第 7 章 MATLAB 数值积分与数值微分

Lecturer: 白煌

杭州师范大学
信息科学与技术学院

2022.12.9

本章要点

- MATLAB 数值积分
- MATLAB 数值微分
- MATLAB 离散傅里叶变换

目录

7.1.1 数值积分基本原理

数值积分是将整个积分区间 [a,b] 分成 n 个子区间 $[x_i, x_{i+1}]$，$i=1,2,\cdots,n$，其中 $x_1=a$，$x_{n+1}=b$。求定积分问题分解为求和问题：

$$S=\int_a^b f(x)dx=\sum_{i=1}^{n}\int_{x_i}^{x_{i+1}} f(x)dx$$

7.1.2 数值积分的实现方法

1． 自适应积分法

MATLAB 提供了基于全局自适应积分算法的 integral 函数来求定积分。函数的调用格式为：

```
I=integral(@fname,a,b)
```

其中，I 是计算得到的积分；fname 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。

7.1.2 数值积分的实现方法

例 7-1：求 $I = \int_0^\infty e^{-x^2} (\ln x)^2 dx$。

7.1.2 数值积分的实现方法

例 7-1：求 $I=\int_0^\infty e^{-x^2}(\ln x)^2 dx$。

```
>> fun=@(x)exp(-x.^2).*log(x).^2;     % 用匿名函数定义被积函数
>> q=integral(fun,0,Inf)              % 求定积分
```

7.1.2 数值积分的实现方法

2. 变步长辛普生法

基于变步长辛普生法，MATLAB 给出了 quad 函数和 quadl 函数来求定积分。函数的调用格式为:

[I,n]=quad(@fname,a,b,tol,trace)

[I,n]=quadl(@fname,a,b,tol,trace)

其中，fname 是被积函数名。a 和 b 分别是定积分的下限和上限。tol 用来控制积分精度，默认时取 tol=10^{-6}。trace 控制是否展现积分过程，若取非 0 则展现积分过程，取 0 则不展现，默认时取 trace=0。返回参数 I 即定积分值，n 为被积函数的调用次数。

7.1.2 数值积分的实现方法

例 7-2：设 $f(x) = e^{-0.5x}\sin(x+\frac{\pi}{6})$，求 $S = \int_0^{3\pi} f(x)dx$。

7.1.2 数值积分的实现方法

例 7-2：设 $f(x)=e^{-0.5x}\sin(x+\frac{\pi}{6})$，求 $S=\int_0^{3\pi}f(x)dx$。

（1）建立被积函数文件 fesin.m

```
function f=fesin(x)
f=exp(-0.5*x).*sin(x+pi/6);
```

7.1.2 数值积分的实现方法

例 7-2：设 $f(x)=e^{-0.5x}\sin(x+\frac{\pi}{6})$，求 $S=\int_0^{3\pi}f(x)dx$。

（1）建立被积函数文件 fesin.m

```
function f=fesin(x)
f=exp(-0.5*x).*sin(x+pi/6);
```

（2）调用数值积分函数 quad 求定积分

```
>> [S,n]=quad(@fesin,0,3*pi)
```

7.1.2 数值积分的实现方法

例 7-2：设 $f(x) = e^{-0.5x}\sin(x+\frac{\pi}{6})$，求 $S=\int_0^{3\pi} f(x)dx$。

（1）建立被积函数文件 fesin.m

```
function  f=fesin(x)
f=exp(-0.5*x).*sin(x+pi/6);
```

（2）调用数值积分函数 quad 求定积分

```
>>  [S,n]=quad(@fesin,0,3*pi)
```

也可不建立关于被积函数的函数文件，而使用匿名函数求解：

```
>>  f=@(x)exp(-0.5*x).*sin(x+pi/6);   % 用匿名函数定义被积函数
>>  [S,n]=quad(f,0,3*pi)               % 注意函数名不加 @ 号
```

7.1.2 数值积分的实现方法

例 7-3：分别用 quad 函数和 quadl 函数求 $\int_1^{2.5} e^{-x}dx$ 的近似值，并在相同的积分精度下，比较函数的调用次数。

7.1.2 数值积分的实现方法

例 7-3：分别用 quad 函数和 quadl 函数求 $\int_1^{2.5} e^{-x}dx$ 的近似值，并在相同的积分精度下，比较函数的调用次数。

（1）调用函数 quad 求定积分

```
>> format long
>> fx=@(x)exp(-x);
>> [I,n]=quad(fx,1,2.5,1e-10)
```

7.1.2 数值积分的实现方法

例 7-3：分别用 quad 函数和 quadl 函数求 $\int_1^{2.5} e^{-x}dx$ 的近似值，并在相同的积分精度下，比较函数的调用次数。

（1）调用函数 quad 求定积分

```
>> format long
>> fx=@(x)exp(-x);
>> [I,n]=quad(fx,1,2.5,1e-10)
```

（2）调用函数 quadl 求定积分

```
>> format long
>> fx=@(x)exp(-x);
>> [I,n]=quadl(fx,1,2.5,1e-10)
>> format short
```

7.1.2 数值积分的实现方法

3. 高斯-克朗罗德法

MATLAB 提供了基于自适应高斯-克朗罗德法的 quadgk 函数来求振荡函数的定积分。该函数的调用格式为:

[I,err]=quadgk(@fname,a,b)

其中，err 返回近似误差范围，其他参数的含义和用法与 quad 函数相同。积分上下限可以是 -Inf 或 Inf，也可以是复数。如果积分上下限是复数，则 quadgk 在复平面上求积分。

7.1.2 数值积分的实现方法

例 7-4: 求 $\int_0^{\pi} \frac{x\sin x}{1+\cos^2 x}dx$。

7.1.2 数值积分的实现方法

例 7-4：求 $\int_0^\pi \frac{x\sin x}{1+\cos^2 x}dx$。

（1）建立被积函数文件 fsx.m

```
function  f=fsx(x)
f=x.*sin(x)./(1+cos(x).*cos(x));
```

7.1.2 数值积分的实现方法

例 7-4：求 $\int_0^\pi \frac{x\sin x}{1+\cos^2 x}dx$。

（1）建立被积函数文件 fsx.m

```
function f=fsx(x)
f=x.*sin(x)./(1+cos(x).*cos(x));
```

（2）调用函数 quadgk 求定积分

```
>> I=quadgk(@fsx,0,pi)
```

7.1.2 数值积分的实现方法

4. 梯形积分法

在科学实验和工程应用中，函数关系往往是不知道的，只有实验测定的一组样本点和样本值，人们是无法使用 quad 等函数计算其定积分的。在 MATLAB 中，对由表格形式定义的函数关系的求定积分问题用梯形积分函数 trapz。该函数调用格式为:

I=trapz(X,Y)

其中，等长向量 X，Y 定义函数关系 Y=f(X)。

7.1.2 数值积分的实现方法

例 7-5：用 trapz 函数计算定积分 $\int_1^{2.5} e^{-x} dx$。

7.1.2 数值积分的实现方法

例 7-5：用 trapz 函数计算定积分 $\int_1^{2.5} e^{-x}dx$。

```
>> X=1:0.01:2.5;
>> Y=exp(-X);        % 生成函数关系数据向量
>> trapz(X,Y)
```

7.1.3 多重定积分的数值求解

MATLAB 提供的 dblquad 函数用于求二重积分的数值解，triplequad 函数用于求三重积分的数值解。函数的调用格式为:

```
dblquad(@fun,a,b,c,d,tol)
triplequad(@fun,a,b,c,d,e,f,tol)
```

其中，fun 为被积函数的函数文件名，[a,b] 为 x 的积分区域，[c,d] 为 y 的积分区域，[e,f] 为 z 的积分区域，参数 tol 的用法与函数 quad 完全相同。

7.1.3 多重定积分的数值求解

例 7-6：计算二重定积分

$$I=\int_{-1}^{1}\int_{-2}^{2}e^{-x^2/2}\sin(x^2+y)dxdy$$

7.1.3 多重定积分的数值求解

例 7-6：计算二重定积分

$$I=\int_{-1}^{1}\int_{-2}^{2}e^{-x^2/2}\sin(x^2+y)dxdy$$

（1）建立一个函数文件 fxy.m

```
function f=fxy(x,y)
global ki;
ki=ki+1;        % ki 用于统计被积函数的调用次数
f=exp(-x.^2/2).*sin(x.^2+y);
```

7.1.3 多重定积分的数值求解

例 7-6：计算二重定积分

$$I = \int_{-1}^{1}\int_{-2}^{2} e^{-x^2/2}\sin(x^2+y)dxdy$$

（1）建立一个函数文件 fxy.m

```
function f=fxy(x,y)
global ki;
ki=ki+1;          % ki 用于统计被积函数的调用次数
f=exp(-x.^2/2).*sin(x.^2+y);
```

（2）调用 dblquad 函数求解

```
>> global ki;
>> ki=0;
>> I=dblquad(@fxy,-2,2,-1,1)
>> ki
```

7.1.3 多重定积分的数值求解

例 7-6：计算二重定积分

$$I=\int_{-1}^{1}\int_{-2}^{2}e^{-x^2/2}\sin(x^2+y)dxdy$$

（1）建立一个函数文件 fxy.m

```
function f=fxy(x,y)
global ki;
ki=ki+1;          % ki 用于统计被积函数的调用次数
f=exp(-x.^2/2).*sin(x.^2+y);
```

（2）调用 dblquad 函数求解

```
>> global ki;
>> ki=0;
>> I=dblquad(@fxy,-2,2,-1,1)
>> ki
```

如果使用匿名函数：

```
>> f=@(x,y)exp(-x.^2/2).*sin(x.^2+y);
>> I=dblquad(f,-2,2,-1,1)
```

7.1.3 多重定积分的数值求解

例 7-7：计算三重定积分

$$\int_0^1 \int_0^\pi \int_0^\pi 4xze^{-z^2y-x^2}dxdydz$$

7.1.3 多重定积分的数值求解

例 7-7：计算三重定积分

$$\int_0^1\int_0^\pi\int_0^\pi 4xze^{-z^2y-x^2}dxdydz$$

```
>> fxyz=@(x,y,z)4*x.*z.*exp(-z.*z.*y-x.*x);
>> triplequad(fxyz,0,pi,0,pi,0,1,1e-7)
```

7.2 数值微分

在 MATLAB 中，没有直接提供求数值导数的函数，只有计算向前差分的函数 diff，其调用格式为：

- DX=diff(X)：计算向量 X 的向前差分，DX(i)=X(i+1)-X(i)，i=1,2,$\cdots$,n-1。
- 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（默认状态），按列计算差分；dim=2 时，按行计算差分。

7.3 离散傅里叶变换

一维离散傅里叶变换函数，其调用格式与功能为：

- fft(X)：返回向量 X 的离散傅里叶变换。设 X 的长度（即元素个数）为 N，若 N 为 2 的幂次，则为以 2 为基数的快速傅里叶变换，否则为运算速度很慢的非 2 幂次的算法。对于矩阵 X，fft(X) 应用于矩阵的每一列。
- fft(X,N)：计算 N 点离散傅里叶变换。它限定向量的长度为 N，若 X 的长度小于 N，则不足部分补上零；若大于 N，则删去超出 N 的那些元素。对于矩阵 X，它同样应用于矩阵的每一列，只是限定了向量的长度为 N。
- fft(X,[],dim) 或 fft(X,N,dim)：这是对于矩阵而言的函数调用格式，前者的功能与 fft(X) 基本相同，而后者则与 fft(X,N) 基本相同。只是当参数 dim=1 时，该函数作用于 X 的每一列；当 dim=2 时，则作用于 X 的每一行。

相应地，一维离散傅里叶逆变换函数是 ifft。