Modul praktikum metode numerik IV

Matlab untuk matematika

|  |  |  |  |
| --- | --- | --- | --- |
| Nama |  |  | **Judul praktikum** |
| Nim |  | Limit , differential, Fungsi logaritma , logaritma natural, persamaan eksponensial , gauss |
| Kelas |  |

**

# LIMIT di matlab



syntax

>>Syms x

>> f=4\*x/(sqrt(x^2+4\*x+1)- sqrt(x^2+6\*x+1))

f =

(4\*x)/((x^2 + 4\*x + 1)^(1/2) - (x^2 + 6\*x + 1)^(1/2))

>> limit(f)

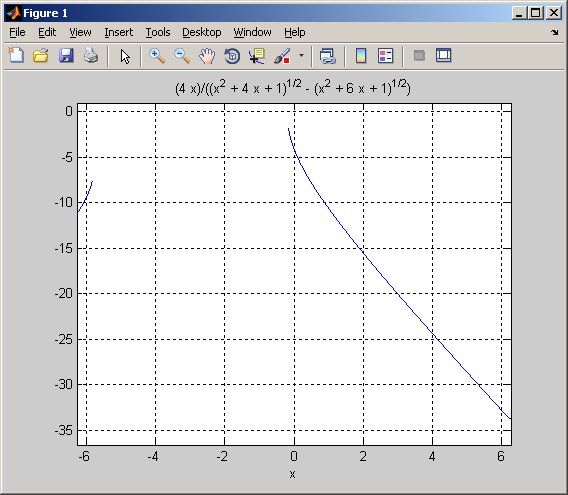
ans = -4

# LIMIT di scilab



# tampilan plot fungsi f

>>ezplot(f)

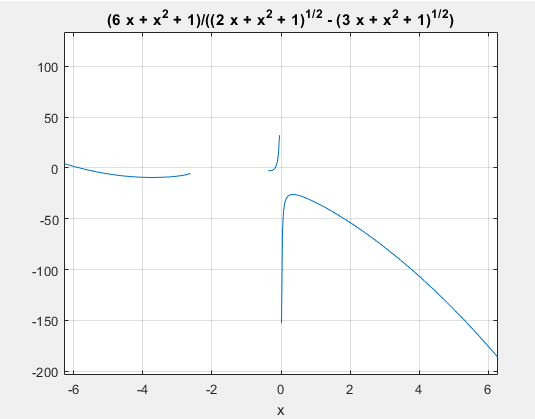
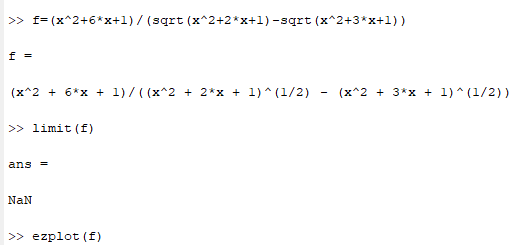


untuk di coba :



Tampilkan proses , hasil dan grafik plotnya :

#### Copas disini proses, hasil, dan grafik pengerjaannya :



# Differential / turunan matlab

Carilah turunan dari persamaan di bawah ini ,



syntax

>> y = x^3+3\*x^2-5\*x+7

y =

x^3 + 3\*x^2 - 5\*x + 7

>> turunanpertama=diff(y,1)

turunanpertama =

3\*x^2 + 6\*x - 5

>> turunankedua=diff(y,2)

turunankedua =

6\*x + 6

# Differential / turunan scilab

di scilab hanya bisa untuk mencari nilai / bukan persmaan nya seperti matlab

-> x = poly(0,'x');

--> y=3^3+3\*x^2-5\*x+7

y =

34 -5x +3x²

--> dy = derivat(y)

dy =

-5 +6x

# Menampilkan 1 grafik dari 2 persamaan yang berbeda

>>ezplot(y)

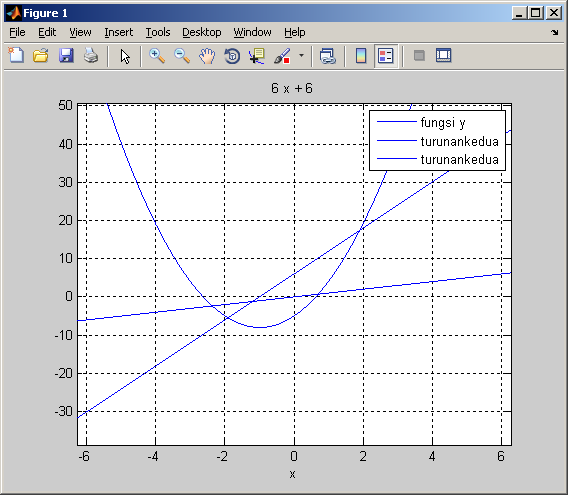
>>Hold on

Ezplot (turunanpertama)

>>hold on

>>ezplot(turunankedua)

>>legend(‘fungsi’,’turunanpertama’,’turunankedua’)



untuk di coba :



|  |
| --- |
| Copas disini proses, hasil, dan grafik pengerjaannya : |

# persamaan dengan Fungsi logaritma (log)

MATLAB bisa menyelesaikan persamaan aljabar menggunakan fungsi logaritma, namu hanya pada basis 10, jadi persamaan logaritma yang bukan basis 10 harus di ubah dahulau menjadi basis 10

# Di matlab



Harus di rubah terlebih dahulu



Syntax

>>solve (‘log10(x-2)=4\*log10(2)’)

# Di scilab

Untuk di coba

#



|  |
| --- |
| Copas disini proses, hasil, dan grafik pengerjaannya : |

# Fungsi logaritma natural (ln)

Fungsi logaritma natural (ln), dalam matlab pun menggunakan fungsi log



Syntax

>>solve(‘log(x)=5’)

Untuk di kerjakan

#



|  |
| --- |
| Copas disini proses, hasil, dan grafik pengerjaannya : |

# Persamaan fungsi eksponensial (*e*)

Penggunaan nya hampir sama dengan menggunakn fungsi logaritma natural secara analitik



Syntax

>>

>> eq=exp(x)-3\*x

eq =

exp(x) - 3\*x

>> solve(eq)

ans =

-lambertw(0, -1/3)

Nilai jika dalam double

>> double(ans)

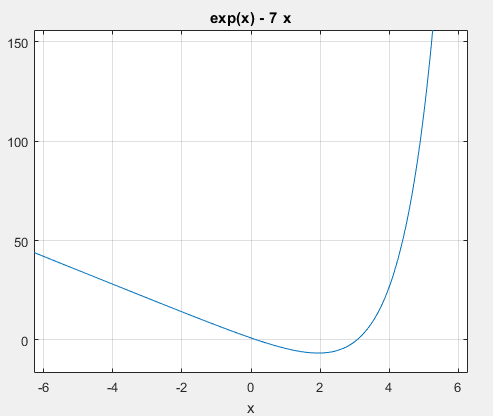
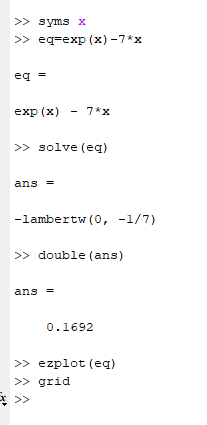
ans =

0.6191

>>

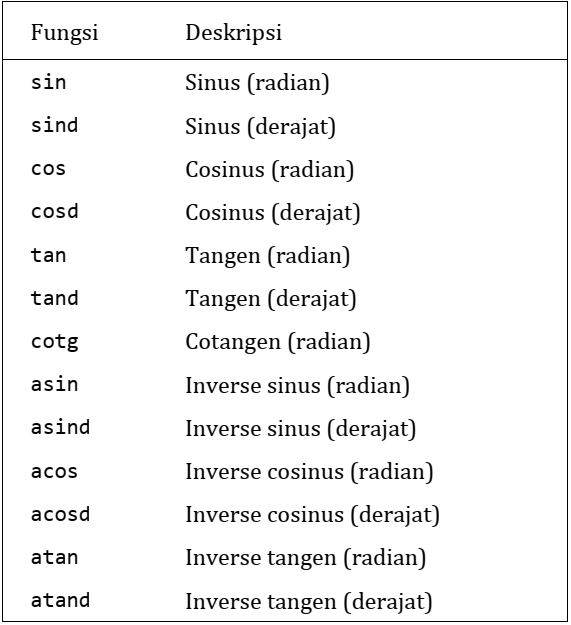
UNTUK DI KERJAKAN

Hitunglah persamaan dibawah ini dan sertakan gambar grafiknya



# Sistem persamaan Trigonometri

Penyelesaian persamaan trigonometri bisa menggunakan beberapa perintah :



Solve, simple , simplify , expand atau collect

## Contoh di matlab :



>> eq='cos(x)=0'

eq =

cos(x)=0

>> solve(eq)

ans =

pi/2

>>

### Tugas kerjakan soal matematika di bawah ini , printscreen proses dan hasilnya,

### Kerjakan 3 dari 6 soal yang ada



|  |
| --- |
| Copas disini proses, hasil, dan grafik pengerjaannya :  13.  14.  15. |

## Kerjakan 10 dari 16 soal yang ada



|  |
| --- |
| Copas disini proses, hasil, dan grafik pengerjaannya :  1.  2.  3.  4.  5.  6.  7.  9.  10.  11. |

# Iterasi gauss Seidell

## Scilab

//function c=f(xy),x=xy(1);y=xy(2);c=[(x+y-8);(2\*x+y-8)];endfunction // Example 8 . 2

// Gauss−S e i d e l Method

// Page no . 274

clc ; clear ; close ;

A=[4 , -1 ,1;4 ,-8 ,1;-2 ,1 ,5]; // e q u a t i o n matrix

B =[7;-21;15] // s o l u t i o n matrix

for i =0:10

X(i+1 ,1)=i;

end

for i =2:4

X(1,i)=0;

end

for r =1:10

for i =1:3

k1 =0;

for j=1:i -1

k1=k1 -A(i,j)\*X(r+1,j +1) ;

end

k2 =0;

for j=i +1:3

k2=k2 -A(i,j)\*X(r,j +1) ;

end

X(r+1,i +1) =( k1+k2+B(i ,1) )/A(i,i);

end

end

disp (A)

disp(B)

printf ( 'r\t x(r)\t\ty(r)\t z(r)');

printf ( ' \n −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− ' )

disp (X)

printf ( '\n\nAfter 10 iterations exact solutionis : \n')

printf('X :%f\t' ,X(10 ,2) )

printf('y :%f\t',X(10 ,3) )

printf('z :%f\t',X(10 ,4) )

## Matlab

clc

clear all

close all

close all hidden

format compact

%

disp('Matrik A')

A = [70 1 0;60 -1 1;40 0 -1]

%

disp('Matrik b')

C = [636 ; 518 ; 307]

%

n = length(C);

X = zeros(n,1);

Error\_eval = ones(n,1);

%

for i = 1:n

j = 1:n;

j(i) = [];

B = abs(A(i,j));

Check(i) = abs(A(i,i)) - sum(B);

if Check(i) < 0

end

end

iteration = 0;

while max(Error\_eval) > 0.001

iteration = iteration + 1;

Z = X;

for i = 1:n

j = 1:n;

j(i) = [];

Xtemp = X;

Xtemp(i) = [];

X(i) = (C(i) - sum(A(i,j) \* Xtemp)) / A(i,i);

end

Xsolution(:,iteration) = X;

Error\_eval = sqrt((X - Z).^2);

end

disp(' ')

disp('Proses Iterasi')

K = [1:iteration;Xsolution]'

%

disp(' ')

disp(['Jumlah Iterasi : ',num2str(iteration)])

disp(' ')

disp(['Tingkat Presisi : '])

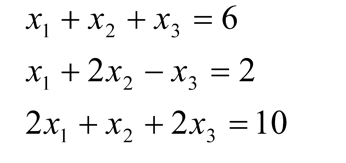
Error\_eval

disp(' ')

disp('Nilai X')

X

# Selesaikan persaman berikut dengan metode gauss seidell



|  |
| --- |
| Copas disini proses, hasil, dan grafik pengerjaannya :  Pada koding Scilab, iterasi akan dihentikan setelah 10 kali iterasi. Sehingga output hasilnya seperti screenshot diatas. Namun, tidak diketahui presisinya.  Pada MATLAB, iterasi akan dihentikan ketika tingkat presisinya <= 0.0001 sehingga iterasi akan looping jika belum mencapai tingkat presisi 0.0001.  Setelah program dijalankan, ternyata iterasi looping hingga 2272 dan menyebabkan hasil akhir menjadi NaN. Bisa jadi persamaan diatas tidak konvergen (divergen). |