

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ - VIỄN THÔNG



ĐỒ ÁN CUỐI KỲ
MÔN PHƯƠNG PHÁP TÍNH & MATLAB
Đề tài: THIẾT KẾ ỨNG DỤNG BẰNG APP DESIGNER
LỚP: 21DTV2

GIẢNG VIÊN HƯỚNG DẪN: HUỲNH QUỐC THỊNH

NHÓM SINH VIÊN THỰC HIỆN: NHÓM 6BRO

STT	HỌ VÀ TÊN	MSSV
1	DANH CHÍ HIỀN (Nhóm trưởng)	21200287
2	NGUYỄN GIANG (Thư ký)	21200284
3	TRẦN NGỌC HUY	20200040
4	TRẦN HỮU HẠNH	21200286
5	NGUYỄN THỪA VŨ HIỆP	21200288
6	TRẦN LÊ NHẬT QUİN	21200342

TP. Hồ Chí Minh – Năm 2023

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ - VIỄN THÔNG

ĐỒ ÁN CUỐI KỲ

MÔN PHƯƠNG PHÁP TÍNH & MATLAB

Đề tài: THIẾT KẾ ỨNG DỤNG BẰNG APP DESIGNER

LỚP: 21DTV2

GIẢNG VIÊN HƯỚNG DẪN: HUỖNH QUỐC THỊNH

NHÓM SINH VIÊN THỰC HIỆN: NHÓM 6BRO

STT	HỌ VÀ TÊN	MSSV
1	DANH CHÍ HIỀN (Nhóm trưởng)	21200287
2	NGUYỄN GIANG (Thư ký)	21200284
3	TRẦN NGỌC HUY	20200040
4	TRẦN HỮU HẠNH	21200286
5	NGUYỄN THỪA VŨ HIỆP	21200288
6	TRẦN LÊ NHẬT QUIN	21200342

TP. Hồ Chí Minh – Năm 2023

LỜI CẢM ƠN

Lời đầu tiên, nhóm 6BRO, xin gửi lời cảm ơn sâu sắc và chân thành nhất đến giảng viên hướng dẫn, Th.S Huỳnh Quốc Thịnh, thầy đã hỗ trợ và hướng dẫn nhóm trong suốt quá trình thực hiện đồ án cuối kỳ môn học Phương pháp tính và Matlab.

Thầy đã không ngần ngại chia sẻ kiến thức, kinh nghiệm và thời gian quý báu của mình để giúp nhóm chúng em hoàn thiện đồ án với đề tài “*Thiết kế ứng dụng bằng App Designer*”. Những lời khuyên và góp ý của thầy đã giúp nhóm chúng em rất nhiều trong việc nắm bắt và áp dụng lý thuyết vào thực tế. Chúng em cũng xin gửi lời cảm ơn đến các bạn trong nhóm đã cùng nhau nỗ lực, cống hiến và hợp tác để hoàn thành đồ án này. Sự cố gắng và đam mê của mỗi thành viên đã tạo nên sự thành công của đồ án. Mặc dù còn nhiều sai sót trong quá trình thực hiện nhưng đó là kết quả của sự nỗ lực không ngừng từ các thành viên.

Nhóm rất mong nhận được những góp ý từ thầy, nhằm giúp chúng em hoàn thiện vốn kiến thức và rút ra những kinh nghiệm quý báu để tiếp tục hoàn thành tốt những dự án sắp tới.

Xin chân thành cảm ơn thầy!

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

....., ngày ... tháng ... năm 2023

Người nhận xét

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN	3
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	4
MỤC LỤC.....	1
DANH MỤC CÁC HÌNH.....	3
DANH MỤC CÁC BẢNG.....	6
TÓM TẮT	7
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	8
1.1. LÝ DO CHỌN ĐỀ TÀI	8
1.2. MỤC TIÊU ĐỀ TÀI	8
1.3. QUY TRÌNH PHÁT TRIỂN CỦA ỨNG DỤNG	9
1.3.1 TỔNG QUAN VỀ MATLAB APP DESIGNER	10
1.3.2 VAI TRÒ CỦA CÁC THÀNH VIÊN TRONG ĐỒ ÁN.....	10
CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI.....	11
PHẦN 1: THUẬT TOÁN CHƯƠNG TRÌNH	11
1.1 TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH.....	11
1.1.1 Phương pháp chia đôi	11
1.1.2 Phương pháp lặp	11
1.1.3 Phương pháp Newton (Tiếp tuyến).....	12
1.2 ĐA THỨC NỘI SUY.....	12
1.2.1 Đa thức nội suy Lagrange	12
1.2.2 Đa thức Newton	13
1.3 HỒI QUY	14
1.3.1 Hồi quy tuyến tính	14
1.3.2 Hồi quy hàm mũ.....	15
1.3.3 Hồi quy Logarit.....	15
1.4 TÍNH GẦN ĐÚNG ĐẠO HÀM.....	16
1.5 TÍNH GẦN ĐÚNG TÍCH PHÂN.....	17
1.5.1 Tích phân hình thang	17
1.5.2 Công thức Simpson (Simpson's 1/3)	18
1.5.3 Công thức Simpson's 3/8.....	18

PHẦN 2: CODE CHƯƠNG TRÌNH	18
2.1 CODE THUẬT TOÁN	18
2.1.1 TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH	18
2.1.2 ĐA THỨC NỘI SUY	20
2.1.3 HỒI QUY	21
2.1.4 TÍNH GẦN ĐÚNG ĐẠO HÀM.....	22
2.1.5 TÍNH GẦN ĐÚNG TÍCH PHẦN	24
2.2 CODE GIAO DIỆN	25
2.2.1 TẠO GIAO DIỆN NGƯỜI DÙNG	25
2.2.2 TẠO SỰ KIỆN CALLBACK.....	34
2.2.2.1 Tab Nghiệm	34
2.2.2.2 Tab Nội Suy	38
2.2.2.3 Tab Hồi Quy	40
2.2.2.4 Tab ĐẠO HÀM	42
2.2.2.5 Tab TÍCH PHẦN	46
PHẦN 3: CHẠY CHƯƠNG TRÌNH TÍNH	49
3.1 TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH.....	50
3.2 ĐA THỨC NỘI SUY.....	56
3.3 HỒI QUY	61
3.4 TÍNH GẦN ĐÚNG ĐẠO HÀM.....	68
3.5 TÍNH GẦN ĐÚNG TÍCH PHẦN.....	77
CHƯƠNG 3: TỔNG KẾT.....	82
3.1. ĐÁNH GIÁ ĐỘ HOÀN THIỆN CỦA ĐỀ TÀI	82
3.1.1 ƯU ĐIỂM	82
3.1.2 HẠN CHẾ	82
3.1.3 ĐỀ XUẤT	82
3.2. ĐÁNH GIÁ NHÓM VÀ CÁC THÀNH VIÊN	83
3.2.1 HOẠT ĐỘNG CỦA NHÓM VÀ CÁC THÀNH VIÊN	83
3.2.2 BẢNG PHÂN CÔNG, ĐÁNH GIÁ CÁC THÀNH VIÊN	84
TÀI LIỆU THAM KHẢO.....	86

DANH MỤC CÁC HÌNH

Hình 1. Thuật toán tìm gần đúng nghiệm theo phương pháp chia đôi	11
Hình 2. Giao diện hiển thị với người dùng	26
Hình 3. Giao diện tìm nghiệm	26
Hình 4. Giao diện Tìm Nghiệm và tên các components	27
Hình 5. Giao diện Nội Suy	28
Hình 6. Giao diện Nội Suy và tên các components	28
Hình 7. Giao diện Hồi Quy	29
Hình 8. Giao diện Hồi Quy và tên các components	30
Hình 9. Giao diện ĐẠO HÀM	31
Hình 10. Giao diện ĐẠO HÀM và tên các components	31
Hình 11. Giao diện TÍCH PHÂN	32
Hình 12. Giao diện TÍCH PHÂN và tên các components	33
Hình 13. Giao diện GIỚI THIỆU NHÓM	34
Hình 14. Đặt tất cả các file vào cùng một thư mục	50
Hình 15. Nhập các thông số vào giao diện	51
Hình 16. Kết quả đạt được bằng phương pháp chia đôi sau khi ấn "THỰC THI"	52
Hình 17. Kết quả đạt được bằng phương pháp lặp sau khi ấn "THỰC THI"	52
Hình 18. Kết quả đạt được bằng phương pháp Newton sau khi ấn "THỰC THI"	53
Hình 19. Nhập các thông số vào giao diện	54
Hình 20. Kết quả đạt được bằng phương pháp chia đôi sau khi ấn "THỰC THI"	54

Hình 21. Kết quả đạt được bằng phương pháp lặp sau khi ấn "THỰC THI"	55
Hình 22. Kết quả đạt được bằng phương pháp Newton sau khi ấn "THỰC THI"	55
Hình 23. Nhập các thông số vào giao diện	56
Hình 24. Kết quả đạt được bằng Nội Suy Newton sau khi ấn "THỰC THI"	57
Hình 25. Kết quả đạt được bằng Nội Suy Lagrange sau khi ấn "THỰC THI"	58
Hình 26. Nhập các thông số vào giao diện	59
Hình 27. Kết quả đạt được bằng Nội Suy Newton sau khi ấn "THỰC THI"	60
Hình 28. Kết quả đạt được bằng Nội Suy Lagrange sau khi ấn "THỰC THI"	61
Hình 29. Nhập các thông số vào giao diện	62
Hình 30. Kết quả đạt được bằng Hồi Quy Tuyến tính sau khi ấn "TÍNH"	63
Hình 31. Kết quả đạt được bằng Hồi Quy Hàm mũ sau khi ấn "TÍNH"	64
Hình 32. Kết quả đạt được bằng Hồi Quy Logarit sau khi ấn "TÍNH"	65
Hình 33. Nhập các thông số vào giao diện	66
Hình 34. Kết quả đạt được bằng Hồi Quy Tuyến tính sau khi ấn "TÍNH"	67
Hình 35. Kết quả đạt được bằng Hồi Quy Hàm mũ sau khi ấn "TÍNH"	67
Hình 36. Kết quả đạt được bằng Hồi Quy Logarit sau khi ấn "TÍNH"	68
Hình 37. Nhập các thông số vào giao diện	69
Hình 38. Xấp xỉ tiến với sai số cắt cụt $O(h)$	70
Hình 39. Xấp xỉ lùi với sai số cắt cụt $O(h)$	70
Hình 40. Xấp xỉ trung tâm với sai số cắt cụt $O(h)$	71
Hình 41. Xấp xỉ trung tâm với sai số cắt cụt $O(h^2)$	71
Hình 42. Xấp xỉ lùi với sai số cắt cụt $O(h^2)$	72

Hình 43. Xấp xỉ tiến với sai số cắt cụt $O(h^2)$	72
Hình 44. Nhập các thông số vào giao diện	73
Hình 45. Xấp xỉ tiến với sai số cắt cụt $O(h)$	74
Hình 46. Xấp xỉ lùi với sai số cắt cụt $O(h)$	74
Hình 47. Xấp xỉ trung tâm với sai số cắt cụt $O(h)$	75
Hình 48. Xấp xỉ trung tâm với sai số cắt cụt $O(h^2)$	75
Hình 49. Xấp xỉ lùi với sai số cắt cụt $O(h^2)$	76
Hình 50. Xấp xỉ tiến với sai số cắt cụt $O(h^2)$	76
Hình 51. Nhập các thông số vào giao diện	77
Hình 52. Tích phân hình thang cho mảng dữ liệu	78
Hình 53. Tích phân Simpson 1/3 cho mảng dữ liệu	78
Hình 54. Tích phân Simpson 3/8 cho mảng dữ liệu	79
Hình 55. Nhập các thông số vào giao diện	80
Hình 56. Tích phân hình thang cho hàm số	80
Hình 57. Tích phân Simpson 1/3 cho hàm số	81
Hình 58. Tích phân Simpson 3/8 cho hàm số	81
Hình 59. Hoạt động của nhóm trên MS Teams	83

DANH MỤC CÁC BẢNG

<i>Bảng 1. Bảng phân công - đánh giá thành viên nhóm 6BRO</i>	<i>85</i>
---	-----------

TÓM TẮT

Đồ án cuối kỳ môn học Phương pháp tính và Matlab của nhóm 6BRO, dưới sự hướng dẫn của giảng viên Th.S Huỳnh Quốc Thịnh, tập trung vào việc thiết kế và phát triển một ứng dụng bằng App Designer trên Matlab để giải quyết các bài toán phương pháp tính.

Chúng tôi đã thành công trong việc thiết kế các giao diện người dùng cho các bài toán Tìm nghiệm, Nội suy, Hồi quy, Đạo hàm và Tích phân. Chúng tôi cũng đã phát triển các thuật toán để giải quyết các bài toán này, bao gồm việc tìm nghiệm bằng các phương pháp Chia đôi, Lặp và Newton; tìm đa thức nội suy Newton và Lagrange; tìm và vẽ phương trình hồi quy tuyến tính, hồi quy hàm mũ và hồi quy mũ e; tính đạo hàm cho dữ liệu x, y và từ hàm số; thay đổi phương pháp tính đạo hàm; và tính tích phân hình thang và bằng phương pháp Simpson.

Đồ án này không chỉ giúp chúng tôi nắm bắt được kiến thức môn học một cách sâu sắc hơn mà còn tạo ra một sản phẩm có giá trị thực tế, có thể được sử dụng trong nhiều ngữ cảnh khác nhau. Chúng tôi rất tự hào về những gì chúng tôi đã đạt được và hy vọng rằng đồ án này sẽ là nguồn cảm hứng cho những người khác trong việc học và ứng dụng Phương pháp tính và Matlab.

Trân trọng, nhóm 6BRO.

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. LÝ DO CHỌN ĐỀ TÀI

Chúng tôi - nhóm 6BRO, đã chọn đề tài “Thiết kế ứng dụng bằng App Designer” cho đồ án cuối kỳ môn học Phương pháp tính và Matlab vì những lý do sau:

Ứng dụng thực tế: Matlab App Designer là công cụ mạnh mẽ cho phép chúng tôi thiết kế các ứng dụng có giao diện đồ họa người dùng (GUI) một cách dễ dàng và hiệu quả. Việc tạo ra một ứng dụng có thể giải quyết các bài toán phương pháp tính không chỉ giúp chúng tôi nắm bắt được kiến thức môn học một cách sâu sắc hơn mà còn tạo ra một sản phẩm có giá trị thực tế, có thể được sử dụng trong nhiều ngữ cảnh khác nhau.

Phát triển kỹ năng lập trình: Qua quá trình thực hiện đồ án, chúng tôi đã có cơ hội phát triển và cải thiện kỹ năng lập trình của mình. Chúng tôi đã học cách sử dụng Matlab App Designer để tạo ra các giao diện người dùng đẹp mắt và dễ sử dụng, cũng như cách viết code để xử lý các bài toán phức tạp.

Tính toán và trực quan hóa dữ liệu: Đồ án của chúng tôi không chỉ giải quyết các bài toán phương pháp tính mà còn cho phép chúng tôi trực quan hóa dữ liệu và kết quả. Điều này giúp chúng tôi hiểu rõ hơn về cách thức hoạt động của các phương pháp và cách chúng tác động đến kết quả.

Hợp tác nhóm: Đồ án cuối kỳ cũng đã tạo ra một cơ hội tuyệt vời để chúng tôi học cách làm việc nhóm một cách hiệu quả. Mỗi thành viên trong nhóm đều có vai trò quan trọng và cùng nhau đóng góp vào sự thành công của đồ án.

1.2. MỤC TIÊU ĐỀ TÀI

Mục tiêu của đồ án cuối kỳ môn học Phương pháp tính và Matlab của nhóm, dưới sự hướng dẫn của giảng viên Huỳnh Quốc Thịnh, là thiết kế và phát triển một

ứng dụng bằng App Designer trên Matlab để giải quyết các bài toán phương pháp tính. Cụ thể, mục tiêu của chúng tôi bao gồm:

Thiết kế Giao diện: Chúng tôi sẽ thiết kế các giao diện người dùng cho các bài toán Tìm nghiệm, Nội suy, Hồi quy, Đạo hàm và Tích phân. Mỗi giao diện sẽ được thiết kế một cách trực quan và dễ sử dụng, giúp người dùng dễ dàng nhập dữ liệu và nhận kết quả.

Phát triển Luận lý: Chúng tôi sẽ phát triển các thuật toán để giải quyết các bài toán phương pháp tính. Điều này bao gồm việc tìm nghiệm bằng các phương pháp Chia đôi, Lặp và Newton; tìm đa thức nội suy Newton và Lagrange; tìm và vẽ phương trình hồi quy tuyến tính, hồi quy hàm mũ và hồi quy mũ e; tính đạo hàm cho dữ liệu x, y và từ hàm số; thay đổi phương pháp tính đạo hàm; và tính tích phân hình thang và bằng phương pháp Simpson.

Trực quan hóa Dữ liệu: Chúng tôi sẽ sử dụng các công cụ trực quan hóa dữ liệu của Matlab để vẽ hàm số cần tìm nghiệm, vẽ phương trình hồi quy và hiển thị kết quả tính toán.

Kiểm tra và Đánh giá: Cuối cùng, chúng tôi sẽ kiểm tra và đánh giá ứng dụng của mình để đảm bảo rằng nó hoạt động chính xác và hiệu quả.

1.3. QUY TRÌNH PHÁT TRIỂN CỦA ỨNG DỤNG

Quy trình phát triển của ứng dụng bao gồm việc xác định yêu cầu, thiết kế giao diện, phát triển luận lý, kiểm thử và triển khai. Trong quá trình này, chúng tôi đã sử dụng Matlab App Designer để thiết kế giao diện và phát triển luận lý cho các bài toán phương pháp tính.

Chúng tôi đã thành công trong việc thiết kế và phát triển một ứng dụng để giải quyết các bài toán phương pháp tính. Cụ thể, chúng tôi đã thiết kế được các giao diện Tìm nghiệm, Nội suy, Hồi quy, Đạo hàm; tìm được nghiệm bằng các

phương pháp Chia đôi, Lặp, Newton; tìm được đa thức nội suy Newton và Lagrange; tìm được và vẽ phương trình hồi quy tuyến tính, hồi quy hàm mũ, hồi quy mũ e; tính được đạo hàm cho dữ liệu x, y và từ hàm số; thay đổi được phương pháp tính đạo hàm; và tính được tích phân hình thang và bằng phương pháp Simpson.

1.3.1 TỔNG QUAN VỀ MATLAB APP DESIGNER

Matlab App Designer là một công cụ mạnh mẽ cho phép chúng tôi tạo ra các ứng dụng có giao diện người dùng đồ họa (GUI) một cách dễ dàng và hiệu quả. Với App Designer, chúng tôi có thể tạo ra các giao diện người dùng trực quan và dễ sử dụng, và viết code để xử lý các bài toán phức tạp.

1.3.2 VAI TRÒ CỦA CÁC THÀNH VIÊN TRONG ĐỒ ÁN

Mỗi thành viên trong nhóm đều có vai trò quan trọng trong việc hoàn thành đồ án. Một số thành viên chịu trách nhiệm thiết kế giao diện, trong khi những người khác tập trung vào việc phát triển luận lý. Tất cả các thành viên đều tham gia vào quá trình kiểm thử và đánh giá ứng dụng.

CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI

PHẦN 1: THUẬT TOÁN CHƯƠNG TRÌNH

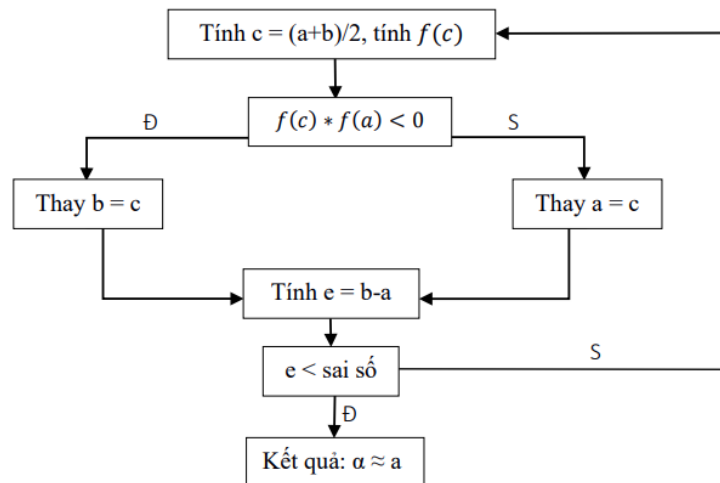
1.1 TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH

1.1.1 Phương pháp chia đôi

Bước 1: Xác định sai số cho phép.

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

Bước 3: Thực hiện theo lưu đồ thuật toán dưới đây:



Hình 1. Thuật toán tìm gần đúng nghiệm theo phương pháp chia đôi

Bước 4: Kết luận nghiệm $= a$ và số lần lặp.

1.1.2 Phương pháp lặp

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 $\varphi(x)$ [tức biến đổi x về vế trái, phần còn lại về vế phải chính là $\varphi(x)$].

Bước 4: Chọn 2 giá trị đầu là $x_1 = b$ và $x_0 = a$.

Bước 5: Tính (lặp) giá trị $x_i = \varphi(x_{i-1})$, với $i = 1, 2, 3 \dots$

đến khi $|x_i - x_{i-1}| < \text{sai số}$ thì dừng.

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

1.1.3 Phương pháp Newton (Tiếp tuyến)

Bước 1: Xác định sai số cho phép.

Bước 2: Tính $f'(x)$ và $f''(x)$.

Bước 3: Xác định khoảng phân li nghiệm $[a, b]$ trong đó f' và f'' không đổi dấu.

Bước 4: Chọn $x_0 = a$. Nếu $f(x_0) * f''(x_0) > 0$ thì sang Bước 5. Nếu không thỏa thì gán $x_0 = \frac{(x_0+b)}{2}$ và lặp lại đến khi thỏa.

Bước 5: Tính (lặp) giá trị $x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})}$ cho đến khi $|x_i - x_{i-1}| < \text{sai số}$ thì dừng.

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

1.2 ĐA THỨC NỘI SUY

1.2.1 Đa thức nội suy Lagrange

x	x_0	x_1	x_2	\dots	x_n
y	y_0	y_1	y_2	\dots	y_n

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

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

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

Tổng quát

$$p_n(x) = \sum_{i=0}^n L_i(x)f(x_i)$$

với

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

1.2.2 Đa thức Newton

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, x_k là: $y[x_i, x_j, x_k] = \frac{y[x_i, x_j] - y[x_j, x_k]}{x_i - x_k}$

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 :

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

Đa thức Newton lùi xuất phát từ nút x_n :

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

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

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

1.3 HỒI QUY

1.3.1 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 có được.

Phân 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

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

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

$$\begin{cases} \frac{\partial S_r}{\partial a_0} = 0 \\ \frac{\partial S_r}{\partial a_1} = 0 \end{cases} \Leftrightarrow \begin{cases} -2\Sigma(y_i - a_0 - a_1x_i) = 0 \\ -2\Sigma[(y_i - a_0 - a_1x_i)x_i] = 0 \end{cases}$$

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

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

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

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

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

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

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 khớp. r càng gần 1 thì kết quả hồi quy càng khớp

1.3.2 Hồi quy hàm mũ

Dạng $y = ax^b$

Lấy Logarit cơ số 10 hai vế: $\lg y = \lg a + b \lg x$

Đặt $Y = \lg y$; $A_0 = \lg a$; $A_1 = b$; $X = \lg x$

Đưa về dạng $Y = A_0 + A_1 X$

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

1.3.3 Hồi quy Logarit

Hồi quy hàm Logarit có dạng

$$y = a + b \cdot \ln(x)$$

, nơi mà a và b là các hằng số cần tìm. Đây là dạng của hồi quy tuyến tính, vì vậy chúng ta có thể sử dụng phương pháp hồi quy tuyến tính để tìm a và b .

Cần đảm bảo rằng các giá trị mà chúng ta lấy logarit phải luôn luôn dương. Nếu không, chúng ta cần thực hiện một số bước tiền xử lý dữ liệu trước khi áp dụng các phương pháp hồi quy này.

1.4 TÍNH GẦN ĐÚNG ĐẠO HÀM

Tính gần đúng đạo hàm nhờ áp dụng công thức Taylor, đạo hàm bậc 1 được tính bình thường như sau:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Từ công thức Taylor:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f^{(3)}(x) + \dots$$

Khi $|h|$ bé thì các số hạng cuối ở vế phải rất bé, ta có thể bỏ qua và có gần đúng:

$$f(x+h) \approx f(x) + hf'(x)$$

Như vậy:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Phương pháp sai phân thuận: xấp xỉ theo chiều tiến (từ $x_i \rightarrow x_{i+1}, x_{i+2} \dots$)

Sai số cắt cụt $O(h)$:

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

Sai số cắt cụt $O(h^2)$:

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

Phương pháp sai phân ngược, xấp xỉ theo chiều lùi (từ $x_i \rightarrow x_{i-1}, x_{i-2} \dots$)

Sai số cắt cụt $O(h)$:

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

Sai số cắt cụt $O(h^2)$:

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

Phương pháp sai phân trung tâm $(x_{i+1} - x_{i-1}, x_{i+2} - x_{i-2}, \dots)$

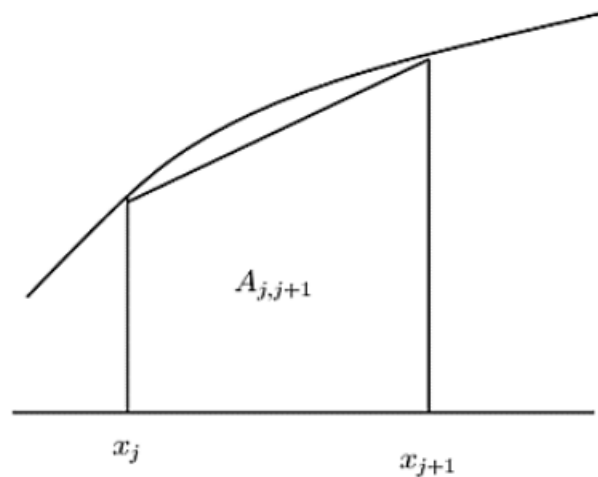
Sai số cắt cụt $O(h^2)$:

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

1.5 TÍNH GẦN ĐÚNG TÍCH PHÂN

1.5.1 Tích phân 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.



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

Công thức hình thang chia khoảng $[a, b]$ thành N đoạn con bằng nhau:

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

1.5.2 Công thức Simpson (Simpson's 1/3)

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.

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

Công thức Simpson's 1/3 chia đoạn $[a, b]$ thành $2N$ đoạn con bằng nhau:

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

1.5.3 Công thức Simpson's 3/8

$$I = \int_a^b f(x) dx \approx \frac{3h}{8} \left[f(a) + f(b) + 2 \sum_{\substack{i=1 \\ i \bmod 3 = 0}}^{N-1} f(a + ih) + 3 \sum_{\substack{i=2 \\ i \bmod 3 \neq 0}}^{N-1} f(a + ih) \right]$$

Công thức Simpson's 3/8 chia đoạn $[a, b]$ thành $2N$ đoạn con bằng nhau:

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

PHẦN 2: CODE CHƯƠNG TRÌNH

2.1 CODE THUẬT TOÁN

2.1.1 TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH

Hàm MATLAB để tính gần đúng nghiệm phương trình bằng cách sử dụng phương chia đôi:

```
function [root, iteration] = Nghiem_PhuongPhapChiaDoi(f, a, b, epsilon)
    iteration = 0;
    while (b - a) > epsilon
        c = (a + b)/2;
        if f(c) == 0
            break;
        end
        %Nếu f(c) > 0 --> sign(f(c)) == 1 || f(c) < 0 --> sign(f(c)) == -1
        if sign(f(c)) == sign(f(a))
            a = c;
        else
            b = c;
        end
        iteration = iteration + 1;
    end
    root = (a + b)/2;
end
```

Hàm MATLAB để tính gần đúng nghiệm phương trình bằng cách sử dụng phương lặp:

```
function [root, iteration, message] = Nghiem_PhuongPhapLap(g, x0, b, epsilon, maxIter)
    message = '';
    syms x;
    gx = g(x);
    g_prime = diff(gx);
    g_prime_negative_sub = -g_prime;
    g_prime_negative = matlabFunction(g_prime_negative_sub);
    [~, g_max] = fminbnd(g_prime_negative, x0, b);
    if (g_max >= 1 || g_max <= -1)
        root = 'NaN';
        iteration = 'NaN';
        message = 'Hàm g(x) không hội tụ! Hãy tìm g(x) theo cách khác.';
        return;
    end

    iteration = 0;
    while iteration < maxIter
        x1 = g(x0);
        if abs(x1 - x0) < epsilon
            break;
        end
        x0 = x1;
        iteration = iteration + 1;
    end
    root = x1;
end
```

Hàm MATLAB để tính gần đúng nghiệm phương trình bằng cách sử dụng Newton (Tiếp tuyến):

```
function [root, iteration] = Nghiem_PhuongPhapNewton(f, x0, epsilon, maxIter)
    df = @(x) (f(x + epsilon) - f(x)) / epsilon; % đạo hàm xấp xỉ
    iteration = 0;
    while iteration < maxIter
        x1 = x0 - f(x0)/df(x0);
        if abs(x1 - x0) < epsilon
            break;
        end
        x0 = x1;
        iteration = iteration + 1;
    end
end
```

```
    root = x1;
end
```

2.1.2 ĐA THỨC NỘI SUY

Hàm (function) trong MATLAB để tìm giá trị nội suy sử dụng đa thức nội suy Lagrange:

```
function result = Lagrange(xa, ya, x)
    n = length(xa);
    sum = 0;
    for i = 1:n
        product = ya(i);
        for j = 1:n
            if i ~= j
                product = product * (x - xa(j)) / (xa(i) - xa(j));
            end
        end
        sum = sum + product;
    end
    result = sum;
end
```

Chương trình tìm đa thức nội suy Lagrange sử dụng kiểu dữ liệu syms trong MATLAB:

```
function L = LagrangeSymbolic(xa, ya)
    syms x;
    n = length(xa);
    L = 0;
    for i = 1:n
        P = 1;
        for j = 1:n
            if i ~= j
                P = P * (x - xa(j)) / (xa(i) - xa(j));
            end
        end
        L = L + P * ya(i);
    end
    L = simplify(L);
end
```

Hàm (function) trong MATLAB để tìm giá trị nội suy sử dụng đa thức nội suy Newton:

```
function result = NewtonInterpolation(xa, ya, x)
    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
    result = d;
end
```



```

        end
    end
    n = length(d);
    result = d(n);
    for i = n-1:-1:1
        result = result * (x - xa(i)) + d(i);
    end
end
end

```

Tìm biểu thức nội suy Newton sử dụng kiểu dữ liệu syms trong MATLAB.

```

function L = NewtonSymbolic(xa, ya)
    syms x;
    n = length(xa);
    L = ya(n);
    for i = n-1:-1:1
        L = L * (x - xa(i)) + ya(i);
    end
    L = simplify(L);
end
end

```

2.1.3 HỒI QUY

Hàm (function) Hồi Quy trong MATLAB:

```

function [a1, a0, r2] = HoiQuy_Regress(x, y)
    n = length(x);
    sumx = 0;
    sumy = 0;
    sumxy = 0;
    sumx2 = 0;
    st = 0;
    sr = 0;
    for i = 1:n
        sumx = sumx + x(i);
        sumy = sumy + y(i);
        sumxy = sumxy + x(i)*y(i);
        sumx2 = sumx2 + x(i)*x(i);
    end

    xm = sumx/n;
    ym = sumy/n;
    a1 = (n*sumxy - sumx*sumy) / (n*sumx2 - sumx*sumx);
    a0 = ym - a1*xm;
    for i = 1:n
        st = st + (y(i) - ym)^2;
        sr = sr + (y(i) - a1*x(i) - a0)^2;
    end
    r2 = (st - sr)/st;
end
end

```

Hàm `HoiQuy_Regress(x, y)` dùng để tính các hệ số hồi quy tuyến tính và hệ số xác định cho một mảng x, y . Hàm này sử dụng phương pháp bình phương tối thiểu để tìm ra đường thẳng có dạng $y = a_1 \cdot x + a_0$ sao cho tổng bình phương các sai số giữa các giá trị thực tế và dự đoán là nhỏ nhất. Hàm này cũng tính hệ số xác định r^2 để đánh giá mức độ phù hợp của mô hình hồi quy với dữ liệu.

Hàm này trả về ba giá trị: a_1 là hệ số góc của đường thẳng, a_0 là hệ số chặn của đường thẳng, và r^2 là hệ số xác định của mô hình. Ta có thể sử dụng hàm này để phân tích mối quan hệ tuyến tính giữa hai biến x, y và dự đoán giá trị của y khi biết giá trị của x .

2.1.4 TÍNH GẦN ĐÚNG ĐẠO HÀM

Hàm (function) trong MATLAB để tính đạo hàm của một hàm số tại một điểm bằng cách sử dụng các công thức xấp xỉ tiến, lùi, hoặc trung tâm:

```
function [dx] = tinhdaohamtuhamso(fx,a,h,pp)
    syms x
    fx = str2sym(fx);
    fx = symfun(fx, [x]);
    if pp == "Xấp xỉ tiến"
        dx = (4*fx(a+h) - 3*fx(a) - fx(a+2*h)) / (2*h);
    elseif pp == "Xấp xỉ trung tâm"
        dx = (fx(a+h)-fx(a-h)) / (2*h);
    else
        dx = (3*fx(a) - 4*fx(a-h) + fx(a-2*h)) / (2*h);
    end
end
```

Hàm này tính đạo hàm của một hàm số fx tại điểm a với bước h nhỏ. Tham số pp xác định phương pháp xấp xỉ được sử dụng: “Xấp xỉ tiến”, “Xấp xỉ trung tâm” hoặc phương pháp khác.

```
function [dx] = tinhdaohamtuhamso2(fx,a,h,pp)
    syms x
    fx = str2sym(fx);
    fx = symfun(fx, [x]);
    if pp == "Xấp xỉ tiến"
        dx = (fx(a+h) - fx(a)) / h;
    elseif pp == "Xấp xỉ lùi"
        dx = (fx(a) - fx(a-h)) / h;
    else
        dx = 0;
    end
end
```

Hàm này cũng tính đạo hàm của một hàm số $f(x)$ tại điểm a với bước h nhỏ. Tuy nhiên, nó chỉ sử dụng phương pháp “Xấp xỉ tiến” và “Xấp xỉ lùi”.

```
function [dx] = tinhdaohamtumang(mangX,mangY,a,h,pp)
[c] = find(mangX == a);
if pp == "Xấp xỉ tiến"
    if c > (length(mangX) - 2)
        msgbox("Error");
        dx = 0;
    else
        dx = (4*mangY(c+1) - 3*mangY(c) - mangY(c+2))/(2*h);
    end
elseif pp == "Xấp xỉ trung tâm"
    dx = (mangY(c+1)-mangY(c-1))/(2*h);
else
    if c <= 2
        msgbox("Chọn giá trị cần tính đạo hàm từ thứ 3 trở đi trong mảng X");
        dx = 0;
    else
        dx = (3*mangY(c) - 4*mangY(c-1) + mangY(c-2))/(2*h);
    end
end
end
```

Hàm này tính đạo hàm của một mảng dữ liệu mangY tại điểm a (với a là một phần tử trong mangX) với bước h nhỏ. Tham số pp xác định phương pháp xấp xỉ được sử dụng: “Xấp xỉ tiến”, “Xấp xỉ trung tâm” hoặc phương pháp khác.

```
function [dx] = tinhdaohamtumang2(mangX,mangY,a,h,pp)
[c] = find(mangX == a);
if pp == "Xấp xỉ tiến"
    if c == length(mangX)
        msgbox("Error");
        dx = 0;
    else
        dx = (mangY(c+1) - mangY(c))/h;
    end
elseif pp == "Xấp xỉ lùi"
    if c == 1
        msgbox("Chọn giá trị cần tính đạo hàm từ thứ 2 trở đi trong mảng X");
        dx = 0;
    else
        dx = (mangY(c) - mangY(c-1))/h;
    end
else
    dx = 0;
end
end
```

Hàm này cũng tính đạo hàm của một mảng dữ liệu mangY tại điểm a (với a là một phần tử trong mangX) với bước h nhỏ. Tuy nhiên, nó chỉ sử dụng phương pháp “Xấp xỉ tiến” và “Xấp xỉ lùi”.

Lưu ý rằng, trong tất cả các hàm trên, nếu không thể tính toán đạo hàm (ví dụ: nếu a nằm ở rìa của mangX), hàm sẽ trả về lỗi và giá trị dx sẽ được gán bằng 0.

2.1.5 TÍNH GẦN ĐÚNG TÍCH PHÂN

Hàm MATLAB tính tích phân của một hàm số $f(x)$ từ a đến b sử dụng phương pháp hình thang:

```
function y = TichPhan_HinhThang(fx,a,b,N)
    syms x
    f = str2sym(fx);
    f = symfun(f, [x]);
    tong = 0;
    h = (b-a)/N;

    for i = 1:1:N-1
        tong = tong + 2*f(a + i*h);
    end
    y = (h/2)*(f(a)+f(b)+tong);
end
```

Hàm MATLAB tính đa thức nội suy Lagrange của một mảng dữ liệu y_a tại các điểm x_a .

```
function result = TichPhan_Lagrange(xa, ya)
    syms x
    n = length(xa);
    sum = 0;

    for i = 1:n
        product = ya(i);
        for j = 1:n
            if i ~= j
                product = product*(x - xa(j))/(xa(i) - xa(j));
            end
        end
        sum = sum + product;
    end
    result = simplify(sum);
end
```

Hàm MATLAB tính tích phân của một hàm số $f(x)$ từ a đến b sử dụng phương pháp Simpson 1/3:

```
function y = TichPhan_Simpson1phan3(fx,a,b,N)
    syms x
    f = str2sym(fx);
    f = symfun(f, [x]);
    tongLe = 0;
    tongChan = 0;
    h = (b-a)/N;
    for i = 1:N-1
        if mod(i,2) ~= 0
            tongLe = tongLe + 4*f(a+i*h);
        else
            tongChan = tongChan + 2*f(a+i*h);
        end
    end
    y = (h/3)*(f(a)+f(b)+tongLe+tongChan);
end
```

```

        end
    end
    y = (h/3)*(f(a)+f(b)+tongLe+tongChan);
end

```

Hàm MATLAB tính tích phân của một hàm số $f(x)$ từ a đến b sử dụng phương pháp Simpson 3/8:

```

function y = TichPhan_Simpson3phan8 (fx,a,b,N)
    syms x
    f = str2sym(fx);
    f = symfun(f, [x]);
    tong3 = 0;
    tongconlai = 0;
    h = (b-a)/N;
    for i = 1:N-1
        if(mod(i,3) == 0)
            tong3 = tong3 + f(a+i*h);
        else
            tongconlai = tongconlai + f(a+i*h);
        end
    end
    y = (h*3/8)*(f(a)+f(b)+2*tong3+3*tongconlai);
end

```

Lưu ý rằng: trong tất cả các hàm trên, N là số lượng phần tử trong mảng hoặc số lượng phân đoạn cần chia khi tính tích phân.

2.2 CODE GIAO DIỆN

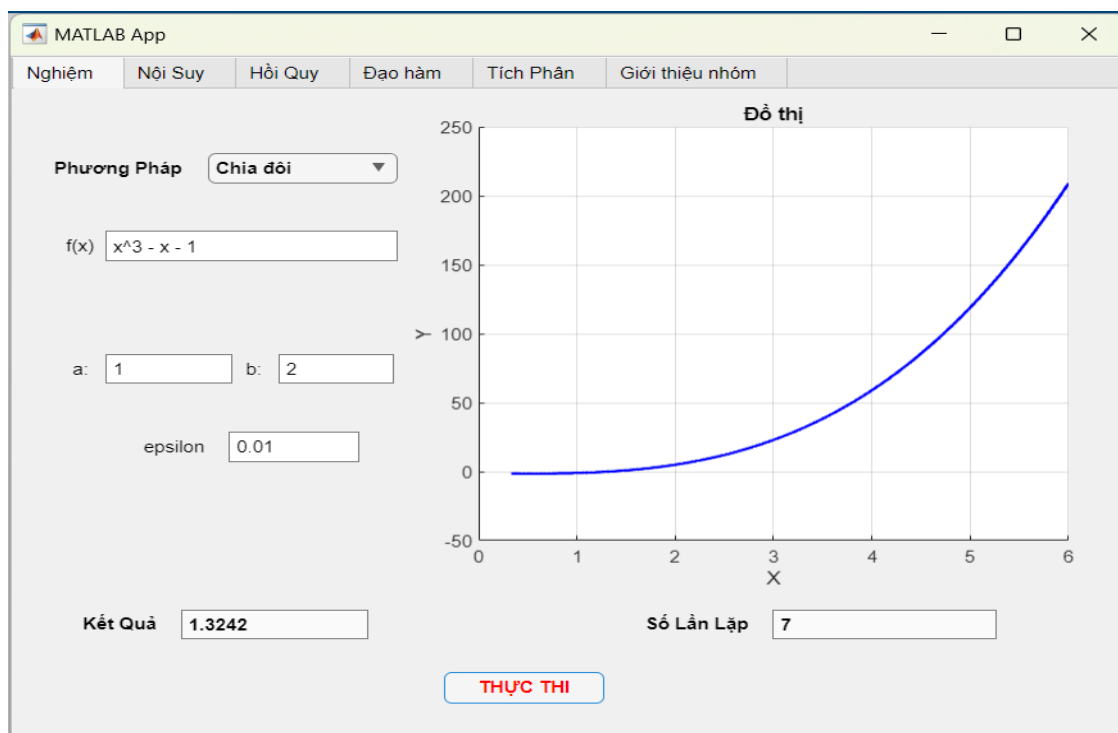
2.2.1 TẠO GIAO DIỆN NGƯỜI DÙNG

Trong App Designer, ta kéo và thả các thành phần từ bảng Components vào cửa sổ trống để tạo giao diện người dùng. Dưới đây là hình ảnh về giao diện người dùng trong đồ án của nhóm:



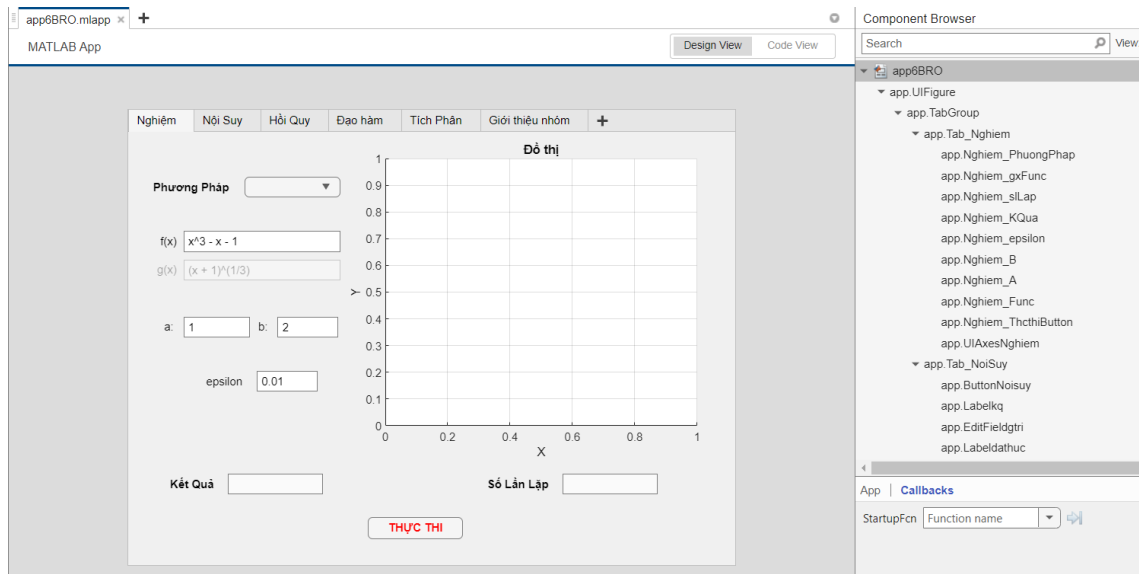
Hình 2. Giao diện hiển thị với người dùng

GIAO DIỆN TÌM NGHIỆM:



Hình 3. Giao diện tìm nghiệm

Giao diện gồm các đối tượng:



Hình 4. Giao diện Tìm Nghiệm và tên các components

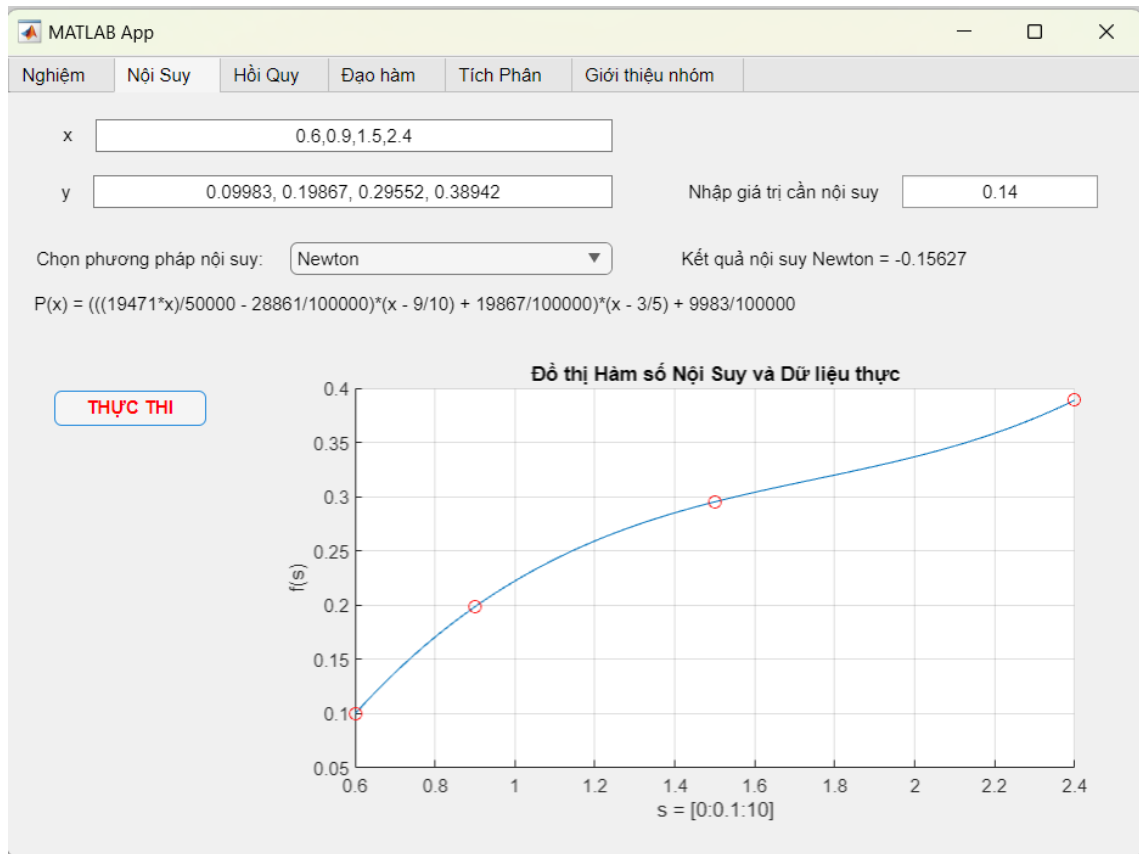
7 Edit Field (Text) để nhập phương trình $f(x)$, hàm lặp $g(x)$, khoảng phân li nghiệm a, b , sai số cho phép ϵ , kết quả tìm nghiệm và số lần lặp.

1 Drop Down để chọn phương pháp tìm nghiệm: Chia đôi, Lặp, Newton.

1 Axes dùng để vẽ hàm số cần tìm nghiệm.

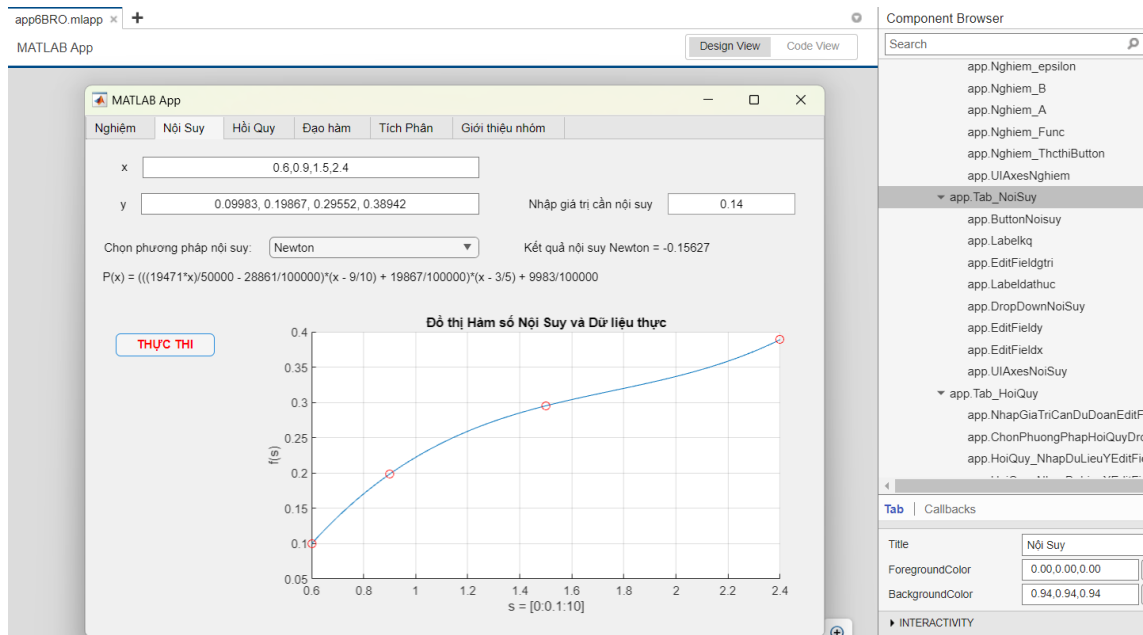
1 Button với thuộc tính text là “THỰC THI” có chức năng tính toán khi chạy chương trình.

GIAO DIỆN NỘI SUY:



Hình 5. Giao diện Nội Suy

Giao diện gồm các đối tượng:



Hình 6. Giao diện Nội Suy và tên các components

2 Edit Field (Text) để nhập dữ liệu x, dữ liệu y.

1 Edit Field (Numeric) dùng để nhập giá trị cần nội suy.

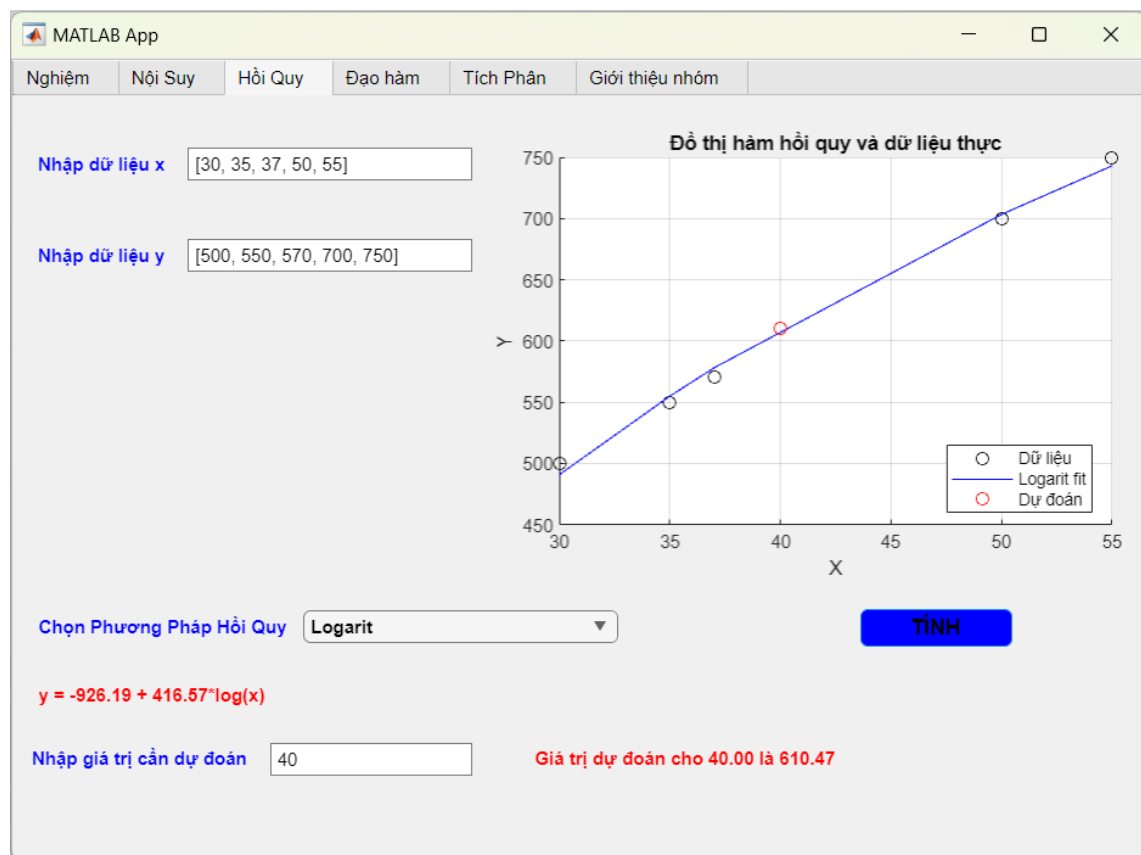
1 Drop Down để chọn phương pháp nội suy: Newton, Lagrange.

2 Label dùng để hiển thị kết quả nội suy và kết quả đa thức nội suy.

1 Axes để vẽ đồ thị hàm số nội suy và dữ liệu thực.

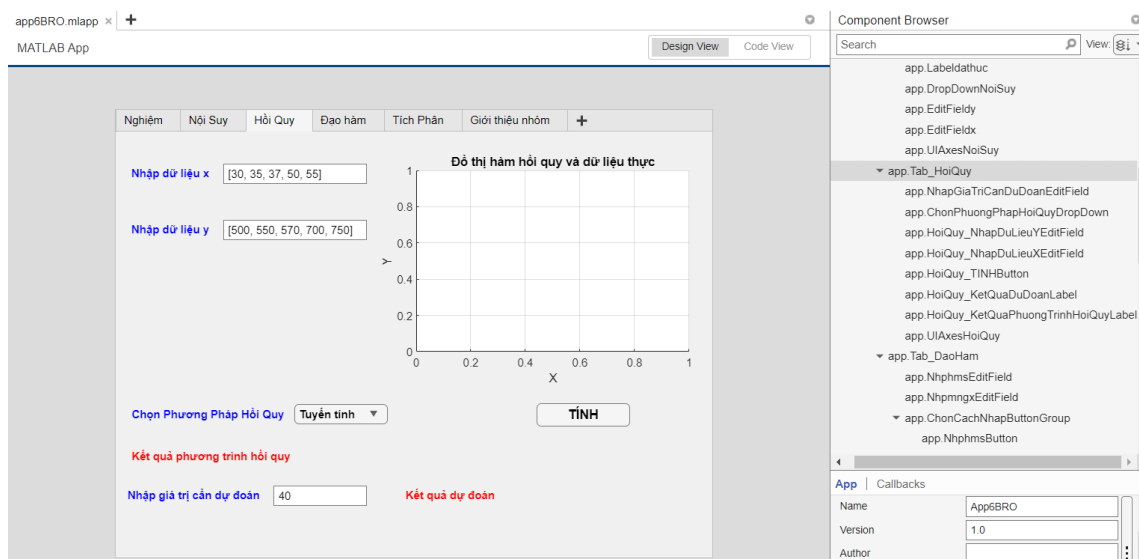
1 Button với thuộc tính text “THỰC THI” khi ấn chương trình sẽ được chạy.

GIAO DIỆN HỒI QUY:



Hình 7. Giao diện Hồi Quy

Giao diện gồm các đối tượng:



Hình 8. Giao diện Hồi Quy và tên các components

3 Edit Field (Text) để nhập mảng dữ liệu x, mảng y và nhập giá trị cần dự đoán.

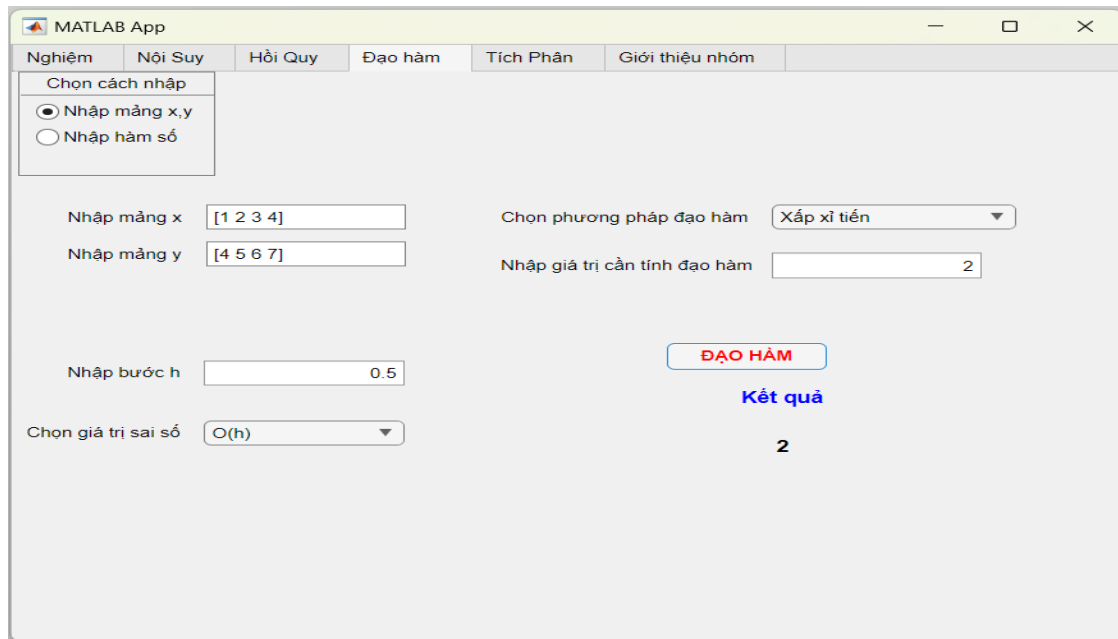
1 Drop Down để chọn phương pháp hồi quy: Tuyến tính, hàm mũ, Logarit.

2 Label dùng để hiển thị kết quả phương trình hồi quy và kết quả dự đoán.

1 Axes dùng để vẽ đồ thị hàm hồi quy và dữ liệu thực.

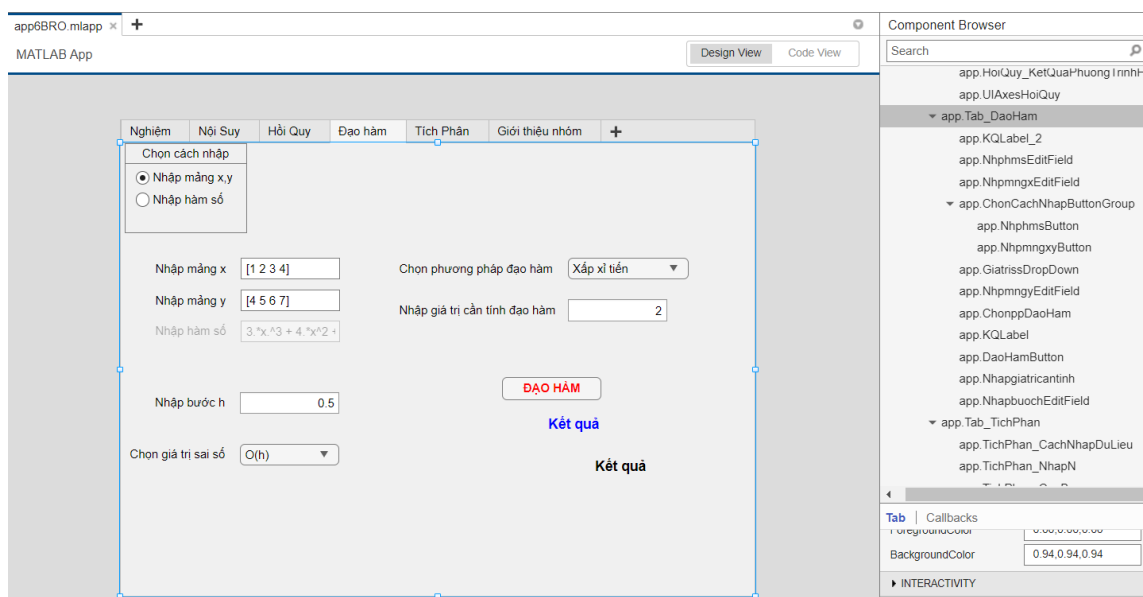
1 Button với thuộc tính text là “TÍNH” khi ấn thì chương trình sẽ thực thi.

GIAO DIỆN TÍNH ĐẠO HÀM:



Hình 9. Giao diện ĐẠO HÀM

Giao diện gồm các đối tượng:



Hình 10. Giao diện ĐẠO HÀM và tên các components

3 Edit Field (Text) để nhập mảng x, mảng y và nhập hàm số cần tính đạo hàm.

2 Edit Field (Numeric) để nhập bước h và nhập giá trị cần tính đạo hàm.

2 Drop Down, một dùng để chọn giá trị sai số $O(h)$ hoặc $O(h^2)$; một dùng để chọn phương pháp đạo hàm: Xấp xỉ tiến, Xấp xỉ lùi, Xấp xỉ trung tâm.

1 Radio Button Group dùng để chọn phương thức tính đạo hàm: tính đạo hàm bằng mảng x, y hoặc tính đạo hàm của hàm số.

1 Label hiển thị kết quả tính đạo hàm.

1 Button với thuộc tính text “ĐẠO HÀM” khi ấn sẽ thực thi chương trình.

GIAO DIỆN TÍNH TÍCH PHÂN:

MATLAB App

Nghiem Noi Suy Hoi Quy Dao ham **Tich Phan** Gioi thieu nhom

Cach Nhap: Mảng dữ liệu

Mảng x: 1, 2, 3, 4, 5

Mảng y(x): 1/2, 1/5, 1/10, 1/17, 1/26

Hoặc

Nhap ham so: $1/(1+x^2)$

Chon phuong phap Tich phan

☒ Hinh thang

☐ Simpson 1/3

☐ Simpson 3/8

Can a: 1

Can b: 5

Nhap N (n doan): 4

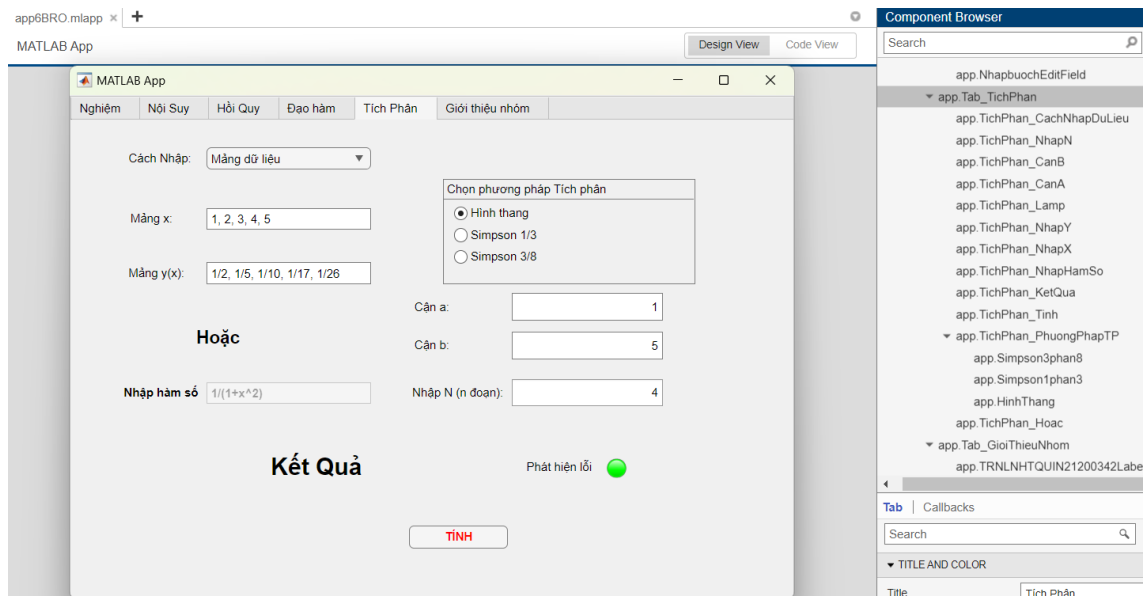
0.62805

Phat hien loi

TINH

Hình 11. Giao diện TÍCH PHÂN

Giao diện gồm các đối tượng:



Hình 12. Giao diện TÍCH PHÂN và tên các components

5 Edit Field (Text) để nhập mảng dữ liệu x, mảng y(x), nhập cận [a, b] để tính tích phân, nhập hàm số cần tính tích phân.

1 Edit Field (Numeric) để nhập N (n đoạn).

1 Lamp để phát hiện lỗi.

1 Radio Button Group dùng để chọn phương pháp tính tích phân: Hình thang, Simpson 1/3, Simpson 3/8.

1 Drop Down dùng để chọn phương thức tính tích phân: tính tích phân từ mảng dữ liệu hoặc tính tích phân của hàm số.

1 Label dùng để hiển thị kết quả tính tích phân.

1 Button với thuộc tính text “TÍNH” khi ấn chương trình sẽ được thực thi.

GIAO DIỆN GIỚI THIỆU NHÓM:



Hình 13. Giao diện GIỚI THIỆU NHÓM

2.2.2 TẠO SỰ KIỆN CALLBACK

2.2.2.1 Tab Thử nghiệm

Tạo sự kiện Callback cho nút nhấn “THỰC THI”:

Chuột phải vào nút “THỰC THI” → Chọn Callbacks → chọn Add ButtonPushedFcn callback để tạo function `Nghiem_ThcthiButtonPushed`. Nội dung function này như sau:

```
% Button pushed function: Nghiem_ThcthiButton
function Nghiem_ThcthiButtonPushed(app, event)
    maxIter = 20; %Số lần lặp tối đa

    % Kiểm tra xem có ô nào rỗng không
    if isempty(app.Nghiem_PhuongPhap.Value)
        uialert(app.UIFigure, 'Chưa chọn phương pháp', 'Lỗi');
        return;
    end

    % Kiểm tra ô g(x) và f(x) có rỗng hoặc không nhập dung định
```

```

% đang hay không?
if strcmp(app.Nghiem_PhuongPhap.Value, 'Lặp')
    try
        g = str2func(['@(x)', app.Nghiem_gxFunc.Value]);

        catch
            uialert(app.UIFigure, 'Nhập sai định dạng (function g(x)) hoặc đang
rỗng!', 'Lỗi');
            return;
        end
    else
        try
            f = str2func(['@(x)', app.Nghiem_Func.Value]);

            catch
                uialert(app.UIFigure, 'Nhập sai định dạng (function f(x)) hoặc đang
rỗng!', 'Lỗi');
                return;
            end
        end

        % Gán Các Value Từ Text Field Vào Các Variable
        %Kiểm tra có nhập đúng định dạng f(x) không

        % Kiểm tra có nhập đúng định dạng a và b hoặc đang rỗng không
        try
            a = str2double(app.Nghiem_A.Value);
            if isnan(a)
                goto catch;
            end
        catch
            uialert(app.UIFigure, 'Nhập sai định dạng (a (integer)) hoặc đang
rỗng!', 'Lỗi');
            return;
        end

        try
            b = str2double(app.Nghiem_B.Value);
            if isnan(b)
                goto catch;
            end
        catch
            uialert(app.UIFigure, 'Nhập sai định dạng (b (integer)) hoặc đang
rỗng!', 'Lỗi');
            return;
        end

        % Kiểm tra có nhập đúng định dạng epsilon hoặc đang rỗng không?
        try
            epsilon = str2double(app.Nghiem_epsilon.Value);
            if isnan(epsilon)
                goto catch;
            end
        catch
            uialert(app.UIFigure, 'Nhập sai định dạng (epsilon (integer)) hoặc đang
rỗng!', 'Lỗi');
            return;
        end

        % kiểm tra khoảng phân ly có hợp lệ hay không?
        if a >= b
            app.Nghiem_KQua.Value = 'NaN';
            app.Nghiem_slLap.Value = 'NaN';
            message = 'Khoảng phân ly không hợp lệ hoặc không tồn tại khoảng phân ly
với các giá trị a và b';
            uialert(app.UIFigure, message, 'Lỗi');
            return;
        end

        % kiểm tra trong khoảng phân ly có nghiệm hay không?
        if strcmp(app.Nghiem_PhuongPhap.Value, 'Lặp')
            if sign(g(a)) == sign(g(b))

```

```

%               app.Nghiem_KQua.Value = 'NaN';
%               app.Nghiem_slLap.Value = 'NaN';
%               message = 'Khoảng phân ly không chứa nghiệm hoặc chứa một số
nghiệm chẵn. Vui lòng chọn khoảng phân ly khác.';
%               uialert(app.UIFigure, message, 'Lỗi');
%               return;
%           end
elseif sign(f(a)) == sign(f(b))
    app.Nghiem_KQua.Value = 'NaN';
    app.Nghiem_slLap.Value = 'NaN';
    message = 'Khoảng phân ly không chứa nghiệm hoặc chứa một số nghiệm
chẵn. Vui lòng chọn khoảng phân ly khác.';
    uialert(app.UIFigure, message, 'Lỗi');
    return;
end

% Chọn hàm để function của phương pháp nào để giải phương trình
switch app.Nghiem_PhuongPhap.Value
    case 'Chia đôi'
        [root, iteration] = Nghiem_PhuongPhapChiaDoi(f, a, b, epsilon);

    case 'Lặp'
        [root, iteration, message] = Nghiem_PhuongPhapLap(g, a, b, epsilon,
maxIter);
        if ~isempty(message)
            uialert(app.UIFigure, message, 'Lỗi');
            return;
        end

    case 'Newton'
        [root, iteration] = Nghiem_PhuongPhapNewton(f, a, epsilon, maxIter);
end

app.Nghiem_KQua.Value = num2str(root);
app.Nghiem_slLap.Value = num2str(iteration);

% Vẽ đồ thị
x = linspace(a/3, b*3, 1000);
if strcmp(app.Nghiem_PhuongPhap.Value, 'Lặp')
    y = arrayfun(g, x);
else
    y = arrayfun(f, x);
end
plot(app.UIAxesNghiem, x, y, 'b-', 'LineWidth', 1.5);
grid(app.UIAxesNghiem, 'on');
end

```

Hàm `Nghiem_ThcthiButtonPushed` được gọi khi người dùng nhấn nút “THỰC THI” trên giao diện người dùng. Hàm này sẽ thực hiện các công việc sau:

Kiểm tra xem người dùng đã nhập đầy đủ và đúng định dạng các thông số cần thiết cho phương pháp giải chọn, bao gồm: hàm số $f(x)$ hoặc $g(x)$, khoảng phân ly $[a, b]$, sai số ϵ và phương pháp giải.

Nếu có bất kỳ ô nào rỗng hoặc sai định dạng, hàm sẽ hiển thị thông báo lỗi và dừng lại. Nếu không, hàm sẽ gọi hàm con tương ứng với phương pháp giải chọn, là `Nghiem_PhuongPhapChiaDoi` hoặc `Nghiem_PhuongPhapLap`, để tìm nghiệm của phương trình trong khoảng phân ly với sai số cho trước.

Sau đó, hàm sẽ hiển thị kết quả nghiệm và số lần lặp trên giao diện người dùng. Dựa vào giá trị của `app.Nghiem_PhuongPhap.Value`, chọn một trong ba hàm con để giải phương trình bằng phương pháp chia đôi, lặp hoặc Newton. Các hàm con này sẽ trả về nghiệm root và số lần lặp iteration của phương trình. Nếu phương pháp lặp không hội tụ, hàm con này còn trả về một thông báo lỗi message.

Nếu có thông báo lỗi từ phương pháp lặp, hiển thị thông báo lỗi trên giao diện người dùng và dừng lại. Nếu không, hiển thị kết quả nghiệm và số lần lặp trên giao diện người dùng.

Vẽ đồ thị của hàm số $f(x)$ hoặc $g(x)$ tùy theo phương pháp giải chọn, trên một khoảng rộng hơn khoảng phân ly ban đầu. Sử dụng các hàm `linspace`, `arrayfun`, `plot` và `grid` để tạo và hiển thị đồ thị.

Tạo sự kiện Callback cho Drop Down “Phương Pháp”:

```
% Value changed function: Nghiem_PhuongPhap
function Nghiem_PhuongPhapValueChanged(app, event)
    selectedOption = app.Nghiem_PhuongPhap.Value;

    % Kiểm tra giá trị của dropdown và ẩn/hiển thị EditField dựa trên
    điều kiện
    if strcmp(selectedOption, 'Lặp') % Thay 'Option1' bằng giá trị của
    phần tử bạn muốn kiểm tra
        app.Nghiem_gxFunc.Visible = 'on'; % Ẩn EditField
        app.gxEtEditFieldLabel.Visible = 'on';

        app.Nghiem_Func.Enable = "off";

        uialert(app.UIFigure, "Người dùng phải tự tìm hàm lặp khi sử
        dụng phương pháp lặp!!!", "Lưu Ý");

    else
        app.Nghiem_gxFunc.Visible = 'off'; % Hiện thị EditField
        app.gxEtEditFieldLabel.Visible = 'off';

        app.Nghiem_Func.Enable = "on";
    end
end
```

Hàm `Nghiem_PhuongPhapValueChanged` được gọi khi người dùng thay đổi giá trị của một Drop Down menu có tên là `Nghiem_PhuongPhap` trên giao diện người dùng. Hàm này sẽ thực hiện các công việc sau:

Lấy giá trị của Drop Down menu và gán cho biến `selectedOption`.

Kiểm tra xem giá trị của `selectedOption` có phải là “Lặp” hay không. Nếu có, thì:

Hiển thị một EditField có tên là `Nghiem_gxFunc` và một Label có tên là `gxEditFieldLabel` để người dùng nhập hàm lặp $g(x)$.

Vô hiệu hóa một EditField có tên là `Nghiem_Func` để người dùng không thể nhập hàm số $f(x)$.

Hiển thị một thông báo cảnh báo trên giao diện người dùng rằng người dùng phải tự tìm hàm lặp khi sử dụng phương pháp lặp.

Nếu không, thì:

Ẩn một EditField có tên là `Nghiem_gxFunc` và một Label có tên là `gxEditFieldLabel` để người dùng không thể nhập hàm lặp $g(x)$.

Kích hoạt một EditField có tên là `Nghiem_Func` để người dùng có thể nhập hàm số $f(x)$.

2.2.2.2 Tab Nội Suy

Tạo sự kiện Callback cho nút nhấn:

Chuột phải vào nút “THỰC THI” → Chọn Callbacks → chọn Add ButtonPushedFcn callback để tạo function `ButtonNoisuyPushed`. Nội dung function này như sau:

```
% Button pushed function: ButtonNoisuy
function ButtonNoisuyPushed(app, event)
    value = app.DropDownNoiSuy.Value;
    if strcmp(value, 'Newton')
        xa = str2num(app.EditFieldx.Value);
        ya = str2num(app.EditFielddy.Value);
        x = app.EditFieldgtri.Value;
        L = NewtonSymbolic(xa, ya);
        app.Labelmathuc.Text = "P(x) = " + char(L);
        result = NewtonInterpolation(xa, ya, x);
        app.Labelkq.Text = "Kết quả nội suy Newton = " + num2str(result);
        % Tạo một dãy giá trị x từ min(xa) đến max(xa)
        x_plot = linspace(min(xa), max(xa), 1000);

        % Tính giá trị y tương ứng với mỗi giá trị x_plot
        if strcmp(value, 'Newton')
            y_plot = arrayfun(@(x) NewtonInterpolation(xa, ya, x), x_plot);
        end

        % Vẽ đồ thị hàm số nội suy trên UIAxes1
        plot(app.UIAxesNoiSuy, x_plot, y_plot);

        % Đánh dấu dữ liệu thực trên đồ thị
        hold(app.UIAxesNoiSuy, 'on');
```

```

        plot(app.UIAxesNoiSuy, xa, ya, 'ro');
        hold(app.UIAxesNoiSuy, 'off');

    else
        xa = str2num(app.EditFieldx.Value);
        ya = str2num(app.EditFiieldy.Value);
        x = app.EditFieldgtri.Value;
        L = LagrangeSymbolic(xa, ya);
        app.Labeldathuc.Text = "P(x) = " + char(L);
        result = Lagrange(xa, ya, x);
        app.Labelkq.Text = "Kết quả nội suy Lagrange = " + num2str(result);

        % Tạo một dãy giá trị x từ min(xa) đến max(xa)
        x_plot = linspace(min(xa), max(xa), 1000);

        % Tính giá trị y tương ứng với mỗi giá trị x_plot
        y_plot = arrayfun(@(x) Lagrange(xa, ya, x), x_plot);

        % Vẽ đồ thị hàm số nội suy trên UIAxes1
        plot(app.UIAxesNoiSuy, x_plot, y_plot);

        % Đánh dấu dữ liệu thực trên đồ thị
        hold(app.UIAxesNoiSuy, 'on');
        plot(app.UIAxesNoiSuy, xa, ya, 'ro');
        hold(app.UIAxesNoiSuy, 'off');
    end
end

```

Hàm `ButtonNoisuyPushed` được gọi khi người dùng nhấn nút “THỰC THI” trên giao diện người dùng. Hàm này sẽ thực hiện các công việc sau:

Lấy giá trị của Drop Down menu có tên là `DropDownNoiSuy` và gán cho biến `value`. Kiểm tra xem giá trị của `value` có phải là “Newton” hay không.

Nếu có, thì:

Lấy các giá trị của các `EditField` có tên là `EditFieldx`, `EditFiieldy` và `EditFieldgtri` và chuyển đổi chúng thành các mảng số `xa`, `ya` và số `x`.

Gọi hàm con `NewtonSymbolic` để tìm đa thức nội suy Newton của mảng `ya` tại các điểm `xa` và gán cho biến `L`.

Hiển thị đa thức nội suy Newton trên một `Label` có tên là `Labeldathuc`.

Gọi hàm con `NewtonInterpolation` để tìm giá trị nội suy Newton của hàm số tại điểm `x` và gán cho biến `result`.

Hiển thị kết quả nội suy Newton trên một `Label` có tên là `Labelkq`.

Tạo một dãy giá trị `x_plot` từ giá trị nhỏ nhất đến giá trị lớn nhất của `xa`. Tính giá trị `y_plot` tương ứng với mỗi giá trị `x_plot` bằng cách gọi hàm `arrayfun` với hàm `NewtonInterpolation`.

Vẽ đồ thị của hàm số nội suy trên một UIAxes có tên là UIAxesNoiSuy bằng cách gọi hàm plot. Đánh dấu dữ liệu thực xa và ya trên đồ thị bằng cách gọi hàm hold và plot.

Nếu không, thì:

Lấy các giá trị của các EditField có tên là EditFieldx, EditFiely và EditFieldgtri và chuyển đổi chúng thành các mảng số xa, ya và số x.

Gọi hàm con LagrangeSymbolic để tìm đa thức nội suy Lagrange của mảng ya tại các điểm xa và gán cho biến L.

Hiển thị đa thức nội suy Lagrange trên một Label có tên là Labeldathuc.

Gọi hàm con Lagrange để tìm giá trị nội suy Lagrange của hàm số tại điểm x và gán cho biến result. Hiển thị kết quả nội suy Lagrange trên một Label có tên là Labelkq.

Tạo một dãy giá trị x_plot từ giá trị nhỏ nhất đến giá trị lớn nhất của xa.

Tính giá trị y_plot tương ứng với mỗi giá trị x_plot bằng cách gọi hàm arrayfun với hàm Lagrange.

Vẽ đồ thị của hàm số nội suy trên một UIAxes có tên là UIAxesNoiSuy bằng cách gọi hàm plot. Đánh dấu dữ liệu thực xa và ya trên đồ thị bằng cách gọi hàm hold và plot.

2.2.2.3 Tab Hồi Quy

Tạo sự kiện Callback cho nút nhấn:

Chuột phải vào nút “TÍNH” → Chọn Callbacks → chọn Add ButtonPushedFcn callback để tạo function HoiQuy_TINHButtonPushed. Nội dung function này như sau:

```
% Button pushed function: HoiQuy_TINHButton
function HoiQuy_TINHButtonPushed(app, event)
    % Chuyển đổi giá trị nhập vào từ người dùng thành số
    x = str2num(app.HoiQuy_NhapDuLieuXEditField.Value);
    y = str2num(app.HoiQuy_NhapDuLieuYEditField.Value);
    x_pred = str2num(app.NhapGiaTriCanDuDoanEditField.Value);

    %Kiểm tra kích thước của ma trận x và y
    if length(x) ~= length(y)
```

```

        uialert(app.UIFigure, 'Kích thước của ma trận x và y phải bằng nhau!',
"Lỗi!!!");
        return;
    end

    % Khởi tạo x_pos và yfit
    x_pos = x;
    yfit = zeros(size(x));

    % Lấy phương pháp hồi quy từ người dùng
    method = app.ChonPhuongPhapHoiQuyDropDown.Value;

    switch method
        case 'Tuyến tính'
            % Hồi quy tuyến tính
            [a1, a0, r2] = HoiQuy_Regress(x, y);
            yfit = a0 + a1*x;
            y_pred = a0 + a1*x_pred;
            app.HoiQuy_KetQuaPhuongTrinhHoiQuyLabel.Text = sprintf('y = %.2f +
%.2f*x, \t R^2 = %.2f', a0, a1, r2);
            app.HoiQuy_TINHButton.BackgroundColor = [255 0 0]/255;

        case 'Hàm mũ'
            % Hồi quy hàm mũ
            X = [ones(length(x),1) x'];
            B = X\log(y)';
            a = exp(B(1));
            b = B(2);
            yfit = a*exp(b*x);
            y_pred = a*exp(b*x_pred);
            app.HoiQuy_KetQuaPhuongTrinhHoiQuyLabel.Text = sprintf('y =
%.2f*exp(%.2f*x)', a, b);
            app.HoiQuy_TINHButton.BackgroundColor = [0 255 0]/255;

        case 'Logarit'
            % Hồi quy logarit

            % Lọc ra các giá trị x dương
            idx = x > 0;
            x_pos = x(idx);
            y_pos = y(idx);

            % Nối các mảng cột
            X = [ones(length(x_pos),1) log(x_pos)'];

            % Tính B
            B = X\y_pos';
            a = B(1);
            b = B(2);
            yfit = a + b*log(x_pos);
            y_pred = a + b*log(x_pred);
            app.HoiQuy_KetQuaPhuongTrinhHoiQuyLabel.Text = sprintf('y = %.2f +
%.2f*log(x)', a, b);
            app.HoiQuy_TINHButton.BackgroundColor = [0 0 255]/255;
        end

        % Cập nhật Label với giá trị dự đoán
        app.HoiQuy_KetQuaDuDoanLabel.Text = sprintf('Giá trị dự đoán cho %.2f là
%.2f', x_pred, y_pred);

        % Vẽ hàm hồi quy và dữ liệu lên đồ thị
        plot(app.UIAxesHoiQuy, x, y, 'ko', x_pos, yfit, 'b-');
        hold(app.UIAxesHoiQuy, 'on');
        plot(app.UIAxesHoiQuy, x_pred, y_pred, 'ro');
        hold(app.UIAxesHoiQuy, 'off');
        legend(app.UIAxesHoiQuy, 'Dữ liệu', sprintf('%s fit', method), 'Dự đoán',
'Location', 'best');
    end

```

Hàm `HoiQuy_TINHButtonPushed` được gọi khi người dùng nhấn nút “TÍNH” trên giao diện người dùng. Hàm này sẽ thực hiện các công việc sau:

Chuyển đổi các giá trị nhập vào từ người dùng thành các mảng số x , y và số x_pred . Kiểm tra xem kích thước của mảng x và y có bằng nhau hay không. Nếu không, hiển thị thông báo lỗi và dừng lại.

Khởi tạo một mảng số x_pos bằng với x và một mảng số $yfit$ bằng với một mảng số không có kích thước bằng với x . Lấy giá trị của Drop Down menu có tên là `ChonPhuongPhapHoiQuyDropDown` và gán cho biến `method`.

Dựa vào giá trị của `method`, chọn một trong ba hàm con để hồi quy bằng phương pháp tuyến tính, hàm mũ hoặc logarit. Các hàm con này sẽ trả về các hệ số hồi quy, giá trị $yfit$ tương ứng với mỗi giá trị x_pos và giá trị y_pred tương ứng với giá trị x_pred . Nếu phương pháp logarit được chọn, chỉ lấy các giá trị x và y dương.

Hiển thị phương trình hồi quy và giá trị dự đoán trên các Label có tên là `HoiQuy_KetQuaPhuongTrinhHoiQuyLabel` và `HoiQuy_KetQuaDuDoanLabel`. Thay đổi màu nền của nút “TÍNH” tùy theo phương pháp hồi quy.

Vẽ đồ thị của hàm số hồi quy và dữ liệu trên một UIAxes có tên là `UIAxesHoiQuy` bằng cách gọi các hàm `plot`, `hold` và `legend`. Đánh dấu giá trị dự đoán trên đồ thị bằng một điểm màu đỏ.

2.2.2.4 Tab ĐẠO HÀM

Tạo sự kiện Callback cho nút nhấn:

Chuột phải vào nút “ĐẠO HÀM” → Chọn Callbacks → chọn Add ButtonPushedFcn callback để tạo function `DaoHamButtonPushed`. Nội dung function này như sau:

```
% Button pushed function: DaoHamButton
function DaoHamButtonPushed(app, event)
    pp = app.ChonppDaoHam.Value;
    ss = app.GiatriissDropDown.Value;
    mangX = str2num(app.NhpmngxEditField.Value);
    mangY = str2num(app.NhpmngyEditField.Value);
    x = app.Nhapgiatricantinh.Value;
    fx = app.NhphmsEditField.Value;
    h = app.NhapbuochEditField.Value;
    c = find(mangX == x );
```

```

if (app.NhphmsButton.Value == 1)
    if (ss == "O(h^2)")
        [dx] = tinhdaohamtuhamso(fx,x,h,pp);
    else
        [dx] = tinhdaohamtuhamso2(fx,x,h,pp);
    end
else
    if (ss == "O(h^2)")
        [dx] = tinhdaohamtumang(mangX,mangY,x,h,pp);
    else
        [dx] = tinhdaohamtumang2(mangX,mangY,x,h,pp);
    end
end
dx = double(dx);
if (ss == "O(h)")
    if (pp == "Xấp xỉ trung tâm")
        app.KQLabel.Text = "Không thực hiện được";
    elseif (pp == "Xấp xỉ lùi")
        if c == 1
            app.KQLabel.Text = "Error";
        else
            app.KQLabel.Text = num2str(dx);
        end
    else
        if c == length(mangX)
            app.KQLabel.Text = "Error";
        else
            app.KQLabel.Text = num2str(dx);
        end
    end
else
    if (pp == "Xấp xỉ lùi")
        if c <= 2
            app.KQLabel.Text = "Error";
        else
            app.KQLabel.Text = num2str(dx);
        end
    else
        if c > (length(mangX)-2)
            app.KQLabel.Text = "Error";
        else
            app.KQLabel.Text = num2str(dx);
        end
    end
end
end
end

```

Hàm DaoHamButtonPushed được gọi khi người dùng nhấn nút “ĐẠO HÀM” trên giao diện người dùng. Hàm này sẽ thực hiện các công việc sau:

Lấy các giá trị của các đối tượng giao diện người dùng và chuyển đổi chúng thành các biến số hoặc chuỗi ký tự. Các đối tượng giao diện người dùng bao gồm:

ChonppDaoHam: một Drop Down menu để chọn phương pháp xấp xỉ đạo hàm, có thể là “Xấp xỉ trung tâm”, “Xấp xỉ tiến” hoặc “Xấp xỉ lùi”.

GiattrissDropDown: một Drop Down menu để chọn sai số của phương pháp xấp xỉ, có thể là “O(h)” hoặc “O(h^2)”.

NhpmngxEditField: một EditText để nhập mảng x của dữ liệu.

NhpmngyEditField: một EditText để nhập mảng y của dữ liệu.

Nhapgiatricantinh: một EditText để nhập giá trị x cần tính đạo hàm.

NhphmsEditField: một EditText để nhập hàm số cần tính đạo hàm.

NhapbuochEditField: một EditText để nhập bước h của phương pháp xấp xỉ.

NhphmsButton: một Button để chọn nhập hàm số hay không.

Tìm vị trí của giá trị x cần tính đạo hàm trong mảng x và gán cho biến c .

Kiểm tra xem người dùng có chọn nhập hàm số hay không bằng cách kiểm tra giá trị của NhphmsButton. Nếu có, thì: Kiểm tra xem sai số của phương pháp xấp xỉ là gì bằng cách kiểm tra giá trị của GiatrissDropDown.

Nếu là " $O(h^2)$ ", thì: Gọi hàm con tinhdaohamtuhamso để tính đạo hàm của hàm số tại x bằng phương pháp xấp xỉ có sai số $O(h^2)$ và gán cho biến dx . Hàm con này cần các tham số là hàm số, giá trị x , bước h và phương pháp xấp xỉ.

Nếu là " $O(h)$ ", thì: Gọi hàm con tinhdaohamtuhamso2 để tính đạo hàm của hàm số tại x bằng phương pháp xấp xỉ có sai số $O(h)$ và gán cho biến dx . Hàm con này cần các tham số là hàm số, giá trị x , bước h và phương pháp xấp xỉ.

Nếu không, thì: Kiểm tra xem sai số của phương pháp xấp xỉ là gì bằng cách kiểm tra giá trị của GiatrissDropDown.

Nếu là " $O(h^2)$ ", thì: Gọi hàm con tinhdaohamtumang để tính đạo hàm của mảng dữ liệu tại x bằng phương pháp xấp xỉ có sai số $O(h^2)$ và gán cho biến dx . Hàm con này cần các tham số là mảng x , mảng y , giá trị x , bước h và phương pháp xấp xỉ.

Nếu là " $O(h)$ ", thì: Gọi hàm con tinhdaohamtumang2 để tính đạo hàm của mảng dữ liệu tại x bằng phương pháp xấp xỉ có sai số $O(h)$ và gán cho biến dx . Hàm con này cần các tham số là mảng x , mảng y , giá trị x , bước h và phương pháp xấp xỉ.

Chuyển đổi biến dx thành kiểu số thực.

Kiểm tra xem sai số của phương pháp xấp xỉ là gì bằng cách kiểm tra giá trị của `GiatriissDropDown`.

Nếu là “ $O(h)$ ”, thì: Kiểm tra xem phương pháp xấp xỉ là gì bằng cách kiểm tra giá trị của `ChonppDaoHam`.

Nếu là “Xấp xỉ trung tâm”, thì: Hiện thị chuỗi ký tự “Không thực hiện được” trên một Label có tên là `KQLabel`. Đây là vì phương pháp xấp xỉ trung tâm không thể có sai số $O(h)$.

Nếu là “Xấp xỉ lùi”, thì: Kiểm tra xem vị trí của giá trị x cần tính đạo hàm có bằng 1 hay không. Nếu có, thì: Hiện thị chuỗi ký tự “Error” trên một Label có tên là `KQLabel`. Đây là vì phương pháp xấp xỉ lùi không thể tính đạo hàm tại điểm đầu tiên của mảng x . Nếu không, thì: Hiện thị giá trị của dx trên một Label có tên là `KQLabel`.

Nếu là “Xấp xỉ tiến”, thì: Kiểm tra xem vị trí của giá trị x cần tính đạo hàm có bằng độ dài của mảng x hay không. Nếu có, thì: Hiện thị chuỗi ký tự “Error” trên một Label có tên là `KQLabel`. Đây là vì phương pháp xấp xỉ tiến không thể tính đạo hàm tại điểm cuối cùng của mảng x . Nếu không, thì: Hiện thị giá trị của dx trên một Label có tên là `KQLabel`.

Nếu là “ $O(h^2)$ ”, thì: Kiểm tra xem phương pháp xấp xỉ là gì bằng cách kiểm tra giá trị của `ChonppDaoHam`.

Nếu là “Xấp xỉ lùi”, thì: Kiểm tra xem vị trí của giá trị x cần tính đạo hàm có nhỏ hơn hoặc bằng 2 hay không. Nếu có, thì: Hiện thị chuỗi ký tự “Error” trên một Label có tên là `KQLabel`. Đây là vì phương pháp xấp xỉ lùi không thể tính đạo hàm tại hai điểm đầu tiên của mảng x . Nếu không, thì: Hiện thị giá trị của dx trên một Label có tên là `KQLabel`.

Nếu là “Xấp xỉ tiến” hoặc “Xấp xỉ trung tâm”, thì: Kiểm tra xem vị trí của giá trị x cần tính đạo hàm có lớn hơn độ dài.

Tạo sự kiện Callback cho Radio Button Group “Chọn cách nhập”:

```
% Selection changed function: ChonCachNhapButtonGroup
function ChonCachNhapButtonGroupSelectionChanged(app, event)
    selectedButton = app.ChonCachNhapButtonGroup.SelectedObject;
    if (app.NhphmsButton.Value == 1)
        app.NhpmngxEditField.Visible = "off";
        app.NhpmngxEditFieldLabel.Visible = "off";
        app.NhpmngyEditField.Visible = "off";
        app.NhpmngyEditFieldLabel.Visible = "off";
        app.NhphmsEditField.Visible = "on";
        app.NhphmsEditFieldLabel.Visible = "on";
```

```

else
    app.NhpmngxEditField.Visible = "on";
    app.NhpmngxEditFieldLabel.Visible = "on";
    app.NhpmngyEditField.Visible = "on";
    app.NhpmngyEditFieldLabel.Visible = "on";
    app.NhphmsEditField.Visible = "off";
    app.NhphmsEditFieldLabel.Visible = "off";
end
end

```

Hàm ChonCachNhapButtonGroupSelectionChanged được gọi khi người dùng thay đổi lựa chọn của một ButtonGroup có tên là ChonCachNhapButtonGroup trên giao diện người dùng. Hàm này sẽ thực hiện các công việc sau:

Lấy đối tượng Button được chọn trong ButtonGroup và gán cho biến selectedButton.

Kiểm tra xem Button được chọn có phải là NhphmsButton hay không bằng cách kiểm tra giá trị của NhphmsButton. Nếu có, thì:

Ẩn các EditField có tên là NhpmngxEditField và NhpmngyEditField và các Label có tên là NhpmngxEditFieldLabel và NhpmngyEditFieldLabel để người dùng không thể nhập mảng x và y.

Hiển thị một EditField có tên là NhphmsEditField và một Label có tên là NhphmsEditFieldLabel để người dùng có thể nhập hàm số.

Nếu không, thì:

Hiển thị các EditField có tên là NhpmngxEditField và NhpmngyEditField và các Label có tên là NhpmngxEditFieldLabel và NhpmngyEditFieldLabel để người dùng có thể nhập mảng x và y.

Ẩn một EditField có tên là NhphmsEditField và một Label có tên là NhphmsEditFieldLabel để người dùng không thể nhập hàm số.

Đoạn chương trình này thực hiện chức năng là cho phép người dùng chọn cách nhập liệu là hàm số hay mảng dữ liệu bằng cách bật tắt các đối tượng giao diện người dùng tương ứng.

2.2.2.5 Tab TÍCH PHÂN

Tạo sự kiện Callback cho nút nhấn:

Chuột phải vào nút “TÍNH” → Chọn Callbacks → chọn Add ButtonPushedFcn callback để tạo function TichPhan_TinhButtonPushed. Nội dung function này như sau:

```
% Button pushed function: TichPhan_Tinh
function TichPhan_TinhButtonPushed(app, event)
    hamSo = app.TichPhan_NhapHamSo.Value;
    canA = str2num(app.TichPhan_CanA.Value);
    canB = str2num(app.TichPhan_CanB.Value);
    selectedOption = app.TichPhan_CachNhapDuLieu.Value;
    X = str2num(app.TichPhan_NhapX.Value);
    Y = str2num(app.TichPhan_NhapY.Value);
    N = app.TichPhan_NhapN.Value;
    % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    if strcmp(selectedOption, 'Hàm số')
        fx = app.TichPhan_NhapHamSo.Value;
        flag = 1; %bật cờ
    elseif strcmp(selectedOption, 'Mảng dữ liệu')
        fx = TichPhan_Lagrange(X, Y);
        fx = char(fx);
        flag = 1; %bật cờ
    end

    % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    app.TichPhan_Lamp.Color = 'red';
    if flag == 1
        app.TichPhan_Lamp.Color = 'green';

        if (app.Simpson3phan8.Value)
            ketqua = double(TichPhan_Simpson3phan8(fx, canA, canB, N));
            app.TichPhan_KetQua.Text = num2str(ketqua);
        elseif (app.HinhThang.Value)
            ketqua = double(TichPhan_HinhThang(fx, canA, canB, N));
            app.TichPhan_KetQua.Text = num2str(ketqua);
        elseif (app.Simpson1phan3.Value)
            ketqua = double(TichPhan_Simpson1phan3(fx, canA, canB, N));
            app.TichPhan_KetQua.Text = num2str(ketqua);
        end
    end

    clear fx;
end
```

Hàm TichPhan_TinhButtonPushed được gọi khi người dùng nhấn nút “TÍNH” trên giao diện người dùng. Đoạn chương trình này thực hiện chức năng là cho phép người dùng tính tích phân xác định của một hàm số hoặc một mảng dữ liệu bằng các phương pháp xấp xỉ và hiển thị kết quả trên giao diện người dùng.

Lấy các giá trị của các đối tượng giao diện người dùng và chuyển đổi chúng thành các biến số hoặc chuỗi ký tự. Các đối tượng giao diện người dùng bao gồm:

TichPhan_NhapHamSo: một EditField để nhập hàm số cần tính tích phân.

TichPhan_CanA: một EditField để nhập cận dưới của tích phân.

TichPhan_CanB: một EditField để nhập cận trên của tích phân.

TichPhan_CachNhapDuLieu: một Drop Down menu để chọn cách nhập liệu là hàm số hay mảng dữ liệu.

TichPhan_NhapX: một EditField để nhập mảng x của dữ liệu.

TichPhan_NhapY: một EditField để nhập mảng y của dữ liệu.

TichPhan_NhapN: một EditField để nhập số lượng các đoạn chia nhỏ của phương pháp xấp xỉ.

Simpson3phan8: một Button để chọn phương pháp Simpson 3/8.

HinhThang: một Button để chọn phương pháp Hình thang.

Simpson1phan3: một Button để chọn phương pháp Simpson 1/3.

Kiểm tra xem cách nhập liệu là gì bằng cách kiểm tra giá trị của TichPhan_CachNhapDuLieu.

Nếu là “Hàm số”, thì: Lấy giá trị của TichPhan_NhapHamSo và gán cho biến fx. Bật cờ flag bằng 1.

Nếu là “Mảng dữ liệu”, thì: Gọi hàm con TichPhan_Lagrange để tìm đa thức nội suy Lagrange của mảng Y tại các điểm X và gán cho biến fx. Hàm con này cần các tham số là mảng X và mảng Y. Chuyển đổi biến fx thành xâu kí tự. Bật cờ flag bằng 1. Đổi màu của một Lamp có tên là TichPhan_Lamp thành màu đỏ.

Nếu cờ flag bằng 1, thì: Đổi màu của Lamp thành màu xanh.

Kiểm tra xem phương pháp tính tích phân là gì bằng cách kiểm tra giá trị của các Button Simpson3phan8, HinhThang và Simpson1phan3. Dựa vào giá trị của Button được chọn, gọi một trong ba hàm con để tính tích phân xác định bằng phương pháp Simpson 3/8, Hình thang hoặc Simpson 1/3 và gán cho biến ketqua. Các hàm con này cần các tham số là hàm số hoặc đa thức nội suy, cận dưới, cận trên và số lượng các đoạn chia nhỏ.

Hiển thị giá trị của ketqua trên một Label có tên là TichPhan_KetQua. Xóa biến fx.

Tạo sự kiện Callback cho Drop Down “Cách Nhập”:

```
% Value changed function: TichPhan_CachNhapDuLieu
function TichPhan_CachNhapDuLieuValueChanged(app, event)
```

```

selectedOption = app.TichPhan_CachNhapDuLieu.Value;

% Kiểm tra giá trị của dropdown và ẩn/hiển thị EditField dựa trên điều kiện
% Thay 'Option1' bằng giá trị của phần tử bạn muốn kiểm tra
if strcmp(selectedOption, 'Mảng dữ liệu')
    app.TichPhan_NhapX.Enable = 'on'; % Ẩn EditField
    app.TichPhan_NhapY.Enable = 'on';

    app.TichPhan_NhapHamSo.Enable = "off";

else
    app.TichPhan_NhapX.Enable = 'off'; % Hiện thị EditField
    app.TichPhan_NhapY.Enable = 'off';

    app.TichPhan_NhapHamSo.Enable = "on";

end
end

```

Hàm `TichPhan_CachNhapDuLieuValueChanged` được gọi khi người dùng thay đổi giá trị của một Drop Down menu có tên là `TichPhan_CachNhapDuLieu` trên giao diện người dùng. Đoạn chương trình này thực hiện chức năng là cho phép người dùng chọn cách nhập liệu là hàm số hay mảng dữ liệu bằng cách bật tắt các đối tượng giao diện người dùng tương ứng:

Lấy giá trị của Drop Down menu và gán cho biến `selectedOption`.

Kiểm tra xem giá trị của Drop Down menu là gì bằng cách so sánh nó với chuỗi ký tự “Mảng dữ liệu” bằng hàm `strcmp`.

Nếu bằng nhau, thì: Bật các EditField có tên là `TichPhan_NhapX` và `TichPhan_NhapY` để người dùng có thể nhập mảng x và y . Tắt một EditField có tên là `TichPhan_NhapHamSo` để người dùng không thể nhập hàm số.

Nếu không bằng nhau, thì: Tắt các EditField có tên là `TichPhan_NhapX` và `TichPhan_NhapY` để người dùng không thể nhập mảng x và y . Bật một EditField có tên là `TichPhan_NhapHamSo` để người dùng có thể nhập hàm số.

PHẦN 3: CHẠY CHƯƠNG TRÌNH TÍNH

Để chạy được đoạn chương trình, ta cần phải đặt tất cả các file liên quan vào cùng một thư mục. Cụ thể:

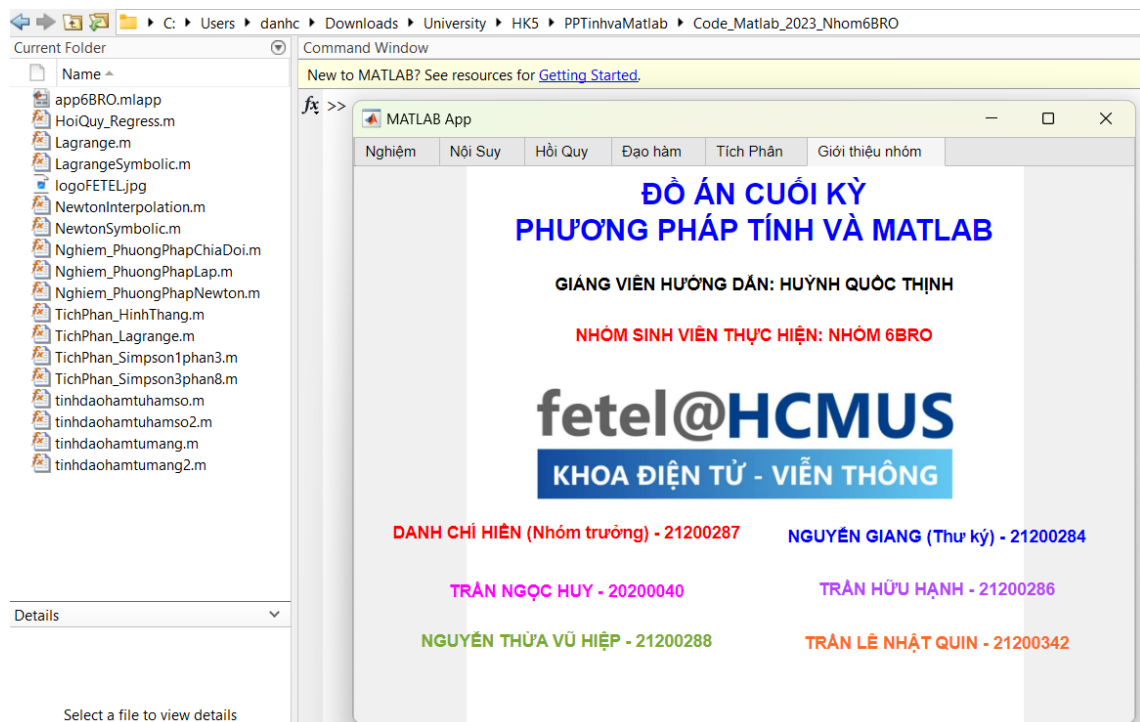
File ứng dụng App Designer: Đây là file chứa mã nguồn của ứng dụng, bao gồm cả các hàm Callbacks đã định nghĩa. File này thường có phần mở rộng là `(.mlapp)`.

Các hàm bên ngoài: Đây là các hàm mà ta gọi trong các hàm Callbacks, bao gồm: `Nghiem_PhuongPhapChiaDoi`, `Nghiem_PhuongPhapLap`,

Nghiem_PhuongPhapNewton;Lagrange,LagrangeSymbolic;Newton Interpolation, NewtonSymbolic, HoiQuy_Regress, tinhdaohamtumang, tinhdaohamtumang2, tinhdaohamtuhams0, tinhdaohamtuhams02;TichPhan_HinhThang,TichPhan_Lagrange, TichPhan_Simpson1phan3,TichPhan_Simpson3phan8. Mỗi hàm này được định nghĩa trong một file riêng biệt với tên giống như tên hàm, và phần mở rộng là (.m).

Thư mục làm việc của Matlab: Khi chạy ứng dụng, Matlab sẽ tìm kiếm các hàm trong thư mục làm việc hiện tại. Do đó, cần đảm bảo rằng thư mục làm việc của Matlab chính là thư mục chứa tất cả các file trên.

Sau khi đã đặt đúng các file, ta có thể mở file ứng dụng App Designer (.mlapp) và chạy ứng dụng. Khi bạn nhấn các Button, các hàm Callbacks sẽ được gọi và thực hiện các tính toán theo như ta đã định nghĩa.



Hình 14. Đặt tất cả các file vào cùng một thư mục

3.1 TÍNH GẦN ĐÚNG NGHIỆM PHƯƠNG TRÌNH

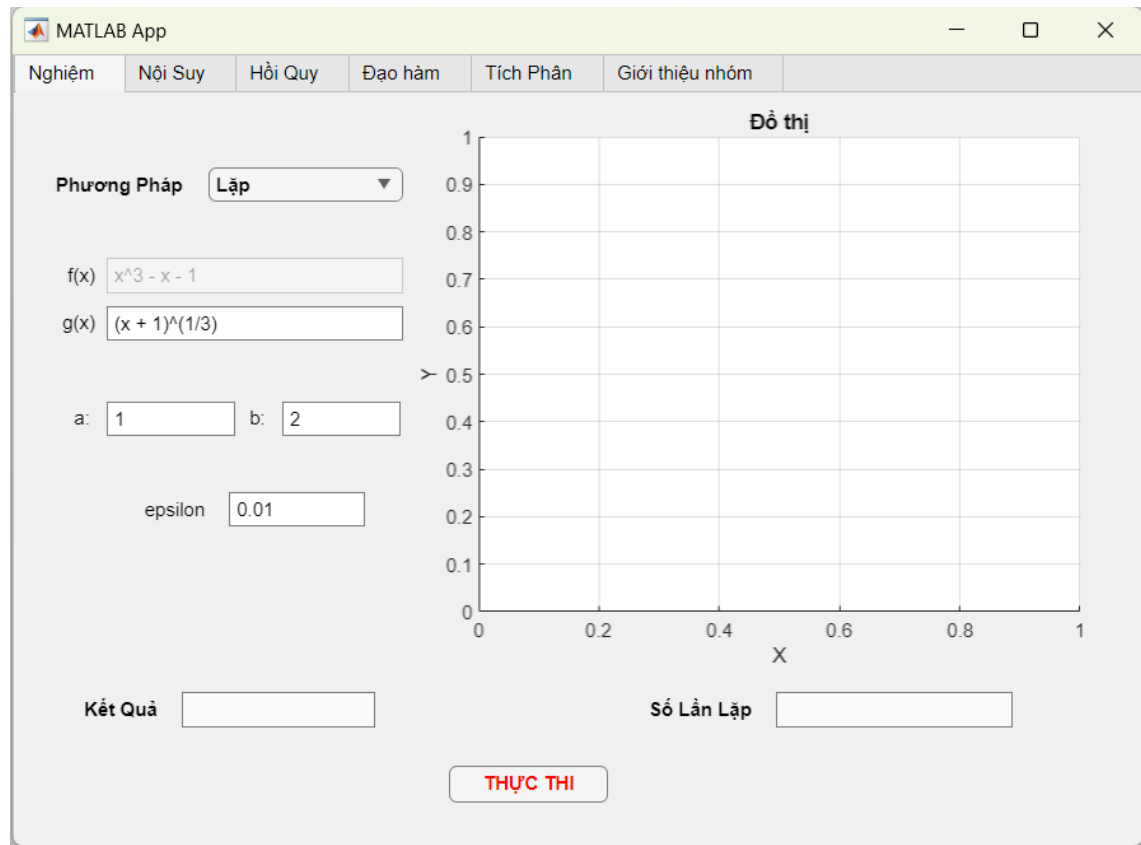
VD1: Tìm nghiệm gần đúng và số lần lặp của phương trình $x^3 - x = 1$ bằng Tab Nghiệm, với khoảng phân li nghiệm [1, 2] và sai số không lớn hơn 0.01. Phương pháp nào cho số lần lặp ít nhất. Vì sao?

Thực hành:

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

$$f(x): x^3 - x - 1 \quad ; \quad g(x): (x + 1)^{1/3}$$

$$a: 1 \quad ; \quad b: 2 \quad ; \quad \text{epsilon: } 0.01$$

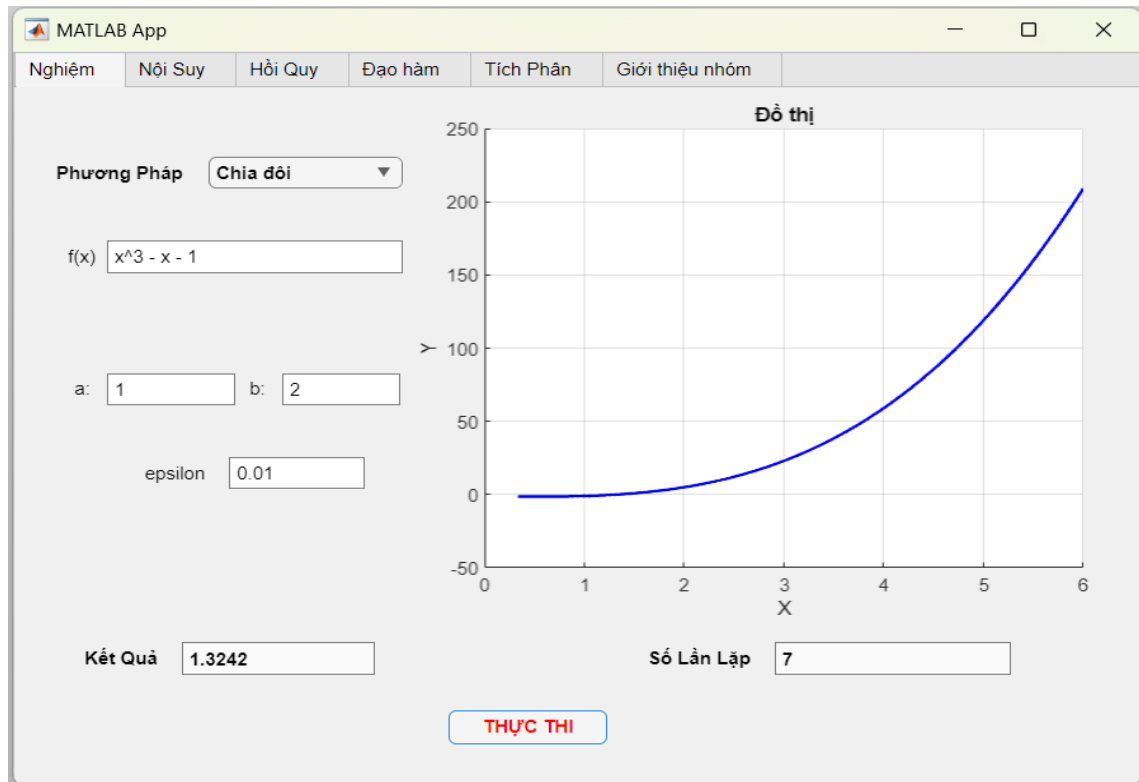


Hình 15. Nhập các thông số vào giao diện

Tại Drop Down Menu (Chọn Phương Pháp): "Chia đôi" hoặc "Lặp" hoặc "Newton".

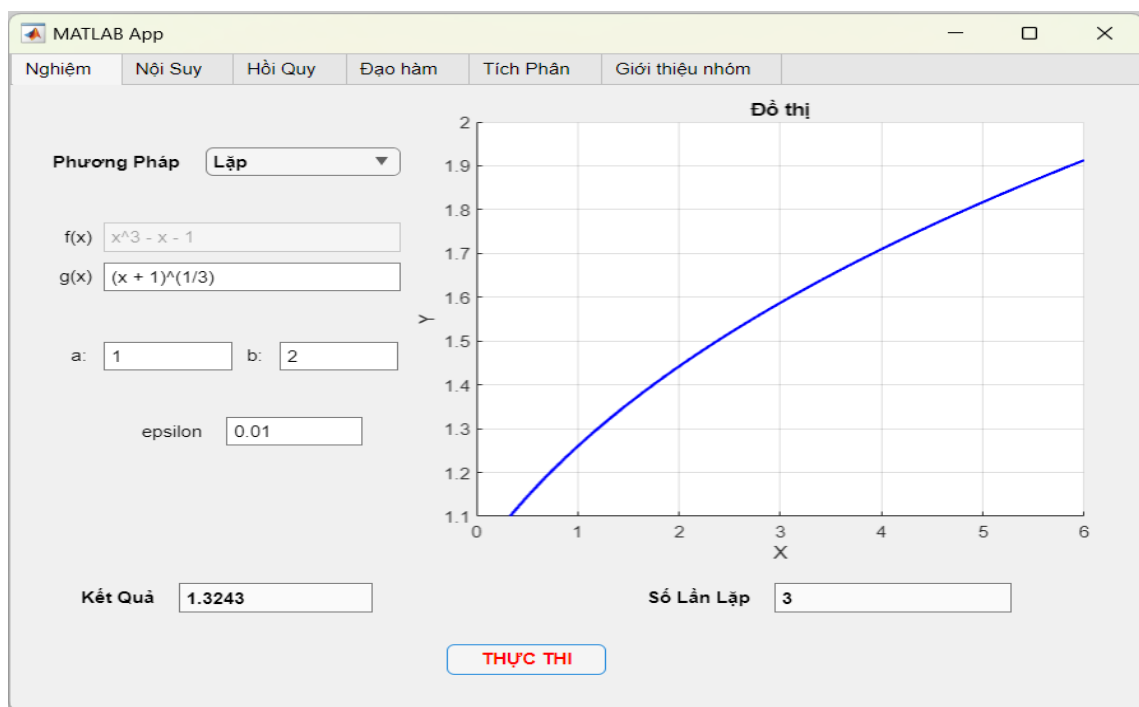
Nhấn Button “THỰC THI”. Đoạn code sẽ được thực thi, tính toán giá trị nghiệm, số lần lặp của từng phương pháp và vẽ được hàm số cần tìm nghiệm theo từng phương pháp.

Kết quả sẽ được hiển thị trên các Label và Axes tương ứng.

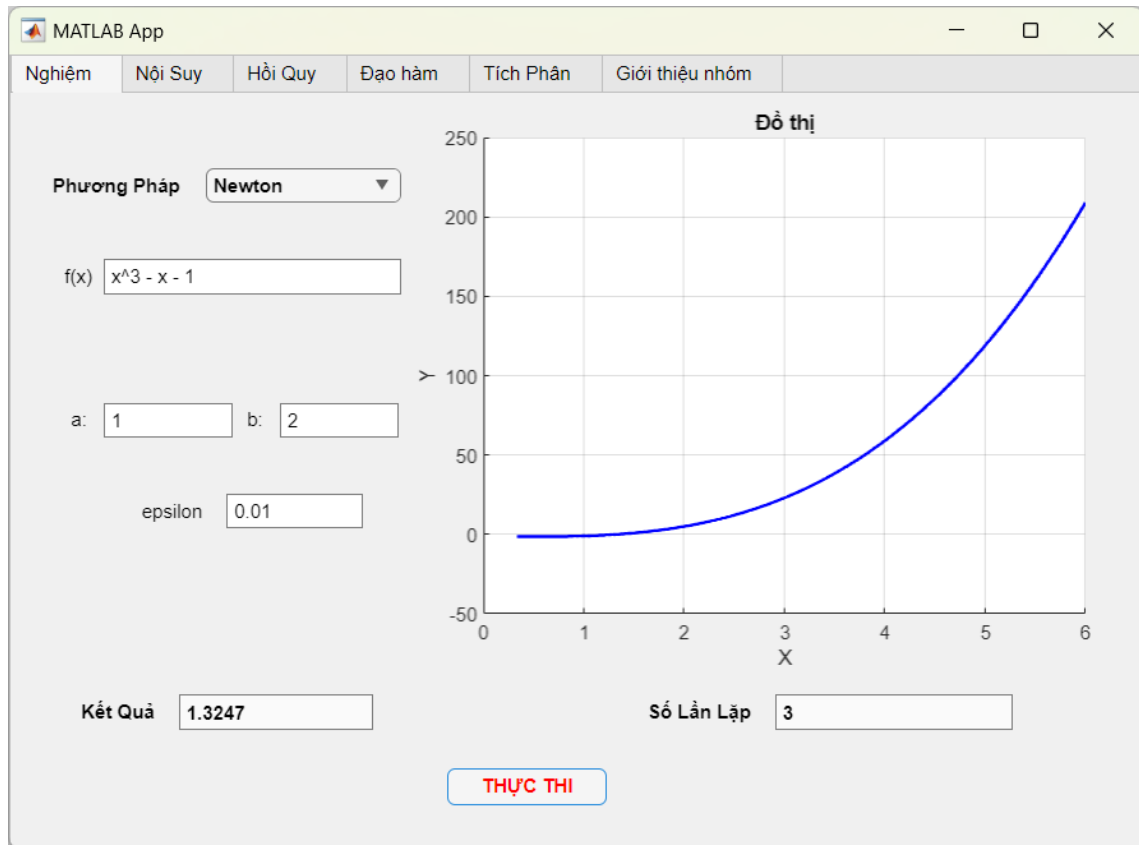


Hình 16. Kết quả đạt được bằng phương pháp chia đôi sau khi ấn "THỰC THI"

Lưu ý: Khi chuyển sang Phương Pháp “Lặp”, ta phải nhập hàm lặp $g(x)$.



Hình 17. Kết quả đạt được bằng phương pháp lặp sau khi ấn "THỰC THI"



Hình 18. Kết quả đạt được bằng phương pháp Newton sau khi ấn "THỰC THI"

Sau khi thực hiện các bước trên, chúng ta sẽ có nghiệm gần đúng và số lần lặp cho mỗi phương pháp. Phương pháp cho số lần lặp ít nhất là phương pháp tiếp tuyến (Newton). Vì về nguyên tắc, phương pháp tiếp tuyến (Newton) tìm nghiệm của một hàm số bằng cách xấp xỉ hàm số đó bằng một đường thẳng tiếp tuyến và sử dụng giao điểm của đường thẳng đó với trục hoành làm nghiệm tiếp theo. Phương pháp này thường hội tụ nhanh nếu điểm khởi tạo gần với nghiệm thực sự và hàm số có đạo hàm không quá nhỏ tại điểm đó.

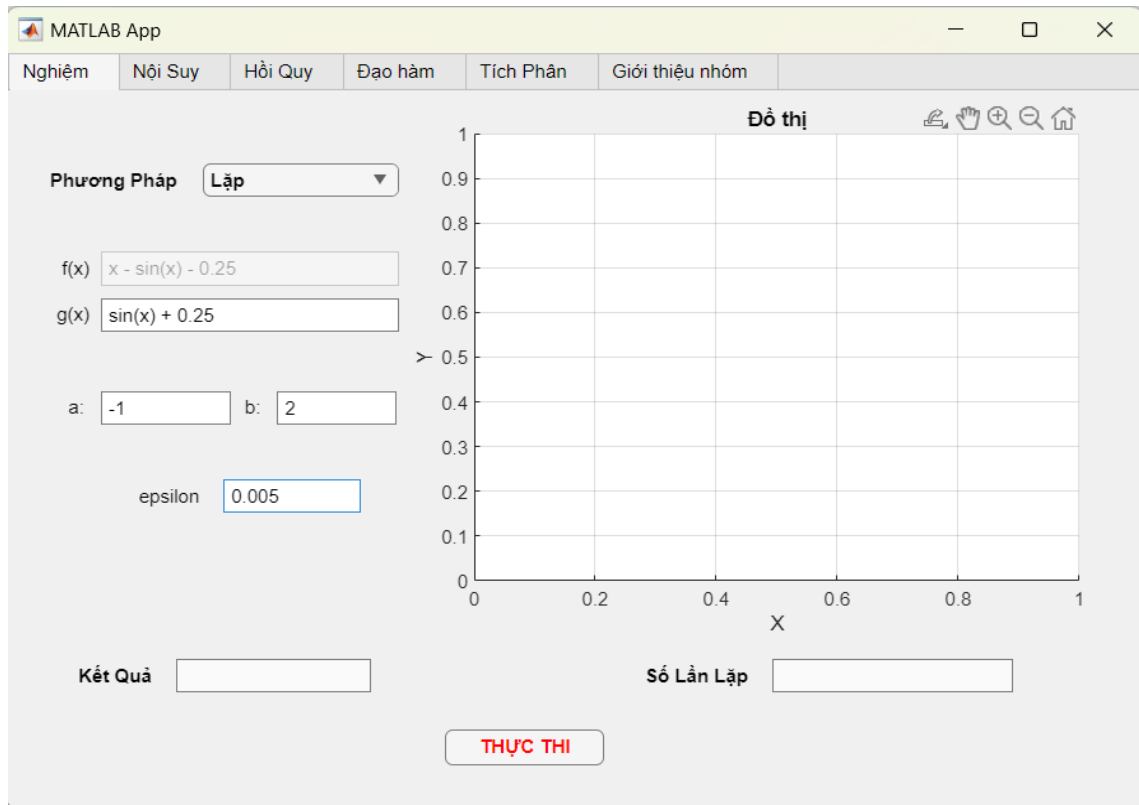
VD2: Tìm nghiệm gần đúng và số lần lặp của phương trình $x - \sin x = 0.25$ bằng Tab Nghiệm, với khoảng phân li nghiệm $[-1, 2]$ và sai số không lớn hơn 0.005. Phương pháp nào cho số lần lặp nhiều nhất. Vì sao?

Thực hành:

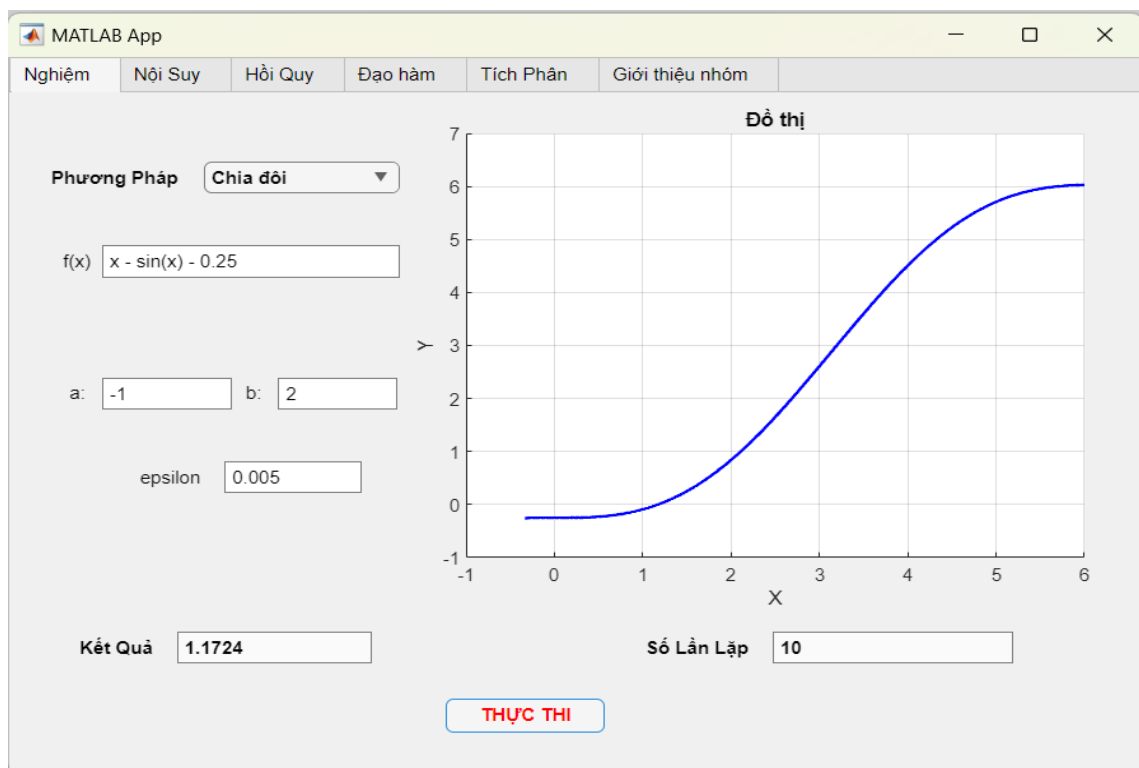
Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

$$f(x): x - \sin(x) - 0.25 \quad ; \quad g(x): \sin(x) + 0.25$$

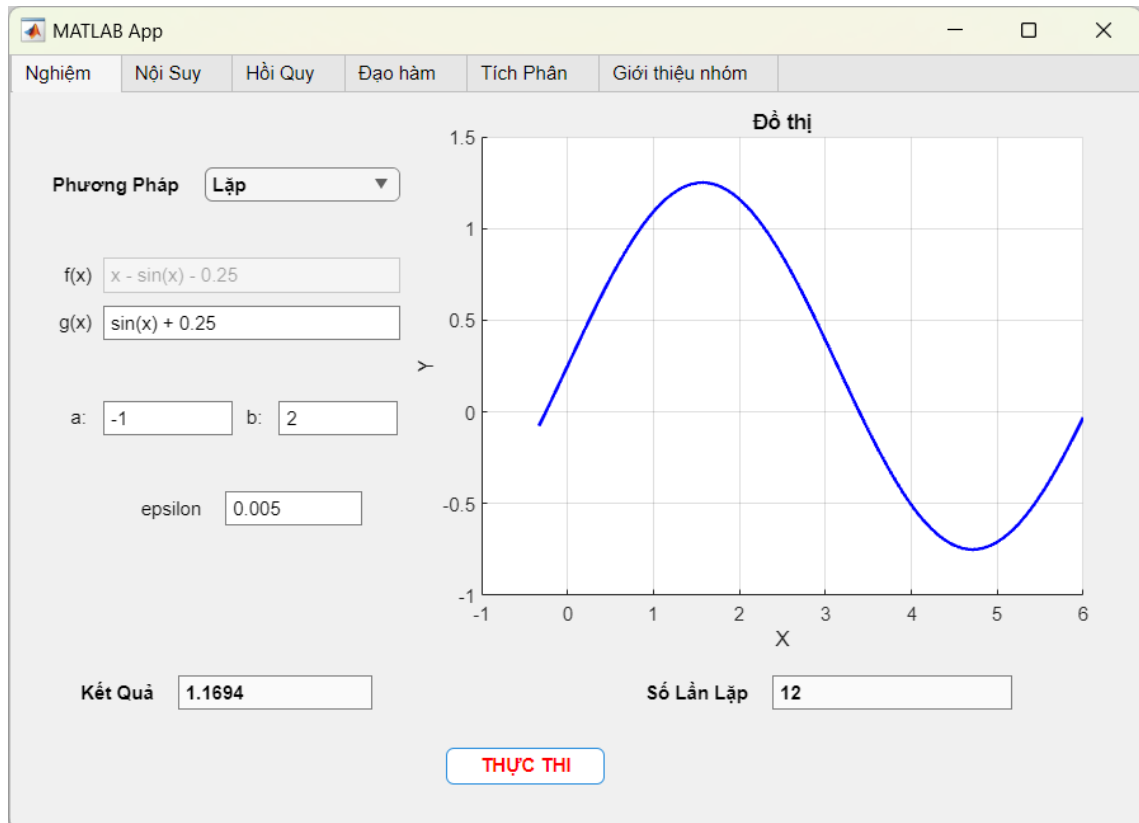
$$a: -1 \quad ; \quad b: 2 \quad ; \quad \text{epsilon}: 0.005$$



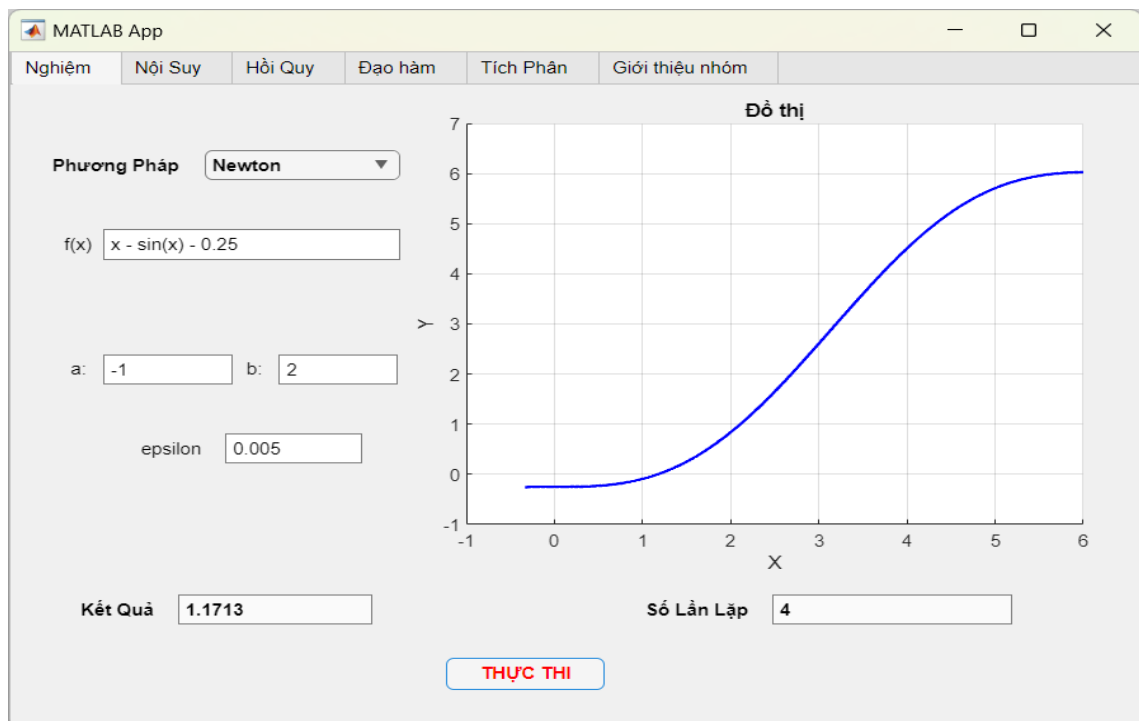
Hình 19. Nhập các thông số vào giao diện



Hình 20. Kết quả đạt được bằng phương pháp chia đôi sau khi ấn "THỰC THI"



Hình 21. Kết quả đạt được bằng phương pháp lặp sau khi ấn "THỰC THI"



Hình 22. Kết quả đạt được bằng phương pháp Newton sau khi ấn "THỰC THI"

Sau khi thực hiện các bước trên, chúng ta sẽ có nghiệm gần đúng và số lần lặp cho mỗi phương pháp. Phương pháp cho số lần lặp nhiều nhất là phương pháp tiếp tuyến (Newton). Vì *điểm khởi tạo xa nghiệm thực, đạo hàm gần 0* (nếu đạo hàm gần 0, sự tiến bộ từng bước sẽ rất nhỏ, dẫn đến việc cần nhiều vòng lặp để đạt được nghiệm), *nghiệm gần điểm cực trị* [nếu nghiệm nằm gần một điểm cực trị của hàm số (nơi đạo hàm bằng 0), phương pháp Newton có thể mất nhiều vòng lặp để vượt qua điểm cực trị và tiến tới nghiệm] và *hàm số biến đổi mạnh trong khoảng phân li*, việc xấp xỉ hàm số bằng một đường thẳng tiếp tuyến có thể không chính xác, dẫn đến việc cần nhiều vòng lặp để điều chỉnh.

3.2 ĐA THỨC NỘI SUY

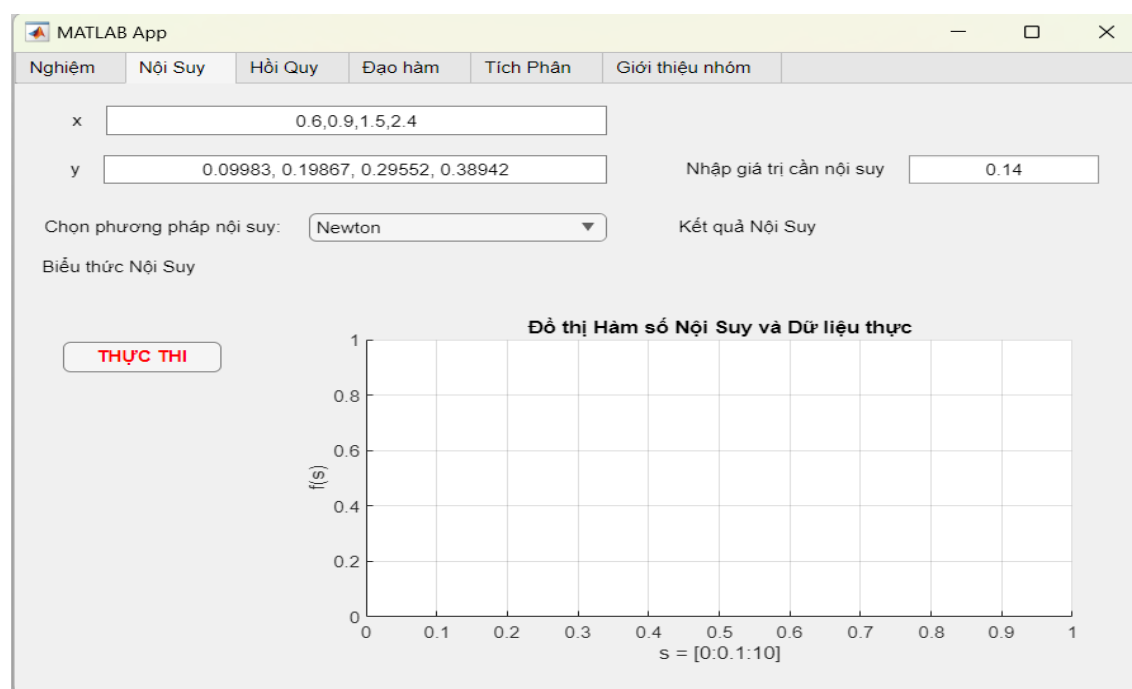
VD1: Áp dụng để tìm giá trị nội suy tại $x = 0.14$. Cho biết các giá trị đầu vào cho nội suy: $x = [0.6, 0.9, 1.5, 2.4]$, $y = [0.09983, 0.19867, 0.29552, 0.38942]$.

Thực hành:

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

Mảng x: [0.6, 0.9, 1.5, 2.4] ; Mảng y: [0.09983, 0.19867, 0.29552, 0.38942]

Giá trị cần nội suy: 0.14

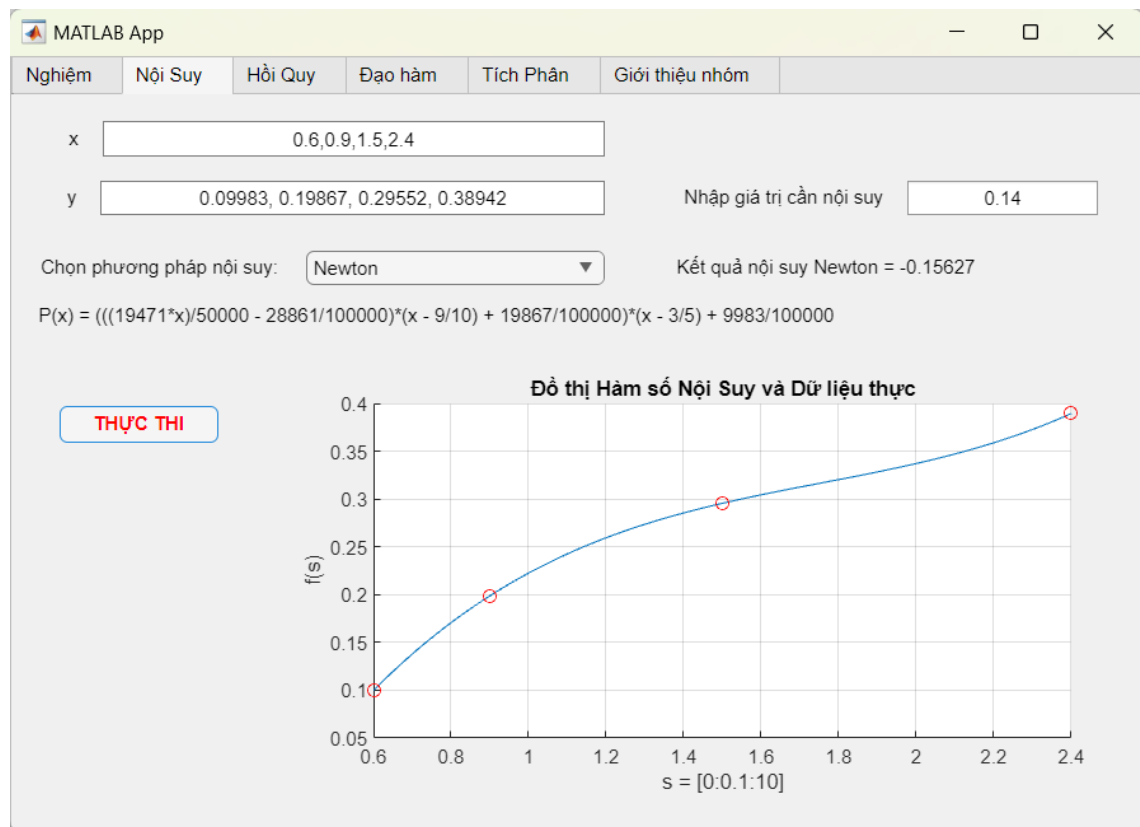


Hình 23. Nhập các thông số vào giao diện

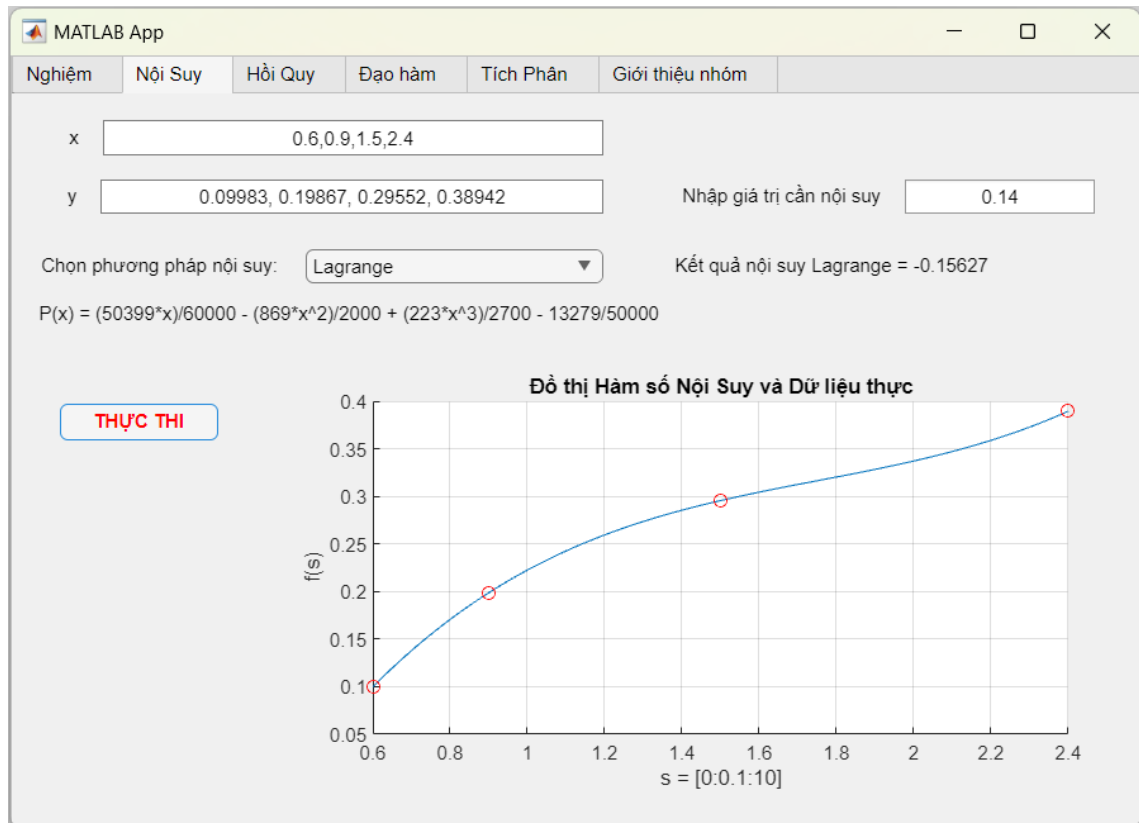
Tại Drop Down Menu (Chọn phương pháp nội suy): "Newton" hoặc "Lagrange".

Nhấn Button “THỰC THI”. Đoạn code sẽ được thực thi, tính toán giá nội suy, biểu thức nội suy của từng phương pháp và vẽ được đồ thị hàm số nội suy và dữ liệu thực theo từng phương pháp.

Kết quả sẽ được hiển thị trên các Label và Axes tương ứng.



Hình 24. Kết quả đạt được bằng Nội Suy Newton sau khi ấn "THỰC THI"



Hình 25. Kết quả đạt được bằng Nội Suy Lagrange sau khi ấn "THỰC THI"

VD2: Áp dụng để tìm giá trị nội suy tại $x = 0.16$. Cho biết các giá trị đầu vào cho nội suy: $x = [0.1, 0.2, 0.3, 0.4]$, $y = [0.09983, 0.19867, 0.29552, 0.38942]$. So sánh sai số giữa kết quả của đạt được với $\sin(0.16)$.

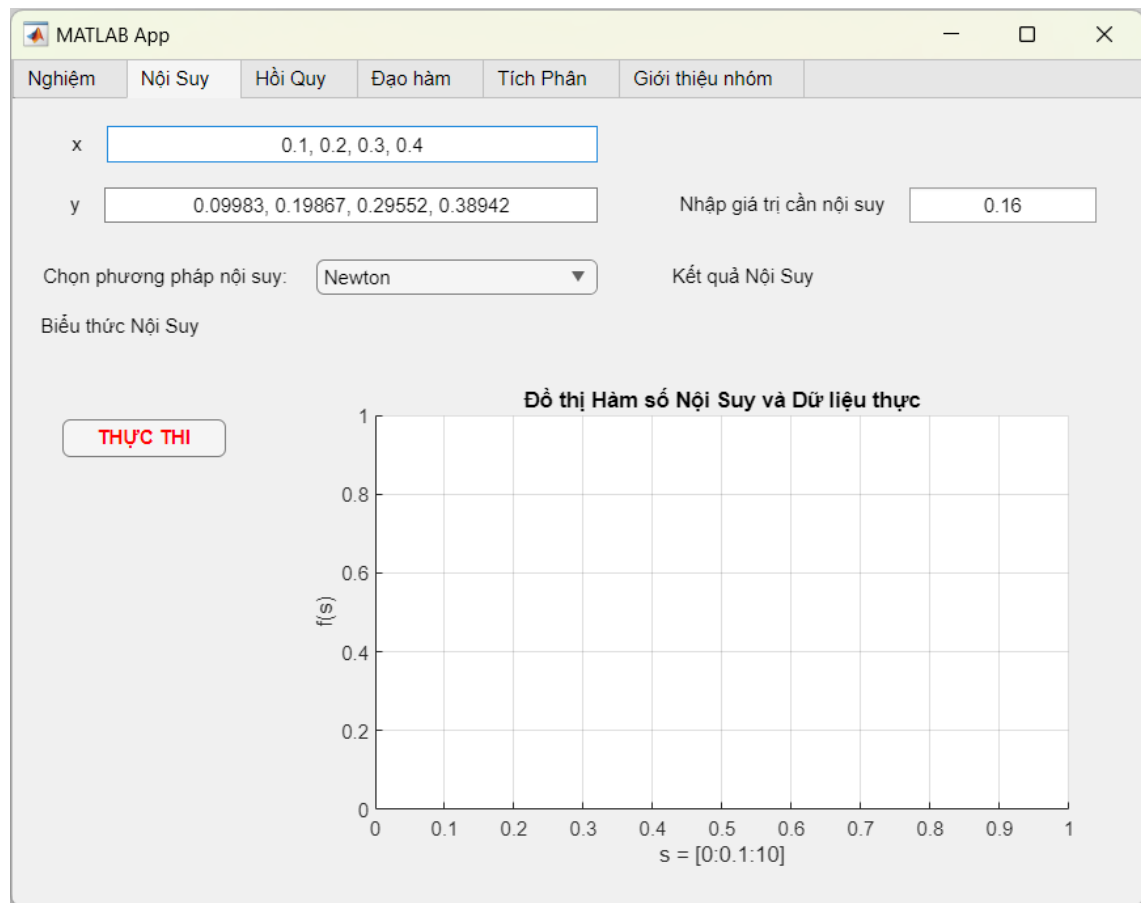
Thực hành:

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

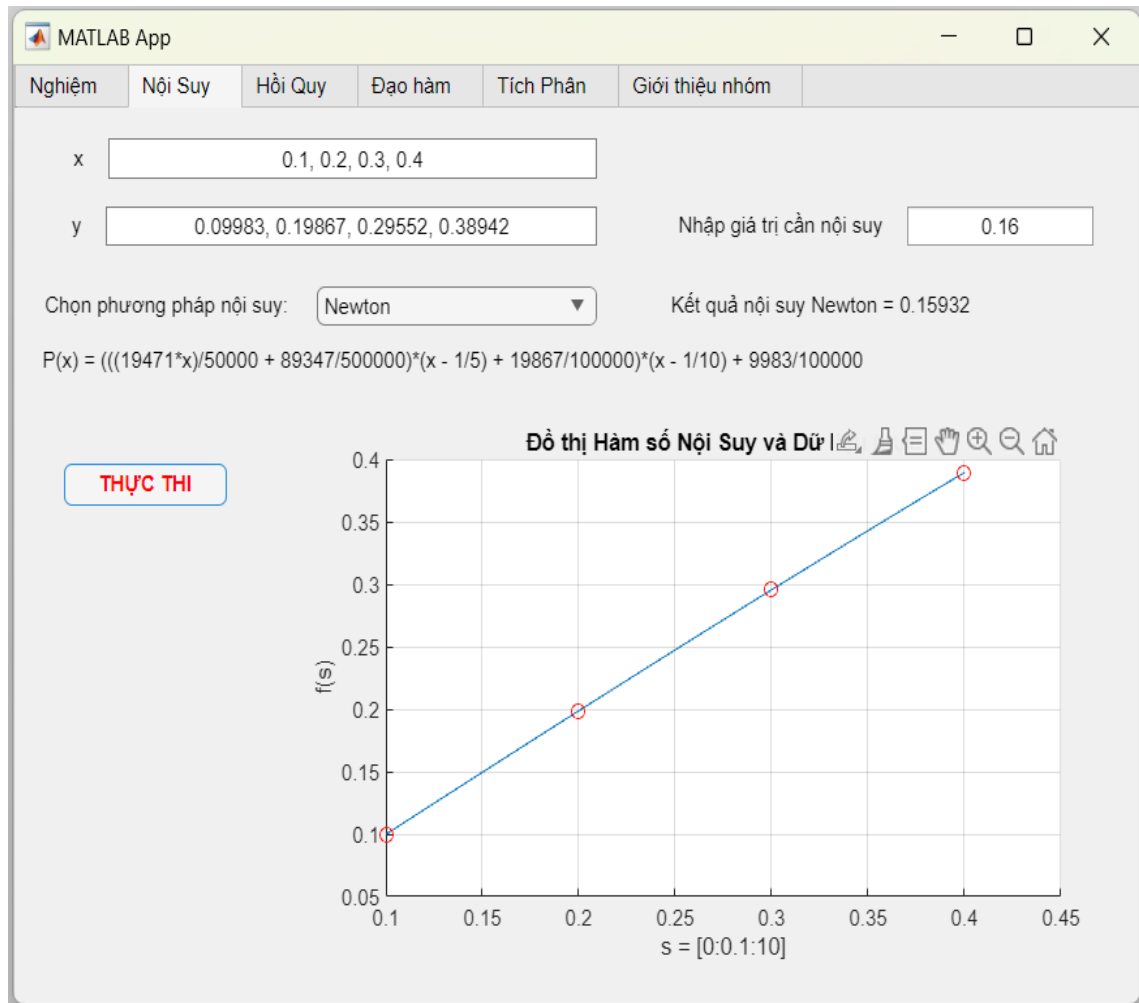
Mảng x: [0.1, 0.2, 0.3, 0.4]

Mảng y: [0.09983, 0.19867, 0.29552, 0.38942]

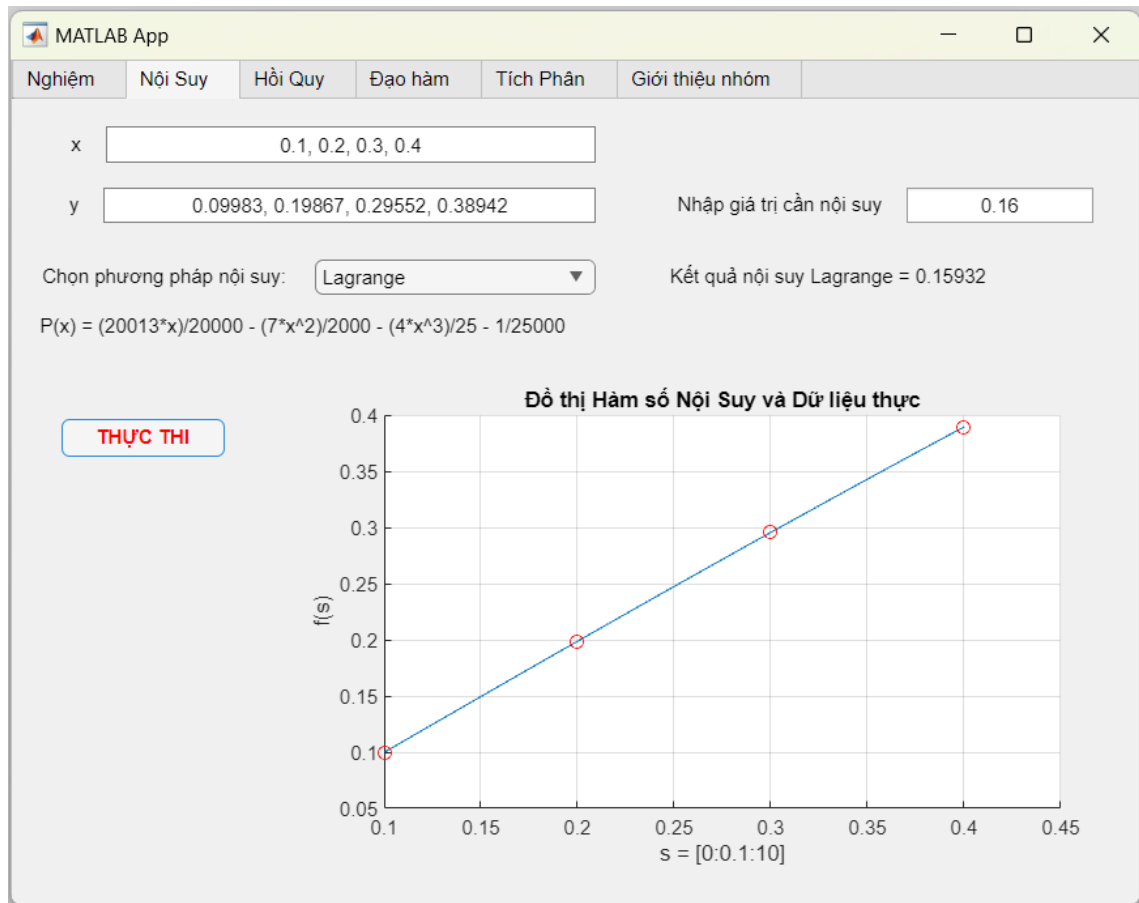
Giá trị cần nội suy: 0.16



Hình 26. Nhập các thông số vào giao diện



Hình 27. Kết quả đạt được bằng Nội Suy Newton sau khi ấn "THỰC THI"



Hình 28. Kết quả đạt được bằng Nội Suy Lagrange sau khi ấn "THỰC THI"

Ta có: $\sin(0.16) \approx 0.159318$

Dựa trên kết quả chạy chương trình, sai số giữa giá trị nội suy và giá trị thực tế ($\sin(x)$) là rất nhỏ ($8.3339e-07$ cho $x = 0.16$). Điều này cho thấy phương pháp nội suy Newton và Lagrange đã cho kết quả khá chính xác trong trường hợp này.

3.3 HỒI QUY

VD1: Sử dụng Tab Hồi Quy, tìm mô hình hồi quy cho số liệu sau đây. Dữ liệu x: [30, 35, 37, 50, 55]; Dữ liệu y: [500, 550, 570, 700, 750]; Dự đoán giá trị của mô hình tại $x = 40$. Vẽ đường thẳng hồi quy và các giá trị đo được lên cùng 1 hình.

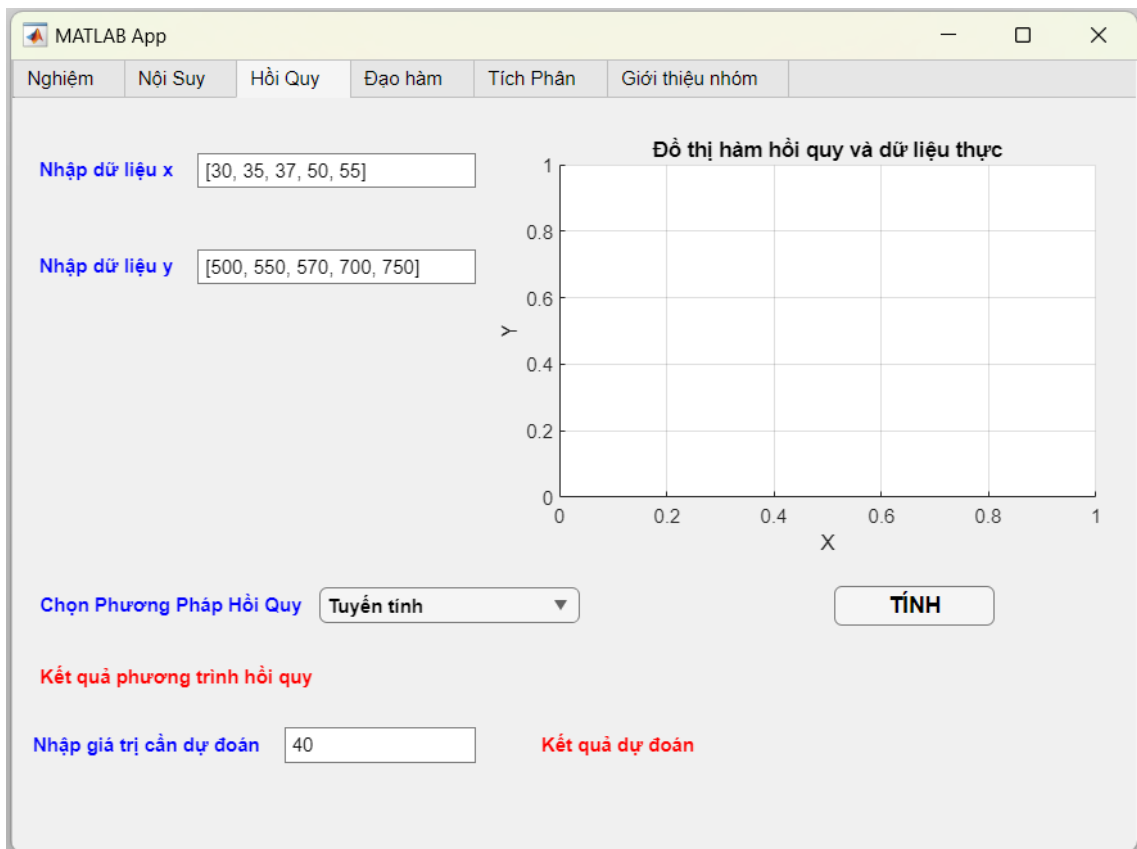
Thực hành:

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

Nhập dữ liệu x: [30, 35, 37, 50, 55]

Nhập dữ liệu y: [500, 550, 570, 700, 750]

Nhập giá trị cần dự đoán: 40



Hình 29. Nhập các thông số vào giao diện

Tại Drop Down Menu (Chọn Phương Pháp Hồi Quy): "Tuyến tính" hoặc "Hàm mũ" hoặc "Logarit".

Nhấn Button “TÍNH”. Đoạn code sẽ được thực thi, tính toán kết quả phương trình hồi quy và hệ số tương quan r^2 (đối với Hồi Quy Tuyến tính), kết quả dự đoán và vẽ đường thẳng hồi quy và các giá trị đo được lên cùng 1 hình.

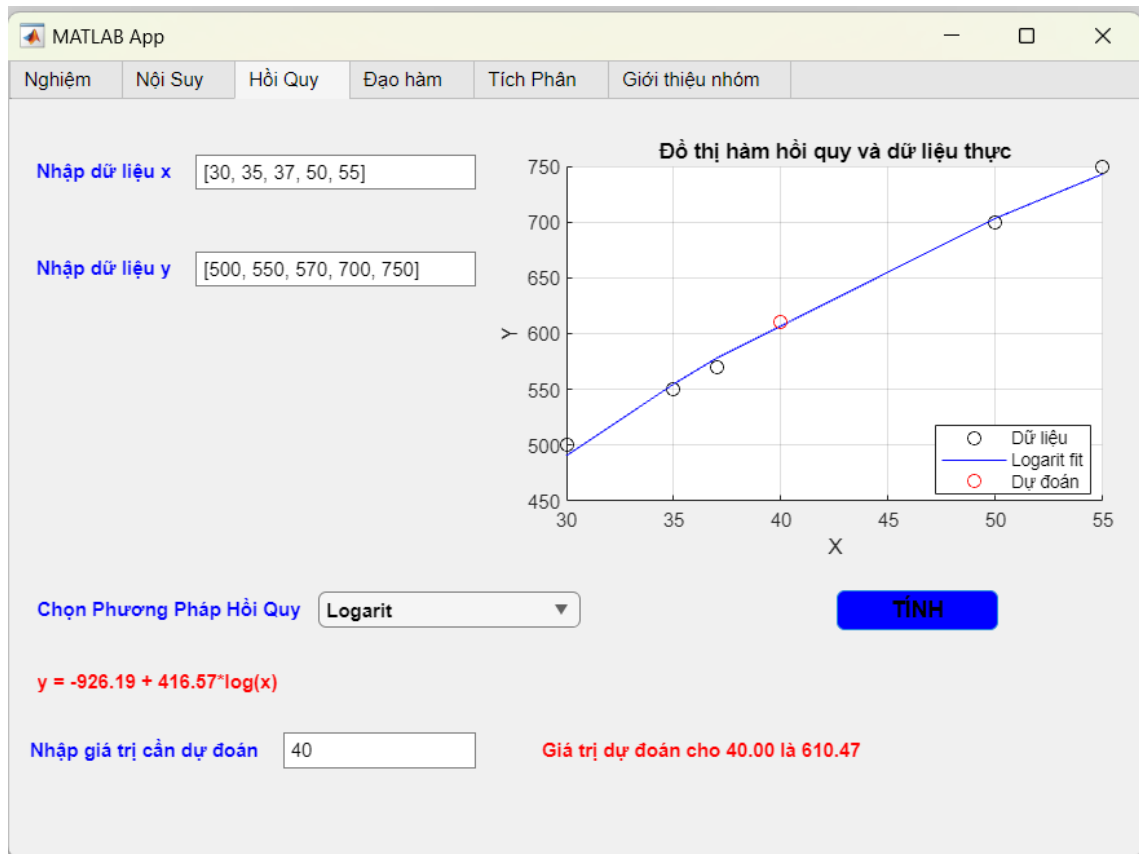
Kết quả sẽ được hiển thị trên các Label và Axes tương ứng.



Hình 30. Kết quả đạt được bằng Hồi Quy Tuyến tính sau khi ấn "TÍNH"



Hình 31. Kết quả đạt được bằng Hồi Quy Hàm mũ sau khi ấn "TÍNH"



Hình 32. Kết quả đạt được bằng Hồi Quy Logarit sau khi ấn "TÍNH"

VD2: Sử dụng Tab Hồi Quy, tìm mô hình hồi quy cho số liệu sau đây. Dữ liệu x: [2.5, 3.5, 5, 6, 7.5, 10, 12.5, 15, 17.5, 20]; Dữ liệu y: [13, 11, 8.5, 8.2, 7, 6.2, 5.2, 4.8, 4.6, 4.3]; Dự đoán giá trị của mô hình tại x = 9. Vẽ đường thẳng hồi quy và các giá trị đo được lên cùng 1 hình.

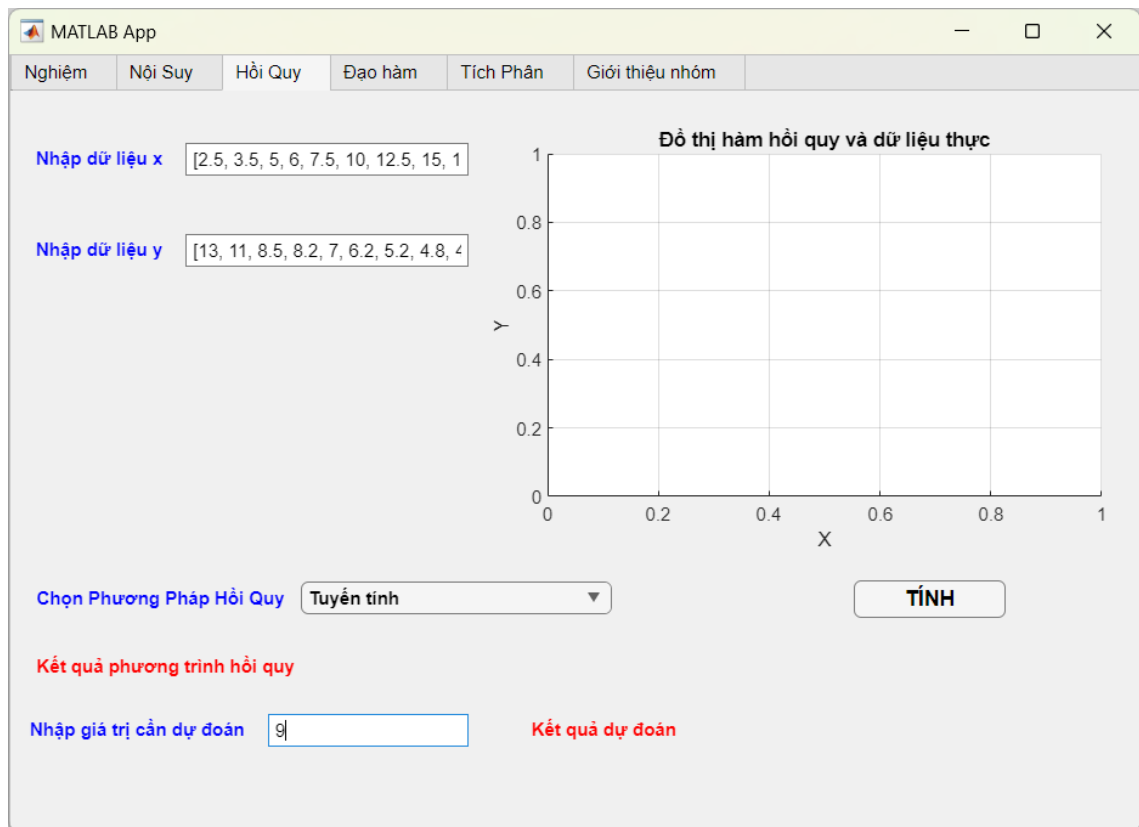
Thực hành:

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

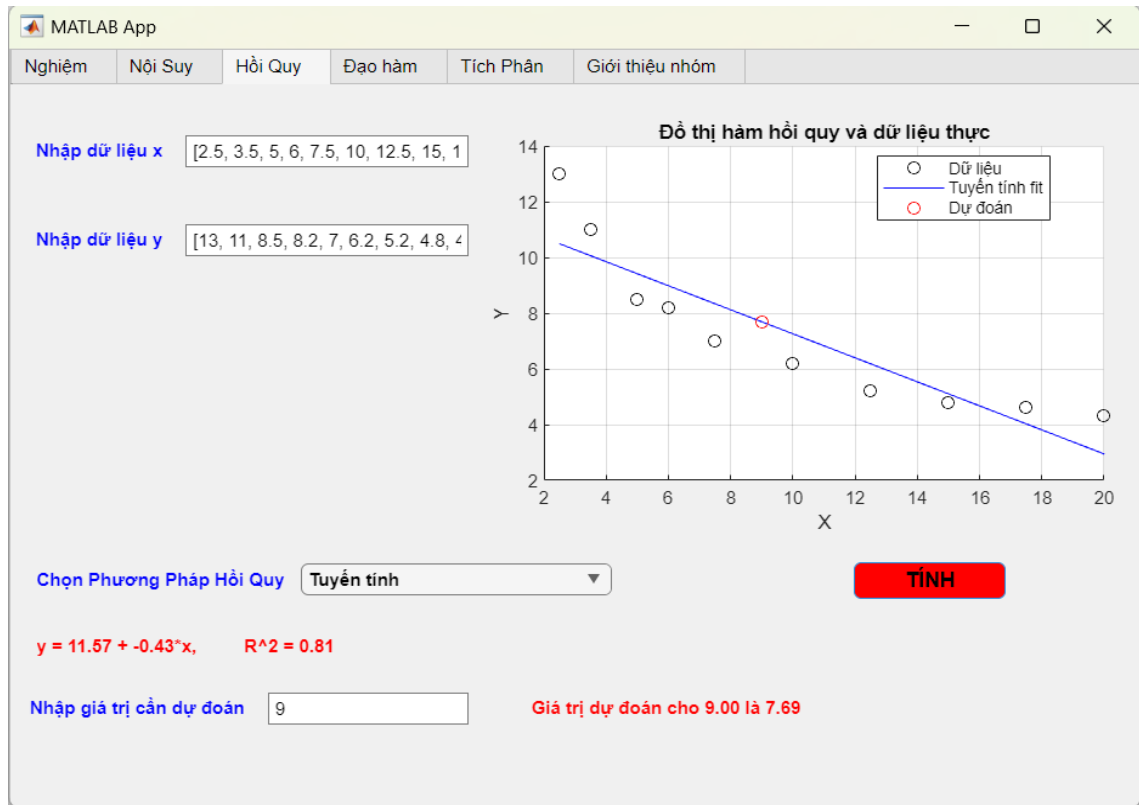
Nhập dữ liệu x: [2.5, 3.5, 5, 6, 7.5, 10, 12.5, 15, 17.5, 20]

Nhập dữ liệu y: [13, 11, 8.5, 8.2, 7, 6.2, 5.2, 4.8, 4.6, 4.3]

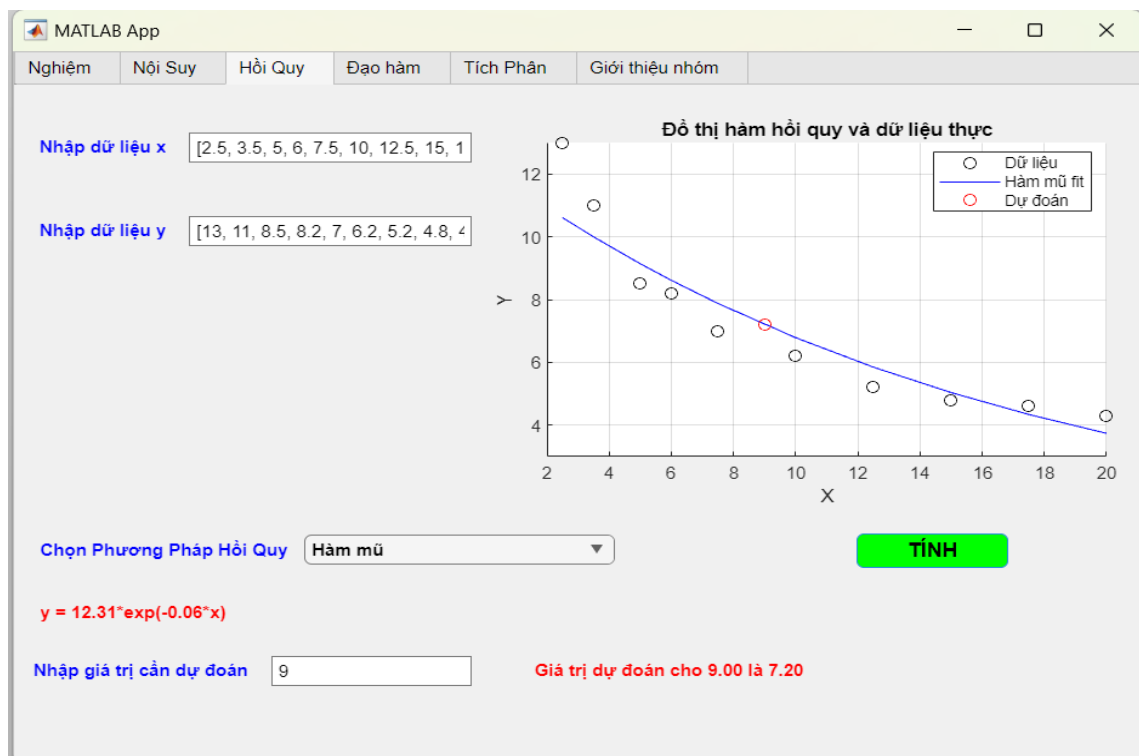
Nhập giá trị cần dự đoán: 9



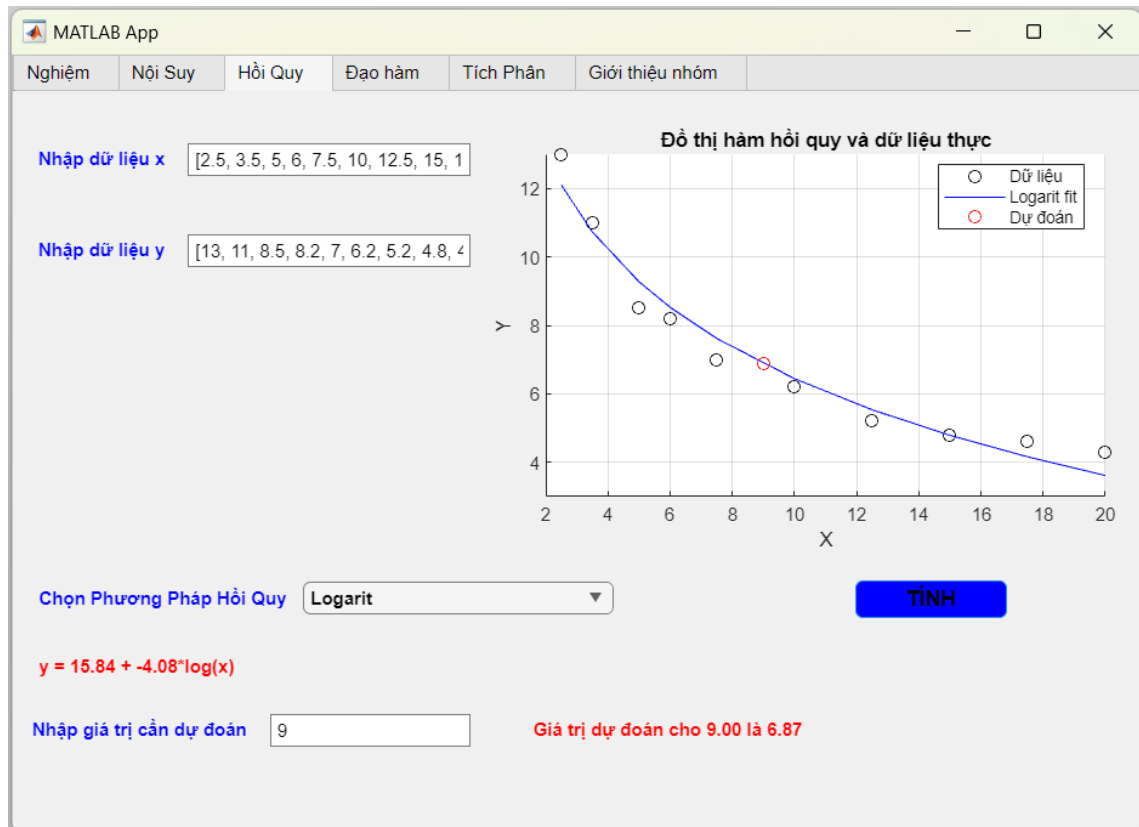
Hình 33. Nhập các thông số vào giao diện



Hình 34. Kết quả đạt được bằng Hồi Quy Tuyến tính sau khi ấn "TÍNH"



Hình 35. Kết quả đạt được bằng Hồi Quy Hàm mũ sau khi ấn "TÍNH"



Hình 36. Kết quả đạt được bằng Hồi Quy Logarit sau khi ấn "TÍNH"

3.4 TÍNH GẦN ĐÚNG ĐẠO HÀM

Cho hàm số $y = \arcsin(x)$ với các giá trị tại: $x = [0.1 \ 0.3 \ 0.5 \ 0.7 \ 0.9]$ và $y = [0.1002 \ 0.3047 \ 0.5236 \ 0.7754 \ 1.1198]$. Tính gần đúng đạo hàm của y tại $x = 0.5$ và so sánh với đạo hàm chính xác $\left(\frac{1}{\sqrt{1-x^2}}\right)$

Thực hành:

Radio Button Group (Chọn cách nhập): “Nhập mảng x, y”

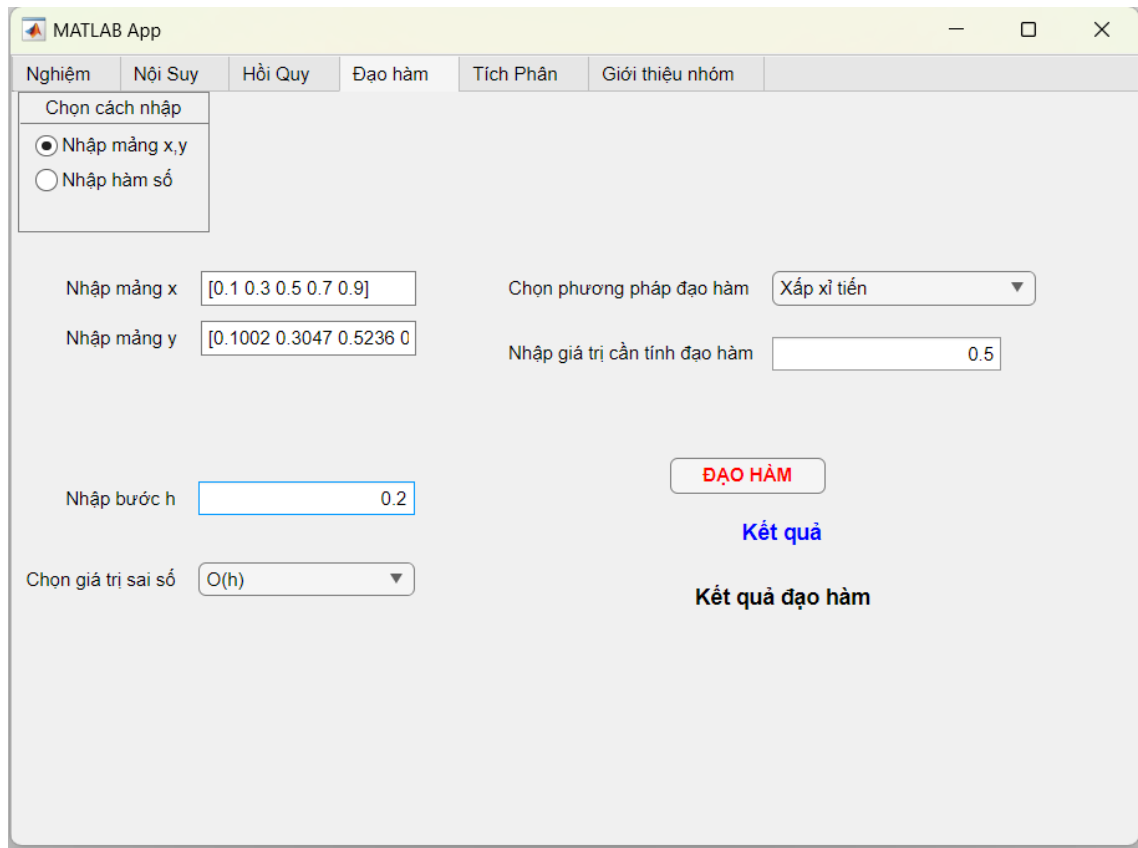
Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

Nhập mảng x: [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y: [0.1002 0.3047 0.5236 0.7754 1.1198]

Nhập giá trị cần tính đạo hàm: 0.5

Bước h: 0.2



The image shows a MATLAB App window titled "MATLAB App". It has a tabbed interface with tabs: "Thử nghiệm", "Nội Suy", "Hồi Quy", "Đạo hàm", "Tích Phân", and "Giới thiệu nhóm". The "Đạo hàm" (Derivative) tab is selected. Inside the app, there is a section "Chọn cách nhập" (Choose input method) with two radio buttons: "Nhập mảng x,y" (selected) and "Nhập hàm số" (unselected). Below this, there are input fields for "Nhập mảng x" (containing [0.1 0.3 0.5 0.7 0.9]), "Nhập mảng y" (containing [0.1002 0.3047 0.5236 0]), "Nhập bước h" (containing 0.2), and "Chọn giá trị sai số" (containing O(h)). There is also a dropdown for "Chọn phương pháp đạo hàm" (containing Xấp xỉ tiến) and an input for "Nhập giá trị cần tính đạo hàm" (containing 0.5). A red button labeled "ĐẠO HÀM" is present. Below the button, the text "Kết quả" (Result) is displayed in blue, and "Kết quả đạo hàm" (Derivative result) is displayed in black.

Hình 37. Nhập các thông số vào giao diện

Tính đạo hàm bằng cách nhập mảng dữ liệu x, y.

Dưới đây là kết quả đạt được sau khi ấn Button “ĐẠO HÀM”.

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☒ Nhập mảng x,y

☐ Nhập hàm số

Nhập mảng x [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y [0.1002 0.3047 0.5236 0]

Chọn phương pháp đạo hàm Xấp xỉ tiến

Nhập giá trị cần tính đạo hàm 0.5

Nhập bước h 0.2

Chọn giá trị sai số O(h)

ĐẠO HÀM

Kết quả

1.259

Hình 38. Xấp xỉ tiến với sai số cắt cụt $O(h)$

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☒ Nhập mảng x,y

☐ Nhập hàm số

Nhập mảng x [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y [0.1002 0.3047 0.5236 0]

Chọn phương pháp đạo hàm Xấp xỉ lùi

Nhập giá trị cần tính đạo hàm 0.5

Nhập bước h 0.2

Chọn giá trị sai số O(h)

ĐẠO HÀM

Kết quả

1.0945

Hình 39. Xấp xỉ lùi với sai số cắt cụt $O(h)$

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☒ Nhập mảng x,y
☐ Nhập hàm số

Nhập mảng x [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y [0.1002 0.3047 0.5236 0]

Chọn phương pháp đạo hàm Xấp xỉ trung tâm

Nhập giá trị cần tính đạo hàm 0.5

Nhập bước h 0.2

Chọn giá trị sai số O(h)

ĐẠO HÀM

Kết quả

Không thực hiện được

Hình 40. Xấp xỉ trung tâm với sai số cắt cụt $O(h)$

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☒ Nhập mảng x,y
☐ Nhập hàm số

Nhập mảng x [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y [0.1002 0.3047 0.5236 0]

Chọn phương pháp đạo hàm Xấp xỉ trung tâm

Nhập giá trị cần tính đạo hàm 0.5

Nhập bước h 0.2

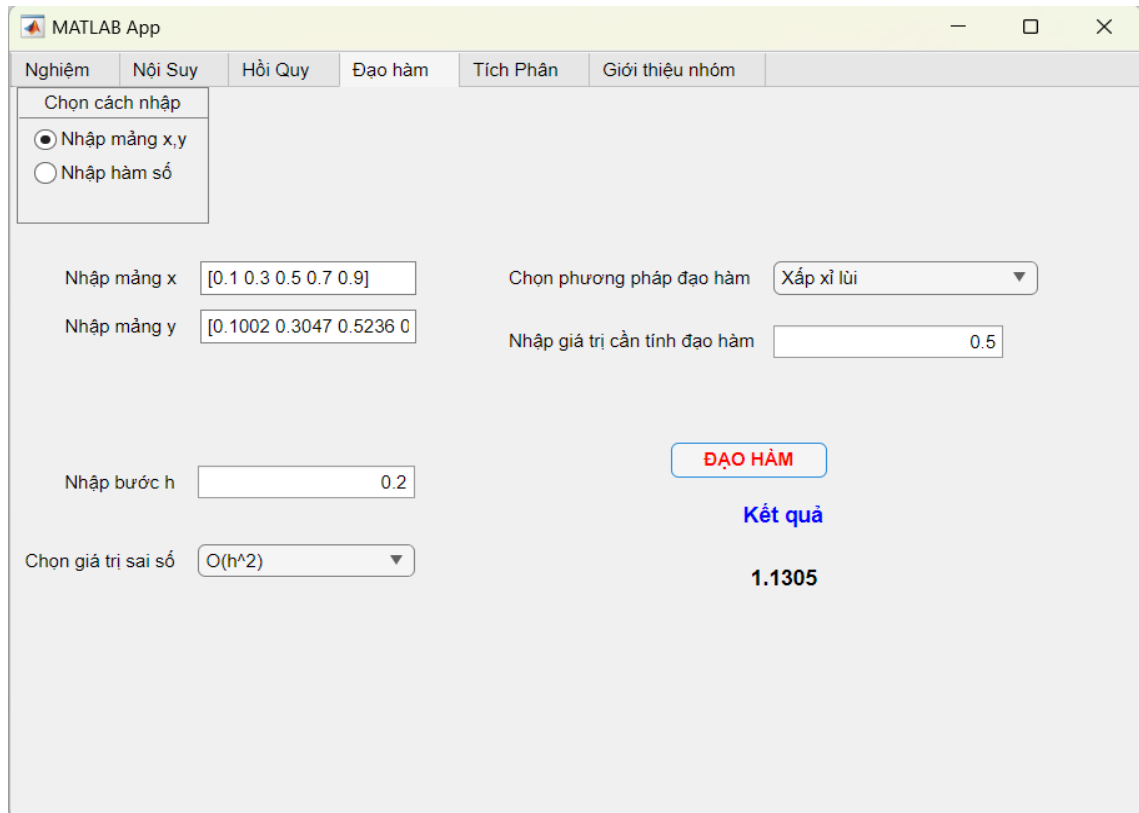
Chọn giá trị sai số O(h^2)

ĐẠO HÀM

Kết quả

1.1767

Hình 41. Xấp xỉ trung tâm với sai số cắt cụt $O(h^2)$



MATLAB App

Chọn cách nhập

☒ Nhập mảng x,y

☐ Nhập hàm số

Nhập mảng x: [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y: [0.1002 0.3047 0.5236 0]

Chọn phương pháp đạo hàm: Xấp xỉ lùi

Nhập giá trị cần tính đạo hàm: 0.5

Nhập bước h: 0.2

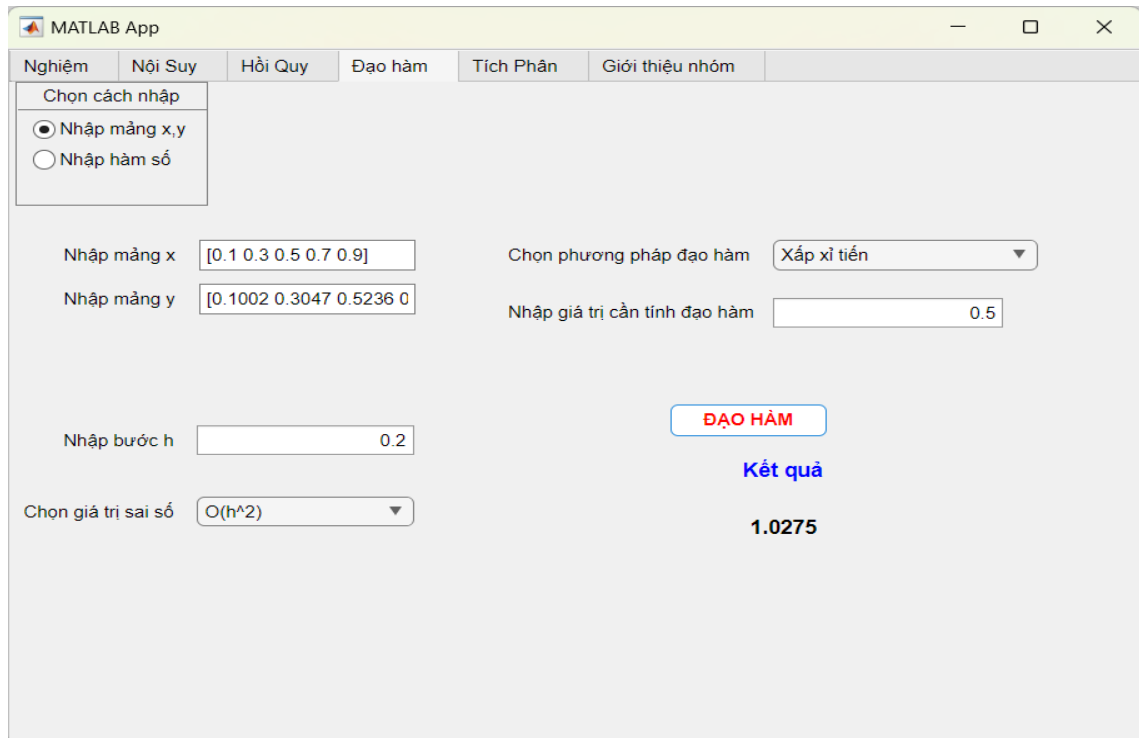
Chọn giá trị sai số: $O(h^2)$

ĐẠO HÀM

Kết quả

1.1305

Hình 42. Xấp xỉ lùi với sai số cắt cụt $O(h^2)$



MATLAB App

Chọn cách nhập

☒ Nhập mảng x,y

☐ Nhập hàm số

Nhập mảng x: [0.1 0.3 0.5 0.7 0.9]

Nhập mảng y: [0.1002 0.3047 0.5236 0]

Chọn phương pháp đạo hàm: Xấp xỉ tiến

Nhập giá trị cần tính đạo hàm: 0.5

Nhập bước h: 0.2

Chọn giá trị sai số: $O(h^2)$

ĐẠO HÀM

Kết quả

1.0275

Hình 43. Xấp xỉ tiến với sai số cắt cụt $O(h^2)$

Ta có: $\left(\frac{1}{\sqrt{1-0.5^2}}\right) \approx 1.1547$

Dựa trên kết quả chạy chương trình, sai số giữa giá trị đạo hàm và giá trị thực tế là rất nhỏ. Điều này cho thấy phương pháp tính gần đúng đạo hàm từ mảng dữ liệu x, y đã cho kết quả khá chính xác trong trường hợp này.

Thực hành:

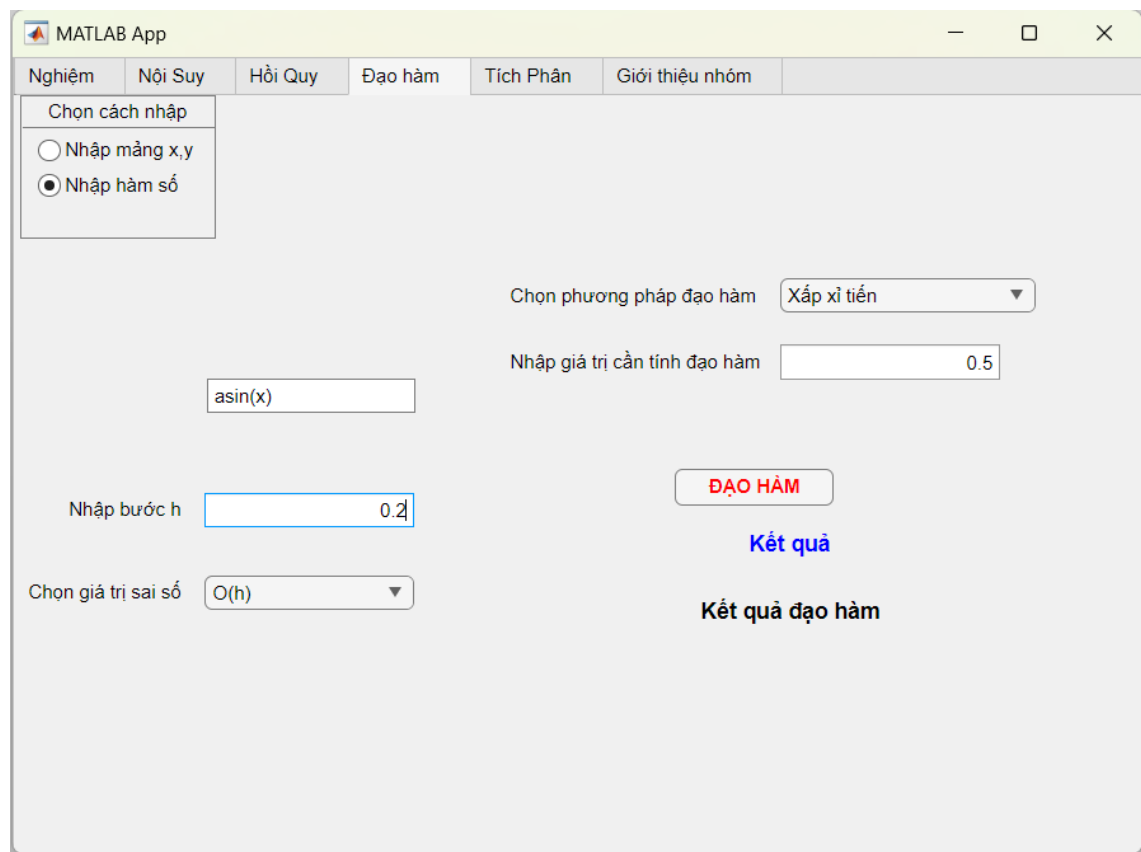
Chuyển Radio Button Group (Chọn cách nhập): “Nhập hàm số”.

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

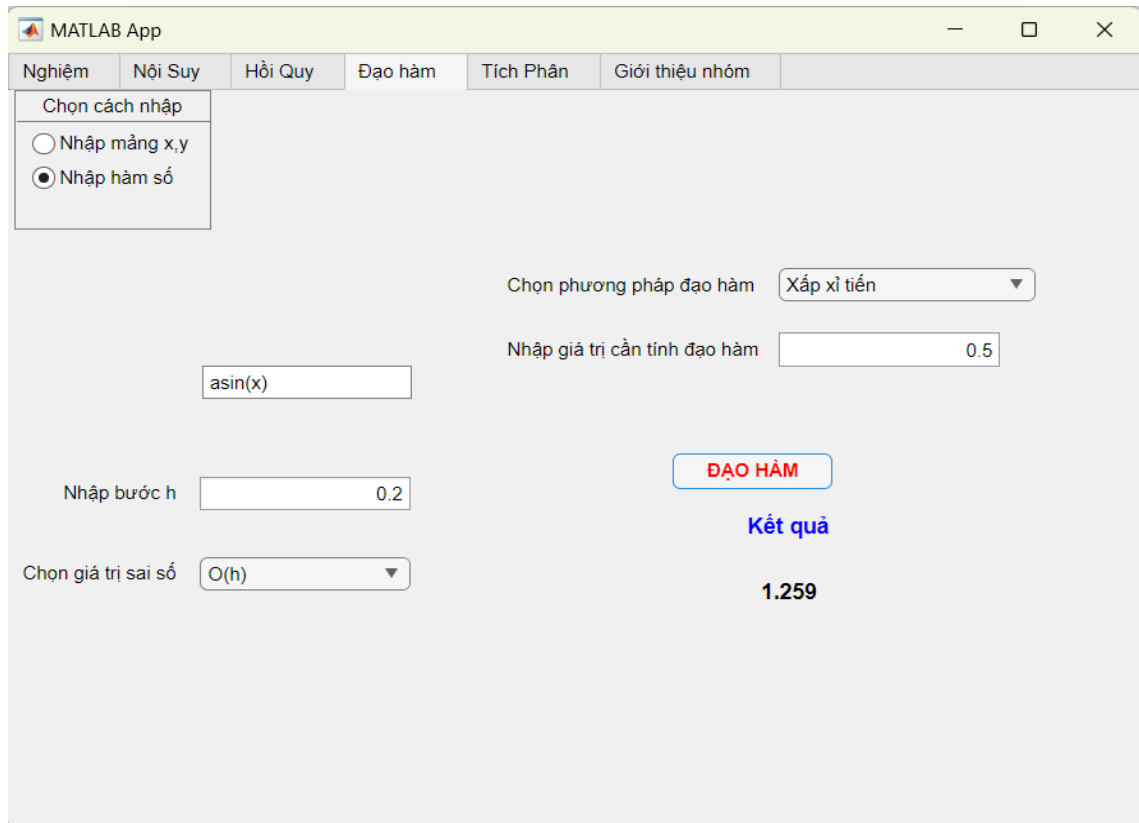
Hàm số cần nhập vào: asin(x)

Nhập giá trị cần tính đạo hàm: 0.5

Bước h: 0.2



Hình 44. Nhập các thông số vào giao diện



MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☐ Nhập mảng x,y

☒ Nhập hàm số

Chọn phương pháp đạo hàm Xấp xỉ tiến

Nhập giá trị cần tính đạo hàm 0.5

asin(x)

Nhập bước h 0.2

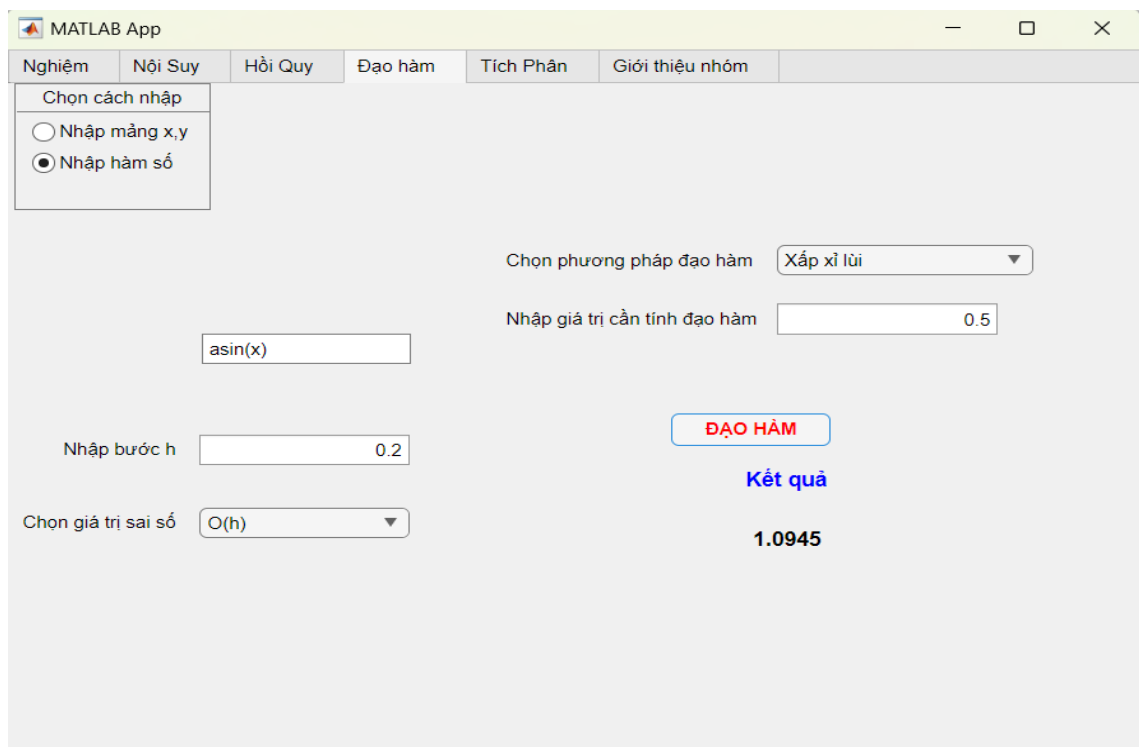
Chọn giá trị sai số O(h)

ĐẠO HÀM

Kết quả

1.259

Hình 45. Xấp xỉ tiến với sai số cắt cụt $O(h)$



MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☐ Nhập mảng x,y

☒ Nhập hàm số

Chọn phương pháp đạo hàm Xấp xỉ lùi

Nhập giá trị cần tính đạo hàm 0.5

asin(x)

Nhập bước h 0.2

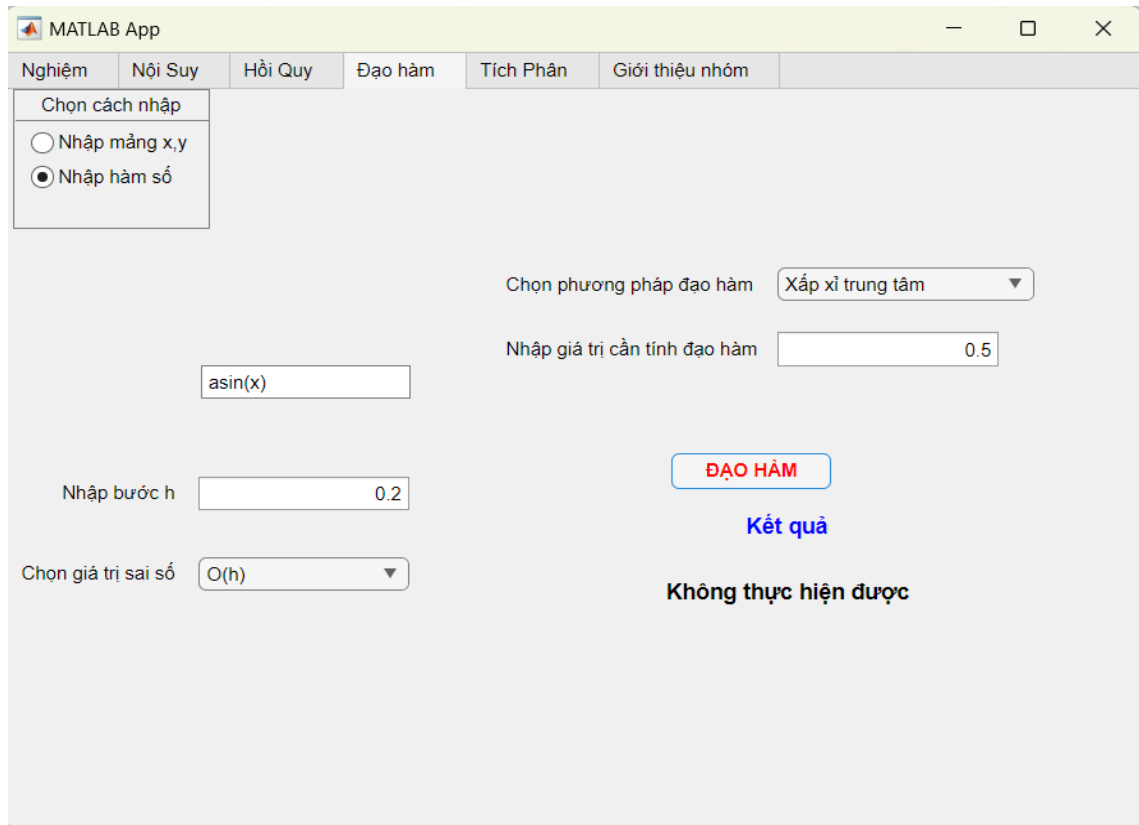
Chọn giá trị sai số O(h)

ĐẠO HÀM

Kết quả

1.0945

Hình 46. Xấp xỉ lùi với sai số cắt cụt $O(h)$



MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☐ Nhập mảng x,y

☒ Nhập hàm số

Chọn phương pháp đạo hàm Xấp xỉ trung tâm

Nhập giá trị cần tính đạo hàm 0.5

asin(x)

Nhập bước h 0.2

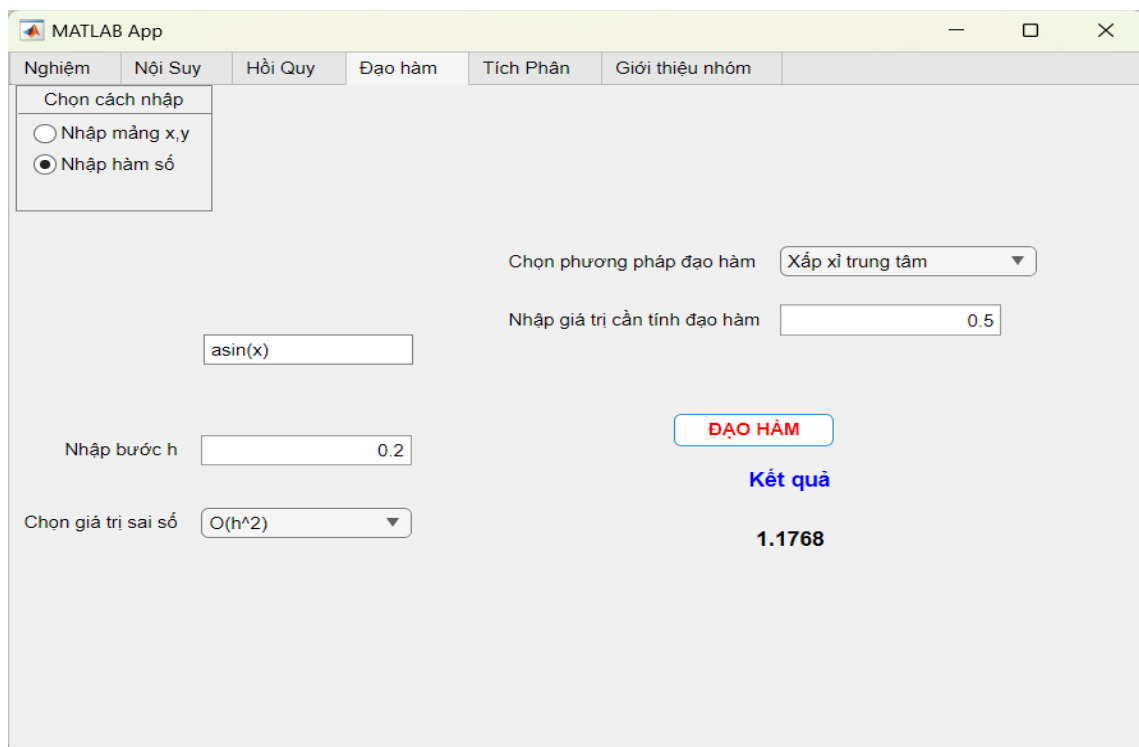
Chọn giá trị sai số O(h)

ĐẠO HÀM

Kết quả

Không thực hiện được

Hình 47. Xấp xỉ trung tâm với sai số cắt cụt $O(h)$



MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☐ Nhập mảng x,y

☒ Nhập hàm số

Chọn phương pháp đạo hàm Xấp xỉ trung tâm

Nhập giá trị cần tính đạo hàm 0.5

asin(x)

Nhập bước h 0.2

Chọn giá trị sai số $O(h^2)$

ĐẠO HÀM

Kết quả

1.1768

Hình 48. Xấp xỉ trung tâm với sai số cắt cụt $O(h^2)$

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☐ Nhập mảng x,y

☒ Nhập hàm số

Chọn phương pháp đạo hàm Xấp xỉ lùi

Nhập giá trị cần tính đạo hàm 0.5

asin(x)

Nhập bước h 0.2

Chọn giá trị sai số $O(h^2)$

ĐẠO HÀM

Kết quả

1.1305

Hình 49. Xấp xỉ lùi với sai số cắt cụt $O(h^2)$

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Chọn cách nhập

☐ Nhập mảng x,y

☒ Nhập hàm số

Chọn phương pháp đạo hàm Xấp xỉ tiến

Nhập giá trị cần tính đạo hàm 0.5

asin(x)

Nhập bước h 0.2

Chọn giá trị sai số $O(h^2)$

ĐẠO HÀM

Kết quả

1.0276

Hình 50. Xấp xỉ tiến với sai số cắt cụt $O(h^2)$

Ta có: $\left(\frac{1}{\sqrt{1-0.5^2}}\right) \approx 1.1547$

Dựa trên kết quả chạy chương trình, sai số giữa giá trị đạo hàm và giá trị thực tế là rất nhỏ. Điều này cho thấy phương pháp tính gần đúng đạo hàm từ hàm số đã cho kết quả khá chính xác trong trường hợp này.

3.5 TÍNH GẦN ĐÚNG TÍCH PHÂN

Cho hàm số $y = \frac{1}{1+x^2}$ với các giá trị tại: $x = [1, 2, 3, 4, 5]$ và $y(x) = [1/2, 1/5, 1/10, 1/17, 1/26]$. Trong đoạn $[1,5]$ với $N = 4$. Áp dụng Tab Tích Phân để tính gần đúng tích phân.

Thực hành:

Drop Down (Cách Nhập): “Mảng dữ liệu”

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

Mảng x: $[1, 2, 3, 4, 5]$ Mảng y(x): $[1/2, 1/5, 1/10, 1/17, 1/26]$

Cận a: 1 Cận b: 5 Nhập N(n đoạn): 4

Hình 51. Nhập các thông số vào giao diện

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Cách Nhập: Mảng dữ liệu

Mảng x: 1, 2, 3, 4, 5

Mảng y(x): 1/2, 1/5, 1/10, 1/17, 1/26

Chọn phương pháp Tích phân

☒ Hình thang

☐ Simpson 1/3

☐ Simpson 3/8

Cận a: 1

Cận b: 5

Nhập hàm số $1/(1+x^2)$

Nhập N (n đoạn): 4

0.62805

Phát hiện lỗi ●

TÍNH

Hình 52. Tích phân hình thang cho mảng dữ liệu

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Cách Nhập: Mảng dữ liệu

Mảng x: 1, 2, 3, 4, 5

Mảng y(x): 1/2, 1/5, 1/10, 1/17, 1/26

Chọn phương pháp Tích phân

☐ Hình thang

☒ Simpson 1/3

☐ Simpson 3/8

Cận a: 1

Cận b: 5

Nhập hàm số $1/(1+x^2)$

Nhập N (n đoạn): 4

0.59125

Phát hiện lỗi ●

TÍNH

Hình 53. Tích phân Simpson 1/3 cho mảng dữ liệu

Hình 54. Tích phân Simpson 3/8 cho mảng dữ liệu

Thực hành:

Chuyển Drop Down (Cách Nhập): “Hàm số”

Trong các Edit Field (Text) tương ứng, nhập các giá trị sau:

Nhập hàm số: $1/(1+x^2)$

Cận a: 1 Cận b: 5 Nhập N(n đoạn): 4

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Cách Nhập: Hàm số

Mảng x: 1, 2, 3, 4, 5

Mảng y(x): 1/2, 1/5, 1/10, 1/17, 1/26

Chọn phương pháp Tích phân

☒ Hình thang

☐ Simpson 1/3

☐ Simpson 3/8

Cận a: 1

Cận b: 5

Nhập hàm số $1/(1+x^2)$

Nhập N (n đoạn): 4

Kết Quả

Phát hiện lỗi ●

TÍNH

Hình 55. Nhập các thông số vào giao diện

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Cách Nhập: Hàm số

Mảng x: 1, 2, 3, 4, 5

Mảng y(x): 1/2, 1/5, 1/10, 1/17, 1/26

Chọn phương pháp Tích phân

☒ Hình thang

☐ Simpson 1/3

☐ Simpson 3/8

Cận a: 1

Cận b: 5

Nhập hàm số $1/(1+x^2)$

Nhập N (n đoạn): 4

0.62805

Phát hiện lỗi ●

TÍNH

Hình 56. Tích phân hình thang cho hàm số

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Cách Nhập:

Mảng x:

Mảng y(x):

Chọn phương pháp Tích phân

☐ Hình thang

☒ Simpson 1/3

☐ Simpson 3/8


Cận a:

Cận b:

Nhập hàm số

Nhập N (n đoạn):

0.59125

Phát hiện lỗi 

TÍNH

Hình 57. Tích phân Simpson 1/3 cho hàm số

MATLAB App

Nghiệm Nội Suy Hồi Quy Đạo hàm Tích Phân Giới thiệu nhóm

Cách Nhập:

Mảng x:

Mảng y(x):

Chọn phương pháp Tích phân

☐ Hình thang

☐ Simpson 1/3

☒ Simpson 3/8


Cận a:

Cận b:

Nhập hàm số

Nhập N (n đoạn):

0.58354

Phát hiện lỗi 

TÍNH

Hình 58. Tích phân Simpson 3/8 cho hàm số

CHƯƠNG 3: TỔNG KẾT

3.1. ĐÁNH GIÁ ĐỘ HOÀN THIỆN CỦA ĐỀ TÀI

Đồ án cuối kỳ Phương pháp tính và Matlab với đề tài “Thiết kế ứng dụng bằng App Designer” đã thực hiện là một công việc tuyệt vời mà nhóm 6BRO và các thành viên trong nhóm cộng tác với nhau trong việc giải quyết các bài toán phương pháp tính. Sau một thời gian thực hiện, dưới đây là một số nhận xét và đánh giá:

3.1.1 ƯU ĐIỂM

Ứng dụng của nhóm đã cung cấp một loạt các giao diện người dùng để giải quyết các bài toán khác nhau, bao gồm tìm nghiệm, nội suy, hồi quy, đạo hàm và tích phân.

Nhóm đã sử dụng nhiều phương pháp khác nhau để giải quyết mỗi bài toán, điều này cho thấy sự linh hoạt và độ chính xác cao.

Ứng dụng của nhóm cũng có khả năng vẽ hàm số và phương trình hồi quy, điều này giúp người dùng dễ dàng hiểu và diễn giải kết quả.

3.1.2 HẠN CHẾ

Mặc dù bạn đã thực hiện rất nhiều công việc, nhưng có thể vẫn còn một số trường hợp mà ứng dụng của nhóm chưa giải quyết được. Ví dụ, ứng dụng có thể không hoạt động chính xác nếu dữ liệu đầu vào không phải là số thực hoặc nếu dữ liệu chứa giá trị không hợp lệ.

Ngoài ra, ứng dụng có thể không đưa ra kết quả chính xác nếu số lượng phần tử trong mảng hoặc số lượng phân đoạn cần chia khi tính tích phân quá nhỏ.

3.1.3 ĐỀ XUẤT

Đối với những hạn chế đã nêu trên, một hướng giải quyết có thể là thêm các kiểm tra dữ liệu đầu vào để đảm bảo rằng chúng phù hợp với yêu cầu của từng phương pháp.

Nhóm cũng có thể cung cấp thêm hướng dẫn chi tiết cho người dùng về cách sử dụng ứng dụng và cách nhập dữ liệu.

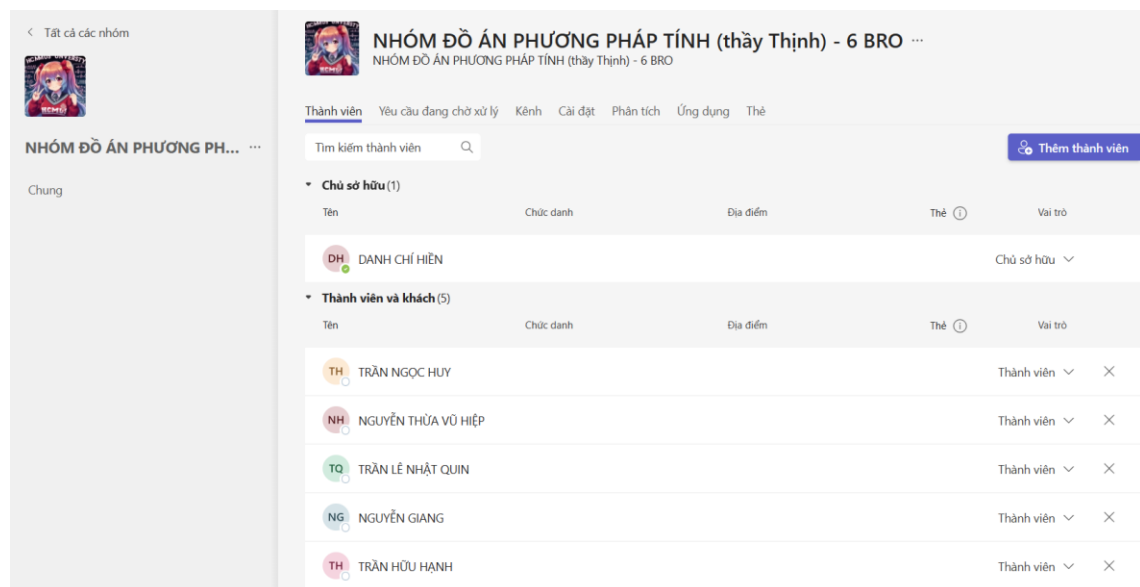
3.2. ĐÁNH GIÁ NHÓM VÀ CÁC THÀNH VIÊN

3.2.1 HOẠT ĐỘNG CỦA NHÓM VÀ CÁC THÀNH VIÊN

Hoạt động nhóm:

Nhóm phân công hoạt động giữa các thành viên, triển khai, phân chia nhiệm vụ trên Microsoft Teams:

<https://teams.microsoft.com/l/team/19%3aLQTJfebmzP1PTJPb34h1XxwkWvEWgpQxvulQdUWF7ZY1%40thread.tacv2/conversations?groupId=7c1a1f1a-951a-4a57-9ce0-517ff642832d&tenantId=40127cd4-45f3-49a3-b05d-315a43a9f033>



Hình 59. Hoạt động của nhóm trên MS Teams

Nhóm cũng triển khai những buổi họp offline trên trường (tại phòng tự học) để hoạt động triển khai viết code và thiết kế giao diện với các thành viên.

Trong các buổi họp online đều có Recording để những thành viên có thể xem lại lúc cần thiết.

Nhóm có sử dụng phần mềm quản lý code để cộng tác (Github): [danhchihien/Code_Matlab_2023_6BRO \(github.com\)](https://github.com/danhchihien/Code_Matlab_2023_6BRO)

3.2.2 BẢNG PHÂN CÔNG, ĐÁNH GIÁ CÁC THÀNH VIÊN

Họ và tên	MSSV	Phân công	Đánh giá
Danh Chí Hiền	21200287	<p>Nhóm trưởng, Quản lý nhóm</p> <p>Khơi gợi yêu cầu (User-story)</p> <p>Đặc tả yêu cầu chức năng, Quản lý mã nguồn</p> <p>Viết các hàm, nêu được thuật toán của Tab Nội Suy, Tab Hồi Quy, cho được ví dụ test đúng chương trình.</p> <p>Thiết kế được giao diện của Tab Nội Suy, Tab Hồi Quy, Tab Giới Thiệu Nhóm</p> <p>Tổng hợp các Tab phân công cho các thành viên lên một Tab Group</p> <p>Hoàn thiện báo cáo</p>	<p>Hoàn thành tốt</p> <p>(20%)</p>
Nguyễn Giang	21200284	<p>Thư ký</p> <p>Viết các hàm và thiết kế được giao diện của Tab Tích Phân</p> <p>Nêu được thuật toán phân tích Tích phân</p> <p>Cho được ví dụ test đúng chương trình</p>	<p>Hoàn thành tốt</p> <p>(16%)</p>
Trần Ngọc Huy	20200040	<p>Viết các hàm và thiết kế được giao diện của Tab Nghiệm</p> <p>Nêu được thuật toán phân tìm Nghiệm</p> <p>Cho được ví dụ test đúng chương trình</p>	<p>Hoàn thành tốt</p> <p>(16%)</p>

		Tổng hợp các Tab phân công cho các thành viên lên một Tab Group	
Trần Hữu Hạnh	21200286	Viết các hàm và thiết kế được giao diện của Tab Đạo Hàm Nêu được thuật toán phân tính Đạo Hàm Cho được ví dụ test đúng chương trình	Hoàn thành tốt (16%)
Nguyễn Thừa Vũ Hiệp	21200288	Viết các hàm và thiết kế được giao diện của Tab Đạo Hàm Nêu được thuật toán phân tính Đạo Hàm Cho được ví dụ test đúng chương trình	Hoàn thành tốt (16%)
Trần Lê Nhật Quin	21200342	Viết các hàm và thiết kế được giao diện của Tab Nội Suy Nêu được thuật toán phân tính Nội Suy. Cho được ví dụ test đúng chương trình	Hoàn thành tốt (16%)

Bảng 1. Bảng phân công - đánh giá thành viên nhóm 6BRO

TÀI LIỆU THAM KHẢO

- [1] Thái Hồng Hải, Huỳnh Quốc Thịnh, "Tài liệu Thực hành Phương pháp tính và MATLAB", *Khoa Điện Tử - Viễn Thông, Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM*.
- [2] Huỳnh Quốc Thịnh, "Slide Bài giảng Phương Pháp tính và MATLAB", *Khoa Điện Tử - Viễn Thông, Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM*.