Аналитическое и численное дифференцирование

<table>
<tr><td>дифференцирование функций, заданных символьными выражениями</td><td>численное дифференцирование</td></tr>
<tr><td>Y=diff(f,order) – производная заданного порядка order функции f, представленная аналитически, от символьного аргумента (одного или нескольких), аргументы требуется декларировать (в классе Sym(bolic)):
syms x y;
перед конструированием функции f в соответствии с синтаксисом ML.</td><td>y=diff(f)/h, g=diff(y)/h,…
f – вектор или матрица, координаты которых являются функциями вектора аргумента x=x0:h:xend постоянным шагом h;
y – вектор, который получен на основе определения первой производной $f' = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$ и численного её представления в виде вектора $y = \left\{\frac{f(x_i+h)-f(x_i)}{h}\right\}_{i=1}^{n}$, n= length(x)= length(f).
<u>Заметим,</u> что точность производной зависит от h и length(y)=n-1, a diff(f) является приращением функции f</td></tr>
<tr><td>В <u>младших версиях ML,</u> содержащих символьное ядро, функция f может задаваться строкой и процедура дифференцирования аналогична, т.е.
Y=diff(f,order)
или строка конвертируется в объект класса Sym:
fsym=sym(f)</td><td>y=diff(f,order,dim)/h^oder
f – вектор или матрица, координаты которых являются функциями вектора аргумента с равноотстоящими узлами (x=x0:h:xend) ;
order – порядок производной; dim – размерность, вдоль которой ведется поиск производной (dim=1, выбирается вектор-столбец, изменяются строки; dim=2, выбирается вектор-строка, изменяются столбцы);
y – вектор или матрица, которые представляют численный аналог производной f порядка order</td></tr>
</table>

Аналитическое и численное интегрирование

<table>
<tr><td>интегрирование int функций, заданных символьными выражениями:</td><td>методы численного интегрирования (f(x) на отрезке [a,b])</td></tr>
<tr><td>а) Y=int(f) – получение в аналитическом виде неопределенного интеграла от функции, заданной аналитически символьным выражением, по умолчанию f интегрируется по первому аргументу, который определяется системой arg= symvar(f,1), заметим, что args=symvar(f) определяет все аргументы символьной функции. Аргументы требуется декларировать (в классе Sym(bolic)) при конструировании выражения функции, например:
syms x y; f=sin(x)*exp(cos(y))

б) Y=int(f,a,b) или Y=int(f,[a b]) – значение определенного интеграла от символьной функции f одной переменной, заданной на отрезке [a,b], если f – функция нескольких переменных, то результат – символьная функция от от 2-йб 3-й и т.д. переменных

в) Y=int(int(f,a,b),args(2),c,d) – значение определенного интеграла от символьной функции двух f(args(1),f(args(2))) (или</td><td>Пусть $X=\{X_i\}$ – разбиение [a,b] узлами такое, что $a=X_1, X_2, \dots, X_n=b$ шагом $h= X_{i+1} - X_i$

а) Метод трапеций: численный аналог интеграла
$I=\sum_{i=1}^{n-1} h(f(X_{i+1}) + f(X_i))/2$
интеграл вычисляется функцией MatLab:
I = trapz(X,Y) – трапеций;
здесь X – вектор узлов, Y – вектор значений функции f в узлах $\{X_i\}$.

б) универсальная функция интегрирования integral:
Y=integral(f,xmin,xmax,’RelTol’, … relative_eps_value, ’AbsTol’, absolute_eps_value) – численное вычисление интеграла, f – функция аноним (handle function) или процедура-функция, опция задания относительной (relative) погрешности ’RelTol’, если задаёте relative_eps_value равным нулю, то вычисление интеграла ориентируется лишь на величину absolute_eps_value – абсолютной погрешности, которая указывается как значение ’AbsTol’, и всё наоборот, при нулевом absolute_eps_value.</td></tr>
</table>

нескольких переменных) заданной на [a,b]× [c,d] аналитически. Здесь g=int(f,a,b) определенный интеграл по первой переменной, а int(g,args(2),c,d) от второй: args(2). *<u>Замечание</u>: пределы интегрирования принадлежат классу Double.*	в) **Y= integral2(f,xmin,xmax, ymin,ymax,…)** – двойной интеграл; **f=f(x,y)** г) **Y= integral 3(f,xmin,xmax, ymin,ymax, zmin,zmax)** – тройной интеграл; **f=f(x,y,z)**; синтаксис integral2,integral3 аналогичен integral *детально, см. Help.*

<u>Замечание</u>: **Метод Симпсона: quad** – устаревшая функция, не рекомендуют в ML v.20, и является численным аналогом интеграла от f(x) на отрезке [a,b], вычисляется в соответствии с квадратурами Симпсона: $I=\sum_{i=1}^{n-1} \frac{h}{6}\left(f(X_i)+4f\left(\frac{X_i+X_{i+1}}{2}\right)+f(X_{i+1})\right),$ а средствами MatLab реализуется процедурой: **I = quad(Func,a,b,tolerance)** (метод Симпсона), здесь **Fun** – функция аноним (handle function) или процедура-функция, Y= **Fun**(X), X – вектор узлов, **a,b** – границы отрезка, **tolerance** – необходимая точность, по умолчанию – 1.0e-6.

Пример 1. Численного дифференцирования

```
clear
h=0.1; x=0:h:2*pi; n=length(x)
% h – определяет качество численного дифференцирования
y=sin(x).*exp(cos(x))
yprime=diff(y,1)/h;
yprime2=diff(y,2)/h^2;    % 1й способ
fprime2compare=diff(yprime,1)/h % 2й способ
sgtitle('Derivatives')
subplot(1,2,1)
m=n-2 % !!! длина вектора при каждом дифференцировании уменьшается на 1
plot(x(1:m),yprime2,'ro',x(1:m),fprime2compare,'b-')
title('y''''')      % y'
fprime3=diff(y,3)/h^3;
prime3compare=diff(yprime2,1)/h
subplot(1,2,2)
m=n-3 !!! длина вектора при каждом дифференцировании уменьшается на 1
plot(x(1:m),fprime3,'ro',x(1:m),prime3compare,'b-')
title('y''''''')   % y''
```

Пример 2. Символьного (аналитического) дифференцирования

```
%% symbolic diff
figure
syms x  % декларирование символьной переменной
f=sin(x)*exp(cos(x))
y=diff(f,3)
fplot(y,[0,2*pi]), legend('y''''''')
```

Пример 3. Символьного (аналитического) интегрирования

```
%% symbolic int
figure
syms x y z
f=x*y*z
%arg=symvar(f,n) % в arg выбираются n первых переменных выражения
arg=symvar(f,3) %
y=int(int(f,arg(1),1,2),arg(3),2,3) % интегрирование по x и z
```