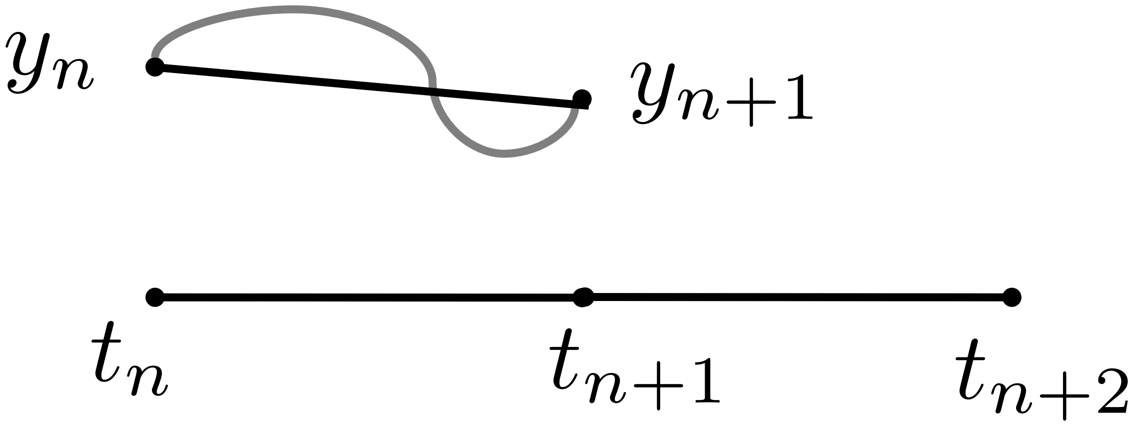
Scientific Computing

Review Parade

Initial Value Problems

$$\frac{d\vec{y}}{dt} = f(t, \vec{y}) \qquad \vec{y}(0) = \vec{y}_0$$

Numerical Solution of IVP

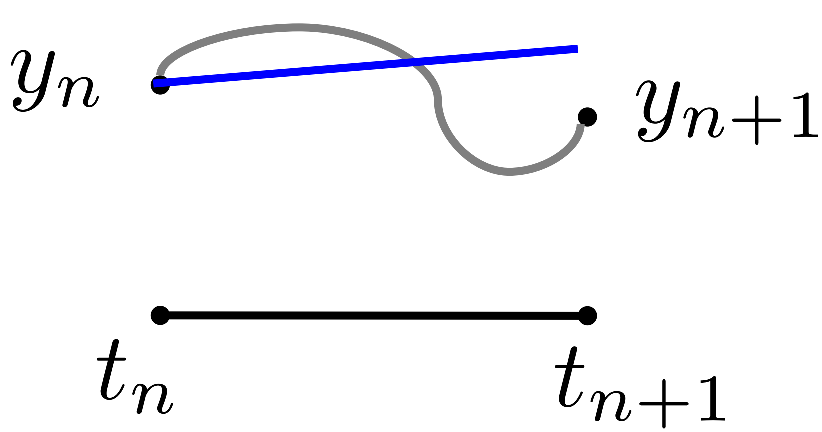


The birth of Num. app. Euler's Method

$$\frac{d\vec{y}}{dt} = f(t, \vec{y})$$

$$lim_{\Delta t \to 0} \frac{\vec{y}_{n+1} - \vec{y}_n}{t_{n+1} - t_n} = f(t_n, \vec{y}_n)$$

$$\boxed{\vec{y}_{n+1} = \vec{y}_n + \Delta t f(t_n, \vec{y}_n)}$$



Accuracy – Taylor series

$$y(t+\Delta t) = y(t) + \Delta t \frac{d\vec{y}}{dt} + \frac{\Delta t^2}{2}\frac{d^2\vec{y}}{dt^2} + \ldots$$

Euler

RK2

RK4

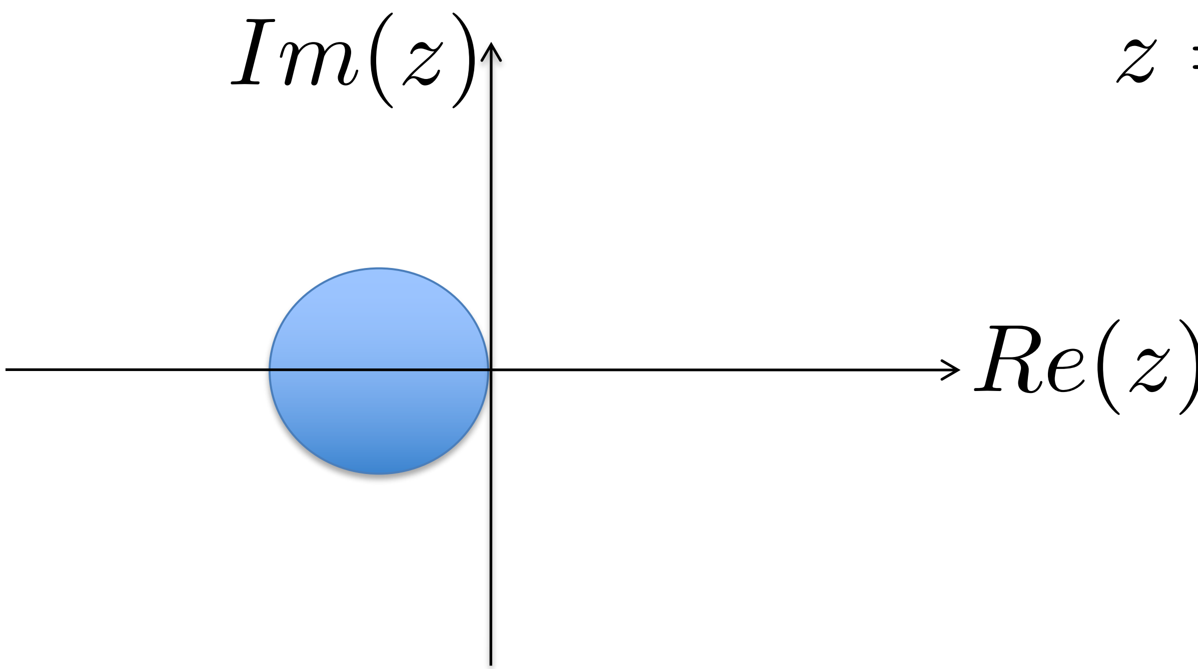
...

Stability

Test Case: $$\frac{dy}{dt} = \lambda y$$

Euler:

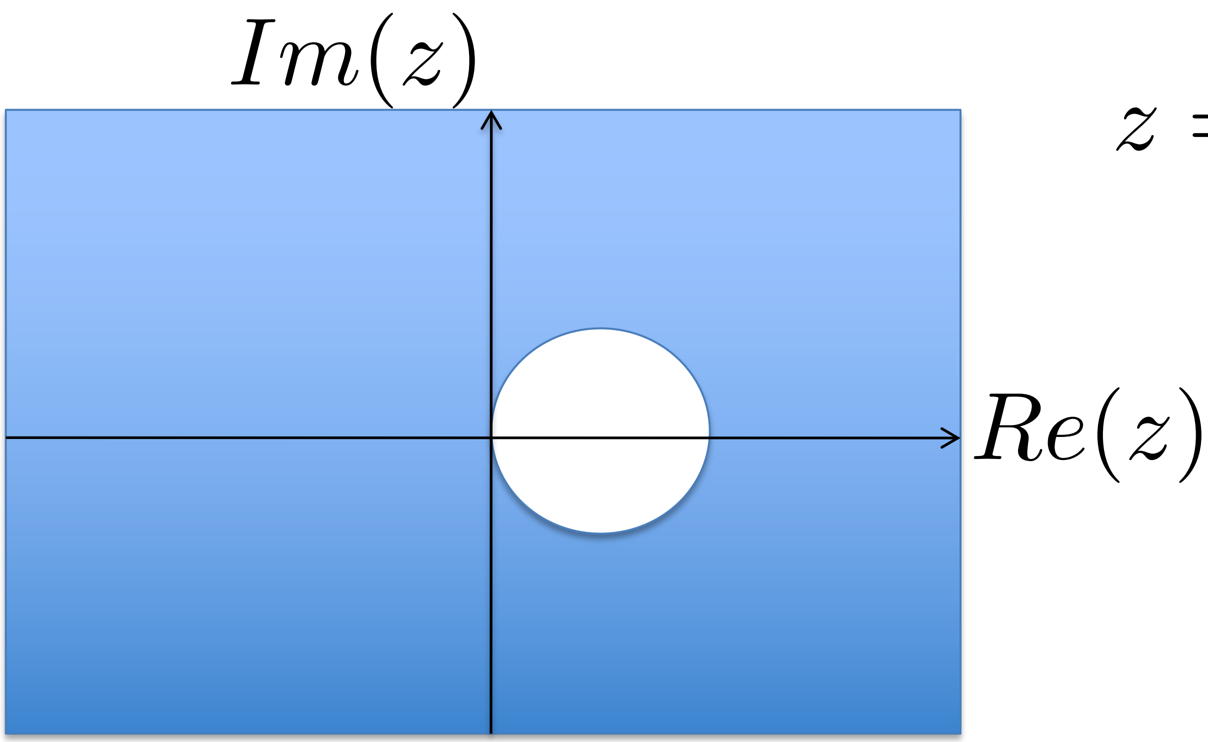
$$z = \Delta t \lambda$$



Implicit Schemes

Backward Euler:

$$\vec{y}_n = \vec{y}_{n+1} - \Delta t f(t_n, \vec{y}_{n+1})$$



$$z = \Delta t \lambda$$

SOLVE IVP Dream Team

LSODA

‘BDF’ ODE15s

‘DOP853’ ODE113

Boundary value problems

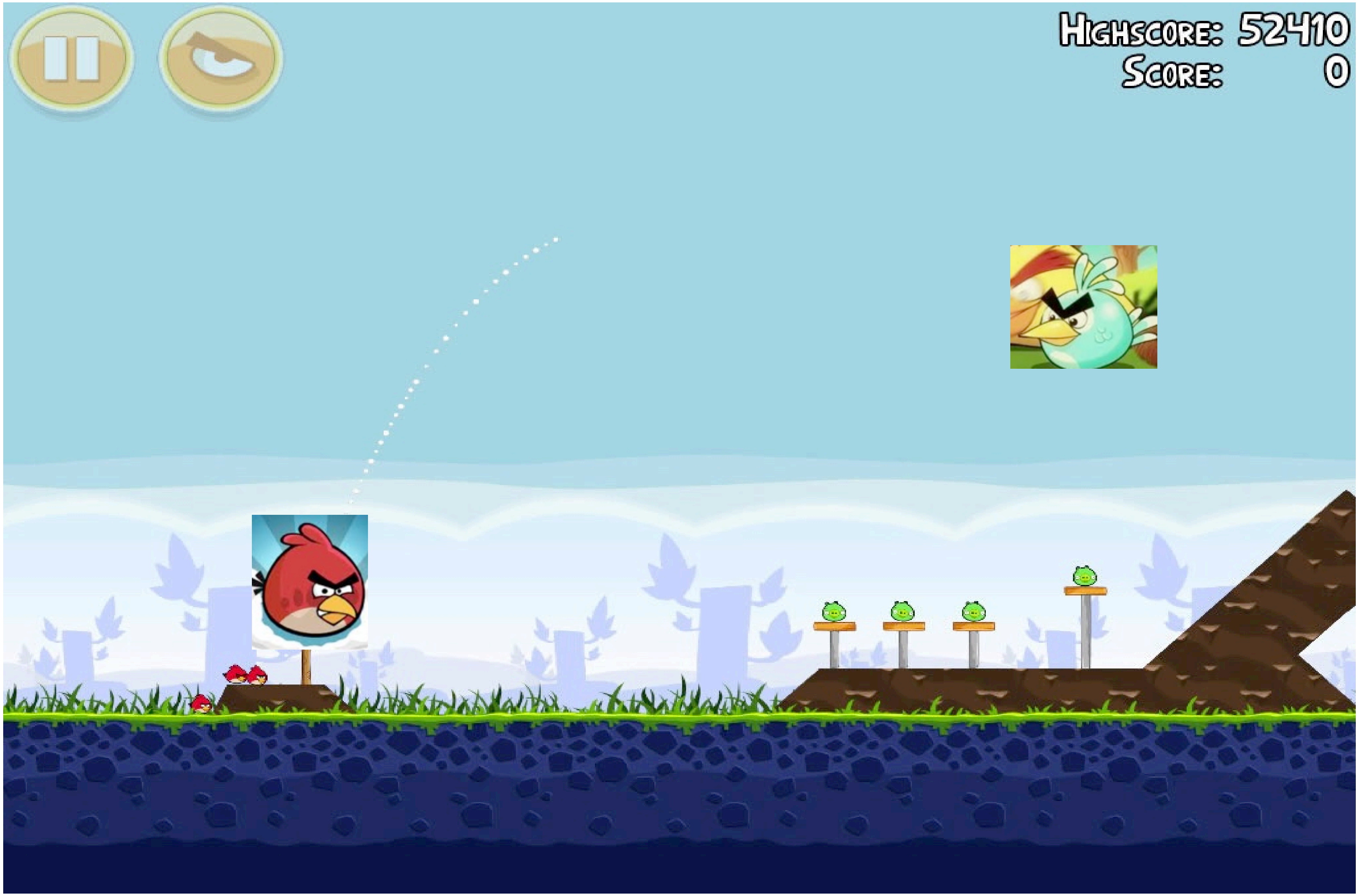
$$\frac{d^2y}{dt^2} = f(t, y, \frac{dy}{dt})$$

Boundary Conditions:

$$y(a) = \alpha$$

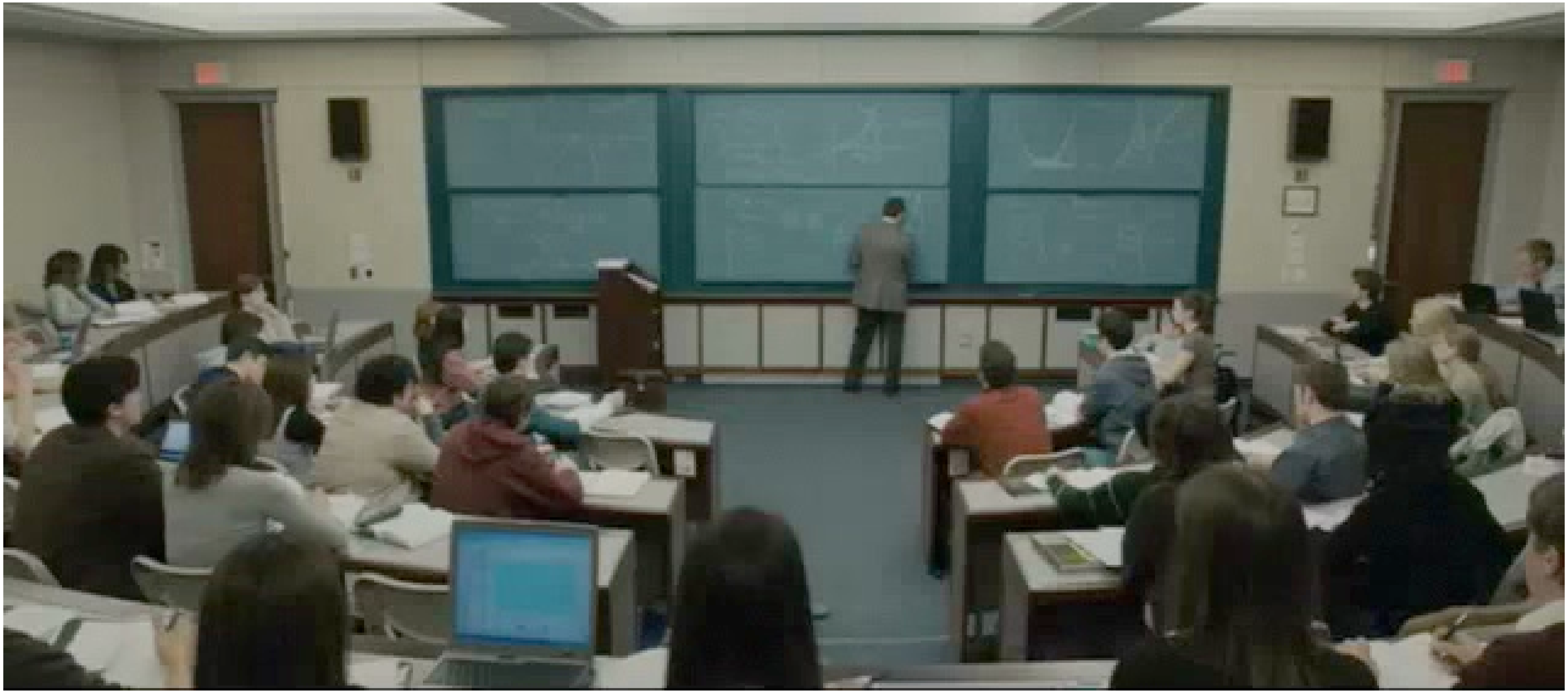
$$y(b) = \beta$$

Shooting



Newton-Raphson Method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



Direct – Linear BVP

$$\frac{d^2y}{dt^2} = p(t)\frac{dy}{dt} + q(t)y + r(t)$$

$$\frac{d^2y(t)}{dt^2} \approx \frac{y(t+\Delta t) - 2y(t) + y(t-\Delta t)}{\Delta t^2}$$

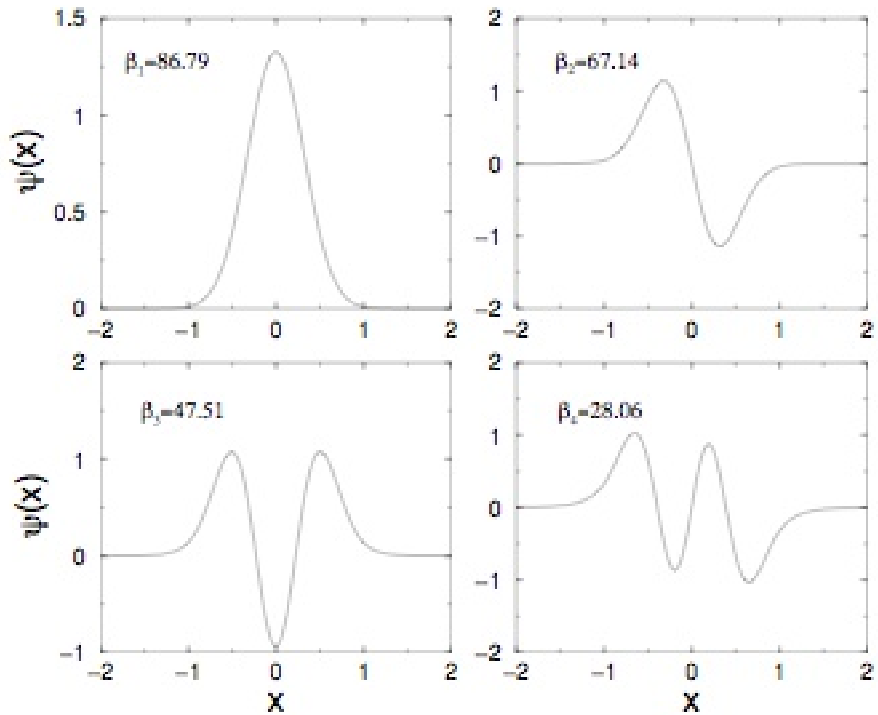
$$\frac{dy(t)}{dt} \approx \frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t}$$

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} y(0) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \\ b \\ \\ \end{bmatrix}$$

Physical Problem

Quantum Harmonic Oscillator:

$$\frac{d^2\psi_n}{dx^2} + n(x)\psi_n = \beta_n\psi_n$$



Linear System of Eqs.

A^{-1}

$O(N^3)$

Gaussian Elim.

$O(N^3)$

$A\vec{x} = \vec{b}$

LbU

$O(N^2)$

Solleve

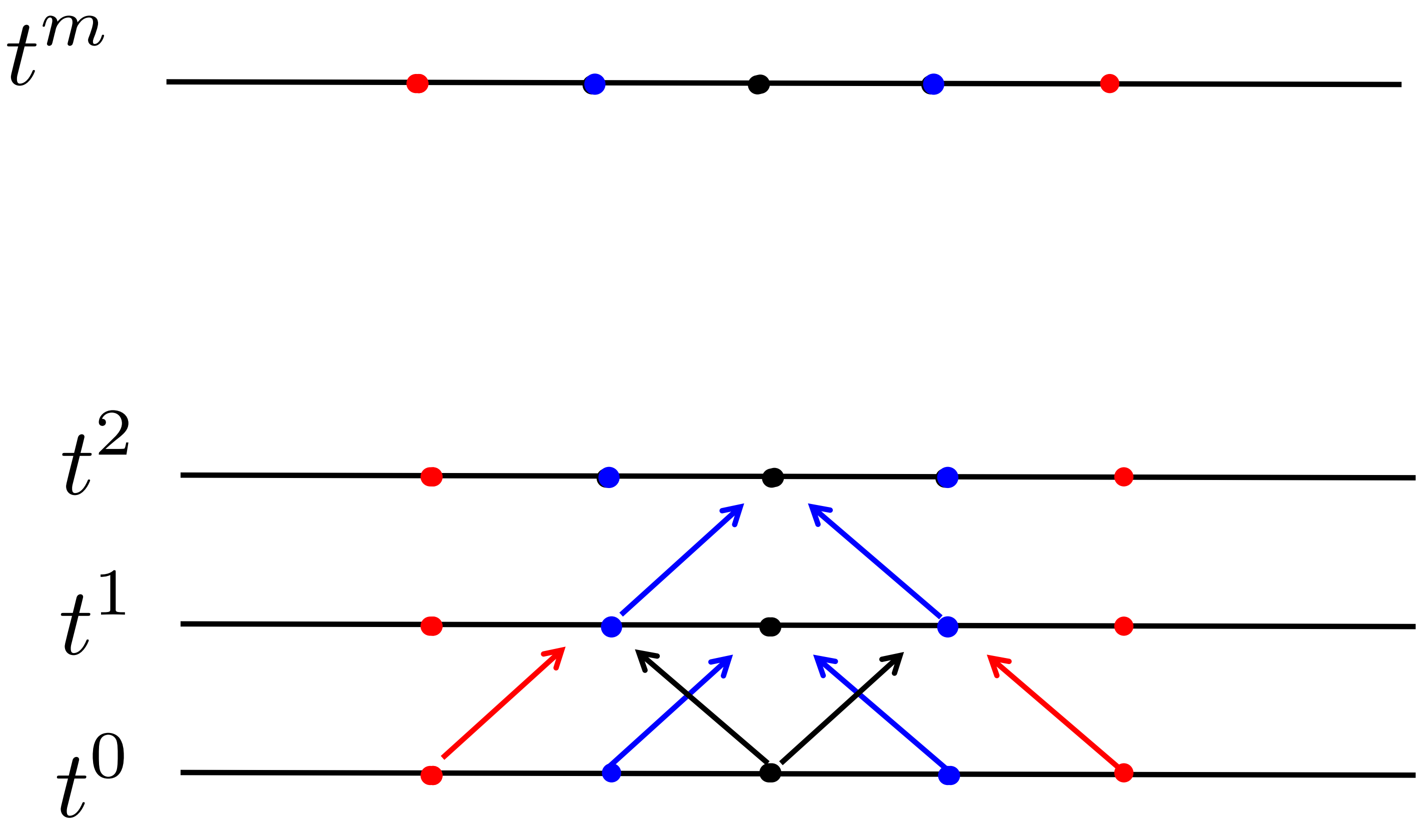
Partial Differential Equations

$$\nabla^2 \psi(x, y) = f(x, y)$$ + b.c.

$$u_t(t, x) = u_x(t, x)$$ + i.c., b.c.

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

Method of Lines



Space and Time

Stability

Von-Neumann:

$$u_n^{(m)} = g^m \exp(i\zeta\Delta x)$$

$$\lim_{m\to\infty} |g|^m =?$$

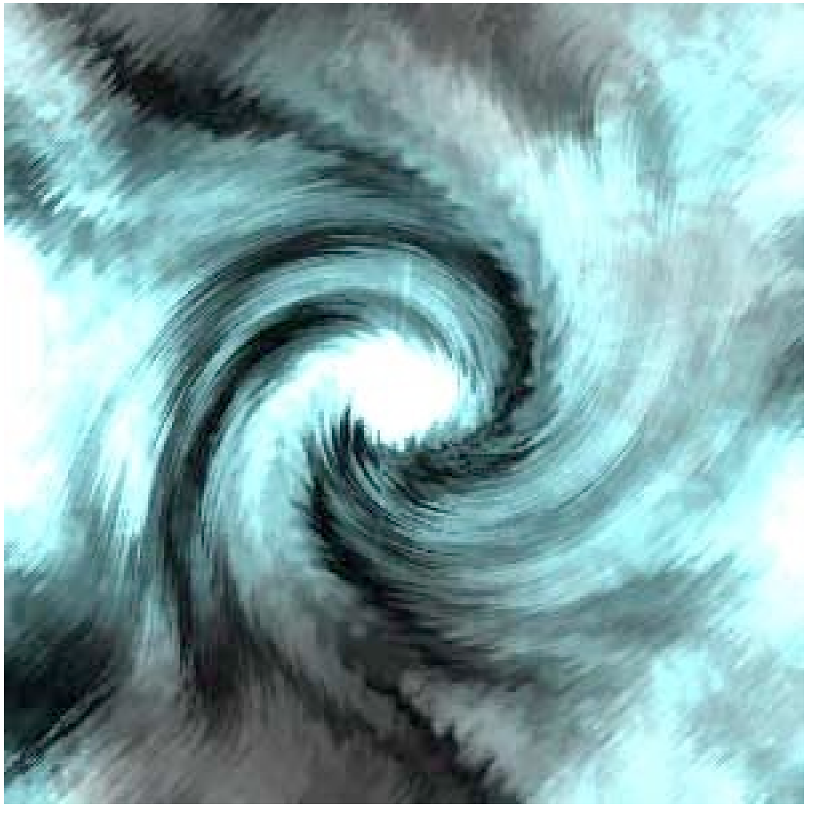
CFL:

$$\lambda = \frac{\Delta t^{(\#t-der)}}{\Delta x^{(\#x-der)}}$$

Physical Example

$$\omega_t + [\psi, \omega] = \nu \nabla^2 \omega$$

$$\nabla^2 \psi = \omega$$



Spectral Methods

Expand u(x,t) :

$$u(x,t) = \sum_{k=1}^{N} a_k(t)\phi_k(x)$$

where $\phi_k(x)$ are orthogonal

Fourier Series

$$\hat{u}(t) = u(k,t) = \sum_{n=1}^{N} u(n,t) e^{-i2\pi(k-1)(n-1)/N}$$

$$u(n,t) = 1/N \sum_{n=1}^{N} \hat{u}(t) e^{i2\pi(k-1)(n-1)/N}$$

Derivative property:

$$u^{(m)}\hat{}(x,t) = (ik)^{(m)} \hat{u}(k,t)$$

The Problem in Fourier Space is **Easier**

PDE in Fourier Space

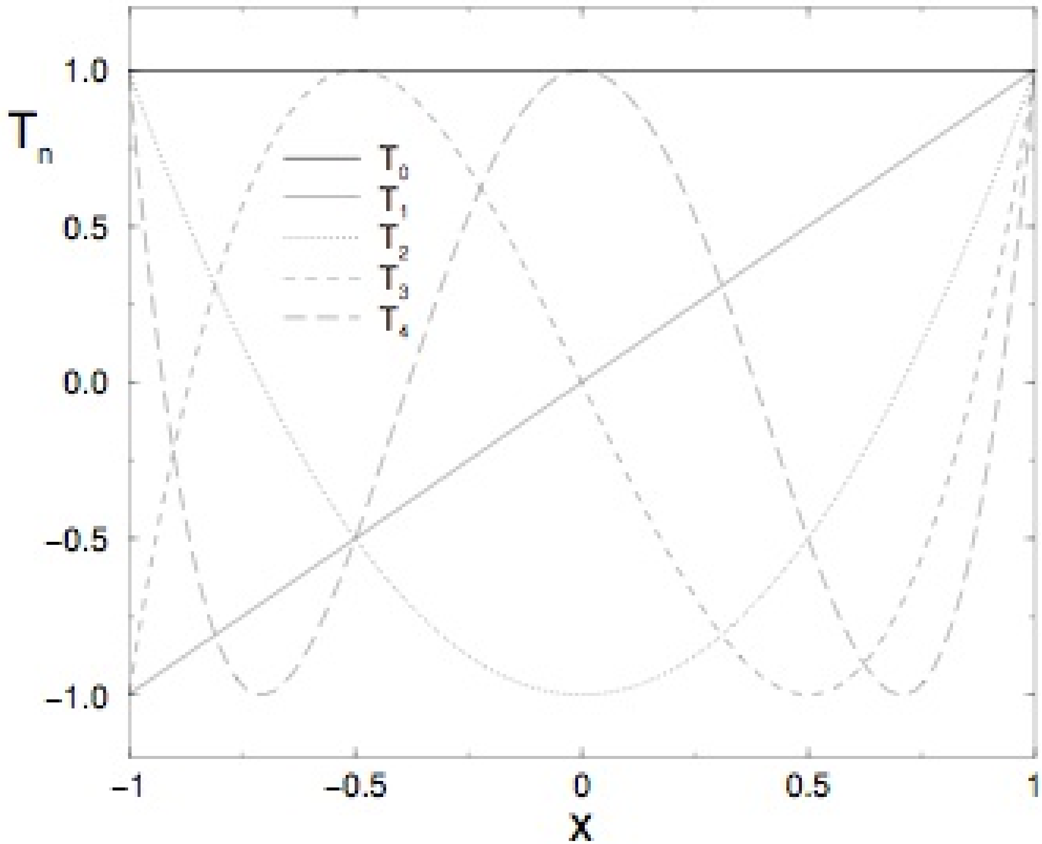
$$u_t = Lu + N(u)$$

$$\Big\Downarrow$$

$$\hat{u}_t = \alpha(k)\hat{u} + \hat{N}(u)$$

Chebishev Polynomials

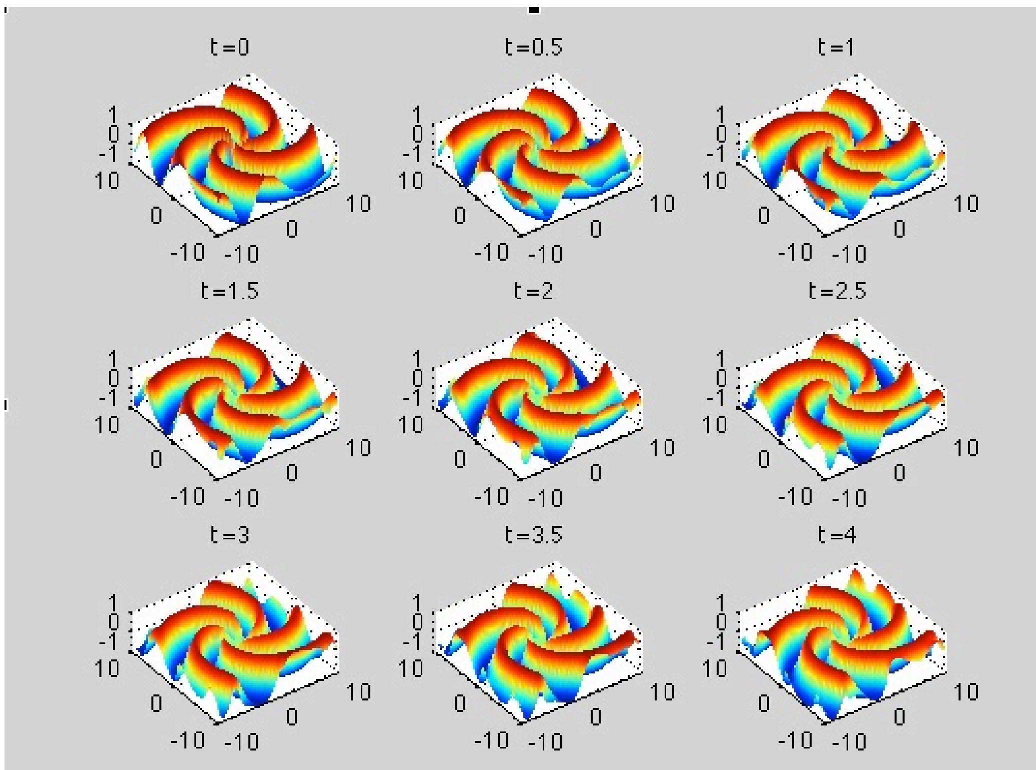
$$\phi_k(x) = T_k(x), \quad T_k(\cos\theta) = \cos k\theta$$



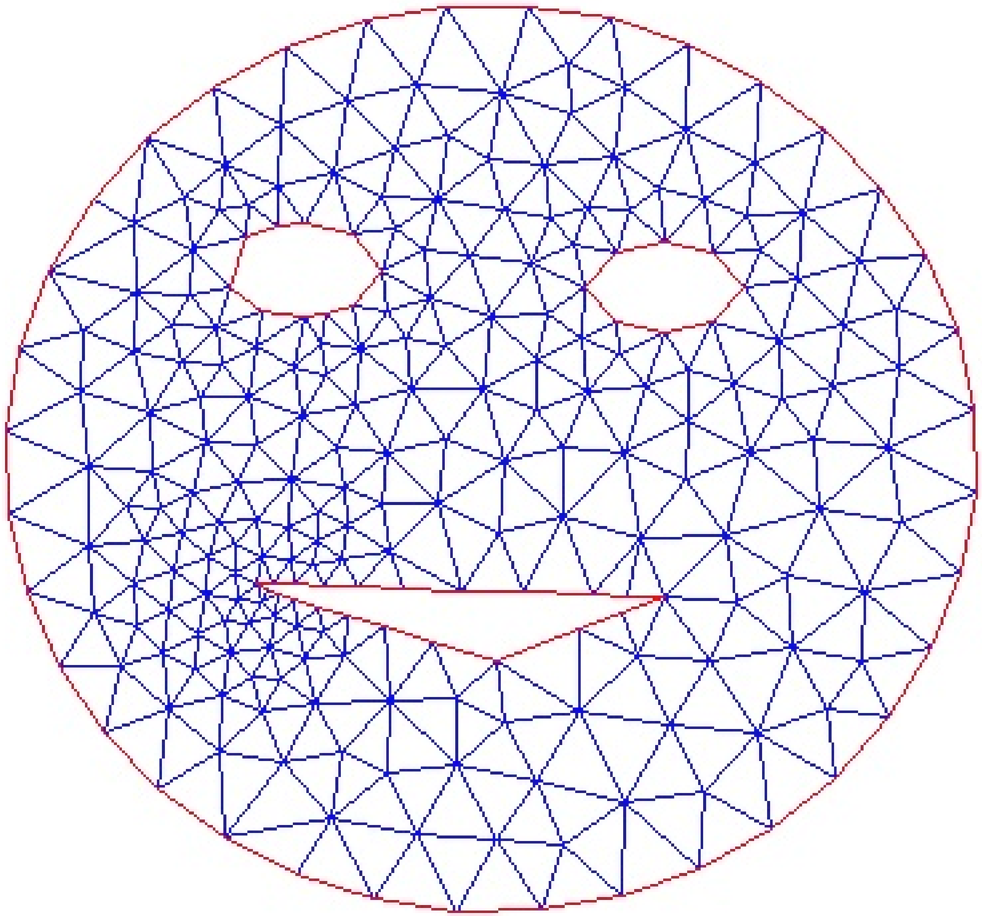
Physical Example: Reaction Diffusion Equations

$$U_t = \lambda(A)U - \omega(A)V + D_1\nabla^2 U$$

$$V_t = \omega(A)U + \lambda(A)V + D_2\nabla^2 V$$



Finite Elements



Competency / Mastery

Skill 1: Python Programming Fundamentals

Skill 2: Initial Value Problems: Forward Integration

Skill 3: Initial Value Problems: Implicit Integration and Stability

Skill 4: Boundary Values Problems: Shooting and Relaxation (Direct)

Skill 5: Finite Difference Derivative Operators for PDE

Skill 6: Finite Difference: Solving Linear Problems

Skill 7: Finite Difference: PDE Solution and Stability

Skill 8: Spectral Methods for PDE

Skill 9: Finite Elements Introduction

It’s YOUR Game

What's Next?

- Scientific Computing with **Data**

  AMATH 482/582

- **High-Performance** Scientific Computing

  AMATH 483/583

- Applied **Analysis**

  AMATH 567

As Cool As It Gets

Please Fill

AMATH 481
https://uw.iasystem.org/survey/312834/

AMATH 581
https://uw.iasystem.org/survey/312842/