

پروژه محاسبات عددی پیشرفته

۱. درونیابی اسپلاین

۲. درونیابی لاگرانژ

۳. برازش منحنی

DO NOT COPY (www.novin-eng.ir)

سوال :

داده‌های زیر مربوط به تراسه یک مدل راکت بر حسب زمان می‌باشد. توابع درونیابی بر اساس روش

لاگرانژ و اسپلاین مرتبه‌ی ۳ را به دست آورده، نتایج را با یکدیگر مقایسه کنید.

t	T
0.00	0.00
0.05	1.00
0.10	5.00
0.15	15.00
0.20	33.50
0.30	33.00
0.40	16.50
0.50	16.00

t	T
0.60	16.00
0.70	16.00
0.80	16.00
0.85	16.00
0.90	6.00
0.95	2.00
1.00	0.00

درونیابی اسپلاین و لاگرانژ

در کد برنامه ابتدا اسپلاین و سپس لاگرانژ بیان شده است

الگوریتم حل اسپلاین:

۰. تعریف کردن x به صورت پارامتری با تابع `syms`

۱. وارد کردن داده‌های مسئله در اکسل

۲. فراخوانی فایل اکسل با تابع `xlsread()`

۳. شناساندن تعداد داده‌ها با تابع `length()`

۴. تعریف ماتریس سه قطری در متلب

۵. حل ماتریس با الگوریتم توماس

۶. در این برنامه از حل کتاب استیون چاپرا برای درونیابی اسپلاین استفاده شده است

۷. محاسبه‌ی چند جمله‌ای اسپلاین

۸. در آخر نمایش نمودار چند جمله‌ای درونیابی را خواهیم داشت

الگوریتم حل لاگرانژ:

۱. وارد کردن داده های مسئله در اکسل
۲. فراخوانی فایل اکسل با تابع `xlsread()`
۳. شناساندن تعداد داده ها با تابع `Length()`
۴. نوشتن فرمول و محاسبه ی ضریب لاگرانژ
۵. محاسبه چند جمله ای لاگرانژ
۶. نمایش نموداری چند جمله ای درونیابی شده

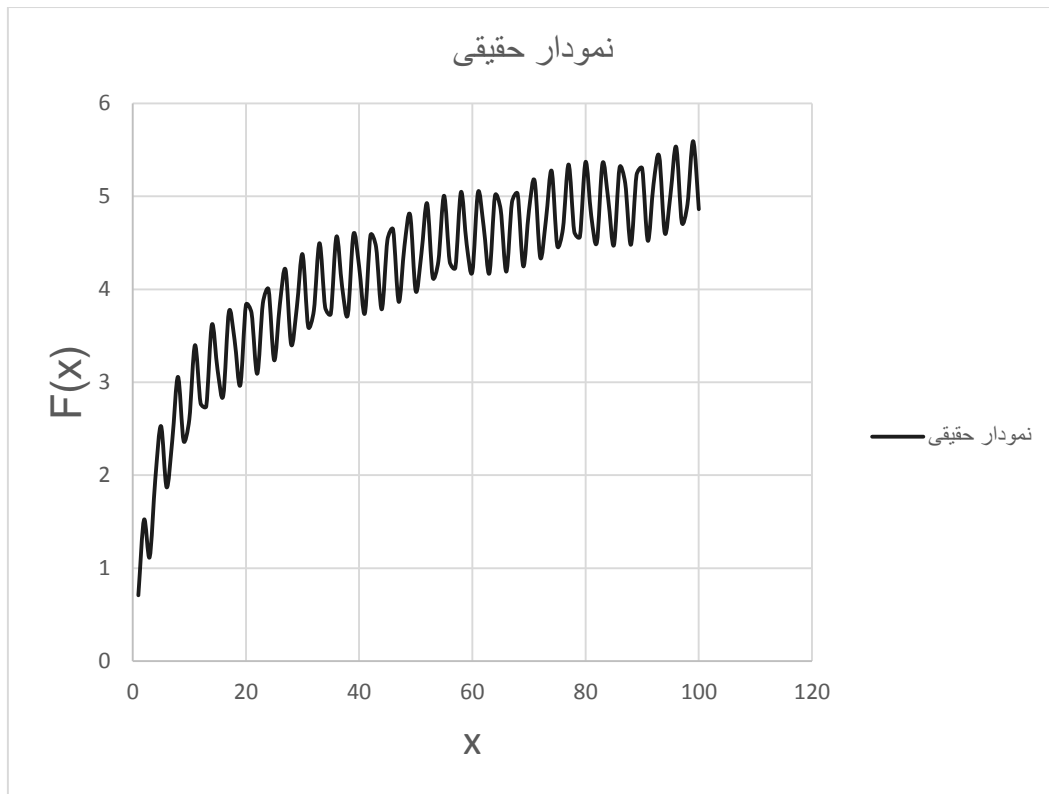
بخش دوم

برازش منحنی شماره ۱ :

سوال:

منحنی زیر را با توابع مناسب تقریب بزیند.

DO NOT COPY (www.novin-eng.ir)

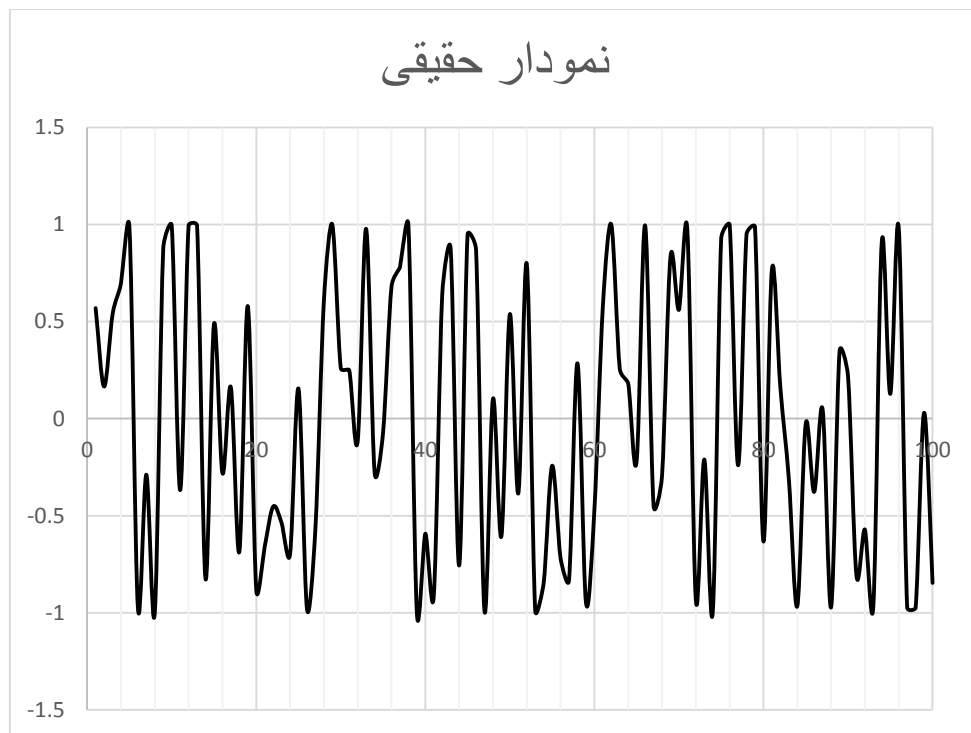


برازش منحنی شماره ۲ :

سوال:

منحنی زیر را با توابع مناسب تقریب بنمید.

DO NOT COPY (www.novin-eng.ir)



الگوریتم حل برازش منحنی ۱ و ۲:

۱. وارد کردن داده های مسئله در اکسل

۲. فراخوانی فایل اکسل با تابع `xlsread()`

۳. شناساندن تعداد داده ها با تابع `Length()`

۴. حدس چندجمله ای به صورت توانی از `x` برای برازش ۱ برنامه و حدس سینوسی برای برازش ۲ برنامه

۵. نوشتن رابطه ماتریسی برنامه و حل آن

۶. محاسبه خطا

۷. نمایش نمودار داده های ورودی و تابع حدس زده شده

کد اسپلاین :

```
%Spline Interpolation Polynomial
```

```
%-----
```

```
clc;
```

```
clear all;
```

```
%taarif parameter x -----
syms x;

%input data -----
Tt=xlsread('data.xlsx');

%number input data -----
n=length(Tt);
t=Tt(:,1);
T=Tt(:,2);

%Define Tridiagonal matrix and calculating Cubic Spline-----
A=zeros(n,n);
A(1,1)=1;
A(n,n)=1;

% Boundary points
c(1)=0;
c(n)=0;
fz(1)=c(1)/A(1,1);
fz(n) = c(n)/A(n,n);

for i=1:n-1
    h(i)=t(i+1)-t(i);
end
for j=1:n-2

%A(j+1,j)= subdiagonal vector
A(j+1,j)=h(j);

% A(j+1,j+1)= diagonal vector
A(j+1,j+1)=2*(h(j)+h(j+1));

% A(j+1,j+2)= superdiagonal vector
A(j+1,j+2)=h(j+1);

% c = right hand side vector
c(j+1)=((3*(T(j+2)-T(j+1)))/h(j+1))-((3*(T(j+1)-T(j)))/h(j));

%The solution of Tridiagonal matrix with Tomas Algorithm

factor =A(j+1,j)/A(j,j);
A(j+1,j+1) = A(j+1,j+1) - factor*A(j,j+1);
c(j+1) = c(j+1) - factor*c(j);
end
```