hồi quy tuyến tính cho cả 3 trường hợp, chỉ khác ở khâu xử lí dữ liệu trước khi truyền dữ liệu vào hàm.

a) Hàm function HoiQuy:

Code	**Mô tả**
function [a0, a1, r2] = HoiQuy(mangX, mangY)	- Khai báo hàm HoiQuy
n = length(mangX);	- Lấy kích thước chuỗi x
sumX = 0;	- Khởi tạo các tham số tính toán:
sumY = 0;	+ Tổng chuỗi x, tổng chuỗi y
sumXY = 0;	+ Tổng tích xy
sumX2 = 0;	+ Tổng các bình phương chuỗi x
st = 0;	- Khởi tạo giá trị st
sr = 0;	- Khởi tạo giá trị sr
for i = 1:n	- Tính các tham số:
sumX = sumX + mangX(i);	+ Tổng chuỗi x, tổng chuỗi y
sumY = sumY + mangY(i);	+ Tổng tích xy
sumXY = sumXY + mangX(i)*mangY(i);	+ Tổng các bình phương chuỗi x
sumX2 = sumX2 + mangX(i)*mangX(i);	
end	
xm = sumX/n;	- Tính giá trị trung bình chuỗi x
ym = sumY/n;	- Tính giá trị trung bình chuỗi y
a0 = (n*sumXY - sumX*sumY)/(n*sumX2-sumX*sumX);	- Tính hệ số a0
a1 = ym - a0*xm;	- Tính hệ số a1
for i = 1:n	- Tính st và sr
st = st + (mangY(i)-ym)^2;	
sr = sr + (mangY(i) - a0*mangX(i) - a1)^2;	
end	
r2 = (st - sr)/st;	- Tính hệ số tương quan
end	

b) Xử lí dữ liệu truyền hàm và kết quả:

Code	**Mô tả**
if(c == "Tuyến tính")	- Chọn phương pháp tuyến tính
[a0, a1, r2] = HoiQuy(x, y);	- Tính giá trị a0, a1 và r2
app.rm.Value = sprintf('%.3f + %.3f*x', a1, a0);	- In hàm tuyến tính lên app.rm
app.rv.Value = sprintf('r2 = %.6f', r2);	- In giá trị r2 lên app.rv
yhq = a1 + a0.*x;	- Tính giá trị yhq (y hồi quy) để vẽ đồ thị đường tuyến tính
end	

Code	**Mô tả**

if(c == "Logarit")
    X = log10(x);
    Y = log10(y);
    [a0, a1, r2] = HoiQuy(X, Y);
    a = 10^a1;
    b = a0;
    app.rm.Value = sprintf('%.3f*x.^%.3f', a, b);
    app.rv.Value = sprintf('r2 = %.6f', r2);
    yhq = a*x.^b;
end
	- Chọn Logarit
	- Chuyển chuỗi x và chuỗi y về dạng logarit cơ số 10
	- Tính a0, a1 và r2
	- Đưa a1 về dạng trước khi truyền
	- Gán a0 vào b
	- In hàm logarit lên thanh app.rm
	- In giá trị r2 lên app.rv
	- Tính yhq để vẽ đồ thị

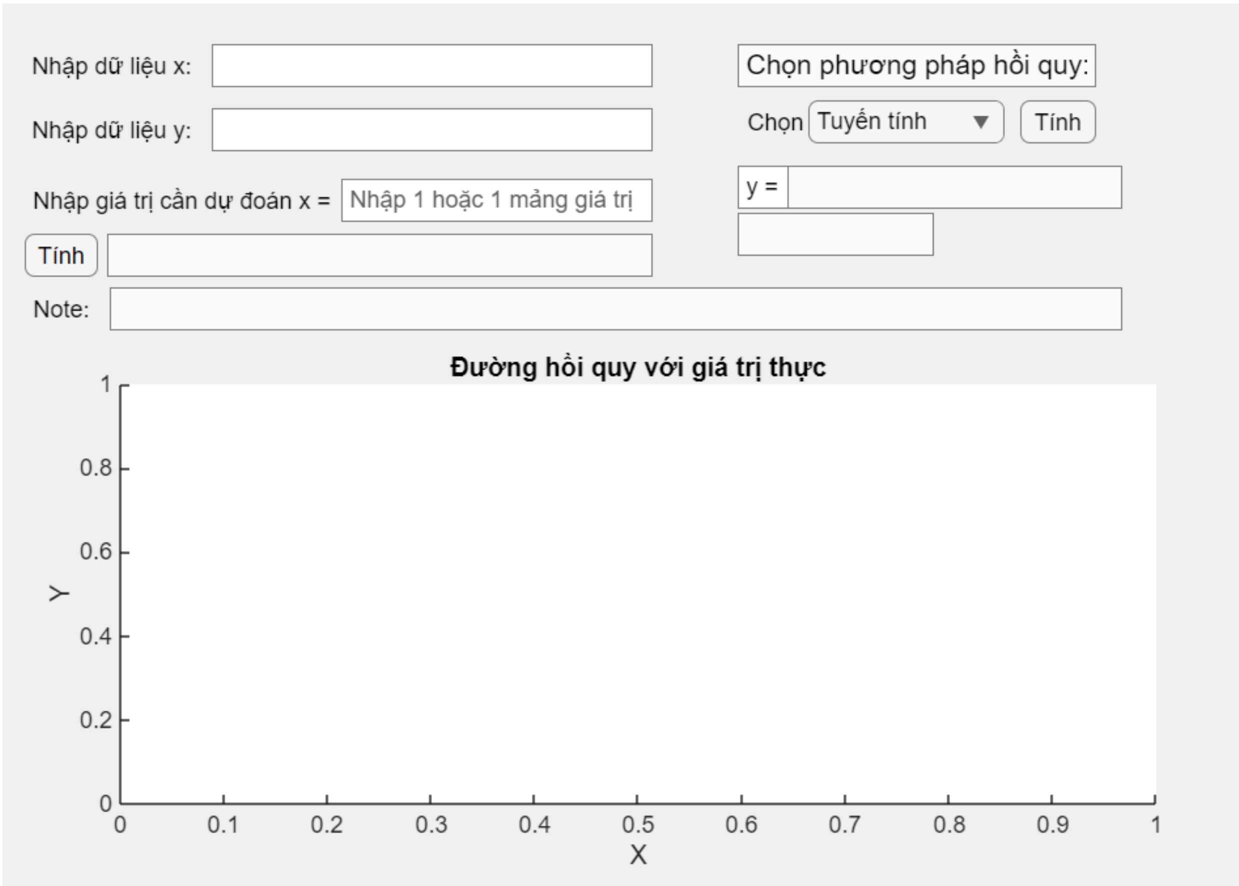
if(c == "Hàm mũ")
    X = x;
    Y = log(y);
    [a0, a1, r2] = HoiQuy(X, Y);
    a = exp(a1);
    b = a0;
    app.rm.Value = sprintf('%.3f*exp(%.3f*x)', a, b);
    app.rv.Value = sprintf('r2 = %.6f', r2);
    yhq = a*exp(b.*x);
end
	- Chọn Hàm mũ
	- Lấy giá trị chuỗi x và đưa chuỗi y về dạng logarit cơ số e
	- Tính a0, a1 và r2
	- Đưa a1 về dạng trước khi truyền
	- Gán a0 vào b
	- In hàm mũ lên thanh app.rm
	- In giá trị r2 lên app.rv
	- Tính yhq để vẽ đồ thị

c) Tính giá trị dự đoán:

Code	Mô tả

Fx = app.rm.Value;
 	
fx = str2func(['@(x)', Fx]);
 	
rp = fx(x);
y = sprintf('%.4f ', rp);
app.pr.Value = y;
	- Lấy hàm tại app.rm, đây là thanh được ghi bởi các hàm xử lí dữ liệu và kết quả ở trên
	- Do được ghi bởi dạng chuỗi kí tự nên cần chuyển sang hàm function
	- Tính tất cả các giá trị f(x), rp có dạng vecto
	- Nối và tạo chuỗi các giá trị dự đoán vào y
	- In các giá trị lên app.pr

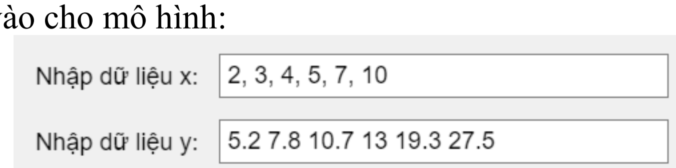
3. Mô tả hoạt động



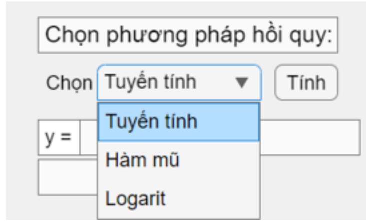
Thiết kế:

- Khung nhập dữ liệu x và y, yêu cầu nhập chuỗi, cách nhau bởi dấu cách hoặc dấu phẩy.
- Khung chọn phương pháp là dạng Drop Down, ngoài tuyến tính còn Logarit và hàm mũ.
- Nút tính cạnh thanh cuộc phương pháp sẽ tính hàm hồi quy, hệ số tương quan và vẽ đồ thị.
- Khung nhập giá trị dự đoán có thể nhập 1 hoặc nhiều giá trị.
- Nút tính ở phần dự đoán sẽ kích hoạt đồng thời nút tính ở thanh chọn phương pháp và tính giá trị cần dự đoán.
- Đồ thị vẽ đường hồi quy dạng nét thẳng và các giá trị thực dạng các điểm.
- Ô note sẽ hiển thị thông báo về tình trạng các thông số nhập vào.

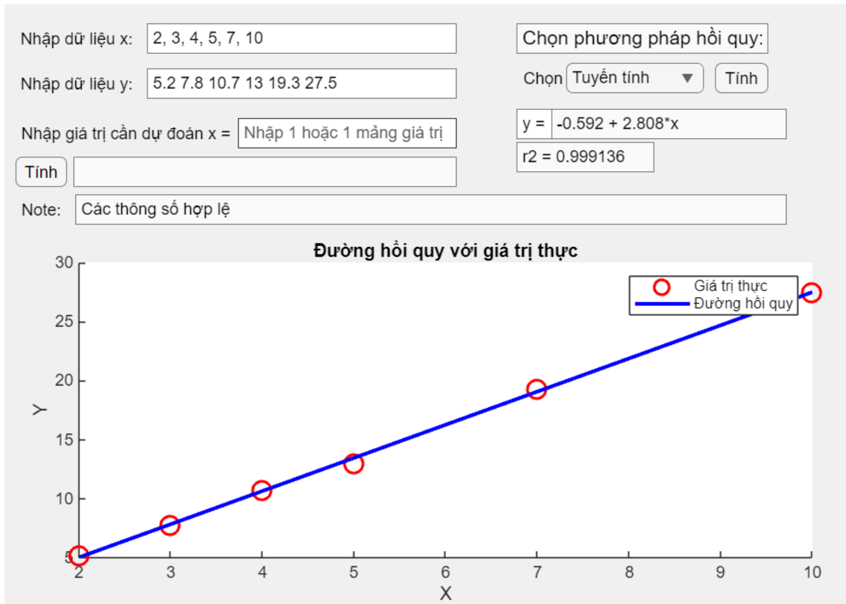
Bước 1: nhập giá trị đầu vào cho mô hình:



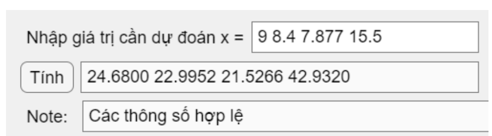
Bước 2: Chọn phương pháp hồi quy và bấm nút Tính để tính hàm hồi quy và hệ số tương quan:



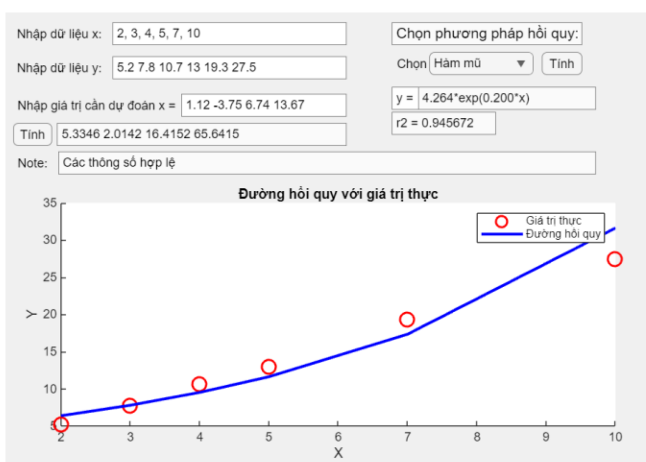
*Kết quả:

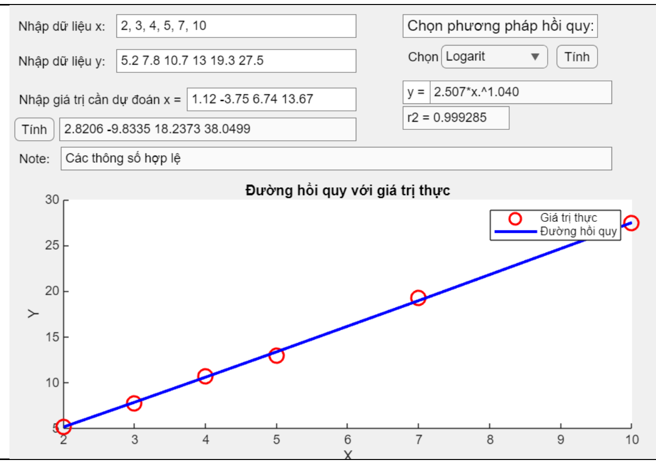
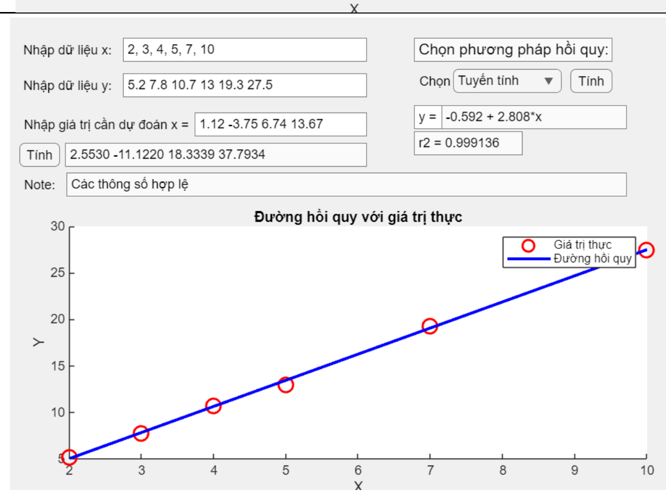


Bước 3: Nhập giá trị cần dự đoán và tính

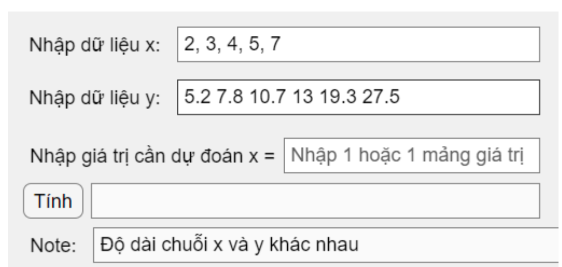
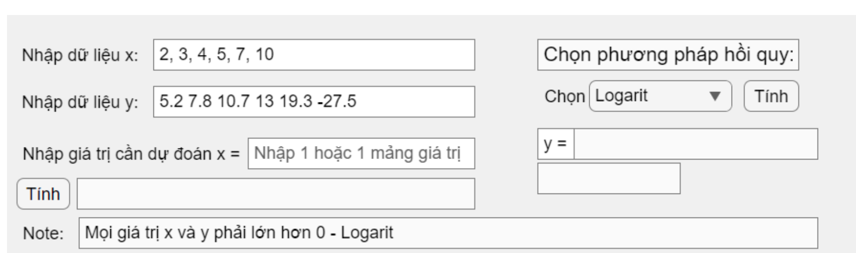


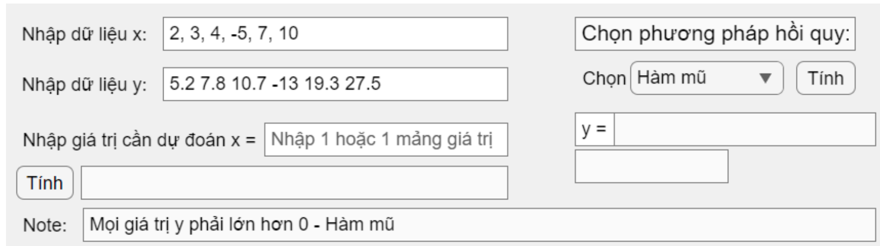
4. Minh họa và kiểm tra kết quả cho từng phương pháp

Hàm mũ	

<table>
<tr><td>Logarit</td><td>
</td></tr>
<tr><td>Tuyến tính</td><td>
</td></tr>
</table>

5. Vấn đề hạn chế và hướng giải quyết

<table>
<tr><td>Lỗi mảng x và y khác độ dài
+ x ngắn hơn y
+ y ngắn hơn x</td><td>
</td></tr>
<tr><td>Chọn Logarit nhưng dữ liệu vào có chứa giá trị âm:
+ x có giá trị âm
+ y có giá trị âm
+ Cả 2 có giá trị âm</td><td>
</td></tr>
</table>

Chọn hàm mũ nhưng chuỗi y đầu vào có chứa giá trị âm:	

*Ngoài ra các trường hợp như:

+ Chưa nhập đầu vào cho x và y khiến mô hình không chạy.

+ Chưa nhập giá trị dự đoán sẽ không tính giá trị hồi quy nhưng vẫn vẽ và tính hàm hồi quy.

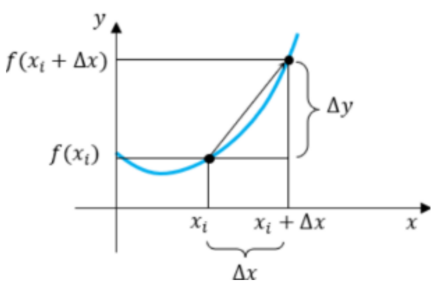
IV. Đạo Hàm

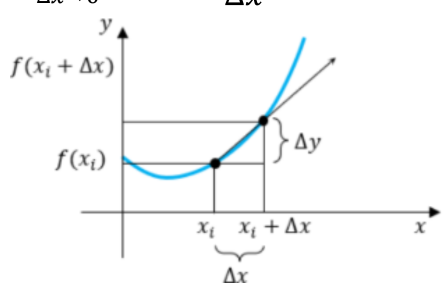
1. Lý thuyết

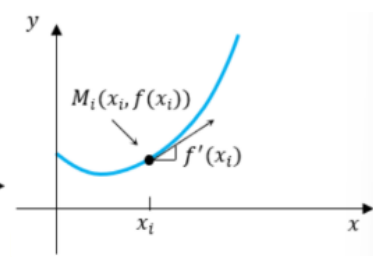
Gọi $f'(x_i)$ là đạo hàm bậc 1 của hàm số $f(x)$ tại điểm x_i nhằm biểu diễn sự biến thiên của hàm số $y = f(x)$ tại điểm $M_i\ (x_i, f(x_i))$

$$\frac{\Delta y}{\Delta x} = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}$$

$$f'(x_i) = \frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}$$







Để thực hiện phương pháp tính đạo hàm (gần đúng) tại một điểm trung gian cần có một số phương pháp nhất định và trong đồ án lần này phương pháp được sử dụng đến là:

- Phương pháp chuỗi Taylor (chiều tiến, chiều lùi, chiều trung tâm)

2. Minh họa thuật toán

Công thức khai triển Taylor bậc n của $f(x_{i+1})$ tại x_i là:

$$f(x_{i+1}) = f(x_i) + \frac{f'(x_i)}{1!}h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + ..$$

Có:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + ..$$

Với $O(h^2)$ là sai số cắt cụt tính từ thành phần đạo hàm bậc 2 (h^2) trở lên:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2!}h^2 + O(h^2)$$

Nếu h rất nhỏ thì ta có thể bỏ qua đạo hàm bậc 2 trở lên và lúc này công thức tính đạo hàm bậc 1 lúc này là:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

Với O(h) là sai số cắt cụt tính từ thành phần đạo hàm bậc 1 (h) trở lên

Trường hợp không bỏ qua thành phần đạo hàm bậc 2 thì công thức đạo hàm bậc 2 và bậc 1 lúc này được biểu diễn như sau:

$$f''(x_i) = \left(\frac{f(x_{i+1}) - f(x_i)}{h} + O(h)\right)'$$

$$= \frac{[f(x_{i+1}) - f(x_i)]'h - [f(x_{i+1}) - f(x_i)]h'}{h^2} + O(h)$$

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h)$$

$$\rightarrow f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2!}h^2 + O(h^2)$$

$$\rightarrow f'(x_i) = \frac{-f(x_{i+2}) - 4f(x_{i+1}) - 3f(x_i)}{2h} + O(h^2)$$

Như đã trình bày ở trên, ta có công thức tính đạo hàm (chiều tiến) từ x_i đến x_{i+1}; x_{i+2};… là:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

$$f'(x_i) = \frac{-f(x_{i+2}) - 4f(x_{i+1}) - 3f(x_i)}{2h} + O(h^2)$$

Đối với phương pháp tính đạo hàm (chiều lùi) từ từ x_i đến x_{i-1}; x_{i-2};… công thức sẽ là:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h)$$

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} + O(h^2)$$

Nếu lấy vị trí x_i làm trung tâm, các sai phân được xác định bằng sai lệch của hàm số $y = f(x_i)$ tại các vị trí trước và sau x_i. Phương pháp này có bậc của sai số cắt cụt lớn gấp đôi so với 2 phương pháp còn lại. Cho nên công thức là:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h} + O(h^4)$$

Từ hệ số hoặc biểu thức trên để lập ra biểu thức $f'(x_i)$ chính xác.

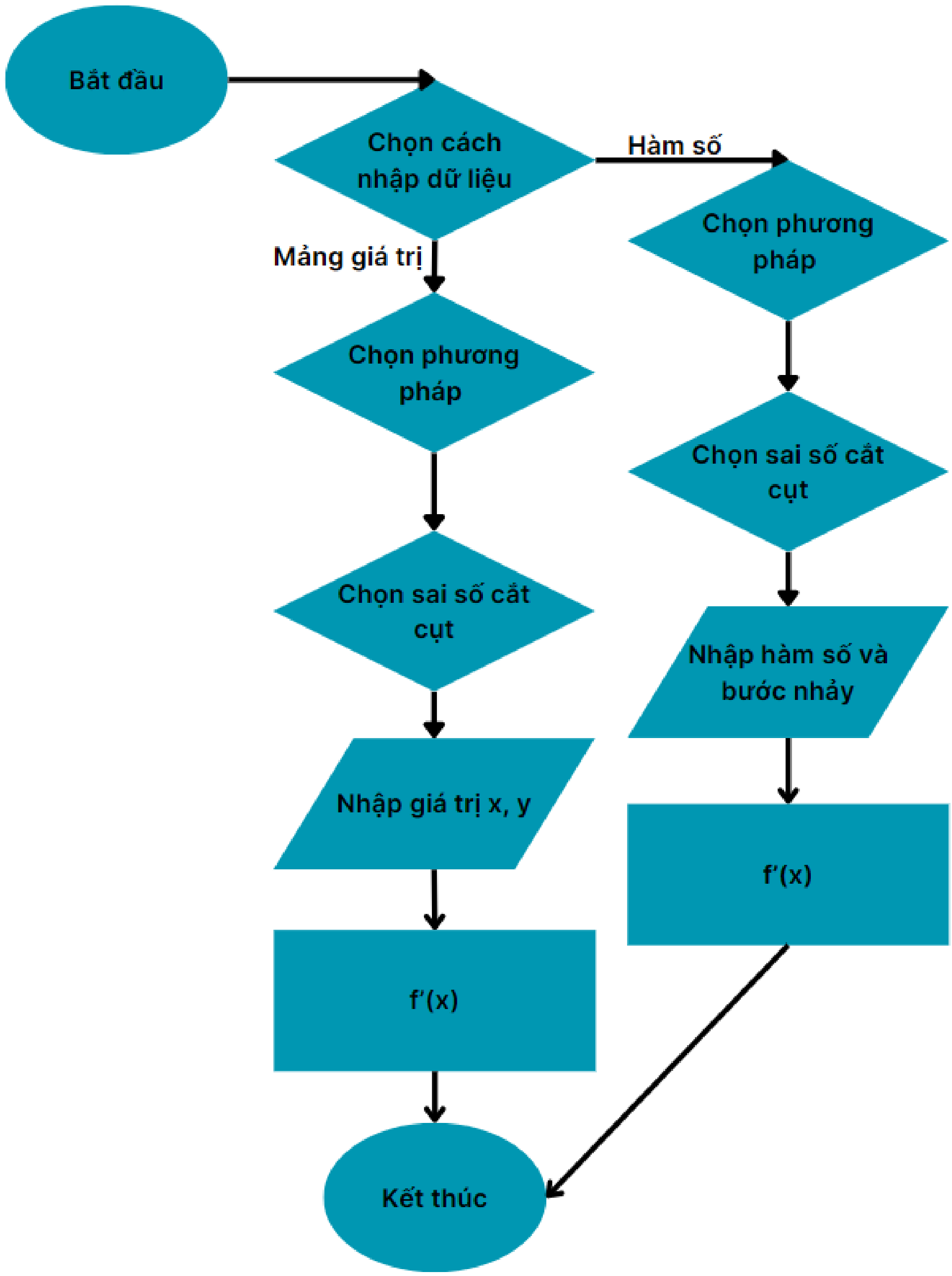
3. Mô tả hoạt động

   Bước 1: Chọn cách nhập dữ liệu là mảng hoặc hàm số.

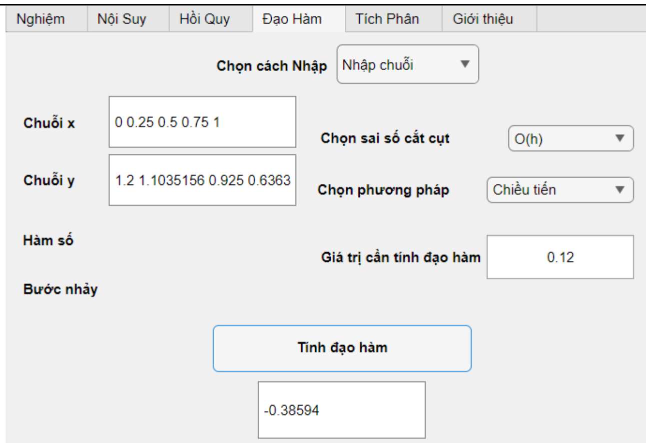
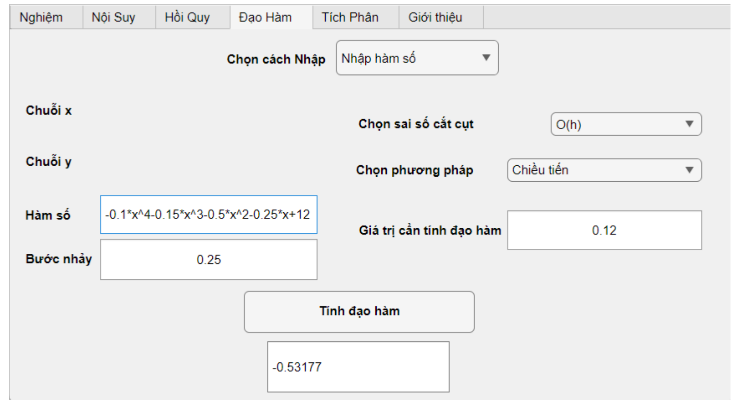
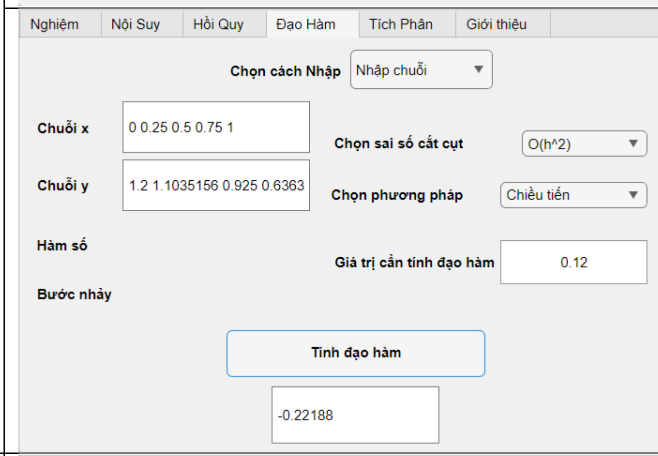
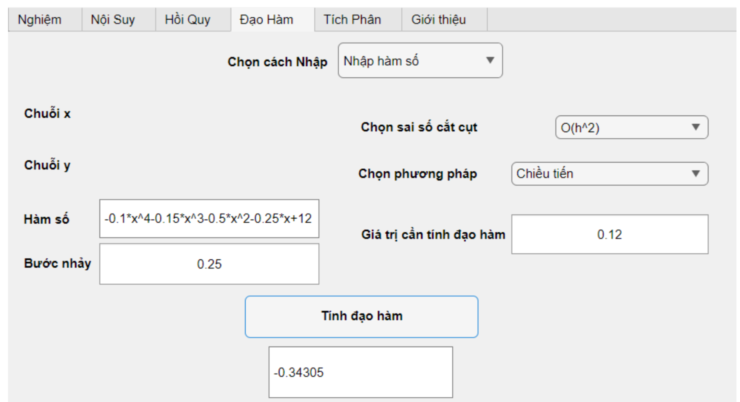
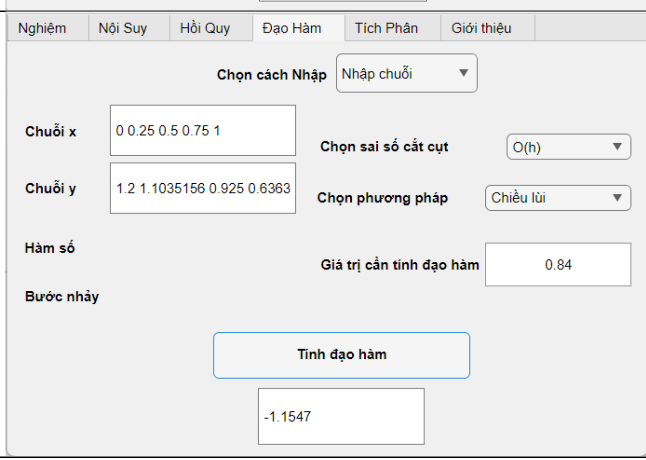
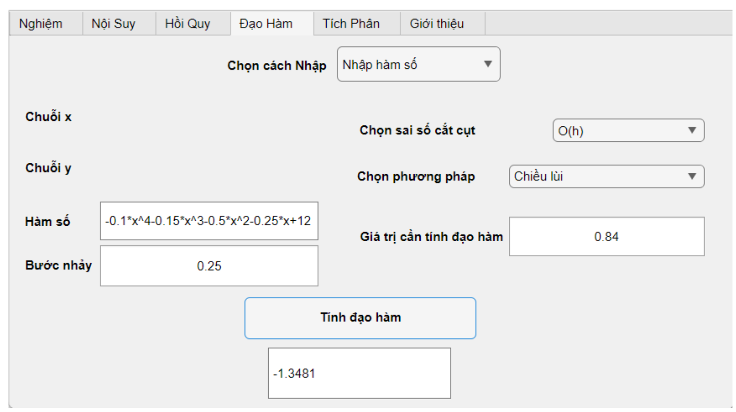
   Bước 2: Chọn 1 trong 3 phương pháp tính đạo hàm: Chiều tiến, chiều lùi, chiều trung tâm.

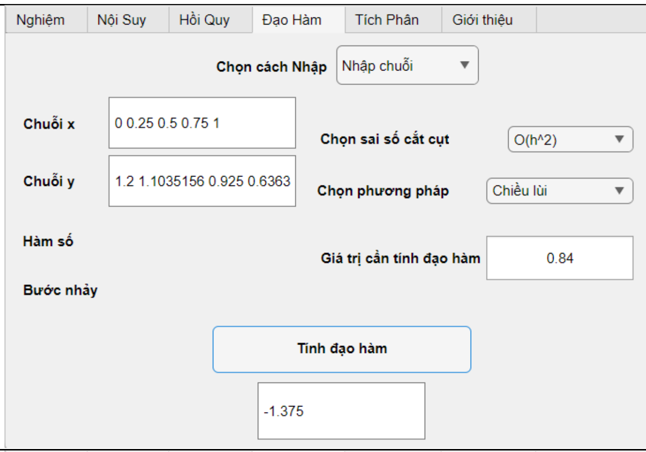
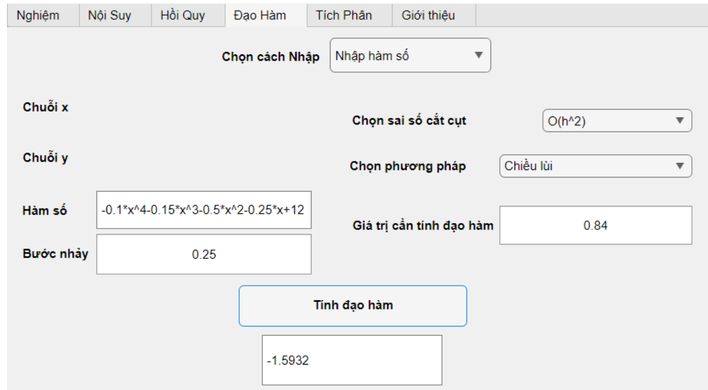
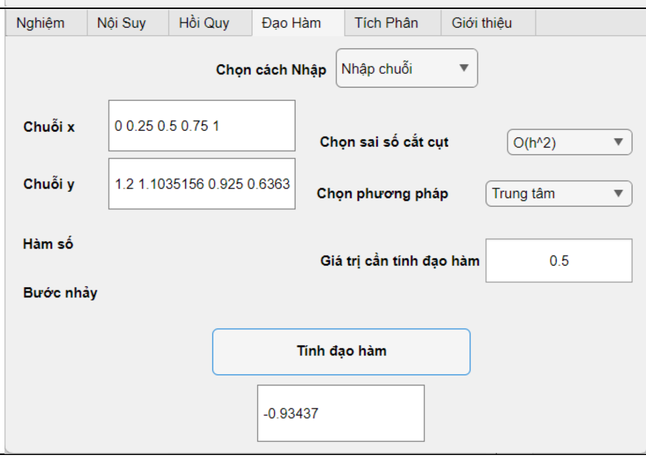
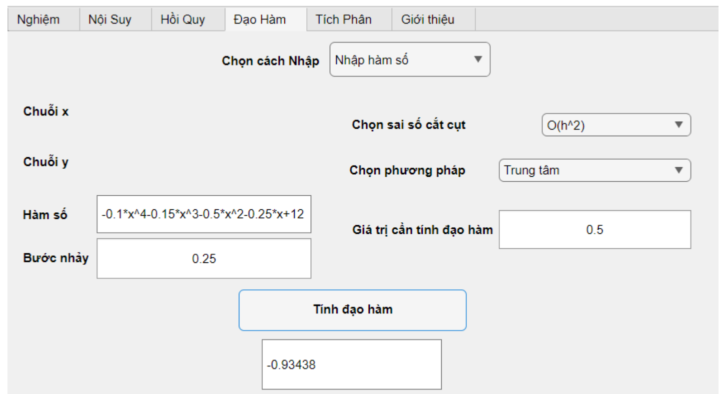
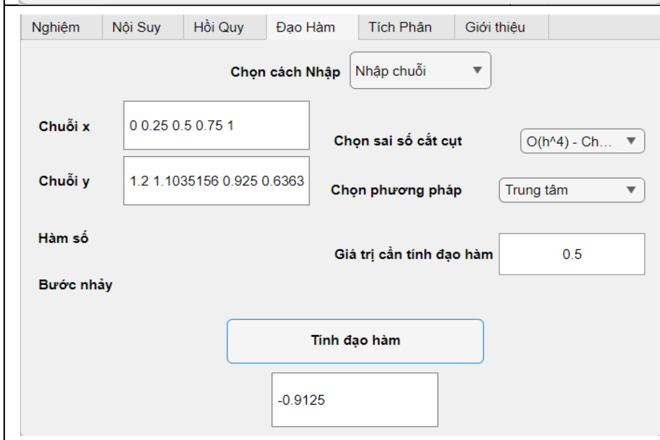
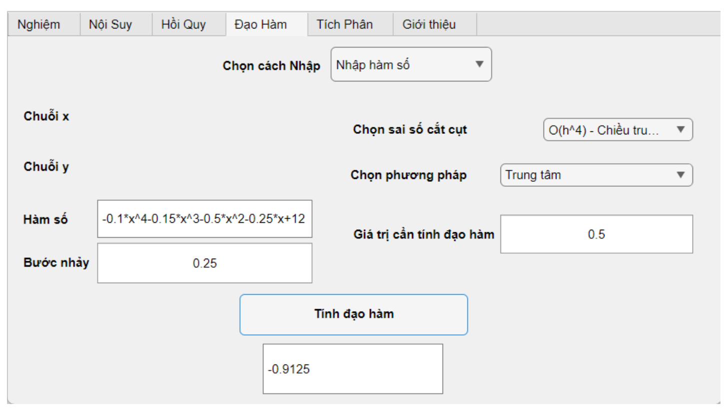
   Bước 3: Chọn giá trị sai số O(h), O(h^2) hoặc O(h^4)(đối với phương pháp trung tâm).

   Bước 4: Nhập giá trị x, y hoặc hàm số $f(x)$ và bước nhảy để tính $f'(x)$, kết quả hiển thị ở ô Kết quả.



4. Minh họa và kiểm tra kết quả cho từng phương pháp

<table>
<tr><th></th><th>Mảng giá trị</th><th>Hàm số</th></tr>
<tr><td rowspan="2">Chiều tiến</td><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Chiều lùi</td><td>
</td><td>
</td></tr>
</table>

<table>
<tr><td></td><td>
</td><td>
</td></tr>
<tr><td rowspan="2">Trung tâm</td><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

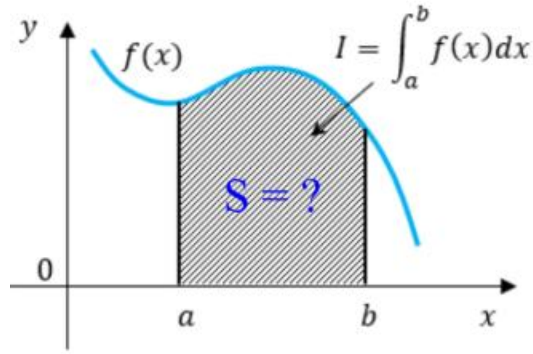
5. Vấn đề hạn chế và hướng giải quyết

- Với mảng x có cái giá trị ngẫu nhiên không có khoảng cách h bằng hoặc gần bằng nhau thì dẫn đến kết quả không đạt độ chính xác cao. Nên bước nhảy h giữa các giá trị phải đồng đều.
- Giá trị tính đạo hàm không phù hợp thì chương trình sẽ không thể chạy được, VD: x = [0.1 0.3 0.5 0.7 0.9] thực hiện tính đạo hàm theo chiều trung tâm tại x = 0.6 với sai số cắt cụt là $O(h^4)$ thì chương trình sẽ không thực hiện được vì y = f(x) không cung cấp đủ giá trị. Nên giá trị tính đạo hàm cần được xem xét kĩ để chương trình có thể thực hiện được.

V. Tích Phân

1. Lý thuyết

Tích phân của hàm số I = $\int_a^b f(x)dx$ được định nghĩa là phần diện tích giới hạn bởi 4 đường bao gồm: đường thẳng x=a, đường thẳng x=b, đường thẳng y=0 và đường cong y = f(x) như hình vẽ dưới đây:

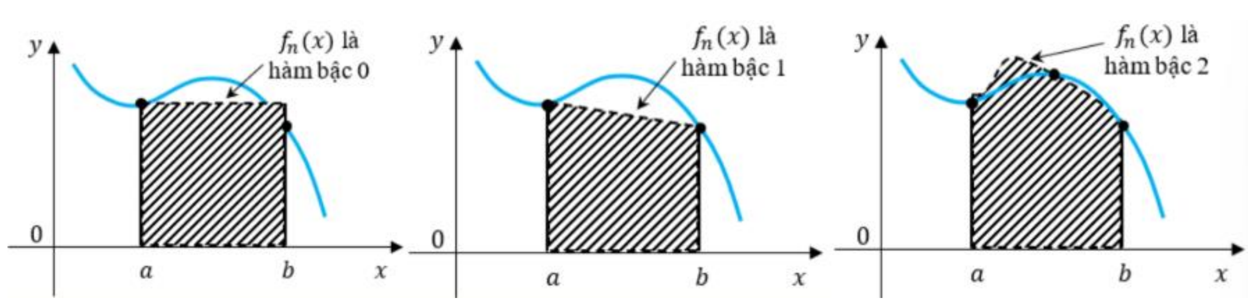


Bài toán đặt ra là thực hiện tính những hàm số f(x) phức tạp, khó biến đổi nên ta có thể dùng những các phương pháp tính gần đúng để tìm được diện tích giới hạn S trên và cũng chính giá trị S này là giá trị của tích phân cần tìm. Trong đồ án này các phương pháp được sử dụng theo công thức Newton Cotes để hiện tính gần đúng tích phân gồm:

- Phương pháp hình thang
- Phương pháp Simpson's $\frac{1}{3}$
- Phương pháp Simpson's $\frac{3}{8}$

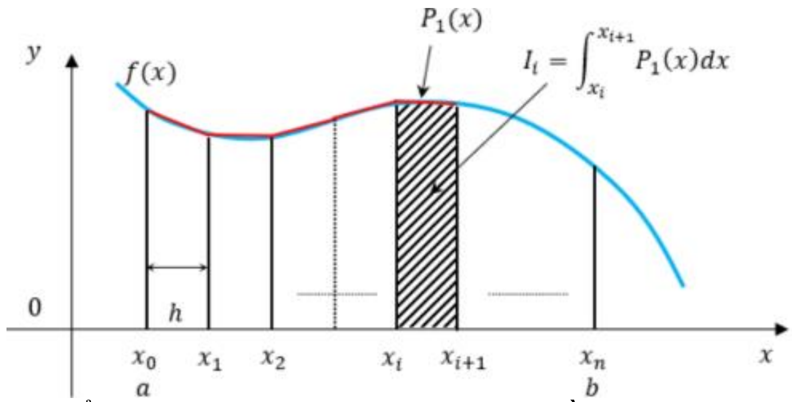
Ý tưởng đề ra là thay các biểu thức phức tạp $f(x)$ hoặc các bảng dữ liệu bằng các hàm xấp xỉ $f_n(x)$ để dễ lấy tích phân. Dạng hàm xấp xỉ thường thấy là dạng đa thức bậc n có dạng $f_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$

Ví dụ về dạng biểu diễn của $f_n(x)$



Tích phân hình thang

2. Minh họa thuật toán
   Với phương pháp hình thang, ta phân chia đoạn [a,b] thành n đoạn nhỏ bằng nhau, sau đó thực hiện nối các điểm $y_i, y_{i+1}, y_{i+2}, ..$ bằng các con đường thẳng



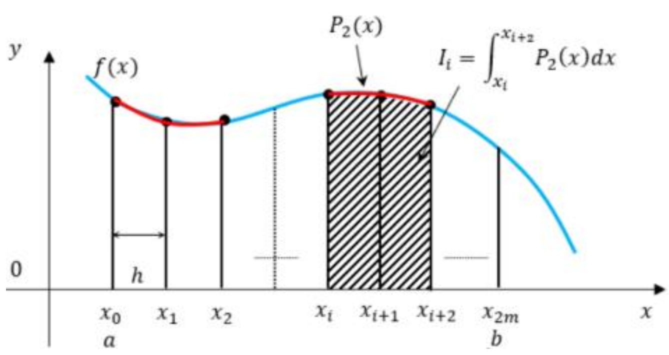
Giá trị tích phân lúc này là tổng diện tích các tích phân từng phần hình thang nhỏ. Có công thức tổng quát của phương pháp hình thang như sau:

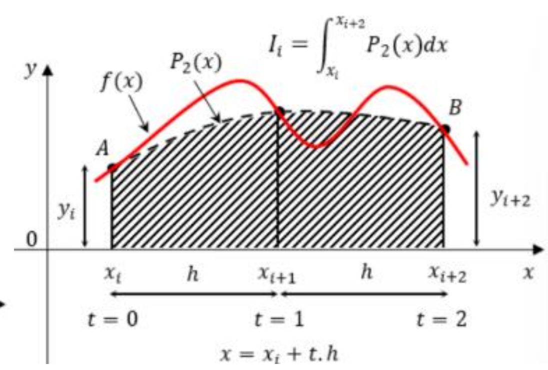
$$I = \int_a^b f(x)dx = \int_{x_0}^{x_1} P_1(x)dx + \int_{x_1}^{x_2} P_1(x)dx + .. + \int_{x_{n-1}}^{x_n} P_1(x)dx$$

$$I = \sum_{i=0}^{n-1} \frac{h}{2}(y_i + y_{i+1}) = \frac{b-a}{2n}(y_0 + y_1 + y_1 + y_2 + \ldots + y_{n-1} + y_{n-1} + y_n)$$

$I = \frac{b-a}{2n}(y_0 + 2y_1 + 2y_2 + \ldots + 2y_{n-1} + y_n)$ với $h = \frac{b-a}{2n}$

Công thức Simpson's $\frac{1}{3}$, thực hiện dựa trên việc phân tách đoạn [a,b] thành n = 2m đoạn con bằng nhau (n là số chẵn) và $h = x_{i+1} - x_i = \frac{b-a}{n}$. Qua 3 điểm liền kề nối các điểm y_i, y_{i+1}và y_{i+2} thay hàm $f(x)$ bằng một đường cong bậc 2 $P_2(x)$.





Có công thức tổng quát của phương pháp Simpson's $\frac{1}{3}$ như sau:

$$I = \int_a^b f(x)dx = \int_{x_0}^{x_2} P_2(x)dx + \int_{x_2}^{x_4} P_2(x)dx + \ldots + \int_{x_{2m-2}}^{x_{2m}} P_2(x)dx$$

Thực hiện khai triển Taylor (Newton tiến) của hàm bậc 2 thì tích phân sẽ là:

$$I_i = \int_{x_i}^{x_{i+2}} P_2(x)dx \approx \int_0^2 \left[y_i + t\Delta y_i + \frac{t(t-1)}{2}\Delta^2 y_i\right] h.dt$$

$$I_i = h\left[y_i t + \frac{t^2}{2}\Delta y_i + \frac{1}{2}\left(\frac{t^3}{3} - \frac{t^2}{2}\right)\Delta^2 y_i\right]\Bigg|\begin{matrix} t = 2 \\ t = 0 \end{matrix}$$

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

$$\to I_i \approx \frac{h}{3}(y_i + 4y_{i+1} + y_{i+2})$$

Tương tự như công thức Simpson's $\frac{1}{3}$, ta thay hàm số $f(x)$ bằng đa thức bậc 3 $P_3(x)$ đi qua 4 điểm. Dùng đa thức nội suy Lagrange bậc 3 ta có dạng tích phân Simpson's $\frac{3}{8}$

$$I = \int_a^b f(x)dx \approx \int_a^b P_3(x)dx$$

$$\to I_i \approx \frac{3h}{8}(y_i + 3y_{i+1} + 3y_{i+2} + y_{i+3})$$

với $h = \frac{b-a}{3}$. Từ các biểu thức trên để lập ra biểu thức chính xác.

3. Mô tả hoạt động

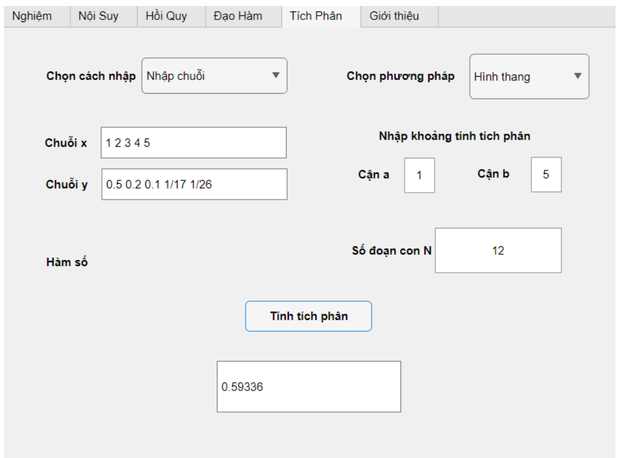
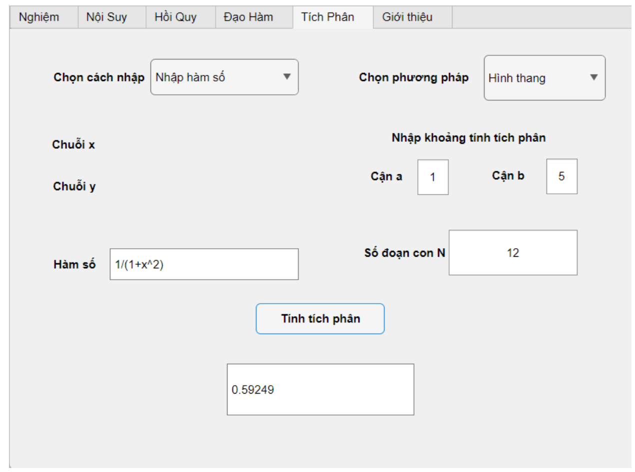
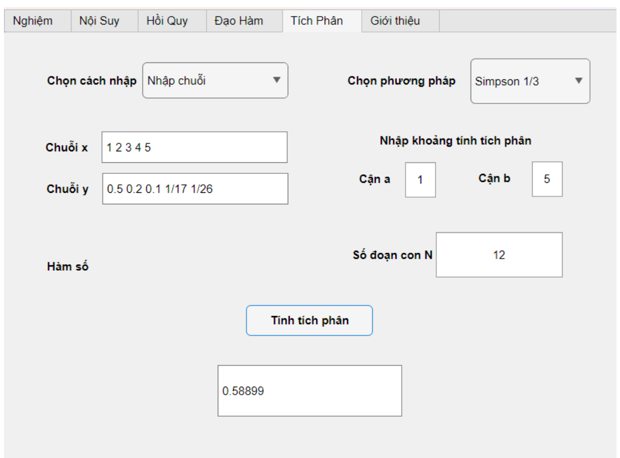
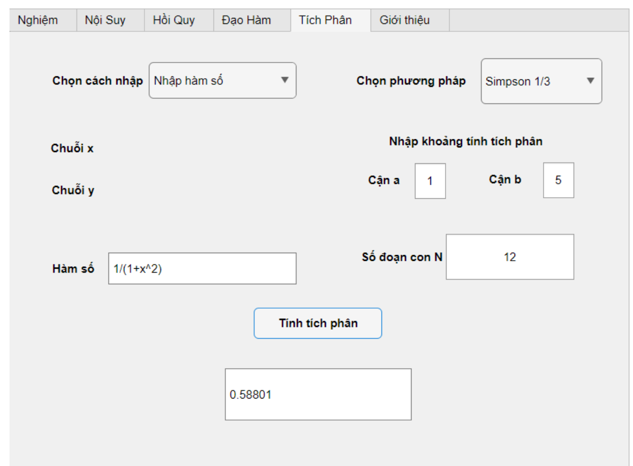
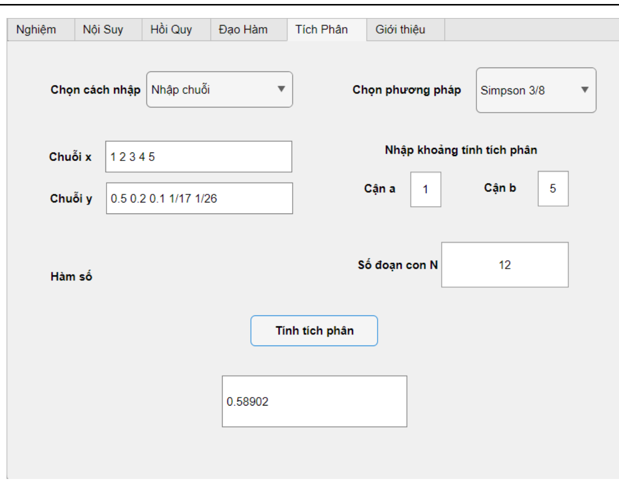
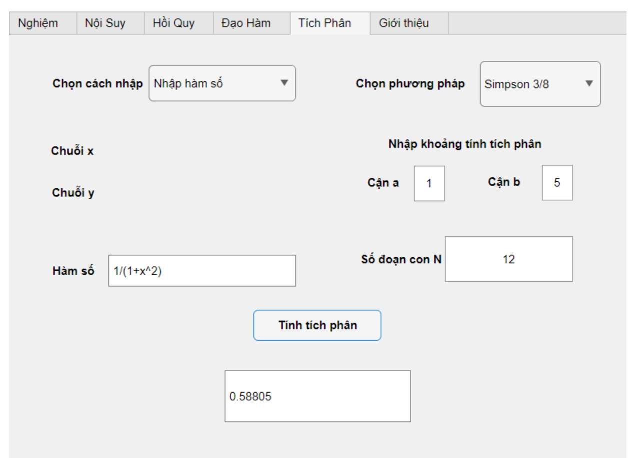
Bước 1: Chọn cách nhập dữ liệu là mảng hoặc hàm số.

Bước 2: Chọn một trong ba phương pháp tính: tích phân hình thang, Simpson's $\frac{1}{3}$ hoặc Simpson's $\frac{3}{8}$.

Bước 3: Nhập các giá trị cận [a, b], số đoạn nhỏ N, sau đó kết quả được trả về sau khi thực hiện các thiết lập và nhấn nút tính toán.

Lưu ý: Nên chọn N là một số chẵn đối với phương pháp Simpson's $\frac{1}{3}$, với Simpson's $\frac{3}{8}$ cần nhập giá trị N là bội số của 3 để kết quả tính ra có giá trị chính xác nhất. Giá trị N càng lớn thì giá trị tính gần đúng càng tiến về giá trị tích phân chính xác.

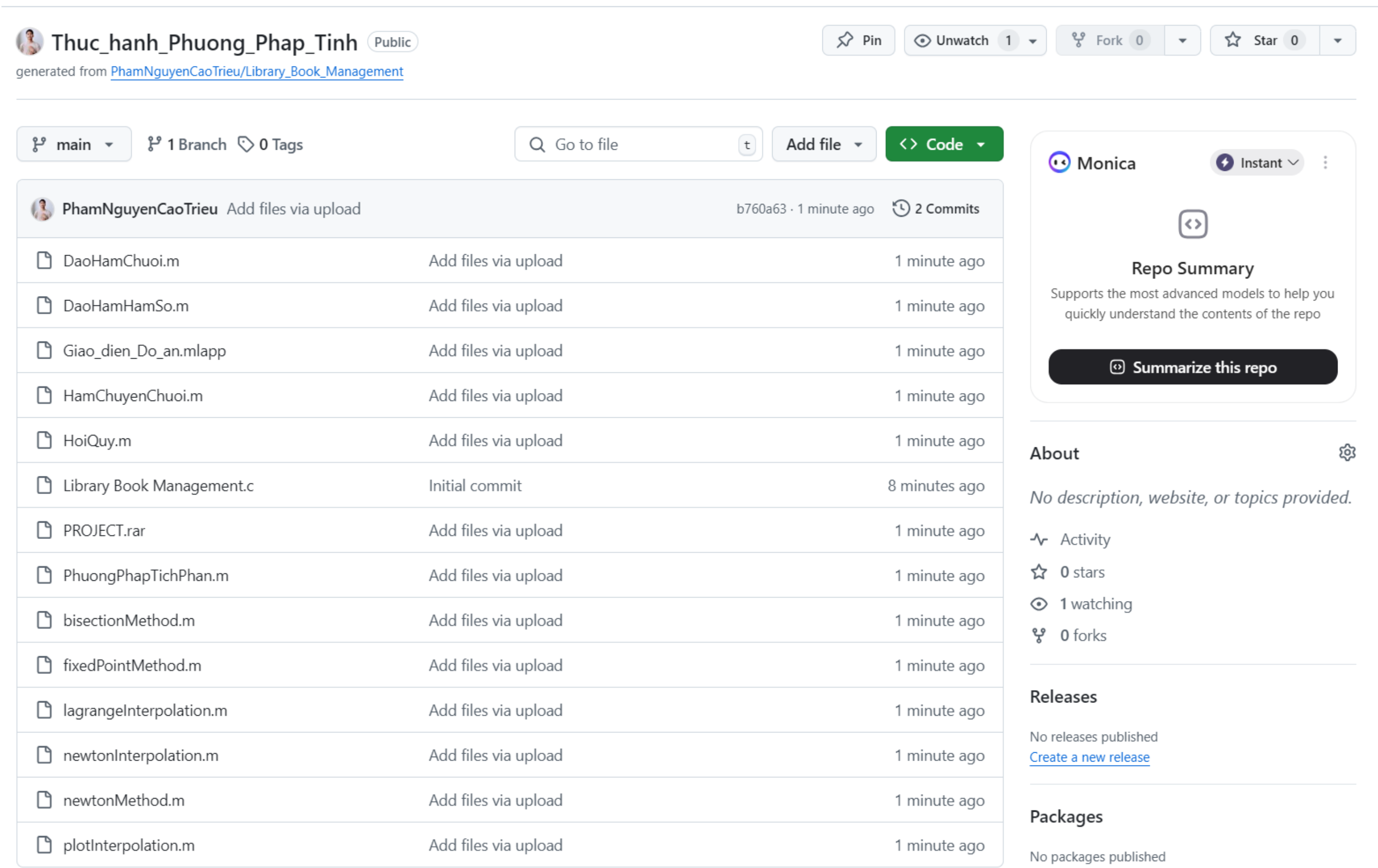
4. Minh họa và kiểm tra kết quả cho từng phương pháp

<table>
<tr><th>Phương pháp</th><th>Mảng dữ liệu</th><th>Hàm số</th></tr>
<tr><td>Hình thang</td><td>
</td><td>
</td></tr>
<tr><td>Simpson’s 1/3</td><td>
</td><td>
</td></tr>
<tr><td>Simpson’s 3/8</td><td>
</td><td>
</td></tr>
</table>

5. Vấn đề hạn chế và hướng giải quyết
Sai số xảy ra càng lớn khi số đoạn chia N càng nhỏ hoặc độ phức tạp của hàm $f(x)$ rất lớn. Nên hướng giải quyết thường là thực hiện chia số đoạn nhỏ N càng lớn nhất có thể và thỏa điều kiện số đoạn N phù hợp cho từng phương pháp Simpson's.

VI. Phần mềm quản lý code (Github)

Github là một nền tảng đám mây nơi bạn có thể lưu trữ, chia sẻ và thực hiện trao đổi nội dung công việc với nhóm trong việc viết code. Github cho phép người dùng lưu trữ mã nguồn để theo dõi và quản lý các thay đổi mới nhất trong xuyên suốt quá trình, đồng thời mã nguồn có thể được chia sẻ công khai cho công đồng hoặc với một số nhóm đối tượng nhất định nhằm cải thiện và nhận sự hợp tác của các thành viên, cá nhân khác.
Nhận thấy sự tiện lợi của phần mềm quản lý code này, nhóm đã sử dụng Github như một công cụ để thực hiện sao lưu các chương trình của đồ án lần này, không chỉ thể các thành viên trong nhóm có thể theo dõi và sửa đổi những lỗi cụ thể hoặc cải tiến một số chức năng nhanh nhất có thể.



Ảnh "Github sau khi up tài liệu và nội dung đồ án"

VII. Đánh giá nhóm

Đánh giá thành viên nhóm			
MSSV	**Họ và tên**	**Nhiệm vụ**	**Tự đánh giá**
22207123	Phạm Nguyễn Cao Triều	Tích phân và Đạo hàm	33.33%
22207094	Huỳnh Phạm Minh Tú	Hồi quy và Thiết kế	33.33%
22207103	Trương Quang Hưng	Tìm nghiệm và Nội suy	33.34%

Nội dung nhiệm vụ:
Đối với các nhiệm vụ thực hiện phát triển chức năng của ứng dụng, nội dung được đề ra là:

- Đảm bảo được thiết kế đầy đủ chức năng theo yêu cầu của giáo viên đưa ra
- Kiểm tra và sửa những lỗi cần được phát triển (nếu có) sớm nhất có thể
- Báo cáo word của những phần này cần được thể hiện đầy đủ những nội dung sau:
  + Có minh họa thuật toán (dựa vào lý thuyết hoặc ý tưởng tự nghĩ ra)
  + Có mô tả hoạt động của từng chức năng
  + Có đầy đủ kết quả kiểm nghiệm của từng phương pháp
  + Đưa ra những hạn chế nhất định hoặc những trường hợp mà không thể giải quyết được hoặc có hướng nhưng chưa thực hiện dược trong đề tài
- Báo cáo word cần có tối thiểu trang bìa, nội dung giới thiệu đề tài, danh sách thành viên cũng như có nội dung tự đánh giá của điểm theo tiêu chí

Bảng đánh giá các tiêu chí			
TT	**Nội dung**	**Điểm**	**Đánh giá**
1	Thiết kế được giao diện Tab Nghiệm	0.4	✓
2	Thiết kế được giao diện Tab Nội Suy	0.4	✓
3	Thiết kế được giao diện Tab Hồi quy	0.4	✓
4	Thiết kế được giao diện Tab Đạo hàm	0.4	✓
5	Thiết kế được giao diện Tab Tích phân	0.4	✓
6	Thiết kế được giao diện Tab Giới thiệu nhóm	0.4	✓
7	Tìm được nghiệm dùng phương pháp Chia đôi	0.4	✓
8	Tìm được nghiệm dùng phương pháp Lặp	0.4	✓
9	Tìm được nghiệm dùng phương pháp Newton	0.4	✓
10	Vẽ được hàm số cần tìm nghiệm	0.4	✓

11	Tìm được đa thức nội suy Newton	0.4	✓
12	Dự đoán được giá trị cần nội suy với nội suy Newton	0.4	✓
13	Tìm được đa thức nội suy Lagrange	0.4	✓
14	Dự đoán được giá trị cần nội suy với nội suy Lagrange	0.4	✓
15	Tìm được và vẽ phương trình hồi quy tuyến tính	0.4	✓
16	Tìm được và vẽ phương trình hồi quy hàm mũ	0.4	✓
17	Tìm được và vẽ phương trình hồi quy mũ e	0.4	✓
18	Tính được đạo hàm cho dữ liệu x, y	0.4	✓
19	Tính được đạo hàm từ hàm số	0.4	✓
20	Thay đổi được phương pháp tính đạo hàm: Xấp xỉ tiến, xấp xỉ lùi, xấp xỉ trung tâm	0.4	✓
21	Tính được tích phân hình thang từ x, y	0.4	✓
22	Tính được tích phân hình thang từ hàm số nhập vào	0.4	✓
23	Tính được tích phân bằng phương pháp Simpson 1/3	0.4	✓
24	Tính được tích phân bằng phương pháp Simpson 3/8	0.4	✓
25	Có sử dụng hàm cho từng phương pháp	0.4	✓

TÀI LIỆU THAM KHẢO

[1] Nguyễn Xuân Vinh (2024). Phần 2 – Phương pháp tính

[2] Lê Trọng Vinh, Trần Minh Toàn (2013). Giáo trình Phương pháp tính và Matlab. NXB Bách Khoa, Hà Nội

[3] Huỳnh Quốc Thịnh (2023). Giáo trình thực hành Phương pháp tính

[4] Huỳnh Quốc Thịnh (2023). DoAnPPT_Matlab

[5] Brian D. Hahn and Daniel T. Valentine (2007). Essential MATLAB for Engineers and Scientists