偏微分方程的有限差分方法：介绍

在课程的这部分，我们将把有限差分方法推广到偏微分方程。我们将研究的方程模型是传导-扩散方程：

$$\frac{\partial T}{\partial t}+\vec{v}\cdot\nabla T=k\nabla^2 T \leftarrow T=T(x,y,t)$$

其中，$\vec{v}=\vec{v}(x,y,t)=\underbrace{U(x,y,t)\vec{i}+V(x,y,t)\vec{j}}_{\text{2维速度}}$ 是一个给定的速度场

k=在这节课中是常数（通常不是）

=等价于粘性或热传导

$$\nabla^2 T=\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}$$

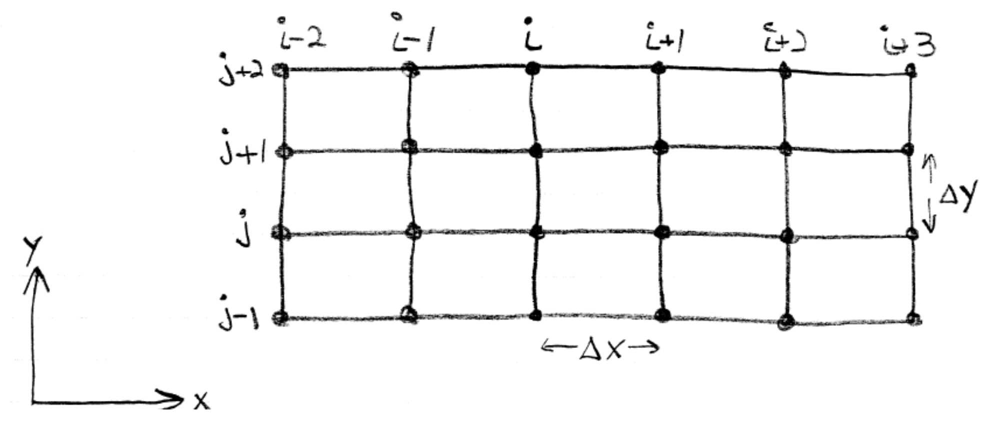
$$\vec{v}\cdot\nabla T=U\frac{\partial T}{\partial x}+V\frac{\partial T}{\partial y}$$

对于许多航空所关心的问题，*k* 是个小量所以无粘性的传导方程近似等价于：

$$\frac{\partial T}{\partial t}+\vec{v}\cdot\nabla T=0$$

有限差分近似及其符号表示

一个偏微分方程的有限差分近似由我们所关心的区间上的网格所构成。在课程的这个部分的大多数时候，我们采取矩形网格结构，其间距为常量。



定义：$T_{ij}^{n} \equiv T(i\Delta x, j\Delta y, n\Delta t)$

现在，我们定义一些常用的有限差分算子：

$$\delta_{2x}T_{ij} \equiv \frac{1}{2\Delta x}(T_{i+1\,j} - T_{i-1\,j}) \leftarrow \text{中心差分}$$

$$\delta_{x}^{-}T_{ij} \equiv \frac{1}{\Delta x}(T_{ij} - T_{i-1\,j}) \leftarrow \text{向后差分}$$

$$\delta_{x}^{+}T_{ij} \equiv \frac{1}{\Delta x}(T_{i+1j} - T_{ij}) \leftarrow \text{向前差分}$$

$$\delta_{x}^{2}T_{ij} \equiv \frac{1}{\Delta x^{2}}(T_{i+1j} - 2T_{ij} + T_{i-1\,j})$$

$$\mu_{x}^{-}T_{ij} \equiv \frac{1}{2}(T_{ij} + T_{i-1\,j})$$

$$\mu_{x}^{+}T_{ij} \equiv \frac{1}{2}(T_{i+1\,j} + T_{ij})$$

$$\mu_{2x}T_{ij} \equiv \frac{1}{2}(T_{i+1\,j} + T_{i-1\,j})$$

同样的，有 δ_{2y}， δ_{y}^{-}， δ_{y}^{+} 等等。

这些算子构成许多偏微分导数有限差分近似。例如：

$$\delta_{2x}T_{ij} = \left.\frac{\partial T}{\partial x}\right|_{ij} + O(\Delta x^{2})$$

为了看清楚，我们采用标准的泰勒级数分析：

$$\delta_{2x}T_{ij} = \frac{1}{2\Delta x}(T_{i+1\,j} + T_{i-1\,j})$$

作替换：

$$T_{i\pm1\,j} = T_{ij} = \Delta x\left.\frac{\partial T}{\partial x}\right|_{ij} + \frac{1}{2}\Delta x^{2}\left.\frac{\partial^{2} T}{\partial x^{2}}\right|_{ij} + \frac{1}{6}\Delta x^{3}\left.\frac{\partial^{3} T}{\partial x^{3}}\right|_{ij} + \frac{1}{24}\Delta x^{4}\left.\frac{\partial^{4} T}{\partial x^{4}}\right|_{ij} + O(\Delta x^{5})$$

可得到：

$$\delta_{2x}T_{ij} = \left.\frac{\partial T}{\partial x}\right|_{ij} + \frac{1}{6}\Delta x^{2}\left.\frac{\partial^{3} T}{\partial x^{3}}\right|_{ij} + O(\Delta x^{4})$$

同样的，截断误差分析可得到：

$$\delta_x^- T_{ij} = \left.\frac{\partial T}{\partial x}\right|_{ij} - \frac{1}{2}\Delta x \left.\frac{\partial^2 T}{\partial x^2}\right|_{ij} + O(\Delta x^2)$$

$$\delta_x^+ T_{ij} = \left.\frac{\partial T}{\partial x}\right|_{ij} + \frac{1}{2}\Delta x \left.\frac{\partial^2 T}{\partial x^2}\right|_{ij} + O(\Delta x^2)$$

$$\delta_x^2 T_{ij} = \left.\frac{\partial^2 T}{\partial x^2}\right|_{ij} + \frac{1}{12}\Delta x^2 \left.\frac{\partial^4 T}{\partial x^4}\right|_{ij} + O(\Delta x^4)$$

我们现在可以合并这些近似来得到一个偏微分方程的有限差分方法：

$$\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} + V\frac{\partial T}{\partial y} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

在空间交点 ij 处做离散：

$$\frac{\partial T_{ij}}{\partial t} + U_{ij}\delta_{2x}T_{ij} + V_{ij}\delta_{2y}T_{ij} = k(\delta_x^2 T_{ij} + \delta_y^2 T_{ij})$$

在这点上，我们可以用任意一种标准常微分方程解法积分这个相连的系统，因为所有 T_{ij}^n 都是随时间向前的。例如，由向前欧拉方法可得到：

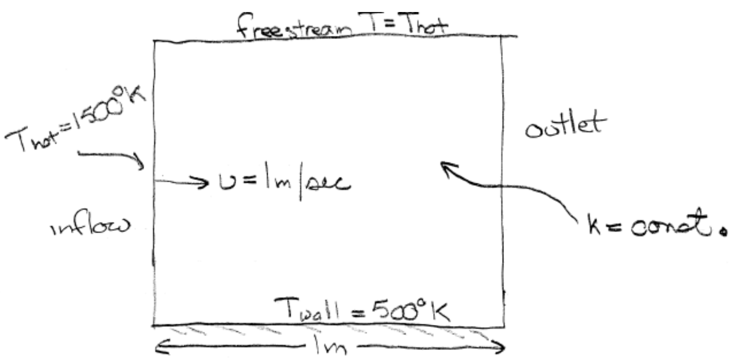
$$\frac{T_{ij}^{n+1} - T_{ij}^n}{\Delta t} + U_{ij}^n\delta_{2x}T_{ij}^n + V_{ij}^n\delta_{2y}T_{ij}^n = k(\delta_x^2 T_{ij}^n + \delta_y^2 T_{ij}^n)$$

对这个算法的一些注释：

 * 局部误差为 0（$\Delta t, \Delta x^2, \Delta y^2$）

 * 在无粘性限制时这个方法就不适用了（就像我们后面讲到的那样），它变得不再稳定。

例子：恒温分布



寻找恒温分布。

$$\Rightarrow U\frac{\partial T}{\partial x} = k(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}) \Rightarrow U_{ij} = k(\delta_x^2 T_{ij} - \delta_y^2 T_{ij})$$

在出口处我们使用抛物边界条件（后面详细介绍），可以推出：

$$U\frac{\partial T}{\partial x} = k\frac{\partial^2 T}{\partial y^2} \quad \text{（在出口处）}$$

这是一个向后差分，在出口处，$\left.\frac{\partial T}{\partial x}\right|_{outlet} = \delta_x^- T$

condif.m 2002 年 2 月 25 日 星期一 10:27:41

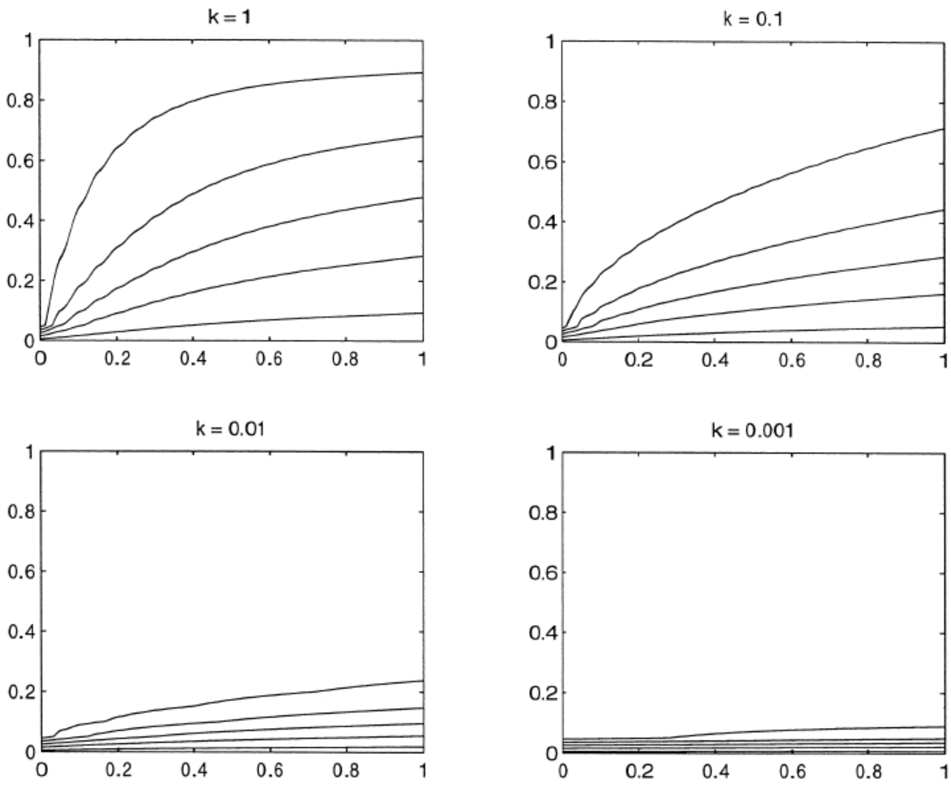
```
Tinlet=1500;
Twall=500;

k=1.0E-3;

L=1.0;
H=1.0;

Nx=20;
Ny=20;

dx=L/Nx;
dy=H/Ny;

for ii=1:Nx+1,
   for jj=1:Ny+1,

      x(ii,jj)=(ii-1)*dx;
      y(ii,jj)=(jj-1)*dy;

      u(ii,jj)=1.0;
      v(ii,jj)=0.0;

   end
end

T=Twall*ones(Nx+1,Ny+1);
```

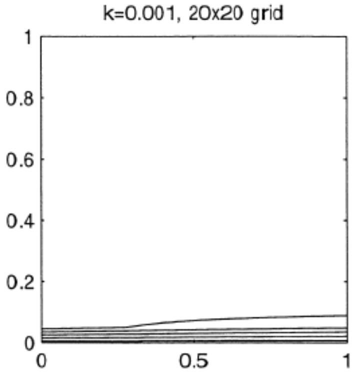
```
% 建立矩阵求解传导-扩散方程

N=(Nx+1)*(Ny+1);
A=zero(N,N);
r=zero(N,1);

%首先，内点离散化

for ii=2:Nx,
    for jj=2:Ny,

        kk0=ii+(jj-1)*(Nx+1);
        kkE=ii+1+(jj-1)*(Nx+1);
        kkW=ii-1+(jj-1)*(Nx+1);
        kkN=ii+(jj)*(Nx+1);
        kkS=ii+(jj-2)*(Nx+1);

        r(kk0)=0.5*u(ii,jj)/dx*(T(ii+1,jj)-T(ii-1,jj)) ...
                    +0.5*v(ii,jj)/dy*(T(ii,jj+1)-T(ii,jj-1)) ...
            -k/dx^2*(T(ii+1,jj)-2*T(ii,jj)+T(ii-1,jj)) ...
                    -k/dy^2*(T(ii,jj+1)-2*T(ii,jj)+T(ii,jj-1));

        A(kk0,kk0)=                                 2*k/dx^2+2*k/dy^2;
        A(kk0,kkE)= 0.5*u(ii,jj)/dx-    k/dx^2                  ;
        A(kk0,kkW)=-0.5*u(ii,jj)/dx-    k/dx^2                  ;
        A(kk0,kkN)= 0.5*v(ii,jj)/dy                  -    k/dy^2;
        A(kk0,kkS)=-0.5*v(ii,jj)/dy                  -    k/dy^2;

    end
end

% 在入口处建立狄利希莱（Dirichlet)条件

ii=1；
for jj=2:Ny+1,
    kk0=ii+(jj-1)*(Nx+1);
    r(kk0)=T(ii,jj)-Tinlet;
    A(kk0,kk0)=1.0;
end

% 在壁底部建立狄利希莱（Dirichlet)条件

jj=1;
for ii=1:Nx+1,
```

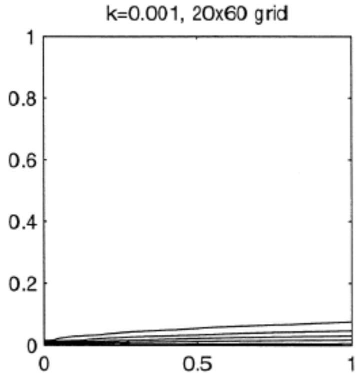
```
    kk0=ii+(jj-1)*(Nx+1);

    r(kk0)=T(ii,jj)-Twall;
    A(kk0,kk0)=1.0;

end

% 在顶部边界建立狄利希莱（Dirichlet)条件

jj=Ny+1;
for ii=1:Nx+1,

    kk0=ii+(jj-1)*(Nx+1);

    r(kk0)=T(ii,jj)-Tinlet;
    A(kk0,kk0)=1.0;

end

%在出口处建立抛物化边界条件

ii=Nx+1；
for jj=2:Ny,

    kk0=ii+(jj-1)*(Nx+1);
    kkW=ii-1+(jj-1)*(Nx+1);
    kkN=ii+(jj)*(Nx+1);
    kkS=ii+(jj-2)*(Nx+1);

    r(kk0)=           u(ii,jj)/dx*(T(ii,jj    )-T(ii-1,jj)) ...
                +0.5*v(ii,jj)/dy*(T(ii,jj+1)-T(ii,jj-1)) ...
          -k/dy^2*(T(ii,jj+1)-2*T(ii,jj)+T(ii,jj-1));

        A(kk0,kk0)=          u(ii,jj)/dx                  +2*k/dy^2;
        A(kk0,kkW)=-        u(ii,jj)/dx                                  ;
        A(kk0,kkN)= 0.5*v(ii,jj)/dy                  -    k/dy^2;
        A(kk0,kkS)=-0.5*v(ii,jj)/dy                  -    k/dy^2;

end

% 解出 dT

dT=-A\r;
```

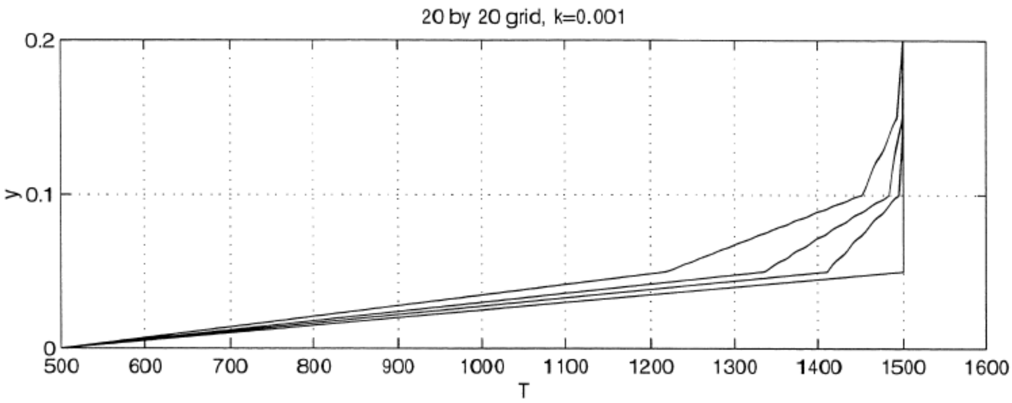
```
%修正 T

for ii=1:Nx+1,
   for jj=1:Ny+1,

      kk0=ii+(jj-1)*(Nx+1);
      T(ii,jj)=T(ii,jj)+dT(kk0);

   end
end

%  画出结果

contour(x,y,T);
```

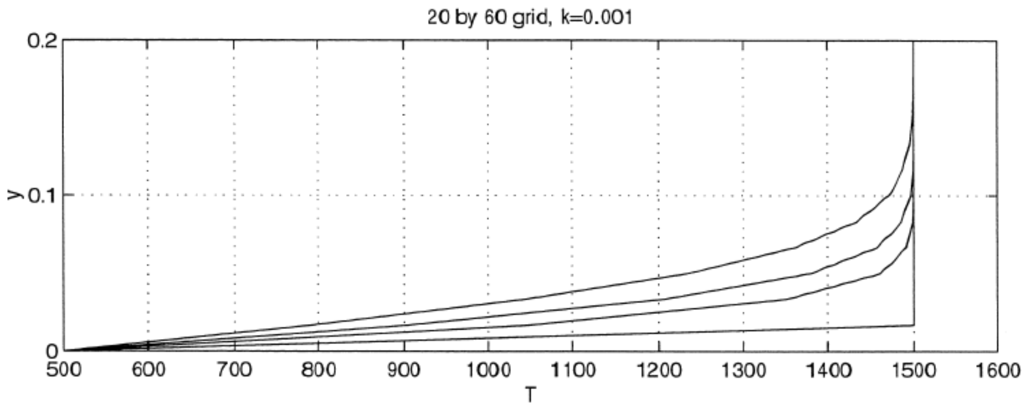
20×20 grid




k=0.001, 20x20 grid
1
0.8
0.6
0.4
0.2
0
0
0.5
1


k=0.001, 20x60 grid
1
0.8
0.6
0.4
0.2
0
0
0.5
1


20 by 20 grid, k=0.001
0.2
y 0.1
0
500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600
T


20 by 60 grid, k=0.001
0.2
y 0.1
0
500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600
T