# 计算物理学作业

艾鑫

2016 年 1 月 13 日

1.(3.10) 用 4 点高斯求积公式编程计算积分

$$I = \int_{1.4}^{2.0} \int_{1.0}^{1.5} \ln(x+2y)\,\mathrm{d}x\,\mathrm{d}y \tag{1}$$

**解**: 将积分区间 $R = \{(x,y)|1.4 \le x \le 2.0,\ 1.0 \le y \le 1.5\}$ 变换到 $R' = \{(u,v)|-1 \le u \le -1,\ -1 \le v \le 1\}$, 即有

$$\begin{cases} x = \frac{1}{2}(b_2+a_2) + \frac{1}{2}(b_2-a_2)u \\ y = \frac{1}{2}(b_1+a_1) + \frac{1}{2}(b_1-a_1)v \end{cases} \tag{2}$$

因此有

$$I = 0.0755 \int_{-1}^{1} \int_{-1}^{1} \ln(0.3u + 0.5v + 4.2)\,\mathrm{d}u\,\mathrm{d}v \tag{3}$$

使用 $n=3$ 的 4 点高斯求积公式, 利用 MATLAB 求解得

$$\begin{aligned} I &= \int_{1.4}^{2.0} \int_{1.0}^{1.5} \ln(x+2y)\,\mathrm{d}x\,\mathrm{d}y \\ &= 0.429554527717559 \end{aligned} \tag{4}$$

MATLAB 代码如下:

```
% 第一题 用四点高斯求积公式求二重积分

node = [0.3399810, 0.8611363];
coef = [0.6521452, 0.3478548];

u = [-node(2),-node(1),node(1),node(2)];
v = u;
A = [coef(2), coef(1), coef(1), coef(2)];
```

```
s = 0;

for i = 1:4
    for j = 1:4
        s = s + A(i)*A(j)*log(0.3*u(i) + 0.5*v(j) + 4.2);
    end
end

I = 0.075 * s;

format long
disp(I)
```

2.(4.7) 应用龙格 -库塔法求初值问题

$$\begin{cases} y'' + 2ty' + t^2 y = e^t, \quad 0 \le t \le 1 \\ y(0) = 1, y'(0) = -1 \end{cases} \tag{5}$$

的数值解, 取步长 $h = 0.1$.

**解**: 令 $y_1 = y, y_2 = y'$, 问题则转化为

$$\begin{cases} y_1' = y_2 \\ y_2' = e^t - 2ty_2 - t^2 y_1, \quad 0 \le t \le 1 \\ y_1(0) = 1,\ y_2(0) = -1 \end{cases} \tag{6}$$

应用龙格 -库塔法求得解如下图所示:

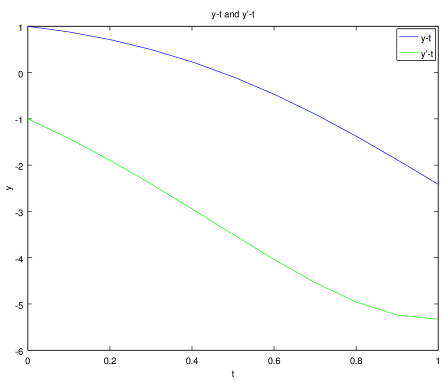



图 1: 龙格 -库塔法求解结果

MATLAB 代码如下:

```
% 第二题 应用龙格—库塔法求初值问题
% 设置初始值
t0 = 0;
h = 0.1;

y1_0 = 1;
y2_0 = -1;

function K = fun(t, y)
  f1 = y(2);
  %f2 = exp(t) - 2*t*y(2) - t^2*y(1);
  f2 = exp(2*t)*sin(t) - 2*y(1) + 2*y(2);
  K = [f1,f2];
endfunction

t = 0:h:1;
n = size(t,2);
Y = zeros(n,2);
Y(1,1) = y1_0;
Y(1,2) = y2_0;

for i = 1:n-1
  K1 = fun(t(i), Y(i,:));
  K2 = fun(t(i) + h/2, Y(i,:) + (h/2)*K1);
  K3 = fun(t(i) + h/2, Y(i,:) + (h/2)*K2);
  K4 = fun(t(i) + h, Y(i,:) + h*K3);
  Y(i+1,:) = Y(i,:) + (K1 + 2*K2 + 2*K3 + K4) * (h/6);
end
% 画图
plot(t,Y(:,1)', 'b',t,Y(:,2)', 'g')
xlabel('t')
ylabel('y')
legend('y-t',"y'-t")
title("y-t and y'-t")
```

3.(8.4) 试采用有限差分法编程求解单位圆域中泊松方程在网格节点处的值

$$\begin{cases} \nabla^2 u = -50r^2\sin(2\phi) \\ u(1,\phi) = 0 \end{cases} \tag{7}$$

其中 $u(1,\phi)=0$ 表示半径为 1 的单位圆边界上 $u$ 值为 1. 取 $h=0.1, \omega=1.25, \varepsilon=10^{-5}, M=16$, 并与解析解 $u=\frac{25}{6}r^2(1-r^2)\sin(2\phi)$ 作比较.

**解**: 在极坐标下泊松方程形式为

$$\nabla^2 u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \varphi^2} = -50r^2\sin(2\phi) \tag{8}$$

用一组等角线和等距离同心圆分割区域, 将半径 $N-1$ 等分, 圆周 $M-1$ 等分, 即有

$$\begin{cases} \Delta r = h = \dfrac{r_0}{N-1} \\ \Delta\varphi = \dfrac{2\pi}{M-1} \end{cases} \tag{9}$$

用误差为 $\mathrm{O}(\Delta r^2)$ 和 $\mathrm{O}(\Delta\varphi^2)$ 的中心差商代替上式中的微商, 可以得到泊松方程的差分格式为

$$\alpha_0(u_{i,j-1}+u_{i,j+1}) + \alpha_1 u_{i-1,j} + \alpha_2 u_{i+1,j} - 2(1+\alpha_0)u_{ij} = h^2 f(r_i,\varphi_j) \tag{10}$$

其中, $\alpha_0=[\Delta\varphi(i-1)]^{-2}, \alpha_1 = 1-(2i-2)^{-1}, \alpha_2 = 1+(2i-2)^{-1}$.

周期性条件有

$$u(r,\varphi)u(r,\varphi+2\pi) \tag{11}$$

即满足 $u_{i1}=u_{iM}, u_{i2}=u_{i,M+1}$.

方程(10)不适合于圆心 $(i=1)$. 利用直角坐标系下的五点差分格式, 有

$$\nabla^2 u_0 \approx \frac{1}{h^2}(u_1+u_2+u_3+u_4-4u_0) \tag{12}$$

在一般情况下, 可将上式写为

$$\nabla^2 u_0 = \frac{4}{h^2}(\bar{u}-u_0) \tag{13}$$

其中 $\bar{u}$ 是半径为 $h$ 的圆周上各结点的平均值, 即

$$\bar{u} = \frac{1}{M-1}\sum_{j=2}^{M} u_{2j}. \tag{14}$$

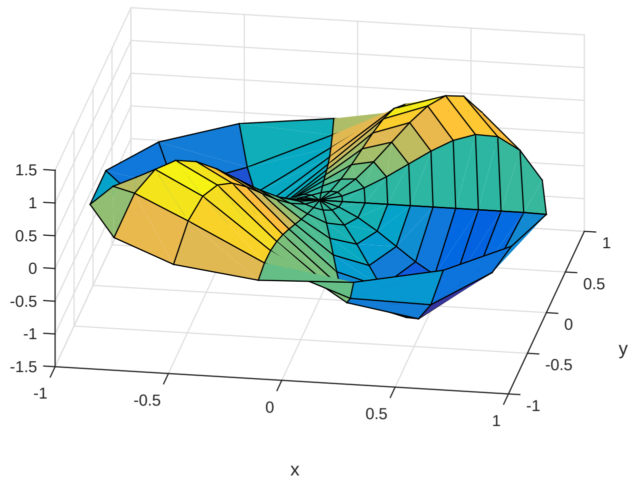
下面对(10)进行简化, 可得

$$u_{ij} = \frac{\alpha_0(u_{i,j-1}+u_{i,j+1}) + \alpha_1 u_{i-1,j} + \alpha_2 u_{i+1,j} - h^2 f(r_i,\varphi_j)}{2+2\alpha_0} \tag{15}$$

引入超松弛因子, 进行超松弛迭代

$$u_{ij}^{k+1} = \omega \times \frac{\alpha_0(u_{i,j-1}^{k+1} + u_{i,j+1}^{k+1}) + \alpha_1 u_{i-1,j}^{k+1} + \alpha_2 u_{i+1,j}^{k+1} - h^2 f(r_i, \varphi_j)}{2 + 2\alpha_0} + (1-\omega) \times u_{ij}^k \tag{16}$$

通过编程求解得结果如下, 与解析解的最大误差为 0.025092.



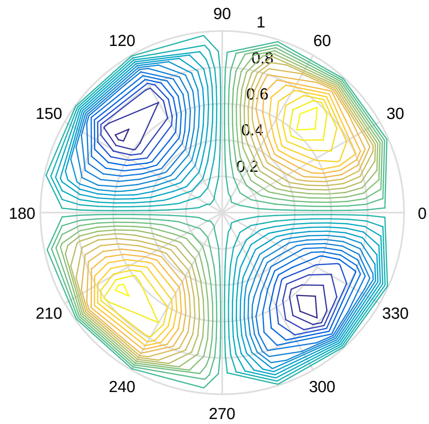




图 2: 有限差分法求解结果

MATLAB 代码如下:

```
% 第三题 用有限差分法求解单位园域中的泊松方程
% 设置初始值
omega = 1.25;
epsilon = 10^(-7);
```

```
r0 = 1.0;
h = 0.1;
deltaR = h;
global N = r0/h+1;
global M = 16;
deltaPhi = 2*pi/(M-1);

R = 0:deltaR:r0;
Phi = 0:deltaPhi:2*pi;
U = zeros(N,M);
% 定义下标i的转换函数，依据周期性条件
function newi = subi(i)
  global N;
  if i==0
    newi = N-1;
  elseif i == N+1
    newi = 2;
  elseif i == N
    newi = 1;
  else
    newi = i;
  end
endfunction
% 定义下标j的转换函数，依据周期性条件
function newj = subj(j)
  global M;
  if j==0
    newj = M-1;
  elseif j == M+1
    newj = 2;
  elseif j == M
    newj = 1;
  else
    newj = j;
  end
endfunction
```

```
function y = fun(r, phi)
  y = -50 * r^2 * sin(2*phi);
endfunction
% 进行迭代
do
  maxDiffU = 0;
  for i = (N-1):-1:1
    if i==1 % 判断是否为圆心
      oldU = U(1,1);
      meanU = mean(U(2,1:(M-1)));
      U0 = omega*(meanU - h^2/4*fun(R(i),Phi(j))) + (1-omega)*oldU;
      U(1,:) = U0;
      diffU = abs(U(1,1)-oldU);
      if diffU > maxDiffU
        maxDiffU = diffU;
      end
    else
      for j = 1:M-1
        oldU = U(i,j);
        alpha_0 = (deltaPhi*(i-1))^(-2);
        alpha_1 = 1 - (2*i-2)^(-1);
        alpha_2 = 1 + (2*i-2)^(-1);
        fenmu = 2*(1+alpha_0);

        temp1 = alpha_0 * ( U(subi(i),subj(j-1)) + \
                            U(subi(i),subj(j+1)) );
        temp2 = alpha_1 * U(subi(i-1),subj(j));
        temp3 = alpha_2 * U(subi(i+1),subj(j));
        U(i,j) = omega*(temp1 + temp2 + temp3 - \
                        h^2*fun(R(i),Phi(j)) )/fenmu + (1-omega)*U(i,j);
        diffU = abs(U(i,j) - oldU);
        if diffU > maxDiffU
          maxDiffU = diffU;
        end
      end
```

```
    U(i,M) = U(i,1);
    end
  end
until maxDiffU < epsilon
% 计算解析解
realU = zeros(size(U));
for i = 1:N
  for j = 1:M
    realU(i,j) = 25/6*(R(i))^2*(1-(R(i))^2)*sin(2*Phi(j));
  end
end
% 输出结果
disp(max(max(abs(U-realU))))
save -ascii U.dat U
save -ascii realU.dat realU
```

4.(11.1) 编程计算例 11.1 中点 3 的电势和 1,2 两点的电量 (见图 11.9).

**解**: 取每个边即为边界元. 首先计算对角线元素 $H_{ii}, G_{ii}$. 由于

$$H_{ii} = -C_i = -\pi \quad (i \in \Gamma\text{光滑点}) \tag{17}$$

$$G_{ii} = \int_{\Gamma_i} u^* \, \mathrm{d}s = \int_{\Gamma_i} \ln r \, \mathrm{d}s = 2\int_0^{s_i/2} \ln r \, \mathrm{d}r = s_i(\ln \frac{s_i}{2} - 1) \tag{18}$$

然后根据下式计算 $H_{ij}$ 非对角元:

$$H_{ij} = \int_{\Gamma_j} q^* \, \mathrm{d}s = \int_{\Gamma_j} \frac{\partial \ln r}{\partial n} \, \mathrm{d}s = \frac{r_d}{|r_d|}\theta_j = \pm\theta_j. \tag{19}$$

根据下式计算 $G_{ij}$ 非对角元:

$$\begin{aligned} G_{ij} &= \int_0^{s_j} \ln r \, \mathrm{d}s = \frac{1}{2}\int_0^{s_j} \ln[r_d^2 + (s-d)^2] \, \mathrm{d}s \\ &= (s_j - d)\ln r_2 + d \ln r_1 - s_j + |r_d|\theta_j \end{aligned} \tag{20}$$

通过计算出来的 $H_{ij}, G_{ij}$ 形成方程

$$\boldsymbol{AX} = \boldsymbol{R}. \tag{21}$$

通过求解方程组求得:

$$q_1 = -2.580, \quad q_2 = 1.72965, \quad u_3 = 0.82492. \tag{22}$$

MATLAB 代码如下:

```
% 第四题 边界元法求解泊松方程的混合边值问题
% 初始化变量
n = 3;
H = zeros(n);
G = zeros(n);
% 计算相关参数
function [d, rd, theta, r1, r2, s] = parafun(i,j)
endPointx = [0,1; 1,0; 0,0];
endPointy = [0,0; 0,1; 1,0];
midPointx = [0.5, 0.5, 0];
midPointy = [0, 0.5, 0.5];
xi = midPointx(i);
yi = midPointy(i);
x1 = endPointx(j,1);
x2 = endPointx(j,2);
y1 = endPointy(j,1);
y2 = endPointy(j,2);
r1Vec = [x1-xi, y1-yi];
r2Vec = [x2-xi, y2-yi];
r21Vec = [x2-x1, y2-y1];
s = sqrt(sum(r21Vec.^2));
r1 = sqrt(sum(r1Vec.^2));
r2 = sqrt(sum(r2Vec.^2));
d = -dot(r1Vec, r21Vec./s);
nVec = [(y2-y1)/s, (x1-x2)/s];
rd = dot(r1Vec, nVec);
theta = acos(dot(r1Vec,r2Vec)/(r1*r2));
endfunction

sVec = [1, sqrt(2), 1];

for i = 1:n
  for j = 1:n
    if i == j % 计算对角元素
      H(i,i) = -pi;
```

```
      G(i,i) = sVec(i)*(log(sVec(i)/2) - 1);
    else % 计算非对角元素
      [d, rd, theta, r1, r2, s] = parafun(i,j);
      H(i,j) = theta;
      G(i,j) = (s-d)*log(r2) + d*log(r1) - s + abs(rd)*theta;
    end
  end
end

% 计算A和R
A = [G(:,1:2),-H (:,3)];
R = [H(:,1:2),-G (:,3)]*[0;1;0];
uq = inv(A)*R;
disp(uq)
```

5.(12.9) 采用蒙特卡罗方法编程求解一下方程.

(1) $e^{-x^3} - \tan x + 800 = 0\,(0 < x < \pi/2)$;

(2) $x + 5e^{-x} - 5 = 0\,(0 < x < 10)$.

**解**: 利用逐步逼近的方法寻找方程 $f(x) = 0$ 的近似根 $x_0$, 使得 $f(x_0) \leq \varepsilon$, 其中 $\varepsilon$ 是事先指定的小量.

首先取一初值 $x_0$, 令 $x_1 = x_0 + b(2R - 1)$, 则有 $x_0 - b < x_1 < x_0 + b$. 即以 $x_0$ 为中心, 以 $b$ 为半径, 随机游走到 $x_1$, 若 $|f(x_1)| < |f(x_0)|$, 就再以 $x_1$ 为中心继续游走, 直至 $|f(x)| \leq \varepsilon$ 为止, 则 $x$ 为所求根.

具体操作时, 刚开始 $b$ 可取得大一些, 然后逐渐减小, 压缩 $x$ 的游动范围. 为了防止无休止的游动, 可先设置一个较大的整数 $N$, 当游走步数 $m = N$ 时, 若 $|f(x)| \leq \varepsilon$ 仍不满足, 则减小 $b$.

通过 MATLAB 求解得方程 (1) 和方程 (2) 的根分别为:

$$x_1 = 1.569546360148, \quad x_2 = 4.965106989195 \tag{23}$$

MATLAB 代码如下:

```
% 第五题 采用蒙特卡罗法解方程


root1 = 0;
root2 = 0;
```

```
% 定义函数
function y = fun1(x)
  y = exp(-x^3) - tan(x) + 800;
endfunction

function y = fun2(x)
  y = x + 5*exp(-x) - 5;
endfunction

function root = findRoot(x0, b, N, epsilon, fun)
  m = 0;
  x = x0;

  f = feval(fun,x);
  if abs(f) < epsilon
    return
  end

  do
    xtemp = x + b*(2*rand()-1);
    ftemp = feval(fun,xtemp);
    m = m + 1;
    if m == N
      b = b/2;
      m = 0;
    end
    if abs(ftemp) < abs(f)
      f = ftemp;
      x = xtemp;
    end
  until abs(f)<epsilon
  root = x;
endfunction

f = @fun1;
root1 = findRoot(1, 0.2, 20, 10^(-5), f);
```

```
disp(root1)
disp(fun1(root1))

f = @fun2;
root2 = findRoot(5, 0.3, 20, 10^(-5), f);
disp(root2)
disp(fun2(root2))
```