多項式微積分與數值微積分

Notebook | MATLAB

https://youtu.be/rhNcguIpFW8?si=8CId-YI7V09mZI5Y

怎样在matlab中呈现多项式

Polynomial Differentiation

- Polynomials are often used in numerical calculations
- For a polynomial

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

the derivative is

$$f'(x) = a_n n x^{n-1} + a_{n-1}(n-1)x^{n-2} + \cdots + a_1$$

> For example, consider the equation
>
> $f(x) = x^3 - 2x - 5$
>
> # To enter this polynomial into MATLAB, use
>
> # p = [1 0 -2 -5];

polyval() 多项式计算

y = polyval(p,x) 计算多项式 p 在 x 的每个点处的值。

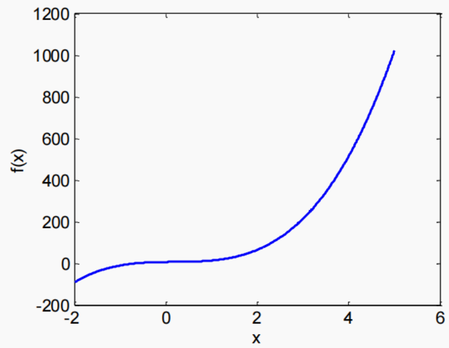
Values of Polynomials: `polyval()`

- Plot the polynomial

$$9x^3 - 5x^2 + 3x + 7$$

for $-2 \le x \le 5$

```
a = [9,-5,3,7];  x = -2:0.01:5;
f = polyval(a,x);
plot(x,f,'LineWidth', 2);
xlabel('x');  ylabel('f(x)');
set(gca, 'FontSize', 14)
```



Y=polyvalm(P,X)
%注意polyvalm函数跟polyval函数的区别，
polyvalm是针对第二个输入变量是矩阵的运算，**polyval**是针对第二个输入变量是标量或者向量的运算。

polyder() 多项式微分（求导）

```
p=[1 1 4 5 1 4];
polyder(p);
```

```
%polynomial differentiation在x=7时对应的数值
polyval(polayder(p),7);
```

📖 **p=polyder(a)：求多项式a的导函数**
p=polyder(a,b)：求a·b的导函数
[p,q]=polyder(a,b)：求a/b的导函数，导函数的分子存入p，分母存入q。

上述函数中，参数a,b是多项式的向量表示，结果p,q也是多项式的向量表示。

多项式四则运算

👉 ***多项式的加减运算：*** 相同次数的，直接相加减；不同次数的，在较低次幂的多项式系数前补0。matlab本身没有对应的函数，可以自编函数进行不同次数的多项式相加。

📖 ***conv()*——计算多项式乘法,**

***deconv()*——计算多项式除法。**

polyint() 多项式积分——Polynomial Integration

```
p=[5 0 -2 0 1];
polyint(p,k);
% k为积分后出现的常数项
```

diff()

计算一个向量中前后相邻元素之间的差异。calculates the differences betweenadjacent elements of a vector。

后面的减前面的。

📖 dx=diff(X,n):计算X的n阶向前差分。例如，diff(X,2)=diff(diff(X))。

dx=diff(A,n,dim):计算矩阵A的n阶差分，dim=1时(缺省状态)，按列计算差分；dim=2，按行计算差分。

dx=diff(X):计算向量X的向前差分，dx(i)=x(i+1)-x(i)，i=1,2,...,n-1。

```
x = [1 2 5 2 1];
diff(x);

>> ans= 1 3 -3 -1
```

```
% Exercise: obtain the slope of a line between 2 points (1,5)
x = [1 2]; y = [5 7];
slope = diff(y)./diff(x) (1,5)
```

Numercial Integration 数值积分

向前差商： $f'(x_0) \approx \dfrac{f(x_0+h)-f(x_0)}{h}$

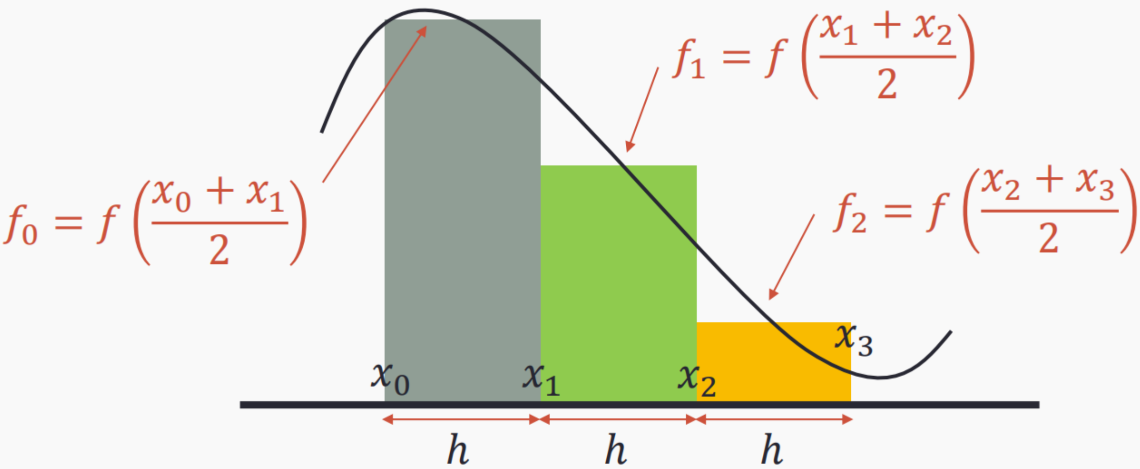
向后差商： $f'(x_0) \approx \dfrac{f(x_0)-f(x_0-h)}{h}$

中心差商： $f'(x_0) \approx \dfrac{f(x_0+h/2)-f(x_0-h/2)}{h}$

方法一：Midpoint Rulefang

Midpoint Rule

$$\int_{x_0}^{x_3} f(x)dx \approx hf_0 + hf_1 + hf_2 = h\sum_{i=0}^{n-1} f_i$$



Midpoint Rule Using `sum()`

- Example:

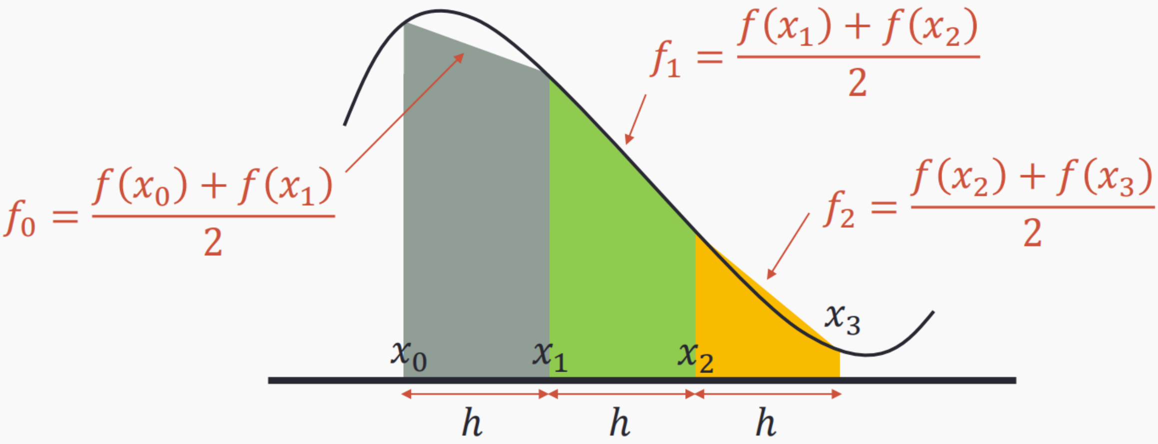
$$A = \int_0^2 4x^3\,dx = x^4|_0^2 = (2)^4 - (0)^4 = 16$$

```
h = 0.05;  x = 0:h:2;
midpoint = (x(1:end-1)+x(2:end))./2;
y = 4*midpoint.^3;
s = sum(h*y)
```

方法二：Trapezoid Rule

Trapezoid Rule

$$\int_{x_0}^{x_3} f(x)dx \approx h\frac{f_0+f_1}{2} + h\frac{f_1+f_2}{2} + h\frac{f_2+f_3}{2}$$



Trapezoid Rule Using `trapz()`

- Example:

$$A = \int_0^2 4x^3\,dx = x^4|_0^2 = (2)^4 - (0)^4 = 16$$

```
h = 0.05; x = 0:h:2; y = 4*x.^3;
s = h*trapz(y)
```

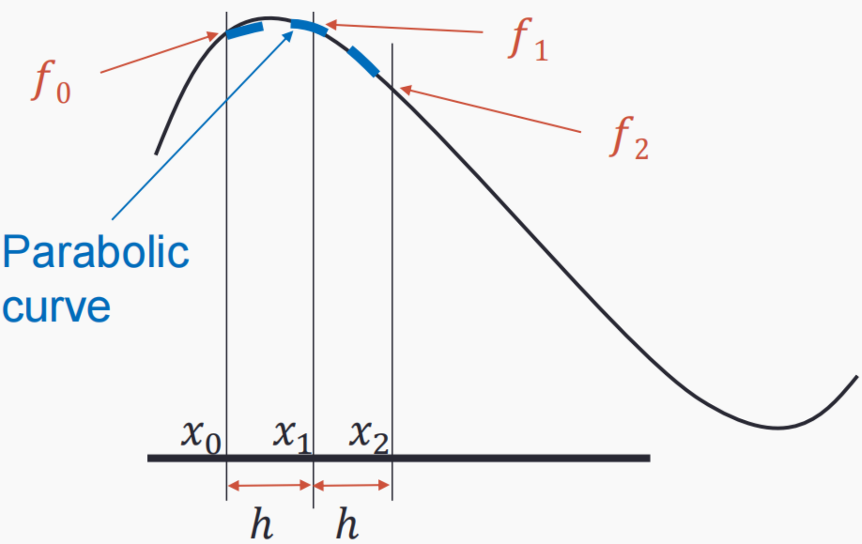
- How accurate is it?
- Alternative:

```
h = 0.05;  x = 0:h:2;  y = 4*x.^3;
trapezoid = (y(1:end-1)+y(2:end))/2;
s = h*sum(trapezoid)
```

方法三：Simpson’s Rule

Second-order Rule: $\frac{1}{3}$ Simpson’s

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(f_0 + 4f_1 + f_2)$$



Simpson’s Rule

- Example:

$$A = \int_0^2 4x^3\,dx = x^4|_0^2 = (2)^4 - (0)^4 = 16$$

```
h = 0.05;  x = 0:h:2;  y = 4*x.^3;
s = h/3*(y(1)+2*sum(y(3:2:end-2))+...
    4*sum(y(2:2:end))+y(end))
```

求解数值积分的函数

quad

```
quad(f,a,b,tol,trace)
```

📖 f:　　被积函数
a,b：　积分区间[a,b]
tol：　计算精度默认为0.001
trace：非零时画出积分图形

guadl内联数值积分函数

> 示例：先建立一个函数文件ex.m：
> function ex=ex(x)
> ex=exp(-x.^2);
> 命令窗口输入：>>I=quad('ex',0,1)
> 或I=quad (@ex,0,1)
> 再或者创建函数时，将函数设为内联函数

integral

- 基于全局自适应积分方法

I=integral(filename,a,b)

其中，I是计算得到的积分；filename是被积函数；a和b分别是定积分的下限和上限，积分限可以为无穷大。

```
integral(filename,a,b)
```

二重积分数值求解

```
 I=dblquad(f,a,b,c,d,tol,trace)
```

review function handle(@)

```
%pointer exzample
xy_plot=plot(@sin,0.0:0.01:2*pi)
```

integral()

- Example: $\int_0^2 \frac{1}{x^3-2x-5}\,dx$

```
y = @(x) 1./(x.^3-2*x-5);
integral(y,0,2)
```

二重积分和三重积分

Double and Triple Integrals

- Example $f(x,y) = \int_0^{\pi} \int_{\pi}^{2\pi} (y \cdot \sin(x) + x \cdot \cos(y))\,dx\,dy$

```
f = @(x,y) y.*sin(x)+x.*cos(y);
integral2(f,pi,2*pi,0,pi)
```

- Example: $f(x,y) = \int_{-1}^{1} \int_0^1 \int_0^{\pi} (y \cdot \sin(x) + z \cdot \cos(y))\,dx\,dy\,dz$

```
f = @(x,y,z) y.*sin(x)+z.*cos(y);
integral3(f,0,pi,0,1,-1,1)
```

📖 mean( )：平均值指算术平均值，即项数之和除以项数。

median( ) ：如果数据个数为奇数，则取值为大小位于中间的元素；如果数据个数为偶数，则取中间两个元素的平均值。

prod( ) 求积函数

cumsum( )：累加和函数

cumprod( )：累乘积函数

std(X)：计算向量X的标准差

std(A)：计算矩阵A的各列的标准差

std(A,flag,dim)：当flag=0时，按s1公式计算样本标准方差；当flag=1时，按s2公式计算样本标准方差；默认flag=0，dim=1

corrcoef(A)：返回由矩阵A所形成的一个相关系数矩阵，其中，第i行第j列的元素表示原矩阵A中第i列第j列的相关系数。

corrcoef(X,Y)：这里X和Y是向量，它们与corrcoef([X,Y])的作用一样，用于求X、Y向量之间的相关系数

sort(X)：对向量X按升序排列

[Y,I]=sort(A,dim,mode)，其中dim指明对A的列还是行进行排序，mode指明按升序还是降序排序，若取ascend则按升序，若取descend则按降序，默认为升序，输出参数中，Y是排序后的矩阵，而I记录Y中的元素在A中的位置。

fminbnd()：用来求局部最小值　x=fminbnd('function name',局部区间或估计值)

如果符号表达式是一个有理分式或可以展开为有理分式，可利用numden函数来提取符号表达式中的分子或分母。其一般调用格式为：

[n,d]=numden(s)

该函数提取符号表达式s的分子和分母，分别将它们存放在n与d中。

collect(f) 对f合并同类项，f是符号表达式或符号矩阵。

collect(f,v) 对f按变量v合并同类项，f是符号表达式或符号矩阵。

expand(f) 对f进行展开，f是符号表达式或符号矩阵。

factor(f) 对f分解因式，f是符号表达式或符号矩阵。

transpose(s) 返回s矩阵的转置矩阵。

det(s) 返回s矩阵的行列式值。

colspace(s) 返回s矩阵列空间的基。

利用函数sym可以将数值表达式变换成它的符号表达式。

函数eval可以将符号表达式变换成数值表达式。

6种关系运算符：<、<=、>、>=、==、~=。
对应的6个函数：lt()、le()、gt()、ge()、eq()、ne()。

3种逻辑运算符：&（与）、|（或）和~（非）。
4个逻辑运算函数：and(a,b)、or(a,b)、not(a)和xor(a,b)。

在进行符号对象的运算前，可用assume函数对符号对象设置值域，函数调用格式为： assume(condition)

assume(expr,set)
第一种格式指定变量满足条件condition，第二种格式指定表达式expr属于集合set。

limit函数的调用格式为：
limit(f,x,a)
f： 函数
x：变量
a：逼近值