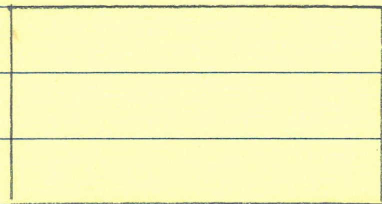
## Application: Unsteady heat equation in two-dimensions

SPRING 2013



Consider a thin rectangular plate with a thermal diffusivity of $c$.

Let $u(t,x,y)$ be the temperature at $(x,y)$ at time $t$. The differential equation that models this situation is

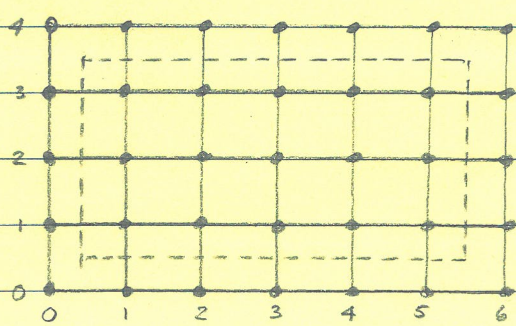
$$\frac{\partial u}{\partial t} = c\nabla^2 u \quad \text{or} \quad \frac{\partial u}{\partial t} = c\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right]$$

There are several ways the boundary information can be supplied — typically either by

1. specifying $u(t,x,y)$ on the boundary (Dirichlet)
2. specifying $\frac{\partial u}{\partial x}$ or $\frac{\partial u}{\partial y}$ on the boundary (Neumann)
3. some combination of these.

We will assume the first case, that $u(t,x,y)$ is given on the boundary.

We can use finite differences by imposing an $(nx+2)\times(ny+2)$ grid on the rectangular domain. This choice gives an $(nx)\times(ny)$ grid on the <u>interior</u> of the domain.




$nx = 5$

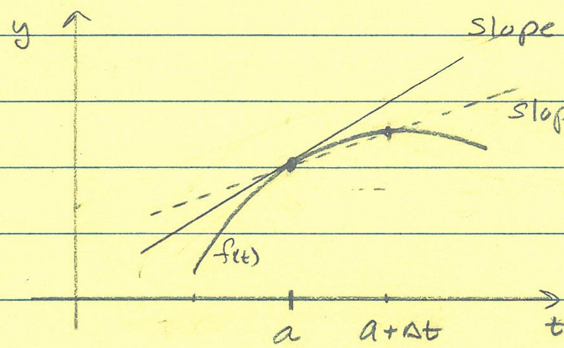
$ny = 3$

The nodes on the boundary have prescribed values — we need to compute the internal node values

## Application: Unsteady heat equation

Solving this problem means determining $u(t, x, y)$ at each of the interior points at a range of time values starting with so initial time.

The simplest approach is to approximate the derivatives at each grid point with formulas similar to the difference quotient in calculus.




slope of this line is $f'(a)$

slope of this line is $\frac{f(a+\Delta t) - f(a)}{\Delta t}$

so

$$f'(a) \approx \frac{f(a+\Delta t) - f(a)}{\Delta t}$$

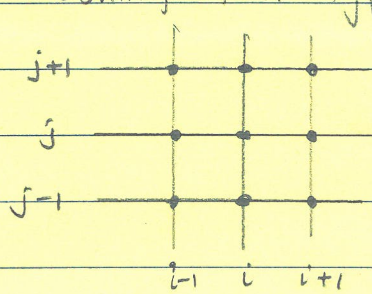
We can approximate $\frac{\partial u}{\partial t}$ with $\frac{u(t+\Delta t, x, y) - u(t, x, y)}{\Delta t}$

It is possible to derive similar formulas for $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(t, x-\Delta x, y) - 2u(t, x, y) + u(t, x+\Delta x, y)}{\Delta x^2}$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u(t, x, y-\Delta y) - 2u(t, x, y) + u(t, x, y+\Delta y)}{\Delta y^2}$$

looking at a typical interior grid node we see




so we can label the node x-coordinates as $x_0, x_1, x_2, \cdots, x_{nx}, x_{nx+1}$ or $x_i$ $i=0, \cdots, nx+1$ and the y-coordinates as $y_j$, $j=0, 1, \cdots, ny+1$

# Application: Unsteady heat equation

Let $u^t_{ij}$ be $u(t, x_i, y_j)$. Then

$$\frac{\partial u}{\partial t} \to \left(u^{t+1}_{ij} - u^t_{ij}\right) / \Delta t$$

$$\frac{\partial^2 u}{\partial x^2} \to \left(u^t_{i-1,j} - 2u^t_{ij} + u^t_{i+1,j}\right) / \Delta x^2$$

$$\frac{\partial^2 u}{\partial y^2} \to \left(u^t_{i,j-1} - 2u^t_{ij} + u^t_{i,j+1}\right) / \Delta y^2$$

The differential equation is thus replaced by a <u>difference equation</u>:

$$\frac{u^{t+1}_{ij} - u^t_{ij}}{\Delta t} = c\left[\frac{u^t_{i-1,j} - 2u^t_{ij} + u^t_{i+1,j}}{\Delta x^2} + \frac{u^t_{i,j-1} - 2u^t_{ij} + u^t_{i,j+1}}{\Delta y^2}\right]$$

Multplying by $\Delta t$ gives

$$u^{t+1}_{ij} - u^t_{ij} = \frac{c\,\Delta t}{\Delta x^2}\left(u^t_{i-1,j} - 2u^t_{ij} + u^t_{i+1,j}\right) + \frac{c\,\Delta t}{\Delta y^2}\left(u^t_{i,j-1} - 2u^t_{ij} + u^t_{i,j+1}\right)$$

If $s_x = c\Delta t / \Delta x^2$ and $s_y = c\Delta t/\Delta y^2$ we have

$$u^{t+1}_{ij} = u^t_{ij} + s_x\left(u^t_{i-1,j} - 2u^t_{ij} + u^t_{i+1,j}\right) + s_y\left(u^t_{i,j-1} - 2u^t_{ij} + u^t_{i,j-1}\right)$$

$$i = 1, \ldots, nx, \quad j = 1, \ldots, ny$$

This is called FTCS (forward time, centered space).

- It is more accurate in the x and y directions ("space") than in t (time)
- need $s_x + s_y \le 1/2$ for method to be <u>stable</u>.

  $\Rightarrow$ as $\Delta x$ and $\Delta y$ get smaller, $\Delta t$ must get smaller as their square! [LOTS OF WORK - SLOW]

## Application: Unsteady heat equation

Example: Domain is 1 unit wide by 0.5 units high

Thermal diffusivity $c = 0.05$

Let $nx = 399$, $ny = 199$ so $\Delta x = \Delta y = \frac{1}{400}$

$$S_x + S_y = \frac{c\,\Delta t}{\Delta x^2} + \frac{c\,\Delta t}{\Delta y^2} = 2\frac{0.05}{\left(\frac{1}{400}\right)^2}\Delta t = 16000\,\Delta t$$

Need $S_x + S_y \le \frac{1}{2}$

so

$$16000\,\Delta t \le \frac{1}{2} \Rightarrow \Delta t \le \frac{1}{32000} = 3.125\times10^{-5}$$

If we want to compute from $t=0$ to $t=1$ we will need at least 32,000 steps!

## Alternating Direction Implicit (ADI) method

One way to improve this is to break our computation of $u_{ij}^{t+1}$ into two steps

$$① \quad \frac{u_{ij}^{t+1/2} - u_{ij}^{t}}{\Delta t/2} = c\,\frac{u_{i-1,j}^{t+1/2} - 2u_{ij}^{t+1/2} + u_{i+1,j}^{t+1/2}}{\Delta x^2} + c\,\frac{u_{i,j-1}^{t} - 2u_{ij}^{t} + u_{ij+1}^{t}}{\Delta y^2}$$

$$② \quad \frac{u_{ij}^{t+1} - u_{ij}^{t+1/2}}{\Delta t/2} = c\,\frac{u_{i-1,j}^{t+1/2} - 2u_{ij}^{t+1/2} + u_{i+1,j}^{t+1/2}}{\Delta x^2} + c\,\frac{u_{i,j-1}^{t+1} - 2u_{ij}^{t+1} + u_{ij+1}^{t+1}}{\Delta y^2}$$

In ① we compute $u_{ij}^{t+1/2}$ values using $u_{ij}^{t}$ values and in ② we compute $u_{ij}^{t+1}$ values using $u_{ij}^{t+1/2}$ values.

Both of these are <u>implicit</u>; we need to solve for multiple values that depend on each other.

# Application: Unsteady heat equation



Let $s_x = \frac{c\Delta t}{\Delta x^2}$ and $s_y = \frac{c\Delta t}{\Delta y^2}$ as before. Then ① and ② can be written as

$$③ \quad -s_x u_{i-1,j}^{t+1/2} + 2(1+s_x)u_{ij}^{t+1/2} - s_x u_{i+1,j}^{t+1/2} = s_y u_{i,j-1}^{t} + 2(1-s_y)u_{ij}^{t} + s_y u_{i,j+1}^{t}$$

$$④ \quad -s_y u_{i,j-1}^{t+1} + 2(1+s_y)u_{ij}^{t+1} - s_y u_{i,j+1}^{t+1} = s_x u_{i-1,j}^{t+1/2} + 2(1-s_x)u_{ij}^{t+1/2} + s_x u_{i+1,j}^{t+1/2}$$

$$\text{for} \quad i = 1, \ldots, nx \qquad j = 1, \ldots, ny$$

Consider $j$ fixed in ③ (i.e. we work on a single row of $u$) then we have a system of equations for $i = 1, 2, \ldots, nx$

$$-s_x u_{0j}^{t+1/2} + 2(1+s_x)u_{1j}^{t+1/2} - s_x u_{2j}^{t+1/2} = s_y u_{1,j-1}^{t} + 2(1-s_y)u_{1j}^{t} + s_y u_{1,j+1}^{t}$$
$$-s_x u_{1j}^{t+1/2} + 2(1+s_x)u_{2j}^{t+1/2} - s_x u_{3j}^{t+1/2} = s_y u_{2,j-1}^{t} + 2(1-s_y)u_{2j}^{t} + s_y u_{2,j+1}^{t}$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$
$$-s_x u_{nx-1,j}^{t+1/2} + 2(1+s_x)u_{nx,j}^{t+1/2} - s_x u_{nx+1,j}^{t+1/2} = s_y u_{nx,j-1}^{t} + 2(1-s_y)u_{nx,j}^{t} + s_y u_{nx,j+1}^{t}$$

This can be written in matrix form as

$$\begin{bmatrix} 2(1+s_x) & -s_x & & & \\ -s_x & 2(1+s_x) & -s_x & & \\ & -s_x & 2(1+s_x) & \ddots & \\ & & \ddots & \ddots & -s_x \\ & & & -s_x & 2(1+s_x) \end{bmatrix} \begin{bmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ \vdots \\ u_{nx,j} \end{bmatrix}^{t+1/2} = \begin{bmatrix} s_y u_{1,j-1}^{t} + 2(1-s_y)u_{1j}^{t} + s_y u_{1,j+1}^{t} + s_x u_{0j}^{t+1/2} \\ s_y u_{2,j-1}^{t} + 2(1-s_y)u_{2j}^{t} + s_y u_{2,j+1}^{t} \\ s_y u_{3,j-1}^{t} + 2(1-s_y)u_{3j}^{t} + s_y u_{3,j+1}^{t} \\ \vdots \\ s_y u_{nx,j-1}^{t} + 2(1-s_y)u_{nx,j}^{t} + s_y u_{nx,j+1}^{t} + s_x u_{nx+1,j}^{t+1/2} \end{bmatrix}$$

where these terms are known since they are boundary values

We need to solve this system for each $j = 1, 2, \dots, ny$ — so there are $ny$ systems to be solved, and that's just half the work ...

Now consider ④ with $i$ fixed (work on a single column of $u$)
This gives rise to the system

$$\begin{bmatrix} 2(1+S_y) & -S_y & & & \\ -S_y & 2(1+S_y) & -S_y & & \\ & -S_y & 2(1+S_y) & \ddots & \\ & & \ddots & \ddots & -S_y \\ & & & -S_y & 2(1+S_y) \end{bmatrix} \begin{bmatrix} u_{i1} \\ u_{i2} \\ u_{i3} \\ \vdots \\ u_{i,ny} \end{bmatrix}^{t+1} = \begin{bmatrix} S_x u_{i-1,1}^{t+1/2} + 2(1+S_x) u_{i1}^{t+1/2} + S_x u_{i+1,1}^{t+1/2} + S_y u_{i0}^{t+1} \\ S_x u_{i-1,2}^{t+1/2} + 2(1+S_x) u_{i2}^{t+1/2} + S_x u_{i+1,2}^{t+1/2} \\ S_x u_{i-1,3}^{t+1/2} + 2(1+S_x) u_{i3}^{t+1/2} + S_x u_{i+1,3}^{t+1/2} \\ \vdots \qquad \vdots \qquad \vdots \\ S_x u_{i-1,ny}^{t+1/2} + 2(1+S_x) u_{i,ny-1}^{t+1/2} + S_x u_{i+1,ny}^{t+1/2} + S_y u_{i,ny+1}^{t+1} \end{bmatrix}$$

As before, $nx$ of these systems must be solved. Thus, for one single update

$$u_{ij}^{t} \rightarrow u_{ij}^{t+1} \qquad i = 1, \dots, nx, \; j = 1, \dots, ny$$

we need to construct and solve $nx + ny$ systems of equations.

Application: Unsteady heat equation

These are a "tridiagonal systems" and are fairly simple to solve. Consider

$$\begin{bmatrix} d_1 & c_1 & & & & \\ a_2 & d_2 & c_2 & & & \\ & a_3 & d_3 & c_3 & & \\ & & a_4 & d_4 & \ddots & \\ & & & \ddots & \ddots & c_{n-1} \\ & & & & a_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}$$

This can be solved using the Thomas algorithm - just regular LU factorization:

```
for i = 2 to n do
    a_i = a_i / d_{i-1}          // multiplier, entry in L
    d_i = d_i - a_i c_{i-1}      // diagonal of U
    b_i = b_i - a_i b_{i-1}
endfor
x_n = b_n / d_n
for i = n-1 downto 1 do
    x_i = (b_i - c_i x_{i+1}) / d_i
endfor
```

Good: $O(n)$ (rather than $O(n^3)$ if matrix was full)

Bad: does not parallelize easily due to dependencies in loops

# Application: Unsteady heat equation

Notice that in the systems arising from ③ and ④ the tridiagonal matrices are both independent of $i$ or $j$ — thus we only need to factor each one once before we start stepping in time.

factor matrices : (presweep)

$xd_1 = 2(1+sx)$
for $i = 2$ to $nx$ do
    $mx_i = sx / xd_{i-1}$
    $xd_i = 2(1+sx) - mx_i\, s_x$
endfor
$yd_1 = 2(1+s_y)$
for $j = 2$ to $ny$ do
    $my_j = sy / yd_{j-1}$
    $yd_j = 2(1+s_y) - my_j\, sy$
endfor

Sweep Rows

for $j = 1$ to $ny$ do
    for $i = 1$ to $nx$ do
        $b_i = s_y(U^t_{i,j-1} + U^t_{i,j+1}) + 2(1-s_y)U^t_{ij}$
    endfor
    $b_1 = b_1 + s_x U^t_{0j}$
    $b_{nx} = b_{nx} + s_x U^t_{nx+1,j}$    these are given boundary values
    for $i = 2$ to $nx$ do
        $b_i = b_i + mx_i\, b_{i-1}$    // Apply factorization to RHS
    endfor

Sweep X (continued)

```
        u^{t+1/2}_{nx,j} = b_nx / xd_nx
        for i = nx-1 down to 1 do
            u^{t+1/2}_{i,j} = (b_i + s_x u^{t+1/2}_{i+1,j}) / xd_i
        endfor
    endfor
```

Sweep Cols

```
    for i = 1 to nx do
        for j = 1 to ny do
            b_j = s_x (u^{t+1/2}_{i-1,j} + u^{t+1/2}_{i+1,j}) + 2(1-s_x) u^{t+1/2}_{ij}
        endfor
        b_1 = b_1 + s_y u^{t+1/2}_{i0}             // use boundary values
        b_ny = b_ny + s_y u^{t+1/2}_{i,ny+1}       //  "        "
        for j = 2 to ny do
            b_j = b_j + my_j b_{j-1}    // apply factorization to RHS
        endfor
        u^{t+1}_{i,ny} = b_ny / yd_ny
        for ny-1 down to 1 do
            u^{t+1}_{i,j} = (b_j + s_y u^{t+1}_{i,j+1}) / yd_j
        endfor
    endfor
```