

# BÀI THỰC HÀNH MATLAB

Họ tên: Nguyễn Anh Tuấn

MSSV: 20200400

Ca học: 9

## Câu 1:

Code	Giải thích
<pre>syms x; f(x) = x^3+5*x^2-9*x+2; fprintf('f(x) = %s \n', f(x)); fprintf('f(2) = %d \n', f(2));  syms x; fd2 = diff(diff(f,x)); fprintf('f''''(x) = %s \n', fd2(x)); fprintf('f''''(5) = %d \n', fd2(5));</pre>	<p><b>Khai báo biến x kiểu symbol</b> <b>Phương trình f(x) dưới dạng symfun</b> <b>In f(x) ra màn hình với thông số s(string or char)</b> <b>In giá trị f(2) ra màn hình với thông số d(integer type)</b></p> <p><b>Khai báo biến x kiểu symbol</b> <b>Ta dc fd2 là đạo hàm bậc 2 của f(x) thông qua 2 lần đạo hàm f(x)</b> <b>In f'' ra màn hình</b></p> <p><b>In kết quả f''(5) ra màn hình với dạng integer</b></p>

## Câu 2:

a.  $3x^3 - 8x^2 - 20x + 16 = 0$   $[-1,2]$  sai số : 0.005

File	Code	Giải thích
main.m	<pre>f= @(x) 3*x^3 - 8*x^2 - 20*x + 16; fp= @(x) 3/20*x^3 - 2/5*x^2 +4/5; a = 0.2; b = 1; saio = 0.005;  fprintf('Nghiem va so lan lap cua pp chia doi: \n') chiadoiRes = chiadoi(f,a,b,saio); nghiemChiaDoi = chiadoiRes(1,1); loopChiaDoi = chiadoiRes(1,2); fprintf('nghiem ~ %3f\n',nghiemChiaDoi); fprintf('so lan lap = %d\n',loopChiaDoi);  fprintf('Nghiem va so lan lap cua pp lap: \n') lapRes = lap(f,a,b,saio); nghiemLap = lapRes(1,1); loopLap = lapRes(1,2); fprintf('nghiem ~ %3f\n',nghiemLap);</pre>	<p><b>Khai báo hàm f dưới dạng function_handle</b> <b>Fp là hàm hội tụ của f</b> <b>Khoảng phân li nghiệm với a ( start ) là 0.2 và b ( end ) là 1</b> <b>Sai số đề cho là 0.005</b> <b>In dòng chữ ra màn hình để phân biệt với các phương pháp khác</b> <b>Tạo biến nhận giá trị từ hàm chiadoi với thông số f, a,b, saio truyền vào. Lúc này biến nhận giá trị gồm 2 phần tử nghiem vs count từ func file chiadoi.m nên ta phải lấy từng phần tử ra từ nghiệm chính rồi thực hiện in ra thông qua hàm fprintf, %3f lấy 3 số sau dấu phẩy, %d lấy số nguyên từ biến</b></p> <p><b>Tương tự như trên với phương pháp lặp</b></p>

	<pre> fprintf('so lan lap = %d\n', loopLap);  fprintf('Nghiem va so lan lap cua pp tiếp tuyen: \n') ttRes = tieptuyen(f,a,b,saio); nghiemTt = ttRes(1,1); loopTt = ttRes(1,2); fprintf('nghiem ~ %3f\n',nghiemTt); fprintf('so lan lap = %d\n', loopTt);  fprintf('Nghiem va so lan lap cua pp day cung: \n') dcRes = daycung(f,a,b,saio); nghiemDc = dcRes(1,1); loopDc = dcRes(1,2); fprintf('nghiem ~ %3f\n',nghiemDc); fprintf('so lan lap = %d\n', loopDc); </pre>	<p><i>Tương tự như trên với phương pháp tiếp tuyến</i></p> <p><i>Tương tự như trên với phương pháp dây cung</i></p>
lap.m	<pre> function answer = lap(fp,a,b,saio)  count = 0; n = a; m = b; nghiem = 0; while ( abs(m-n) &gt;= saio)      count = count + 1;      n = m;     m = fp(n);     nghiem = m; end answer = [nghiem count]; end </pre>	<p>Tạo hàm lặp với các tham số truyền fp ( hàm lặp hội tụ của f) , a là bắt đầu khoảng phân li nghiệm, b là kết thúc khoảng phân li nghiệm, sai số yêu cầu</p> <p>Đặt số vòng lặp là count ban đầu = 0</p> <p>Đặt biến n và m là 2 giá trị ban đầu (x0, x1), khai báo nghiem ban đầu = 0;</p> <p>Chạy vòng lặp kiểm tra đến khi nào giá trị tuyệt đối của <math> x_i - x_{i-1}  &lt; \text{saio}</math> thì dừng (y/c đề)</p> <p>Nếu <math> x_i - x_{i-1}  \geq \text{saio}</math> thì vòng lặp thêm 1</p> <p>Lúc này <math>x_{i-1}</math> nhận giá trị tại x(i)</p> <p>Lúc này x(i) nhận giá trị fp(chính nó)</p> <p>Nghiem cập nhật liên tục tại vị trí x(i) cuối cùng</p> <p>Báo cáo kết quả với tham số thứ nhất là nghiệm và thứ 2 là số lần lặp</p>
chiadoi.m	<pre> function answer = chiadoi(f,a,b,saio)  count = 0; while b-a &gt;= saio      count = count + 1;     c = (a+b)/2;     if f(c)*f(a)&lt;0         b = c;     else         a = c;     end end nghiem = a; answer = [nghiem count]; </pre>	<p>Tạo hàm chiadoi với các tham số truyền hàm f , a là bắt đầu khoảng phân li nghiệm, b là kết thúc khoảng phân li nghiệm, sai số yêu cầu</p> <p>Đặt số vòng lặp là count ban đầu = 0</p> <p>Chạy vòng lặp kiểm tra đến khi nào <math>b-a &lt; \text{saio}</math> thì dừng lại</p> <p>Nếu <math>b-a \geq \text{saio}</math> thì vòng lặp thêm 1</p> <p>Lúc này <math>c = (a+b)/2</math></p> <p>Điều kiện nếu <math>f(c)*f(a) &lt; 0</math> thì:</p> <p>B nhận giá trị tại c</p> <p>Ko thì a nhận giá trị tại c</p> <p>Nghiem cập nhật liên tục tại vị trí a cuối cùng</p>

	end	Báo cáo kết quả với tham số thứ nhất là nghiệm và thứ 2 là số lần lặp
tieptuyen.m	<pre> function answer = tieptuyen(f,a,b,saiso)      syms x;     fd1 = str2func(['@(x)' char(diff(f(x)))]);     fd2 = str2func(['@(x)' char(diff(fd1(x)))]);     x0 = a;     while ( f(x0)*fd2(x0) &lt;= 0)         x0 = (x0+b)/2;     end     count = 0;      x1 = x0 - ( f(x0) / fd1(x0) );     while abs(x1-x0) &gt;= saiso         x0 = x1;         x1 = x0 - ( f(x0) / fd1(x0) );         nghiem = x1;         count = count + 1;     end     answer = [nghiem count]; end </pre>	<p>Tạo hàm chiadoi với các tham số truyền hàm f , a là bắt đầu khoảng phân li nghiệm, b là kết thúc khoảng phân li nghiệm, sai số yêu cầu</p> <p>Khai báo x kiểu symbol</p> <p>Fd1 là đạo hàm bậc 1 tại f thông qua hàm diff thông số f</p> <p>Fd2 là đạo hàm bậc 2 tại f thông qua hàm diff thông số fd1</p> <p>Chọn x0 = a</p> <p>Chạy vòng lặp kiểm tra cho đến khi <math>f(x_0) \cdot f'(x_0) &gt; 0</math> thì dừng lại còn ko thì gán <math>x_0 = (x_0+b)/2</math></p> <p>Lúc này bắt đầu gán lặp = 0</p> <p>Lặp giá trị <math>x_1 = x_0 - f[x_0]/f'[x_0]</math> đến khi giá trị tuyệt đối của <math>x(i) - x(i-1) &lt; \text{sai số}</math> thì dừng còn không thì vẫn cứ gán <math>x_0 = x_1</math> và <math>x_1 = x_1 - f[x_0]/f'[x_0]</math> (y/c đề)</p> <p>Nghiem cập nhật liên tục tại vị trí a cuối cùng</p> <p>Số lần lặp nhận thêm 1 gtri</p> <p>Báo cáo kết quả với tham số thứ nhất là nghiệm và thứ 2 là số lần lặp</p>
daycung.m	<pre> function answer = daycung(f,a,b,saiso)      count = 0;     while abs(b-a) &gt;= saiso          count = count + 1;         c = ( a * f(b) - b * f(a) ) / ( f(b) - f(a) );         if f(c)*f(a) &lt; 0             b = c;         elseif f(c)*f(a) &gt; 0             a = c;         else             break;         end         nghiem = c;     end     answer = [nghiem count]; end </pre>	<p>Tạo hàm dây cung với các tham số truyền hàm f , a là bắt đầu khoảng phân li nghiệm, b là kết thúc khoảng phân li nghiệm, sai số yêu cầu</p> <p>Đặt số vòng lặp là count ban đầu = 0</p> <p>Chạy vòng lặp kiểm tra đến khi nào <math>b-a &lt; \text{saiso}</math> thì dừng lại</p> <p>Nếu <math>b-a \geq \text{saiso}</math> thì vòng lặp thêm 1</p> <p>Lúc này <math>c = (af(b)-bf(a))/(f(b)-f(a))</math></p> <p>Điều kiện nếu <math>f(c) \cdot f(a) &lt; 0</math> thì:</p> <p>B nhận giá trị tại c</p> <p>Ko thì a nhận giá trị tại c</p> <p>Nếu = 0 thì dừng lại, excute</p> <p>Nghiem cập nhật liên tục tại vị trí a cuối cùng</p> <p>Báo cáo kết quả với tham số thứ nhất là nghiệm và thứ 2 là số lần lặp</p>

**Kết quả:**

Phương Pháp	Nghiệm gần đúng	Số lần lặp
Chiadoi	0.665625	8
Lap	0.667130	6
Tiep tuyen	0.666667	1
Day cung	0.666667	9

Phương pháp cho ít số lần lặp nhất là tiếp tuyến vì tại bước gán  $x_1 = x_0 - (f(x_0) / fd1(x_0))$  thứ 2 trong lặp và sau khi đã kiểm tra điều kiện  $f(x_0)*f'(x_0) > 0$  thì giá trị  $x_0$  đã được thay đổi nhiều gần như  $\sim$  là điểm fourier nên số vòng lặp sẽ được hạn chế rất nhiều.

**b.  $x - \sin x - 0.25 = 0$   $[-1,2]$  sai số : 0.005**

File	Code	Giải thích
main.m	<pre> f = @(x) x - sind(x) - 0.25; fp = @(x) sind(x) + 0.25; a = -1; b = 2; saio = 0.005;  fprintf('Nghiệm và số lần lặp của pp chia doi: \n') chiadoiRes = chiadoi(f,a,b,saio); nghiệmChiaDoi = chiadoiRes(1,1); loopChiaDoi = chiadoiRes(1,2); fprintf('nghiệm ~ %3f\n',nghiệmChiaDoi); fprintf('số lần lặp = %d\n',loopChiaDoi);  fprintf('Nghiệm và số lần lặp của pp lap: \n') lapRes = lap(f,a,b,saio); nghiệmLap = lapRes(1,1); loopLap = lapRes(1,2); fprintf('nghiệm ~ %3f\n',nghiệmLap); fprintf('số lần lặp = %d\n',loopLap);  fprintf('Nghiệm và số lần lặp của pp tiep tuyen: \n') ttRes = tieptuyen(f,a,b,saio); nghiệmTt = ttRes(1,1); loopTt = ttRes(1,2); fprintf('nghiệm ~ %3f\n',nghiệmTt); fprintf('số lần lặp = %d\n',loopTt);  fprintf('Nghiệm và số lần lặp của pp day cung: \n') dcRes = daycung(f,a,b,saio); </pre>	<p>Khai báo hàm f dưới dạng function_handle  Fp là hàm hội tụ của f  Khoảng phân li nghiệm với a ( start ) là 0.2 và b ( end ) là 1  Sai số đề cho là 0.005  In dòng chữ ra màn hình để phân biệt với các phương pháp khác  Tạo biến nhận giá trị từ hàm chiadoi với thông số f, a,b, saio truyền vào. Lúc này biến nhận giá trị gồm 2 phần tử nghiệm vs count từ func file chiadoi.m nên ta phải lấy từng phần tử ra từ nghiệm chính rồi thực hiện in ra thông qua hàm fprintf, %3f lấy 3 số sau dấu phẩy, %d lấy số nguyên từ biến</p> <p>Tương tự như trên với phương pháp lặp</p> <p>Tương tự như trên với phương pháp tiếp tuyến</p>

	<pre> ngkiemDc = dcRes(1,1); loopDc = dcRes(1,2); fprintf('ngkiem ~ %3f\n',ngkiemDc); fprintf('so lan lap = %d\n',loopDc); </pre>	<i>Tương tự như trên với phương pháp dây cung</i>
Chiadoi.m, lap.m, tieptuyen.m, daycung.m	<b>Giống như của câu a</b>	<b>Giống như của câu a</b>

**Kết quả:**

Phương Pháp	Nghiệm gần đúng	Số lần lặp
Chiadoi	0.253906	10
Lap	0.254450	3
Tiep tuyen	0.254441	1
Day cung	0.254441	4

Phương pháp nhận số lần lặp nhiều nhất là phương pháp chia đôi bởi các bước so sánh hiệu b-a sau đó nhận giá trị so sánh từ hiệu rồi áp điều kiện thực hiện 1 cách thủ công từng bước không thông qua hàm nên nảy sinh nhiều lần lặp.

### c. Thêm code vào tieptuyen

Code	Giải thích
<pre> function answer = tieptuyen(f,a,b,saio)     syms x;     fd1 = str2func(['@(x)' char(diff(f(x)))]);     fd2 = str2func(['@(x)' char(diff(fd1(x)))]);     nostop = 1;     temp1 = double(solve(diff(f(x))));     temp2 = double(solve(diff(fd1(x))));     if ~isempty(temp1)         for i=1:length(temp1)             if (temp1(i) &lt;= b) &amp;&amp; (temp1(i) &gt;=a)                 disp('Ham f'' doi dau trong khoang phan li ngkiem');                 nostop = 0;                 ngkiem = 'khong xac dinh';                 count = 'khong xac dinh';                 break;             end         end     end     if ~isempty(temp2)         for i=1:length(temp2)             if (temp2(i) &lt;= b) &amp;&amp; (temp2(i) &gt;=a)                 disp('Ham f'' doi dau trong khoang phan li ngkiem');                 nostop = 0;                 ngkiem = 'khong xac dinh';                 count = 'khong xac dinh';                 break;             end         end     end end </pre>	<p><b>-Đoạn code được thêm vào</b></p> <p><b>Nostop là biến điều kiện tiên quyết để quyết định có cho phép chạy thuật toán tính nghiệm vs sll hay không</b>  <b>Temp1 nhận giá trị các nghiệm khi tính hàm f'=0</b>  <b>Temp2 nhận giá trị các nghiệm khi tính f''=0</b>  <b>Thực hiện check xem temp1 vs temp2 có rỗng hay không sau đó xác định chúng có nằm trong khoảng phân li nghiệm hay không, không thì return không xác định</b></p>

<pre> end end if (nostop)     x0 = a;     while ( f(x0)*fd2(x0) &lt;= 0)         x0 = (x0+b)/2;     end     count = 0;     x1 = x0 - ( f(x0) / fd1(x0) );     while abs(x1-x0) &gt;= saiso         x0 = x1;         x1 = x0 - ( f(x0) / fd1(x0) );         nghiem = x1;         count = count + 1;     end     answer = [nghiem count]; end end </pre>	<p>Nếu nostop = 1 thì cho phép chạy logic tính vòng lặp vs nghiệm thông qua phương pháp tiếp tuyến nếu thỏa đk trên</p>
--	---

$x^2 - \sin \pi x = 0$  sai số 0.00005

	Chia đôi	Tiếp tuyến
-0.5 , 0.5	0.496094	Khong xac dinh
-0.05 , 0.5	-0.002734	Khong xac dinh

Sự khác biệt giữa 2 trường hợp: khoảng phân li nghiệm càng hẹp tương ứng với giá trị nghiệm càng bị phân rã

### Câu 3:

a. Tìm đa thức nội suy với

$x = [0.1 \ 0.2 \ 0.3 \ 0.4]$

$y = [0.09983 \ 0.19867 \ 0.29552 \ 0.38942]$

File	Code	Giải thích
lagrange.m	<pre> function dathuc = lagrange(x_arr, y_arr)      n = length(x_arr);     sub_arr = ones(1,n);      t = sym(sub_arr);      syms x;     for i = 1:n         for j = 1:n              if i ~= j                 t(i) = t(i) * (x - x_arr(j))/(x_arr(i) - x_arr(j));             end         end     end </pre>	<p>Tạo hàm lagrange với tham chiếu mảng x vs mảng y</p> <p>N nhận kích thước mảng x</p> <p>Tạo mảng phụ 1 hàng n phần tử = 1</p> <p>Cấu trúc mảng phụ thành symbolic 1 hàng n cột lưu từ biến t</p> <p>Khai báo x symbolic</p> <p>Chạy vòng lặp từ i -&gt; n lồng vào lặp từ j-&gt;n nhằm duyệt các phần tử trong x vs y</p> <p>So sánh nếu i khác j thì:</p> <p>Phần tử tại i của t = t(i) * (x - x_arr(i))/ (x_arr(i) - x_arr(j)) ( tương tự như phép tính hàm l(x) tại index)</p>

	<pre> end dathuc = (t * y_arr'); end </pre>	<p>Lúc này ta được mảng t gồm các l(x) từ index 1 đến n</p> <p>Theo công thức, tiếp tục nhân từng index của y với t, ta được hàm symbol cần tìm</p>
<b>Main.h</b>	<pre> x_arr = [0.1 0.2 0.3 0.4]; y_arr = [0.09983 0.19867 0.29552 0.38942]; syms x; f = lagrange(x_arr,y_arr); disp(f); </pre>	<p><b>Mảng x theo đề</b></p> <p><b>Mảng y theo đề</b></p> <p><b>F là hàm cần tìm</b></p> <p><b>Xuất f ra màn hình</b></p>

**Kết quả :**  $f(x) = (19867*(10*x - 1)*(x - 2/5)*(x - 3/10))/2000 - (3694*(5*x - 1/2)*(x - 1/5)*(x - 2/5))/125 - (9983*(10*x - 2)*(x - 2/5)*(x - 3/10))/6000 + (19471*((10*x)/3 - 1/3)*(x - 1/5)*(x - 3/10))/1000$

**b. Không dùng hàm – dùng simplify – dùng cả simplify vs vpa**

Phương thức	Code	Kết quả
<b>Không dùng hàm</b>	<pre> x_arr = [0.1 0.2 0.3 0.4]; y_arr = [0.09983 0.19867 0.29552 0.38942]; syms x; f = lagrange(x_arr,y_arr); disp(f);  ff = simplify(f); disp(ff);  fff = vpa(simplify(f)); disp(fff); </pre>	<b>Như câu a</b>
<b>Dùng simplify</b>		<b>(20013*x)/20000 - (7*x^2)/2000 - (4*x^3)/25 - 1/25000</b>
<b>Dùng cả simplify vs vpa</b>		<b>- 0.16*x^3 - 0.0035*x^2 + 1.00065*x - 0.00004</b>

**c.  $f(0.14) = 0.13954336$   
 $f(0.46) = 0.44394464$**

**d.  $\sin(0.14) = 0.1395431146$  (sai số: 0.0000004854)  
 $\sin(0.46) = 0.443948107$  (sai số: 0.000003467)**