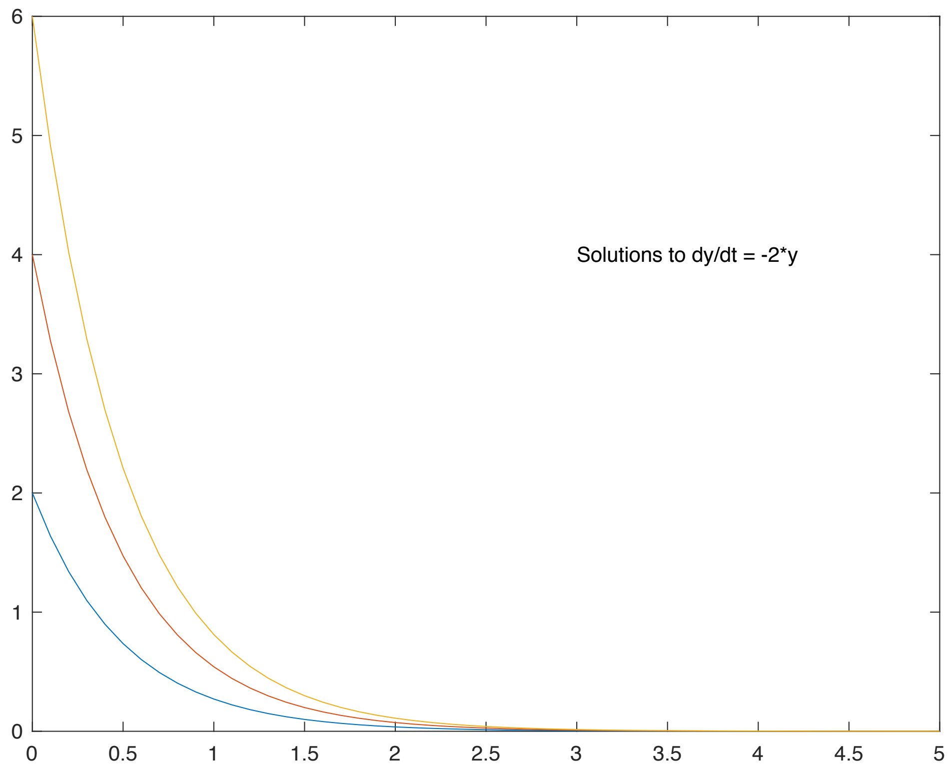
Learning Goals

1. Difference between ODEs and PDEs
2. Classification of PDEs
3. Introduction to finite differences for PDEs

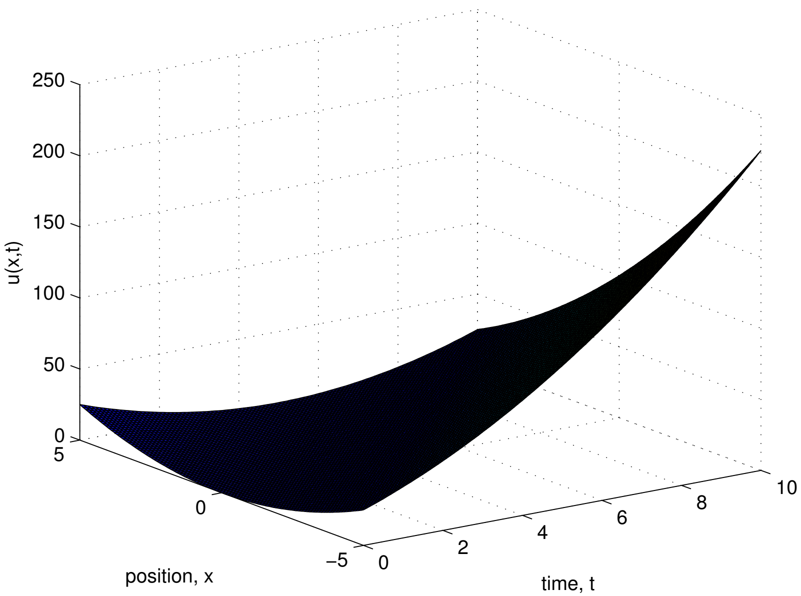
Partial Differential Equations

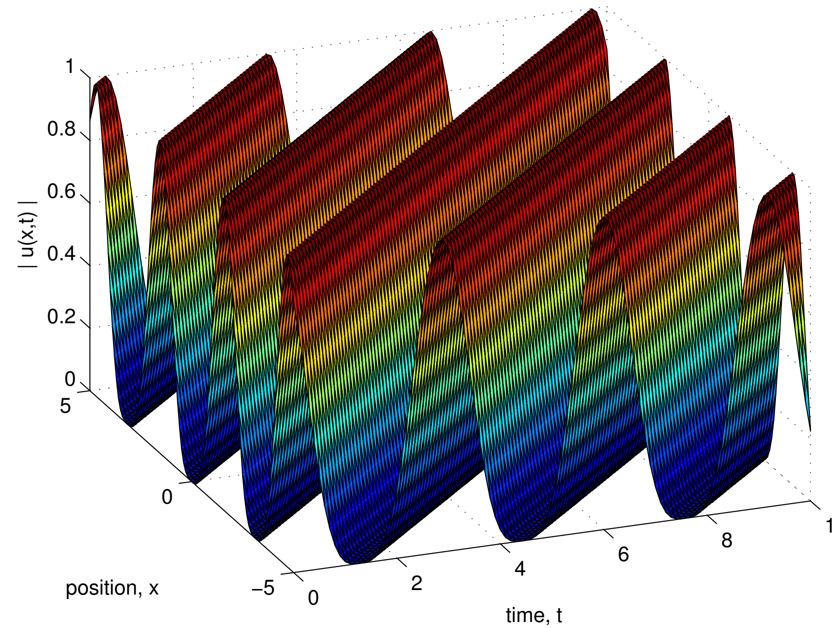
1. Become important in advanced topics in physics
2. Very few are *analytically solvable*, and further, *no one numerical technique exists* that can handle all types of PDEs
3. Most important differences between PDE and ODE.

$$\frac{dy}{dt} = -ay(t) \Rightarrow y(t) = y_o e^{-at}$$



$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$





In this course we will mostly focus on 2^{nd} order, linear *PDEs*. The most general form of this type of PDE is

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial t} + C\frac{\partial^2 u}{\partial t^2} + D\left(x, t, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}\right) = 0$$

where *A,B,C* are functions of *x,t* and *D* is a function of *u,x,t* and the partials of *u.*

1. If $B^2 - 4AC > 0$ the PDE is called *hyperbolic.* An example is the *wave equation.*
2. If $B^2 - 4AC = 0$, the PDE is called *parabolic.* An example is the *diffusion equation*.
3. If $B^2 - 4AC < 0$, the PDE is called *elliptic*. An example of this the *Laplace equation*

While we won't need this classifications, they are important in the theoretical study of these equations

Notation $\quad \dfrac{\partial u}{\partial x} \equiv u_x; \quad \dfrac{\partial^2 u}{\partial x^2} \equiv u_{xx}; \quad \dfrac{\partial^2 u}{\partial x \partial t} \equiv u_{x,t}.$

Do question 1 on the worksheet.

(1) $A = -c^2, B = 0, C = 1, D = 0 \rightarrow B^2 - 4AC = 0 - 4(-c^2)(1) > 0$ so it is *hyperbolic*

PDEs are inherently boundary value problems. They may also involve one or more initial conditions.

Boundary value problems come in two forms, both of which can appear in the same problem.

1. **Dirichelet** Boundary conditions have the *value* of the field (the quantity we are solving for) specified at the boundary.
2. **Neuman** Boundary conditions have the *value of the derivative* of the field at the boundary.

Do question (2) on the worksheet and S T O P

Analytically, about the only way to solve a wide class of *PDEs* is to use separation of variables in which we assume the function $u(x,y,..,t) = X(x)Y(y)\ldots T(t)$, which results in set of *ODEs* which we might be able to solve.

Here we look at numerical approaches only to solve *PDEs*

However, before we do, we need to look a *boundary value problems*

We look at the complications that boundary value problems add to solve equations using the following example:

$$y'' + 4y = 0$$

where the '' indicate spatial derivatives. This *ODE* has the general solutions:

$$y(x) = c_1 \cos(2x) + c_2 \sin(2x); \quad c_1, c_2 \quad \text{constants}$$

Now we are going to use *boundary conditions* rather than initial conditions as our auxiliary equations

Case 1: $y(0) = -2; y(\pi/4) = 10$ When we apply these boundary conditions we get

$$y(0) = -2 = c_1; \quad y(\pi/4) = 10 = c_2$$

$$y(x) = -2\cos(2x) + 10\sin(2x)$$ and things are good

Case 2: $y(0) = -2; y(2\pi) = -2$ When we apply these boundary conditions we get

$$y(0) = -2 = c_1; y(2\pi) = -2 = c_1$$

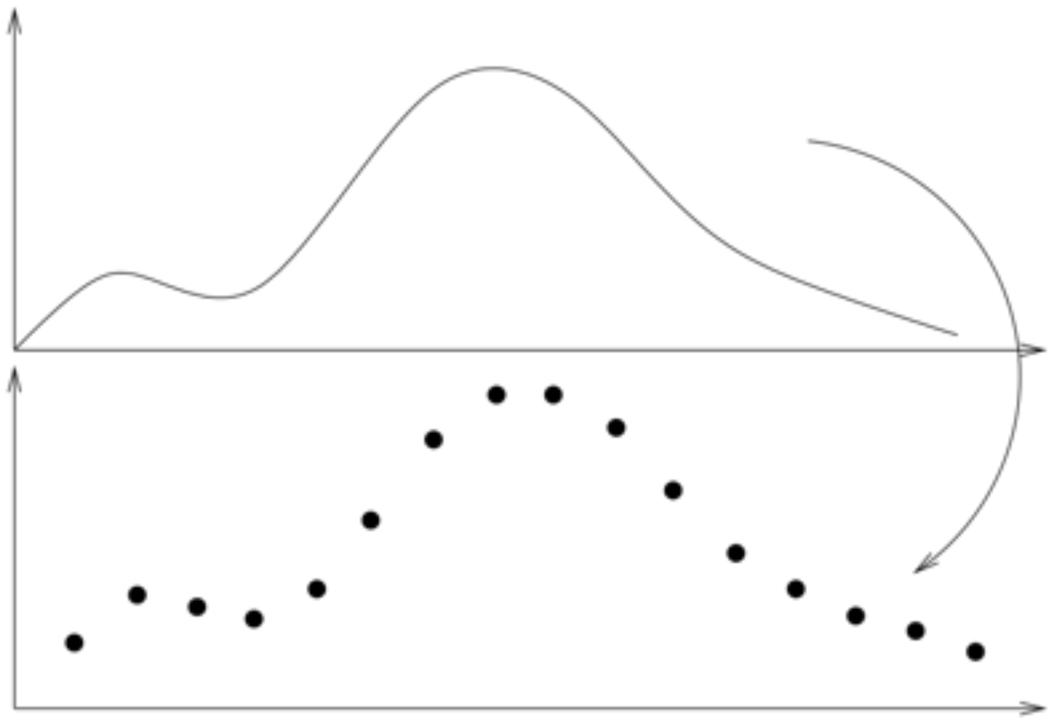
$$y(x) = -2\cos(2x) + c_2\sin(2x)$$ and we've lost uniqueness

Case 3: $y(0) = -2; y(2\pi) = 3$ When we apply these boundary conditions we get

$$y(0) = -2 = c_1; y(2\pi) = 3 = c_1$$ we have two different values for c_1, no solution is for these boundaries even though the system has a general solution

Runge-Kutta methods do not lend themselves as well to boundary value problems because the values at the boundaries never evolve.

Another approach: *finite differences*



In finite differences we seek the solutions to *ODE/PDE* only at *discrete points.* This means we need to replace *derivatives (or partials)* with *differences*

To get the difference equations, we use Taylor expansions

$$y(x+h) = y(x) + hy'(x) + \mathcal{O}(h^2)$$ Forward difference

$$y(x-h) = y(x) - hy'(x) + \mathcal{O}(h^2)$$ Backward difference

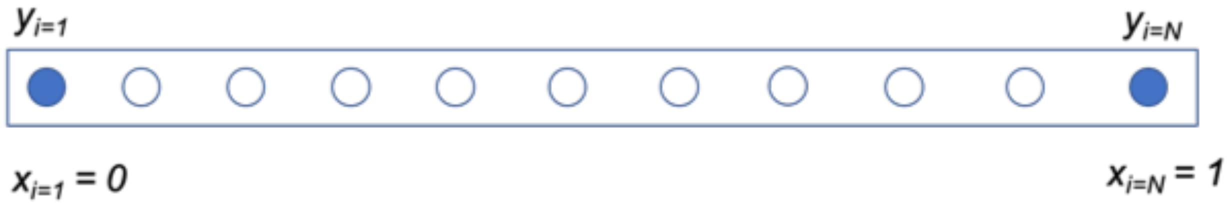
$$\boxed{\begin{aligned} y'(x) &= \frac{y(x+h) - y(x-h)}{2h} + \mathcal{O}(h^2) \\ y''(x) &= \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + \mathcal{O}(h^2). \end{aligned}}$$ Central difference approximation

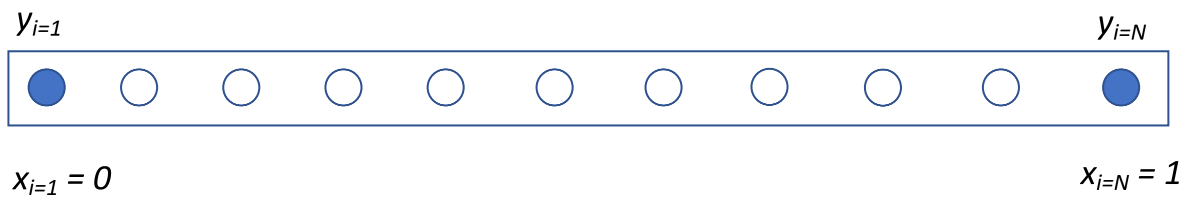
Notation: $y_i \equiv y(x_i); \quad y_{i+1} \equiv y(x_i + h); \quad y_{i-1} \equiv y(x_i - h)$

Do question (3) on the worksheet

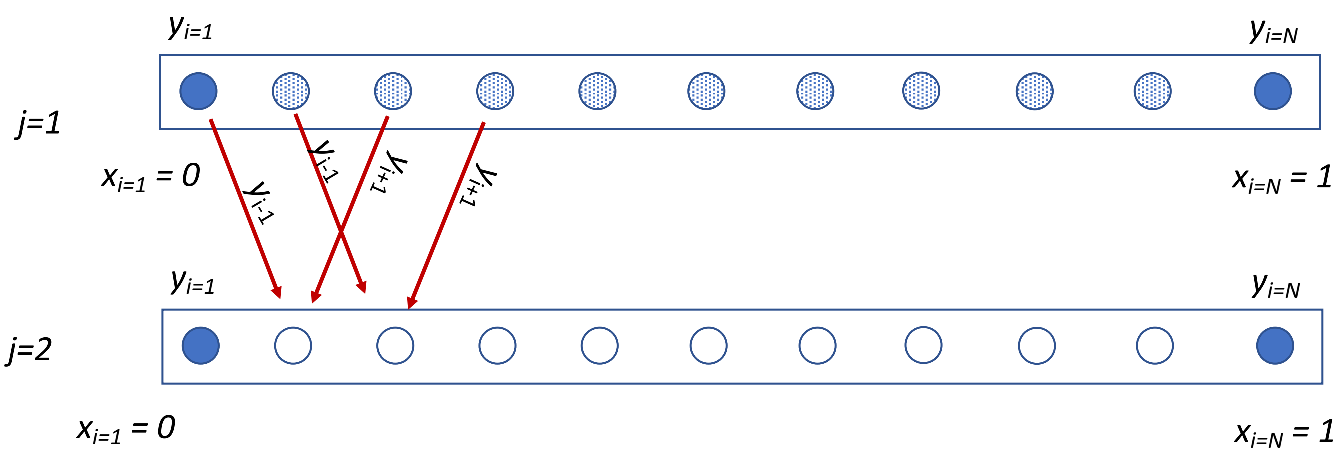
$$\underbrace{\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}}_{y''} - 5\underbrace{\frac{y_{i+1} - y_{i-1}}{2h}}_{y'} + 10y_i = 10x_i$$

$$y_i = \frac{1}{2 - 10h^2}\left[\left(1 - \frac{5h}{2}\right)y_{i+1} + \left(1 + \frac{5h}{2}\right)y_{i-1} - 10h^2 x_i\right].$$





Original problem

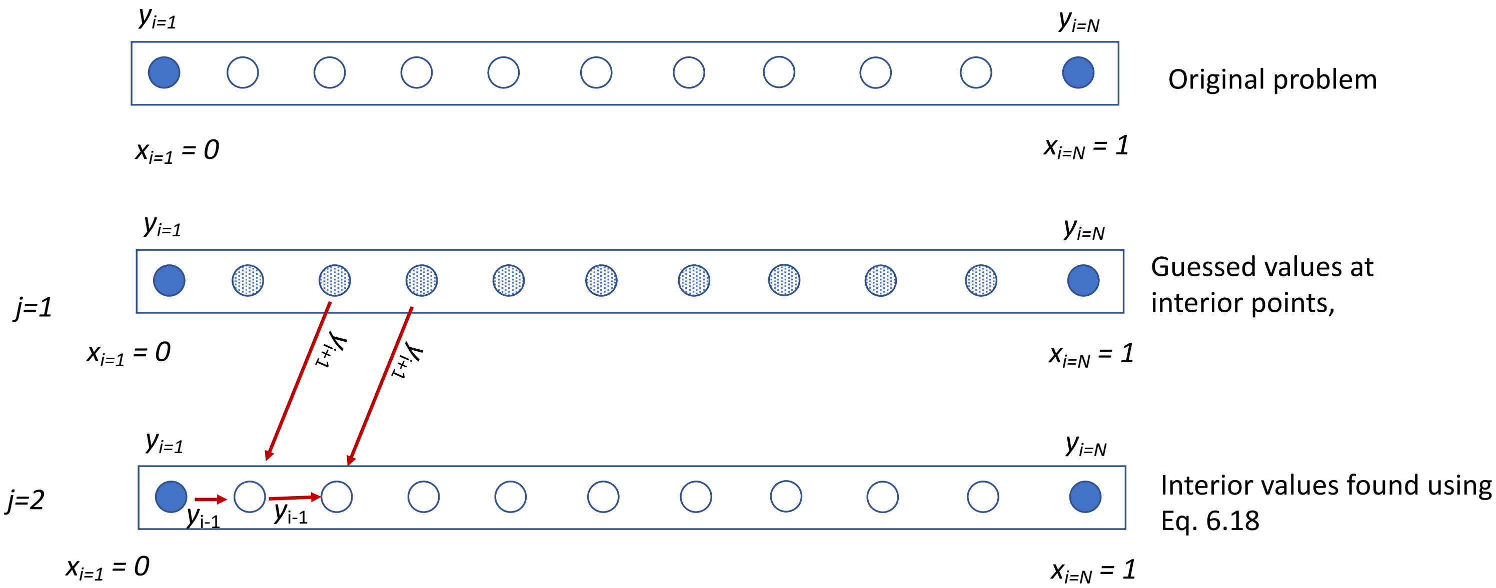


Guessed values at interior points,

Interior values found using Eq. 6.18

$$y_i^j = \frac{1}{2-10h^2}\left[\left(1-\frac{5h}{2}\right) y_{i+1}^{(j-1)} + \left(1+\frac{5h}{2}\right) y_{i-1}^{(j-1)} - 10h^2 x_i\right].$$

Jacobi Scheme—Do question (4) on the worksheet



$$y_i^j = \frac{1}{2 - 10h^2}\left[\left(1 - \frac{5h}{2}\right) y_{i+1}^{(j-1)} + \left(1 + \frac{5h}{2}\right) y_{i-1}^{(j)} - 10h^2 x_i\right].$$

Gauss Seidel algorithm.

Do question (5) on the worksheet

With *ODEs*, we only discretized one independent variable, say space. With *PDEs,* we will often have discretized the spatial variable *and* time.

The same Taylor expansions are used to approximate derivatives in time, but generally the grid step size in time will be *different* than the grid step size in space.

We will begin our study of numerical solutions to *PDEs* by working through the example of the *Laplace equation.* We will use the *steady state heat equation* as the specific physical example of how these PDEs are numerically solved.

The Heat Equation

$$\nabla^2 u(x,y,z,t) = \frac{1}{c^2} u_t, \quad c^2 = \frac{K}{\sigma\rho}$$

where

- u is the temperature at position (x,y) and time t
- K is the thermal conductivity,
- σ is the specific heat,
- ρ is the mass density

In the steady state, $u_t = 0$ so the heat equation becomes

$u_{xx} + u_{yy} = 0$ Note that there is no time

To make the system discrete, we lay out two grids. One in the *x-direction* and one in the *y-direction*. We do this as follows

$$\begin{aligned} x_i &= ih_x, \quad i = 0, 1, \ldots, N_x \\ y_i &= jh_y, \quad j = 0, 1, \ldots, N_y \end{aligned}$$

Using the notation $u_{i,j} = u(x_i, y_j)$ we get

$$\underbrace{\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}}_{u_{xx}} + \underbrace{\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}}_{u_{yy}} = 0.$$

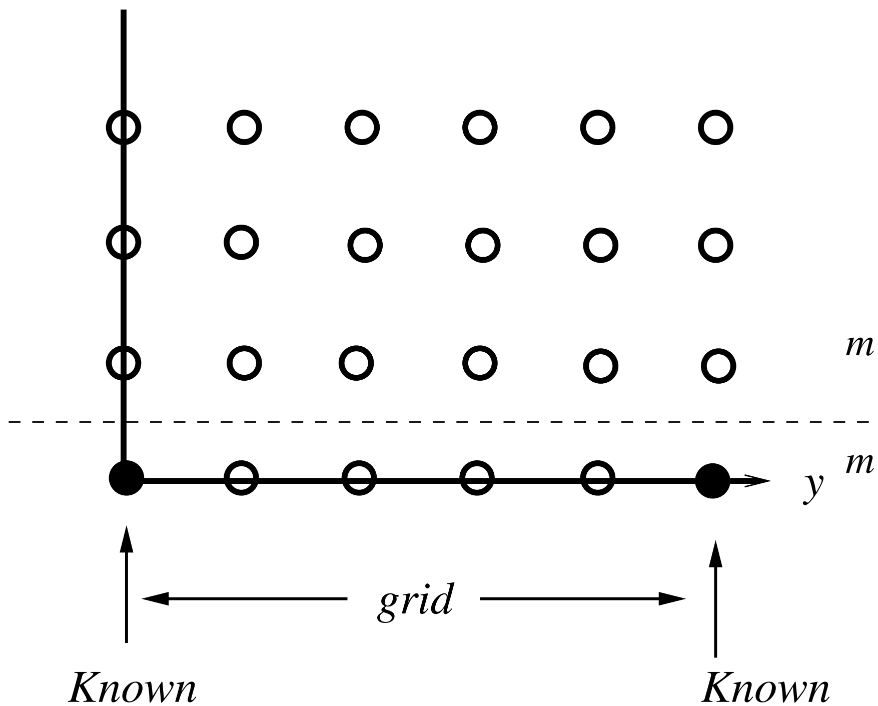
Solving for $u_{i,j}$ we get

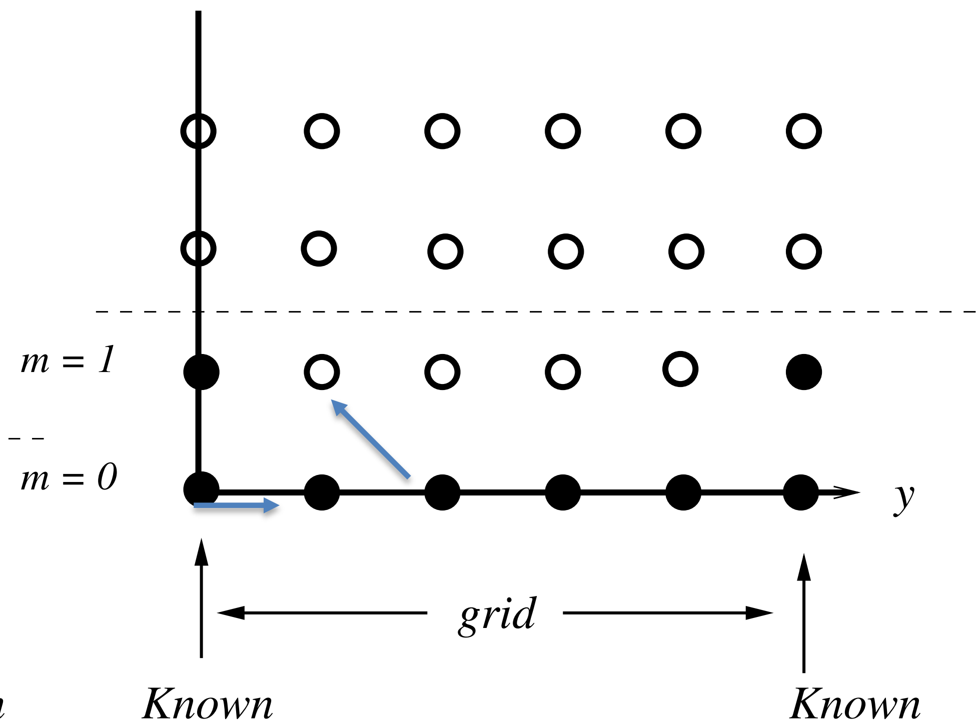
$$u_{i,j} = \frac{h_x^2 h_y^2}{2h_x^2 + 2h_y^2}\left[\frac{u_{i+1,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} + u_{i,j-1}}{h_y^2}\right].$$

This simplifies a bit if we make $h_x = h_y$ in which case we get:

$$u_{i,j} = \frac{1}{4}\left[u_{i+1,\,j} + u_{i-1,\,j} + u_{i,\,j+1} + u_{i,\,j-1}\right]$$

Typically in these kinds of problems all one knows is the boundary conditions. So to solve the heat equation using finite element, we once again resort to an *iterative* process





It turns out that we can speed up the iterative process by using a technique called:
Successive Over Relaxation (SOR)

Here's the idea: We *guess* that converged result is the most recent result *plus* some factor times the *difference* between the *two* most recent results. Calling the solution, $\bar{y}$, we guess

$$\bar{y}_i^{(j)} = y_i^{(j)} + \alpha \left[y_i^{(j)} - y_i^{(j-1)} \right]$$

The idea:

1. The value of the function at each iteration is found using *normal finite differencing*
2. The value of the function is then modified by *SOR*, and it is this value that is used at the next iteration
3. Iteration ends when some *tolerance is met*.
4. The value α is less than 1, and varies from equation to equation

We are ready to solve the *heat equation*.

1. Initialize the problem
2. Apply finite differencing using $u_{i,j} = \frac{1}{4}\left[u_{i+1,\, j} + u_{i-1,\, j} + u_{i,\, j+1} + u_{i,\, j-1}\right]$
3. Adjust $u_{i,j}$ using *SOR*
4. Repeat until tolerance is met

Do questions (6) on the worksheet