

برای محاسبه تفاضل درایه های بردار سطری، ستونی (متغیر تابع) استفاده می شود.

$$A = [2 \ 2 \ 1 \ 0]$$

$$\text{diff}(A),$$

$$\text{ans} = 0 \ -1 \ -1$$

$$\begin{pmatrix} A(2) - A(1) & A(3) - A(2) & A(4) - A(3) \end{pmatrix}$$

تعداد درایه های یک کاهش پیدا کرد.

$$\text{del}x = 0.01;$$

برای متغیر

$$x = 0 : \text{del}x : 1; \quad m = \text{length}(x) \rightarrow \text{تعداد درایه های } x$$

$$dx = \text{diff}(x),$$

تعداد درایه های کمتر از x

$$y = \sin(x);$$

برای تابع:

$$dy = \text{diff}(y)$$

تعداد درایه های کمتر از x

شتن تقریبی

$$yp = \text{diff}(y) ./ \text{diff}(x)$$

معادل با تقسیم تقریبی

$$y'(x_1) = \frac{y_2 - y_1}{x_2 - x_1}$$

$$xp = x(1:m-1); \quad \text{موقع شتن مرتفعه (تغییر شتن)}$$

مرحله $\text{del}x$ کوچکتر باشد، شتن دقیق تر محاسبه می شود

مثال: اندک ترین تابع نویسنده با دریافت متغیر x و تابع y ، به محاسبه شتن تابع پرداخته و خروجی به شکل ماتریس دو سطری باشد که در سطر اول تغییر شتن و در سطر دوم شتن تابع قرار گیرد.

پس در یک برنامه اصلی، با دریافت حد پایین، حد بالا و تعداد تقسیمات، تغییرات تقریبی محاسبه و با فراخوانی تابع، شتن تقریبی تابع $\sin(x) + \cos(x)$ را محاسبه نموده و با شتن دقیق در یک نمودار رسم نماید.

$$\text{function } xpyp = \text{derf}(x, y)$$

$$m = \text{length}(x); \quad \text{نام تابع (این نام دخیوی شود)}$$

$$dx = \text{diff}(x);$$

$$dy = \text{diff}(y);$$

$$xp = x(1:m-1); \quad yp = dy ./ dx;$$

$$xpyp = \text{zeros}(2, m-1);$$

$$xpyp(1,:) = xp;$$

$$xpyp(2,:) = yp;$$

$$\text{function } xpyp = \text{derf}(x, y)$$

$$m = \text{length}(x);$$

$$dx = \text{diff}(x);$$

$$dy = \text{diff}(y);$$

$$xp = x(1:m-1); yp = dy ./ dx;$$

$$xpyp = \text{zeros}(2, m-1);$$

$$xpyp(1,:) = xp;$$

$$xpyp(2,:) = yp;$$

clc

code_der برنامه اصلی

clear

abn=input('Enter [a b n]= ');

a=abn(1); b=abn(2); n=abn(3);

$\text{del}x = \frac{b-a}{n}$;

x=a:delx:b;

y=sin(x)+cos(x);

xpyp=derf(x,y);

xp=xpyp(1,:); yp=xpyp(2,:)

ype=cos(x)-sin(x);

plot(xp,yp,'--')

hold on

plot(x,ype)

hold off

clc

clear

abn=input('Enter [a b n]=');

a=abn(1); b=abn(2); n=abn(3);

delx=(b-a)/n;

x=a:delx:b;

y=sin(x)+cos(x);

xpyp=derf(x,y);

xp=xpyp(1,:); yp=xpyp(2,:);

ype=cos(x)-sin(x);

plot(xp,yp,'--')

hold on

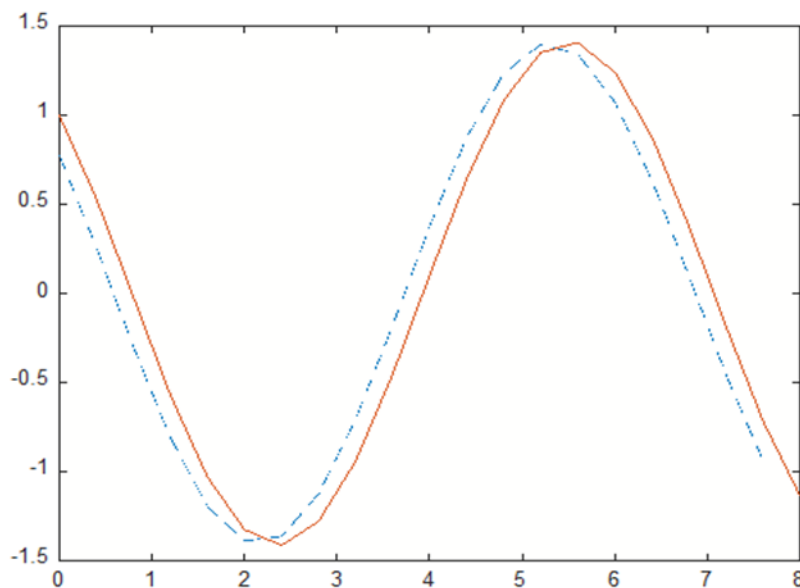
plot(x,ype)

hold off

خروجی 1 :

Enter [a b n]=[0 8 20]

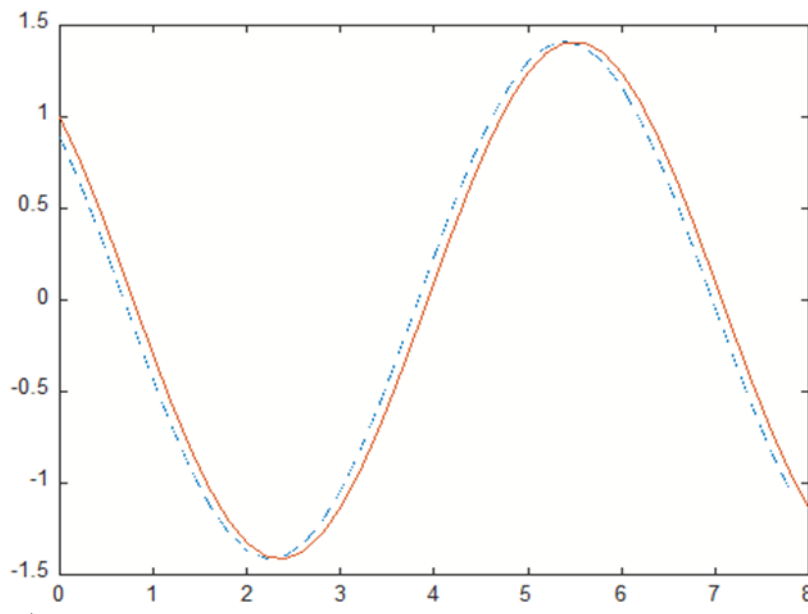
20 تیت



خودار جا غیر هموار و تناوت بین
روش تقریبی و دقیق نامی
نوع است.

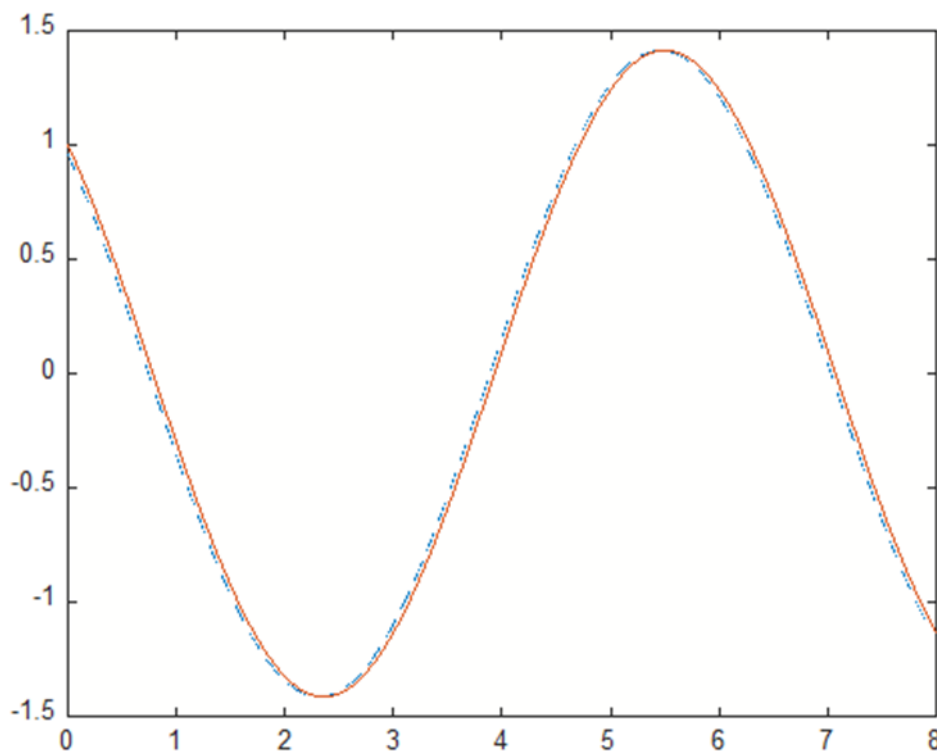
خروجی 2:

Enter [a b n]=[0 8 40]



خروجی 3:

Enter [a b n]=[0 8 100]



تمرین 8: مثال فوق را برای n متغیر دوم انجام دهید.

مثال $y = \sin(x) + \cos(x)$

مثال: با استفاده روش سری و خطای برنامه ای بنویسید تا با دریافت حدس اولیه و میزان خطای
معاده زیر را حل نماید.

$CD(x) = x$

e همان مرجع سنخس خطایی می باشد اگر خطای کمتر از $error$ باشد عملیات تکرار متوقف می شود

clc

clear

ge=input('Enter [g,err0]= ');

g = ge(1); err0=ge(2);

err=err0+1; x1=g;

while err>err0

x=cos(x1);

err=abs((x-x1)/x1);

x1=x;

end

disp(['x=', num2str(x1)])

clc

clear

ge=input('Enter [g,err0]=');

g=ge(1); err0=ge(2);

err=err0+1; x1=g;

while err>err0

x=cos(x1);

err=abs((x-x1)/x1);

x1=x;

end

disp(['x=', num2str(x1)])

سه خروجی:

Enter

[g,err0]=[0.5,0.01]

x=0.74183

>> cos(x)

ans =

0.7372

Enter

[g,err0]=[0.5,0.001]

x=0.73934

>> cos(x)

ans =

0.7389

Enter

[g,err0]=[0.5,0.0001]

x=0.73911

>> cos(x)

ans =

0.7391

خط 2

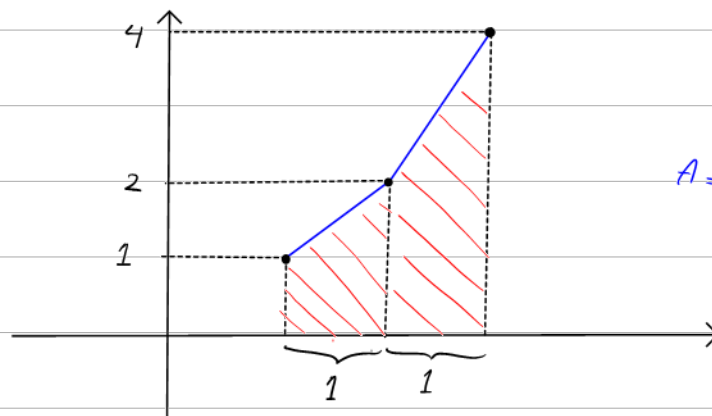
:trapz

این دستور یک تابع را به صورت یک بردار بفرمایید یا ستونی که مرتبه و با در نظر گرفتن تغییراتی با گام 1 (delta x=1) به ناسب مساحت سطح زیر منحنی پردازد.

B = [1 2 4];

K = trapz(B);

K = 4.5



$$A = \frac{3}{2} + 3 = 4.5$$

محاسبه انتگرال معین با استفاده از trapz

$$x = a; \text{del}x : b$$

تابع مشخص $y = f(x) \rightarrow$

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

$$I = \text{trapz}(y) \times \text{del}x,$$

هر مقدار تقسیمات $(\text{del}x)$ کوچکتر پاسخ انتگرال دقیق‌تری شود.

مثال 0.1

```
>> x=0:0.1:pi;
>> y=sin(x);
>> I=trapz(y)*0.1
```

I =

1.9975

مثال 0.01

```
>> x=0:0.01:pi;
y=sin(x);
I=trapz(y)*0.01
```

I =

2.0000

$$\int_0^{\pi} \sin(x) dx = 2$$
 پاسخ دقیق

تمرین 9: ابتدا یک تابع نویسی کرده با دریافت متغیر x و تابع f ، به محاسبه انتگرال تابع پرداخته و خروجی تابع حاصل انتگرال باشد.

پس در یک برنامه اصلی، با دریافت حد پایین، حد بالا و تعداد تقسیمات، متغیرات تعیین نموده و با فراخوانی تابع، انتگرال تقریبی تابع $\sin(x) + \cos(x)$ را محاسبه و چاپ نماید.

$$a=0, \quad b=\pi, \quad \text{del}x = 0.001$$

مثال اجرای برنامه

پس از اجرا با جواب دقیق مقایسه نمایید

مثال: برنامه ای بنویسید که با دریافت مرتبه ماتریس، یک ماتریس مربعی تشکیل دهد به مقدار درایه ها آن $i \times j$ و $(-1)^{i+j}$ (با شماره سطرها و ستون) باشد.

clc

clear

N=input('Enter N = ');

A = zeros(N);

for i=1:N

for j=1:N

A(i,j) = (-1)^(i+j) * i * j;

end

end

disp(A)

clc

clear

N=input('Enter N=');

A=zeros(N);

for i=1:N

for j=1:N

A(i,j)=(-1)^(i+j)*i*j;

end

end

disp(A)

Enter N=3

1	-2	3
-2	4	-6
3	-6	9

مثال: برنامه ای بنویسید که عدد طبیعی N را گرفته و تعیین کند که بر مجموع ارقامش بخش پذیر است یا نه؟

clc

clear

```
n=input('Enter n=');
S=0; m=1; nt=n;
while m>0
    a=rem(nt,10);
    m=floor(nt/10);
    S=S+a;
    nt=m;
end
r=rem(n,S);
if r==0
    disp('True')
```

```
else
    disp('False')
end
```

خروجی

Enter n=456
Sum=15
False

clc

clear

```
n=input('Enter n=');
S=0; m=1; nt=n;
while m>0
    a=rem(nt,10);
    m=floor(nt/10);
    S=S+a;
    nt=m;
end
r=rem(n,S);
disp(['Sum=',num2str(S)])
if r==0
    disp('True')
else
    disp('False')
end
```

خروجی

Enter n=102
Sum=3
True

مثال: برنامه ای بنویسید که یک عدد طبیعی را بگیرد و مقرب آن را چاپ کند.

clc

clear

```
n=input('Enter n=');
i=1; m=1;
while m>0
    a=rem(n,10);
    m=floor(n/10);
    R(i)=a; i=i+1;
    n=m;
end
```

تعیین ارقام
و ذخیره کردن
ارقام در R

K=length(R); تعداد ارقام

clc

clear

```
n=input('Enter n=');
i=1; m=1;
while m>0
    a=rem(n,10);
    m=floor(n/10);
    R(i)=a; i=i+1;
    n=m;
end
k=length(R);
nf=0;
for i=1:k
    nf=nf+R(i)*10^(k-i);
end
disp(nf)
```

$$np = 0;$$

for $i = 1:k$

$$np = np + R(i) \times 10^{(k-i)};$$

end

Enter n=1532540
452351

disp(np) نمایش عدد مطلوب

با استفاده از ارقام به دست آمده

$$R = [6 \ 5 \ 4], \quad k=3$$

مثلاً اگر عدد 456 باشد

$$np = 6 \times 10^2 + 5 \times 10 + 4 \times 10^0 = 654$$

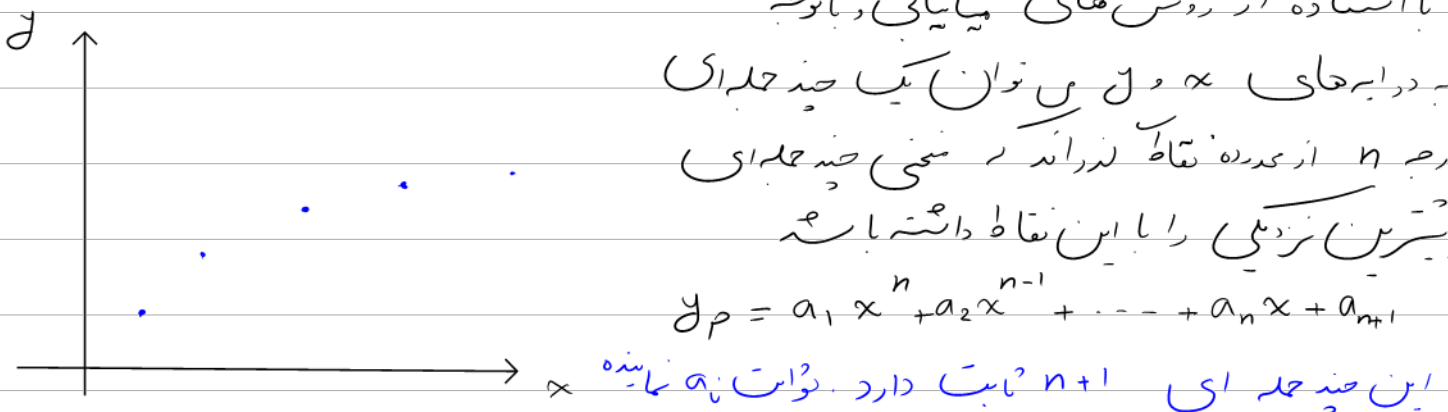
polyfit, polyval

متغیر برداری x تابع برداری y را در نظر بگیرید، در شکل زیر، نقاط متناظر با این تابع در صفحه x نشان داده شده است.

با استفاده از روش های میانجی، رابطه

به درجه های x و y می توان یک چند جمله ای درجه n از مجموعه نقاط گذراند که منحنی چند جمله ای بهترین نزدیکی را با این نقاط داشته باشد.

$$y_p = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1}$$



این چند جمله ای $n+1$ ثابت دارد. ضرایب به نمایشه x

چند جمله ای اند به شکل یک بردار سطر $A = [a_1 \ a_2 \ \dots \ a_n \ a_{n+1}]$ بیان می شوند.

در صورت داشتن بردار ضرایب چند جمله ای تعیین می شود $A = [1 \ 0 \ 0 \ 1] \Rightarrow y_p = x^3 + 1$

polyfit: در متلب با وارد نمودن x و y (متغیر تابع) و درجه چند جمله ای به دستور polyfit

می توان ضرایب چند جمله ای را به شکل بردار دریافت نمود.

* یک بردار سطر $n+1$ درایه می دهد. $A = \text{polyfit}(x, y, n)$

که این بردار نماینده چند جمله ای است.

$$x = [0 \ 1 \ 2], \quad y = [0 \ 1 \ 4];$$

$$A = \text{polyfit}(x, y, 2);$$

$$A = [1 \ 0 \ 0] \Rightarrow y_p = x^2 \Rightarrow \text{در این جا دقیقاً از نقاط تابع می گذرد}$$

```
>> x=[0 1 2];y=[0 1 4];
>> A=polyfit(x,y,2)
A =
    1.0000    0.0000   -0.0000
```

polyval: اگر جواب یک تابع چندجمله‌ای را داشته باشیم می‌توان مقدار تابع را در نقاط مختلف با استفاده از دستور polyval تعیین نمود

$y_p = \text{polyval}(A, x)$

مقدار متغیر در چند جمله‌ای x بردار جواب

x می‌تواند به صورت بردار یا عدد اسکالر باشد. اگر x یک بردار با m درایه باشد مقدار y_p در m نقطه محاسبه شده. y_p به شکل بردار هم مرتبه با x به دست می‌آید.

در $A = [1 \ 0 \ 0]$ و $x = 3$ $y_p = x^2$

```
>> x=3; A=[1 0 0];
>> yp=polyval(A,x)
```

در $y_p = \text{polyval}(A, x)$

$y_p = 9$

$yp =$

9

$y_p = 3^2 = 9$

در $A = [1 \ 0 \ 0 \ 1]$ و $x = [0 \ 1 \ 2 \ 3]$ $y_p = x^3 + 1$

```
>> x=[0 1 2 3];A=[1 0 0 1];
>> yp=polyval(A,x)
```

در $y_p = \text{polyval}(A, x)$

$yp =$

$y_p = 1 \ 2 \ 9 \ 28$

1 2 9 28

تمرین ۱۵: خودی دستورات زیر را محاسبه نمائید و با خودی متب مقایسه کنید

الف)

در $A = [1 \ 2 \ 1]$ و $x = [0 \ 2 \ 3 \ 4]$

در $y_p = \text{polyval}(A, x)$

در $A = [1 \ 0 \ 0 \ 0 \ 2 \ 5 \ 3]$ و $x = 2$

ب)

$y_p = \text{polyval}(A, x)$

polyder و **polyint**

اگر بخواهیم از چند جمله‌ای مشتق یا انتگرال بگیریم کافی است بردار جواب چند جمله‌ای را در دستورات **polyder** و **polyint** وارد نماییم. دو خودی دستورات، بردارهای جواب چند جمله‌ای مشتق و یا انتگرال ظاهر می‌گردد.

$$y_p = 3x^2 + 2x + 1 \quad y_p' = 6x + 2 \quad \int y_p dx = x^3 + x^2 + x$$

مثال:

$$A = [3 \ 2 \ 1];$$

$$dA = \text{polyder}(A),$$

$$dA = 6 \ 2$$

$$IA = \text{polyint}(A),$$

$$IA = 1 \ 1 \ 1 \ 0$$

$$>> A=[3 \ 2 \ 1];$$

$$>> dA=\text{polyder}(A)$$

$$A =$$

$$6 \ 2$$

$$>> IA=\text{polyint}(A)$$

$$IA =$$

$$1 \ 1 \ 1 \ 0$$

* می‌توان با مقدار دادن به بردار ثوابت چند جمله‌ای حاشیو، رانشال، از طریق دستور polyval، مقدار حشو و انشال توابع چند جمله‌ای را به دست آورد.

مثال: تابعی بنویسید که متغیر برداری x، تابع برداری n، و درجه چند جمله‌ای n را به عنوان ورودی دریافت کند و ثوابت چند جمله‌ای مرتبط با y را محاسبه نموده و پس به محاسبه بردار ثوابت حشو و انشال چند جمله‌ای پردازد.

پس در برنامه اصلی با دریافت حد پایین، حد بالا، تعداد نشیات متغیر و درجه چند جمله‌ای با فراخوانی تابع، به محاسبه متادیر تابع، حشو تابع و انشال تابع پرداخته و در سه مقدار متادیر تابع، حشو و انشال تابع را رسم و با جواب دقیق حاصل از انشالگیری و حشوگیری متقیم مقایسه نماید. مثال:

$$y = \sin^2(x) + \cos(x)$$

$$a = 0$$

$$b = \pi$$

$$n = 200$$

مثال: برنامه‌ای بنویسید ساعت ورود و خروج ده که پرسشی n کارمند را بگیرد و براساس بیشترین میزان حضور کارمندان را با نوشتن ده پرسشی مرتب نماید.

```

clc
clear
nt=input('Number of Employees=');
emp(1:nt)=struct('C',[],'d',[]);
n=nt;
for i=1:n
    disp(['Employee',num2str(i),':']);
    C(i)=input('Personnel Code=');
    A(i)=input('Employee entry time (for ex. 0802)=');
    B(i)=input('Employee departure time (for ex. 1634)=');

    e=floor(A(i)/100)*60+mod(A(i),100);
    f=floor(B(i)/100)*60+mod(B(i),100);

    d(i)=f-e;
end
while n>1
    for i=1:n-1
        a=d(i);b=d(i+1);Dp=C(i);Ep=C(i+1);
        if d(i)>d(i+1)
            d(i)=b;d(i+1)=a;C(i)=Ep;C(i+1)=Dp;
        end
    end
    n=n-1;
end
k=1;
disp(['*****Program output:'])
for i=nt:-1:1
    disp(['Employee',num2str(k),':']);
    disp(['Personnel Code=', num2str(C(i))]);
    disp(['Activity duration in minutes=', num2str(d(i))]);
    emp(k).C=C(i);emp(k).d=d(i);
    k=k+1;
end

```

ساختار: اختیاری

زمان حضور به دقیقه

کد پرسنلی

نمایش کارمند به همراه شماره ردیف

تبدیل شماره به رشته (حرف)

کد پرسنلی کارمند نام

زمان ورود

زمان خروج

مدت حضور به دقیقه

مرتب کردن براساس
الگوریتم حبابی

از بزرگ به کوچک

نمایش

امداد ساختار (اختیاری)

دریافت و خروجی:
دریافت

Number of Employees=4
Employee1:
Personnel Code=101
Employee entry time (for ex. 0802)=0800
Employee departure time (for ex. 1634)=1532
Employee2:
Personnel Code=102
Employee entry time (for ex. 0802)=0758
Employee departure time (for ex. 1634)=1525
Employee3:
Personnel Code=103
Employee entry time (for ex. 0802)=0755
Employee departure time (for ex. 1634)=1600
Employee4:
Personnel Code=104
Employee entry time (for ex. 0802)=0745
Employee departure time (for ex. 1634)=1528

*****Program output:

Employee1:
Personnel Code=103
Activity duration in minutes=485
Employee2:
Personnel Code=104
Activity duration in minutes=463
Employee3:
Personnel Code=101
Activity duration in minutes=452
Employee4:
Personnel Code=102
Activity duration in minutes=447

خروجی:

تمرین 11: برنامه ای بنویسید که یک عدد را از طرفه و مقوم علیه های اول آن را محاسبه و در یک بردار سطرین چاپ نماید.

حل دستگاه معادلات جبری:

* برای حل یک دستگاه معادلات جبری خطی یا غیر خطی، ابتدا معادلات متده را داخل یک بردار فرض می کنیم.

$$\begin{cases} x + 2y = 3 \\ 2x + 3y = 4 \end{cases}$$

مثال:

$$x = [x(1) \ x(2)] \quad x(1) = x \quad x(2) = y \quad \text{معادلات}$$

بردار نطری x به تعداد معادلات است

* بنابراین در سبب گننام نوشتن معادلات، معادلات بر حسب درایه‌های x ($x(2), x(1)$) نوشته می‌شود.

معادله اول: $x(1)$

معادله دوم: $x(2)$

* همچنین می‌توان معادلات نیز در سبب بردار نطری قرار می‌گیرد

$$F = [x(1) + 2 * x(2) - 3, \ 2 * x(1) + 3 * x(2) - 4]$$

در معادله داریم بنابراین F یک بردار نطری با دو درایه است که هر درایه عبارت یک معادله می‌باشد.

* برای حل، دستگاه معادلات را به شکل یک تابع با خروجی F به صورت زیر تعریف می‌کنیم:

دردی معادله (معادلات) $\rightarrow$ `function F=myfun1(x)`

$$F = [x(1) + 2 * x(2) - 3, \ 2 * x(1) + 3 * x(2) - 4];$$

حل در برنامه اصلی یا در Command window با توجه به دستور `fsolve` و دارد نمودن تابع و حدس اولیه به شکل بردار می‌توان دستگاه معادله را حل نمود.

$$x = fsolve(@myfun1, [0, 1])$$

$x(1)$ ← حدس برای $x(1)$
 $x(2)$ ~ ~ ~ $x(2)$

$$x_0 = [0, 1];$$

$x = fsolve(@myfun1, x_0)$ ⇒ خروجی شامل پاسخ دستگاه $x(1)$ ، $x(2)$ و ... می‌باشد. تابع:

`function F=myfun1(x)`

$$F = [x(1) + 2 * x(2) - 3, \ 2 * x(1) + 3 * x(2) - 4];$$

مراخوانی تابع و حل در صفحه دستور:

$$x0 = [0 \ 1];$$

$$x = fsolve(@myfun1, x0)$$

x =

خروجی:

-1.0000 2.0000 → z
x

* اگر داده بر مجهولات در دستگاه معادلات یک یا چند پارامتر داشته باشیم به خواهم
تا تغییر آن ها، تاثیر آن پارامتر را بر پاسخ برای سیستم بابت این پارامترها
را در ورودی تابع وارد نمائیم.

مثال:

$$\begin{cases} x + y + cz = 0 \\ x + 2y + z = 1 \\ 2x - y + 2z = d \end{cases}$$

دو پارامتر c و d

$$x = x(1) \quad y = x(2) \quad z = x(3)$$

function F=myfun2(x,c,d)

$$F = [x(1)+x(2)+c*x(3), x(1)+2*x(2)+x(3)-1, 2*x(1)-x(2)+2*x(3)-d];$$

در این حالت، در برنامه اصلی، دستور fsolve به صورت زیر به کار می رود

$x_0 = [0 \ 0 \ 1]; \rightarrow x(3)$ ، $x(2)$ ، $x(1)$ برای

$c = 2$ ، $d = 7$;

$x = fsolve(@ (x) myfun2(x,c,d), x_0); \Rightarrow$ در برنامه اصلی یا

صفحه دستور

$x_f = x(1)$ ، $y_f = x(2)$ ، $z_f = x(3)$ ،

تابع

function F=myfun2(x,c,d)

F=[x(1)+x(2)+c*x(3), x(1)+2*x(2)+x(3)-1, 2*x(1)-x(2)+2*x(3)-d];

خروجی

مراخوانی تابع و حل در صفحه دستور

>> x0=[0,0,1];

>> c=2;d=7;

>> x=fsolve(@ (x) myfun2(x,c,d),x0);

فرم fsolve

>> xf=x(1),yf=x(2),zf=x(3),

xf=

5.0000

yf=

-1.0000

zf=

-2.0000

تمرین 12: برنامه ای بنویسید به با نوشتن یک تابع و برنامه اصلی مقدار x را برای مقادیر مختلف c با اخذ حدس اولیه مناسب محاسبه نموده و مقدار x را بر حسب c در یک نمودار ترسیم نماید.

$$\begin{cases} x + y + cz = 0 \\ x + 2y + z = 1 \\ 2x - y + 2z = 4 \end{cases} \quad c = a : del c : b$$

یابست در برنامه اصلی a ، b و تعداد تئیات و حدس اولیه وارد شود و ورودی c در تابع وارد شود.

راحتی: تابع به تعداد درایه های c فراخوانی و حرکات درایه $(n \times 1)$ در تابع $(n \times 1)$ سود و پس از حل مقدار x به دست آمده در بردار $x(n \times 1)$ جایگزین گردد.

مثال: $n=100$, $b=9$, $a=7$, $x_0 = [1, 0, 0]$

حل معادله دینرانیل با شرایط اولیه:

① برای حل معادله دینرانیل، ابتدا معادلات را به فضای حالت می بریم. بین متغیر، به تعداد مرتبه معادله دینرانیل، متغیرهای فضای حالت قرار داده و معادلات دینرانیل در فضای حالت تعریف می شود.

مثال: $\dot{x} = 2x + 3y + 4z$ متغیر t تغییر شرایط اولیه $x(0)=0$, $y(0)=0$

معادله دینرانیل مرتبه

متغیرهای فضای حالت x_1, x_2
 $x_1 = x$, $x_2 = y$

معادله دینرانیل در فضای حالت:

* دو معادله دینرانیل مرتبه اول:

$$\dot{x}_1 = x_2 \quad \dot{x}_2 = -3x_2 - 4x_1 + 2t \Rightarrow x_2 = -3x_2 - 4x_1 + 2t$$

* در فضای حالت، سمت چپ، شامل مشتقات مرتبه اول متغیرهای فضای سمت راست شامل ترکیبی از متغیرهای فضای حالت است.

مثال فوق

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -4y_1 - 3y_2 + 2t$$

مثال:

مرتبه 3 شرایط اولیه $y(0), \dot{y}(0), \ddot{y}(0)$ به شرط

به تغییر فضای حالت $\ddot{y} + 2\dot{y} + y = t$ $\dot{y}_1 = \dot{y}, \dot{y}_2 = \ddot{y}, \dot{y}_3 = \ddot{y}$

سمت چپ مثال استغاثات تغییرهای فضای حالت

$$\dot{y}_1 = \dot{y}_2$$

$$\dot{y}_2 = \dot{y}_3$$

$$\dot{y}_3 = \ddot{y} = ? \Rightarrow \dot{y}_3 = -2\dot{y} - \dot{y} - 2y + t \Rightarrow \dot{y}_3 = -2y_1 - y_2 - 2y_3 + t$$

$$\begin{cases} \dot{y}_1 = \dot{y}_2 \\ \dot{y}_2 = \dot{y}_3 \\ \dot{y}_3 = -2y_1 - y_2 - 2y_3 + t \end{cases}$$

در حالت کلی $\dot{y}_1 = \dot{y}_2$
 مرتبه n $\Rightarrow \dot{y}_2 = \dot{y}_3$

ماتریس معادله دینامیک تغییرهای فضای حالت

② در ادامه برای معادلات فضای حالت، تابعی بنویسیم. خروجی این تابع شامل استغاثات توابع $y_1, y_2, \dots$ (تغیرهای حالت است) که به صورت یک بردار ستونی باشد

تغیر نام دینامیک $\text{function } dydt = \text{diffreq1}(t, y)$

n سطر برای $\dot{y}_1, \dot{y}_2, \dots, \dot{y}_n$ $dydt = [y(1); y(2); \dots; y(n)]$

برای مثال آخر

$$dydt = [y(2); y(3); -2*y(1) - y(2) - 2*y(3) + t];$$

③ حال با استفاده از دستور ode45 می توان با فراخوانی تابع، وارد نمودن حدود تغییر و شرایط اولیه، معادله دینامیک را حل نمود (در برنامه اصلی یا صفحه دستور)

$$[t, y] = \text{ode45}(@\text{diffreq1}, T, y_0);$$

T حدود تغییر t می باشد (محدوده زمانی که معادله دینامیک در آن محدود حل می شود)

y_0 : شرایط اولیه شامل شرایط اولیه تغییر فضای حالت $y_1(0), y_2(0), \dots, y_n(0)$ است.

خروجی t و y :

t : بردار ستونی تغییر

y : شامل بردارهای ستونی تغییرهای فضای حالت $y_1, y_2, \dots, y_n$ است

ستون اول $y_1 = y(:,1) = y$

ستون دوم $y_2 = y(:,2) = \dot{y}$

ستون n ام $y_n = y(:,n) = y^{(n)}$ مشتق مرتبه n

مثال: من خواهم پاسخ معادله دیفرانسیل زیر را برای بازه $0 \leq t \leq 10$ و شرایط اولیه زیر بدست آورم

$$\ddot{y} + 3\dot{y} + 4y = 2\sin(t)$$

$$y(0) = 0, \dot{y}(0) = 0$$

فضای حالت:

$$\dot{y}_1 = y_2 \quad y_1(0) = 0$$

$$\dot{y}_2 = -4y_1 - 3y_2 + 2\sin(t) \quad y_2(0) = 0$$

Function $dydt = \text{diffeq1}(t,y)$ ذخیره نام تابع

$$dydt = [y(2); -4*y(1) - 3*y(2) + 2*\sin(t)]$$

function dydt=diffeq1(t,y)
dydt=[y(2);-4*y(1)-3*y(2)+2*sin(t)];

clc در برنامه اصلی یا صفحه دستور

clear all

T = input('enter [t0 tf]= ');

y0 = input('enter [y1(0) y2(0)] = ');

[t,y] = ode45(@diffeq1,T,y0);

figure(1) نمایش گراف 1

plot(t,y(:,1)) نمودار y - t

figure(2) نمایش گراف 2

plot(t,y(:,2)) نمودار $\dot{y}$ - t

clc

clear all

T=input('enter [t0,tf]=');

y0=input('enter [y1(0);y2(0)]=');

[t,y]=ode45(@diffeq1,T,y0);

figure(1)

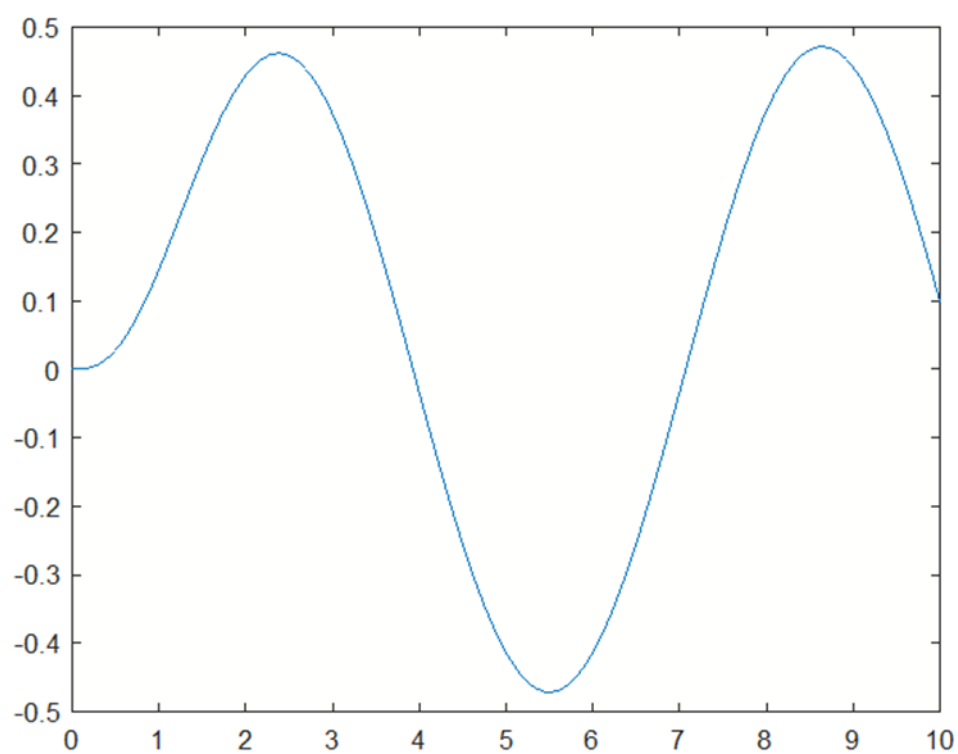
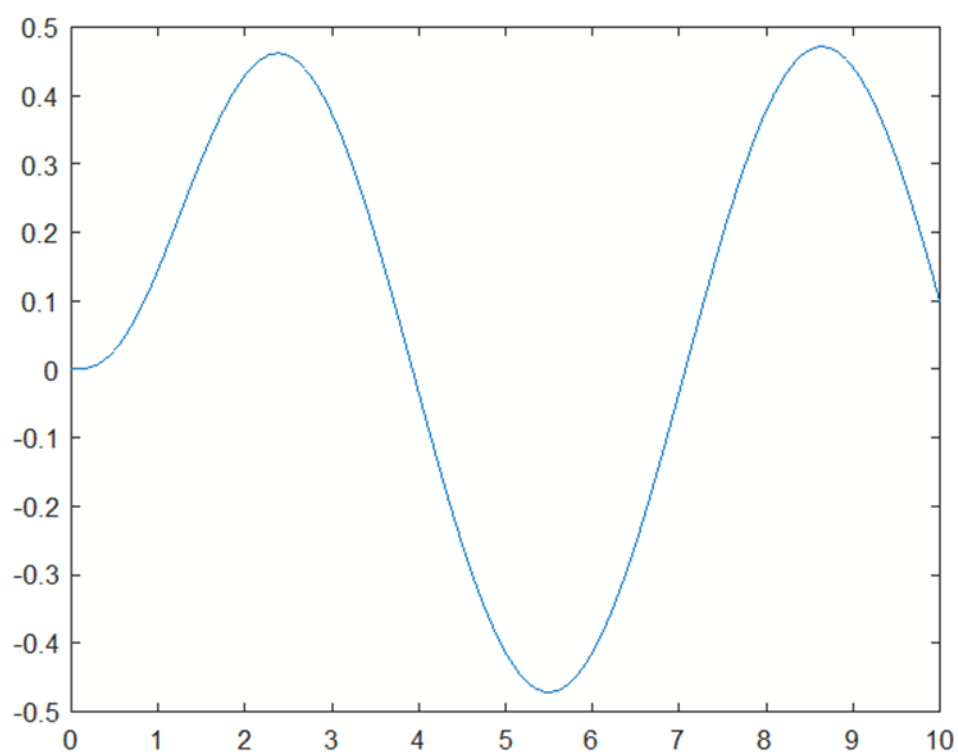
plot(t,y(:,1)) y

figure(2)

plot(t,y(:,2)) $\dot{y}$

```
enter [t0,tf]=[0,10]
```

```
enter [y1(0);y2(2)]=[0;0]
```



تمرین 13: مطابق با مثال قبل برای حل معادلات دیفرانسیل زیر تابع در برنامه اصلی نوشته و آن را مطابق با درودی زیر اجرا کنید:

$$\ddot{y} + 2\dot{y} + y = 0 \quad 0 < t < 5 \quad \text{الف)}$$

$$y(0) = 1, \quad \dot{y}(0) = 0, \quad \ddot{y}(0) = 0$$

$$\begin{cases} \ddot{z} + z + y = 0 \\ \ddot{y} + 2\dot{y} + z = 0 \end{cases} \quad \text{ب)}$$

$$y(0) = 0, \quad \dot{y}(0) = 1, \quad z(0) = 0$$

راحتی : سه متغیر فضای حالت داریم
معادلات ردی 1 مرتبه 2 ردی 2 مرتبه 1 است

$$\begin{aligned} \dot{x}_1 &= x_3 & \dot{x}_1 &= \dot{x}_3 \\ \dot{x}_2 = ? & \Rightarrow \dot{x}_2 = \dot{z} = -x_1 - \dot{x}_2 & \Rightarrow \dot{x}_2 &= -x_1 - \dot{x}_2 \\ \dot{x}_3 = ? & \Rightarrow \dot{x}_3 = \ddot{y} = -\dot{x}_2 - 2x_3 & \dot{x}_3 &= -\dot{x}_2 - 2x_3 \end{aligned} \quad \begin{cases} x_1(0) = 0 \\ x_2(0) = 0 \\ x_3(0) = 1 \end{cases}$$