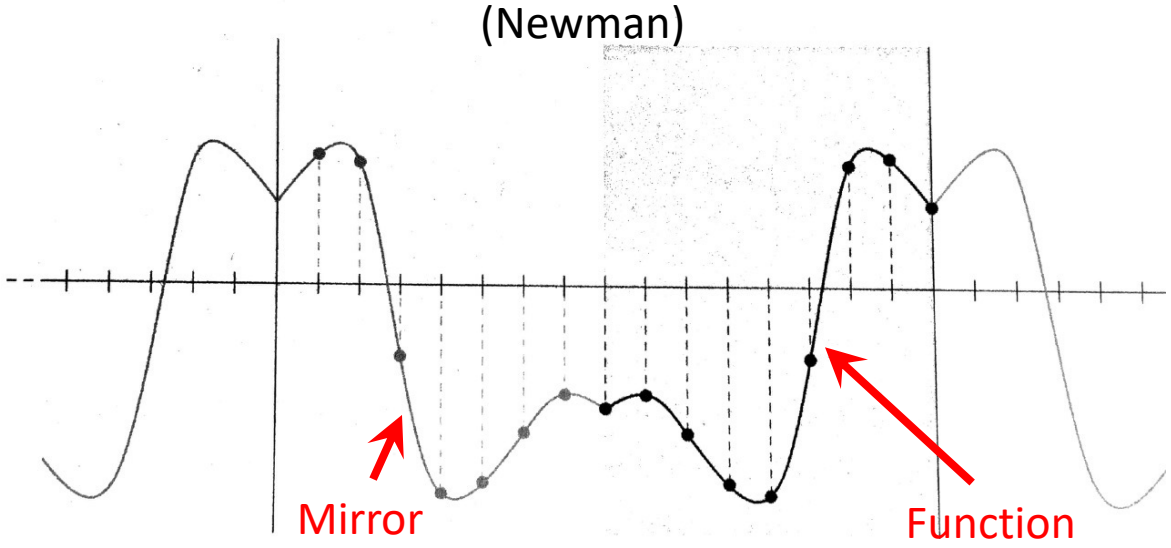
PHY604 Lecture 13

October 5, 2021

Review: Cosine transformation (see Newman Sec. 7.3)



- Can also construct Fourier series from using sine and cosine functions instead of complex exponentials
- Cosine series: Can only represent functions symmetric about the midpoint of the interval
  - Can enforce this for any function by mirroring it, and then repeating the mirrored function
- Different ways of writing it (see Newman):

$$F_k = \sum_{n=0}^{N-1} f_n \cos\left(\frac{\pi k(n+\frac{1}{2})}{N}\right), \quad f_n = \frac{1}{N}\sum_{k=0}^{N-1} F_k \cos\left(\frac{\pi k(n+\frac{1}{2})}{N}\right)$$

Review: Procedure for FFT

- 1. Start with (trivial) FT of single samples:

$$F_0 = \sum_{n=0}^{0} f_n e^0 = f_0$$

- 2. Combine them in pairs using:

$$F_k = F_k^{\text{even}} + e^{-i2\pi k/N} F_k^{\text{odd}}$$

- 3. Continue combining into fours, eights, etc. until the full transform on the full set of samples is reconstructed

Periodicity of FFT

- Last class we claimed that if for *k* in the range [0,N/2] we had:

$$F_k = F_k^{\text{even}} + e^{-i2\pi k/N} F_k^{\text{odd}}$$

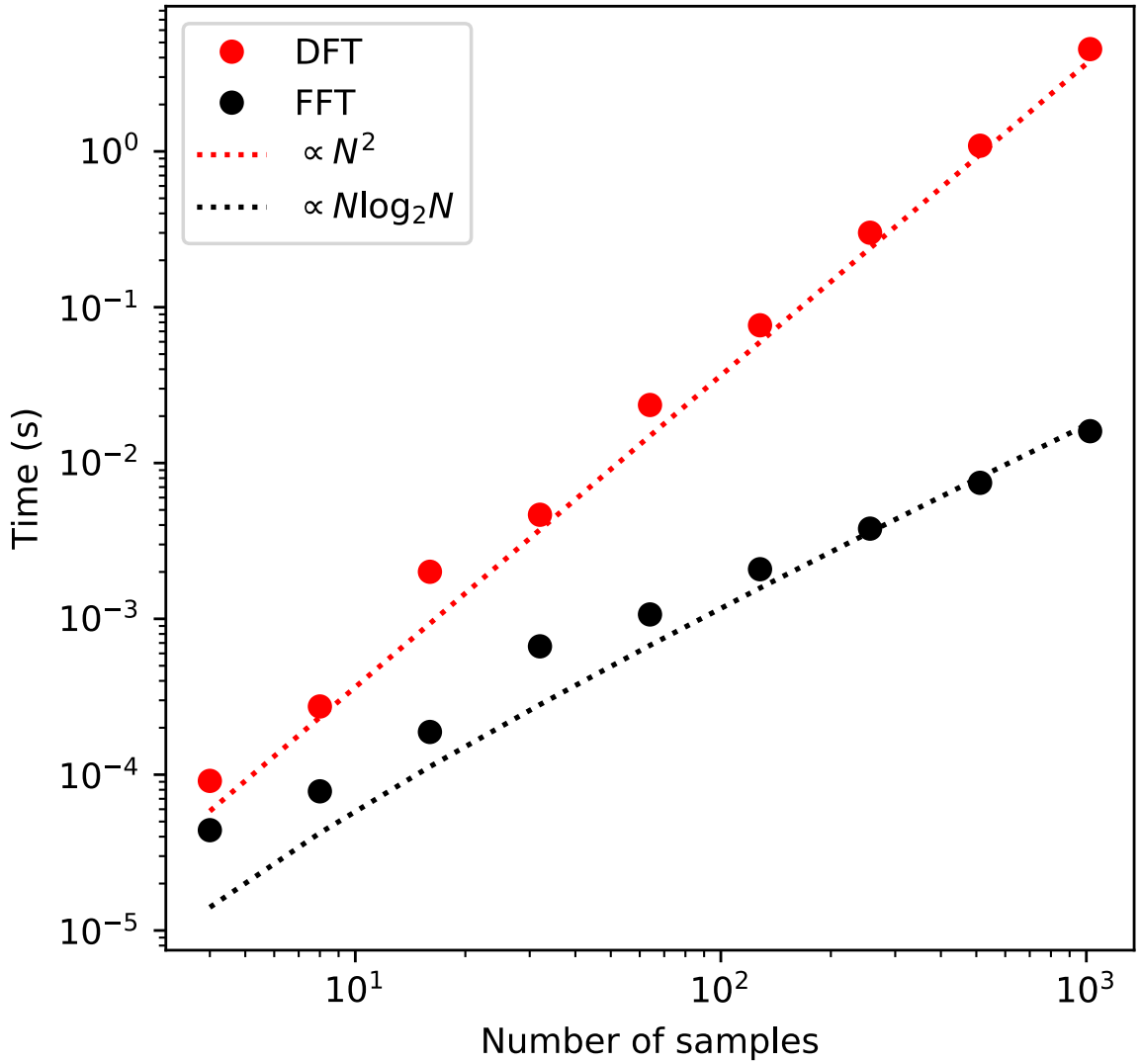
- Then for *k* in [N/2,N-1] we could write

$$F_{k+N/2} = F_k^{\text{even}} - e^{-i2\pi k/N} F_k^{\text{odd}}$$

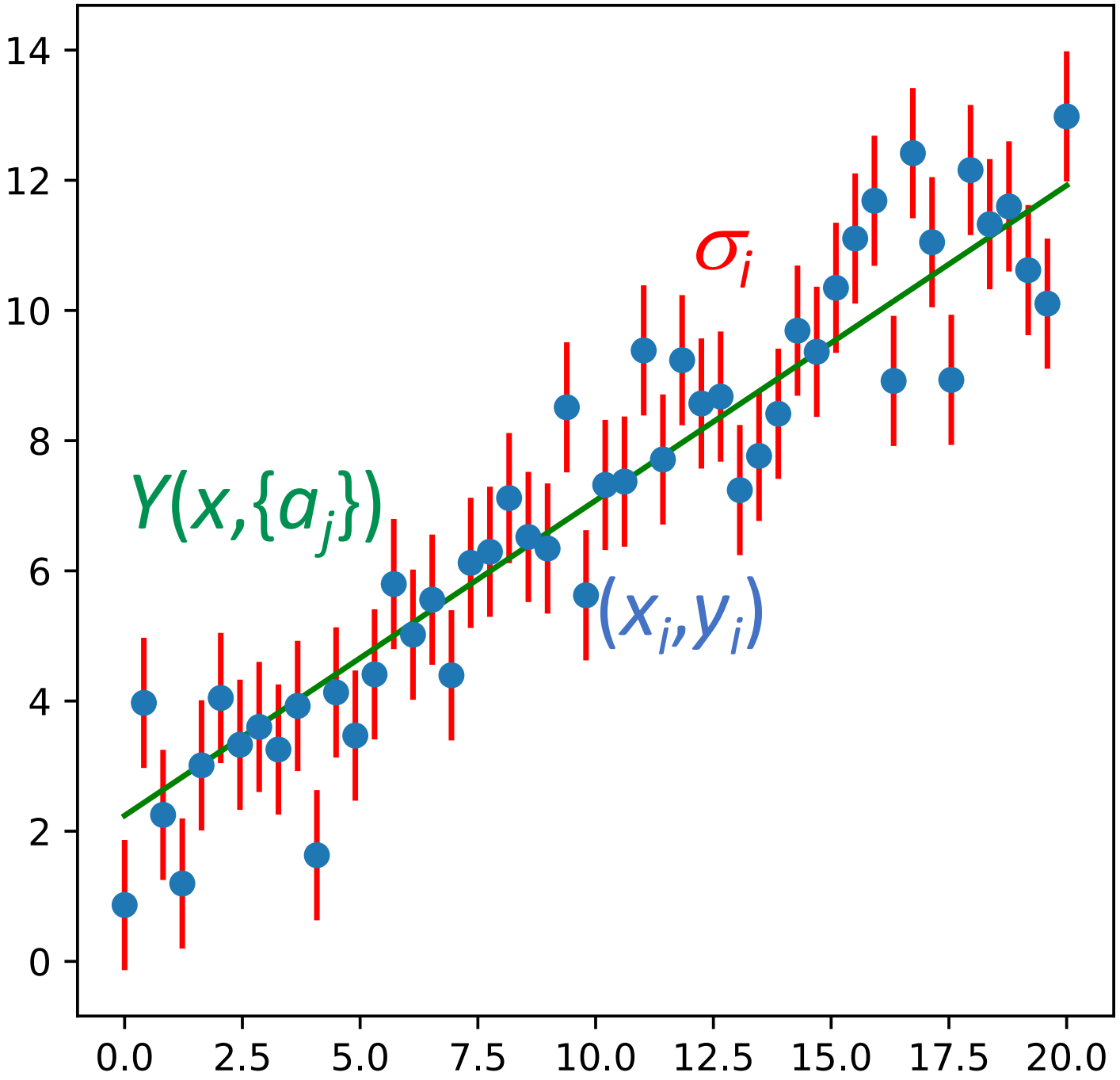
- Recall that we had assumed $N=2^m$. So: $\exp\left[-\frac{2\pi ik}{N}\right] = \exp\left[-\frac{2\pi ik}{2^m}\right]$
  - And $k + N/2 = k + 2^{m-1}$
- Then:

$$\exp\left[-\frac{2\pi i(k+2^{m-1})}{2^m}\right] = \exp\left[-\frac{2\pi ik}{2^m}\right]\exp\left[-\frac{2\pi i2^{m-1}}{2^m}\right] = \exp\left[-\frac{2\pi ik}{2^m}\right]\exp\left[-\pi i\right] = -\exp\left[-\frac{2\pi ik}{2^m}\right]$$

Review: Speed up of FFT vs DFT



Review: Curve fitting



Review: Linear regression: Finding coefficients

- Minimize χ^2 with respect to coefficients:

$$\frac{\partial \chi^2}{\partial a_0} = 2 \sum_{i=0}^{N-1} \frac{a_0 + a_1 x_i - y_i}{\sigma_i^2} = 0, \qquad \frac{\partial \chi^2}{\partial a_0} = 2 \sum_{i=0}^{N-1} x_i \frac{a_0 + a_1 x_i - y_i}{\sigma_i^2} = 0$$

- We can write as:

$$a_0 S + a_1 \Sigma_x - \Sigma_y = 0, \qquad a_0 \Sigma_x + a_1 \Sigma_{x^2} - \Sigma_{xy} = 0$$

- Where coefficients are known:

$$S \equiv \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2}, \quad \Sigma_x \equiv \sum_{i=0}^{N-1} \frac{x_i}{\sigma_i^2}, \quad \Sigma_y \equiv \sum_{i=0}^{N-1} \frac{y_i}{\sigma_i^2}, \quad \Sigma_{x^2} \equiv \sum_{i=0}^{N-1} \frac{x_i^2}{\sigma_i^2}, \quad \Sigma_{xy} \equiv \sum_{i=0}^{N-1} \frac{x_i y_i}{\sigma_i^2}$$

Review: Linear regression: Finding coefficients

- Solving for a_0 and a_1:

$$a_0 = \frac{\Sigma_y \Sigma_{x^2} - \Sigma_x \Sigma_{xy}}{S\Sigma_{x^2} - (\Sigma_x)^2}, \qquad a_1 = \frac{S\Sigma_{xy} - \Sigma_y \Sigma_x}{S\Sigma_{x^2} - (\Sigma_x)^2}$$

- Note that if σ_i is constant, it will cancel out
- Now let's define an error bar for the curve-fitting parameter a_j

$$\sigma_{a_j}^2 = \sum_{i=0}^{N-1} \left(\frac{\partial a_j}{\partial y_i} \right)^2 \sigma_i^2$$

  - See: https://en.wikipedia.org/wiki/Propagation_of_uncertainty

Both independent of y_i

- For our linear case (after some algebra):

$$\sigma_{a_0} = \sqrt{\frac{\Sigma_{x^2}}{S\Sigma_{x^2} - (\Sigma_x)^2}}, \qquad \sigma_{a_1} = \sqrt{\frac{S}{S\Sigma_{x^2} - (\Sigma_x)^2}}$$

Today's lecture:
Curve fitting and PDEs

- General least-squares curve fitting

- Partial differential equations
  - Types of equations
  - Marching methods for parabolic and hyperbolic PDEs

General least squares fit

- No analytic solution to general least squares problem, but can solve numerically
- Generalize to functions of the form:

$$Y(x_i, \{a_j\}) = a_0 Y_0(x) + a_1 Y_1(x) + \cdots + a_{M-1} Y_{M-1}(x) = \sum_{j=0}^{M-1} a_j Y_j(x)$$

- Now minimize χ^2:

$$\frac{\partial \chi^2}{\partial \{a_j\}} = \frac{\partial}{\partial \{a_j\}} \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left[\sum_{k=0}^{M-1} a_k Y_k(x_i) - y_i \right]^2 = 0$$

$$= \sum_{i=0}^{N-1} \frac{Y_j(x_i)}{\sigma_i^2} \left[\sum_{k=0}^{M-1} a_k Y_k(x_i) - y_i \right] = 0$$

General least-squares fit

- From previous slide, we have:

$$\sum_{i=0}^{N-1}\sum_{k=0}^{M-1}\frac{Y_j(x_i)Y_k(x_i)}{\sigma_i^2}a_k = \sum_{i=0}^{N-1}\frac{Y_j(x_i)y_i}{\sigma_i^2}$$

- Set of *j* equations known as normal equations of the least-squares problem (*Y*'s may be nonlinear, but linear in *a*'s)
- Define design matrix with elements $A_{ij} = Y_j(x_i)/\sigma_i$:

$$\mathbf{A} = \begin{bmatrix} \frac{Y_0(x_0)}{\sigma_0} & \frac{Y_1(x_0)}{\sigma_0} & \cdots \\ \frac{Y_0(x_1)}{\sigma_1} & \frac{Y_1(x_1)}{\sigma_1} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

- Only depends on independent variables (not y_i)

General least-squares fit

- With design matrix, we can rewrite: $$\sum_{i=0}^{N-1}\sum_{k=0}^{M-1}\frac{Y_j(x_i)Y_k(x_i)}{\sigma_i^2}a_k=\sum_{i=0}^{N-1}\frac{Y_j(x_i)y_i}{\sigma_i^2}$$

- As: $$\sum_{i=0}^{N-1}\sum_{k=0}^{M-1}A_{ij}A_{ik}a_k=\sum_{i=0}^{N-1}A_{ij}\frac{y_i}{\sigma_i}\quad\Longrightarrow\quad(\mathbf{A}^{\mathrm{T}}\mathbf{A})\mathbf{a}=\mathbf{A}^{\mathrm{T}}\mathbf{b}$$
  - Where $b_i=y_i/\sigma_i$

- Thus: $\mathbf{a}=(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$

- Or, we can solve for **a** via Gaussian elimination

Goodness of fit

- Usually, we have *N* >> *M*, the number of data points is much greater than the number of fitting variables
- Given the error bars, how likely is it that the curve actually describes the data?
- Rule of thumb: If the fit is good, on average the difference should be approximately equal to the error bars

$$|y_i - Y(x_i)| \simeq \sigma_i$$

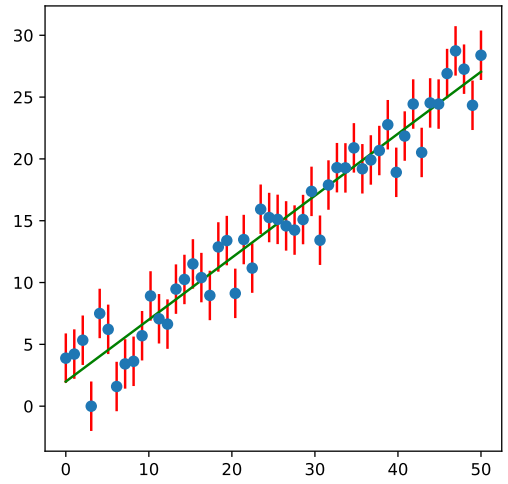
- Since we know we can have a perfect fit for M=N, we postulate:

$$\chi^2 \simeq N - M$$

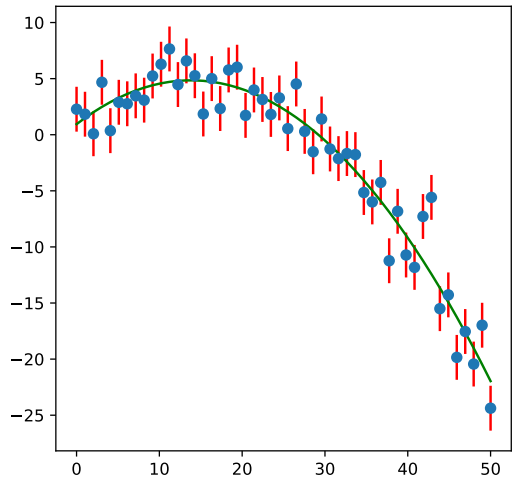
- If $\chi^2 \gg N - M$, probably not an appropriate function (or too small error bars
- If $\chi^2 \ll N - M$, fit is too good, error bars may be too large

Least squares fitting example:

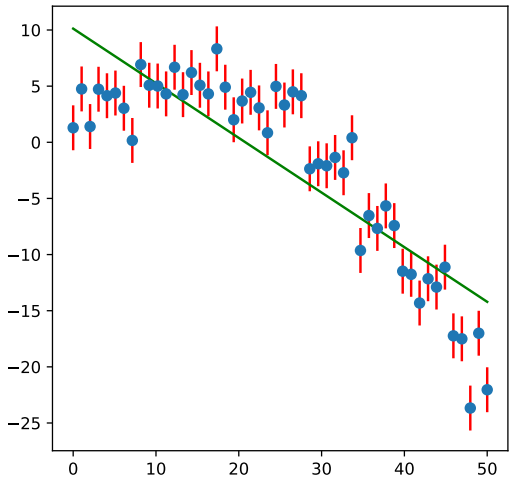
Linear regression, linear function



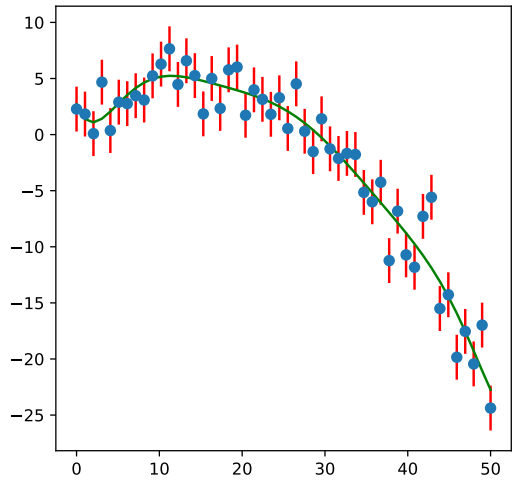
Polynomial regression (order 2), quadratic function



Linear regression, quadratic function



Polynomial regression (order 10), quadratic function



Comments on general least squares

- In the example, we used polynomials as our functions, but can use linear combinations of any functions we would like

- We choose functions strategically to get the best least squares fit
  - Often choosing orthogonal basis functions in the range of the fit will produce better fits

- The matrix $\mathbf{A}^T\mathbf{A}$ is notoriously ill conditioned especially for increased number of basis functions
  - Gaussian substitution will have problems solving (numpy solve uses singular-value decomposition)

- Procedure can be generalized if we also have errors in *x*

Nonlinear least-squares fitting

- Even in the polynomial case, we were using linear combinations of functions
- We can also directly fit a function whose parameters enter nonlinearly
- Consider the function: $f(a_0, a_1) = a_0 e^{a_1 x}$
- Want to minimize: $Q \equiv \sum_{i=1}^{N} (y_i - a_0 e^{a_1 x_i})^2$
- Take derivatives:

$$f_0 = \frac{\partial Q}{\partial a_0} = \sum_{i=1}^{N} e^{a_1 x_i} (a_0 e^{a_1 x_i} - y_i) = 0,$$

$$f_1 = \frac{\partial Q}{\partial a_1} = \sum_{i=1}^{N} x_i e^{a_1 x_i} (a_0 e^{a_1 x_i} - y_i) = 0$$

Nonlinear least-squares fitting

- Produces a nonlinear system—we can use the multivariate root-finding techniques we learned earlier:
  - Compute the Jacobian
  - Take an initial guess for unknown coefficients
  - Use Newton-Raphson techniques to compute the correction:

$$\mathbf{a}_1 = \mathbf{a}_0 - \mathbf{J}^{-1}\mathbf{f}$$

  - Iterate

- Can be very difficult to converge, and highly dependent on the initial guess

Fitting packages

- Fitting is a very sensitive procedure—especially for nonlinear cases
- Lots of minimization packages exist that offer robust fitting procedures

- MINUIT2: the standard package in high-energy physics (Python version: PyMinuit and Iminuit)
- MINPACK: Fortran library for solving least squares problems—this is what is used under the hood for the built in SciPy least squares routine
    - http://www.netlib.org/minpack/
- SciPy optimize: https://docs.scipy.org/doc/scipy/reference/optimize.html

Today's lecture:
Curve fitting and PDEs

- General least-squares curve fitting

- Partial differential equations
  - Types of equations
  - Marching methods for parabolic and hyperbolic PDEs

Partial differential equations (Garcia Chs. 6-9)

- Previously, we studied ordinary differential equations

- Much of physics is involved in solving partial differential equations
  - Schrodinger equation in Quantum mechanics
  - Maxwell's equations in electricity and magnetism
  - Wave equation in optics and acoustics

- For ODEs we developed general methods to solve a variety of problems, e.g., 4^{th} order Runge-Kutta

- For PDEs, we first classify the type of equation, that will tell us what method to use

Examples of PDE types

- Parabolic equations
  - E.g., Time-dependent Schrodinger equation, 1D diffusion equation
  - Consider the Fourier equation with temperature *T* and thermal diffusion coefficient κ:

$$\frac{\partial T(x,t)}{\partial t} = \kappa \frac{\partial^2 T(x,t)}{\partial x^2}$$

- Hyperbolic equations
  - E.g., 1D wave equation with amplitude *A* and speed *c*:

$$\frac{\partial^2 A(x,t)}{\partial t^2} = c^2 \frac{\partial^2 A(x,t)}{\partial x^2}$$

- Elliptic equations
  - E.g., Poisson equation:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = -\frac{1}{\epsilon_0}\rho(x,y)$$

General classification of PDEs

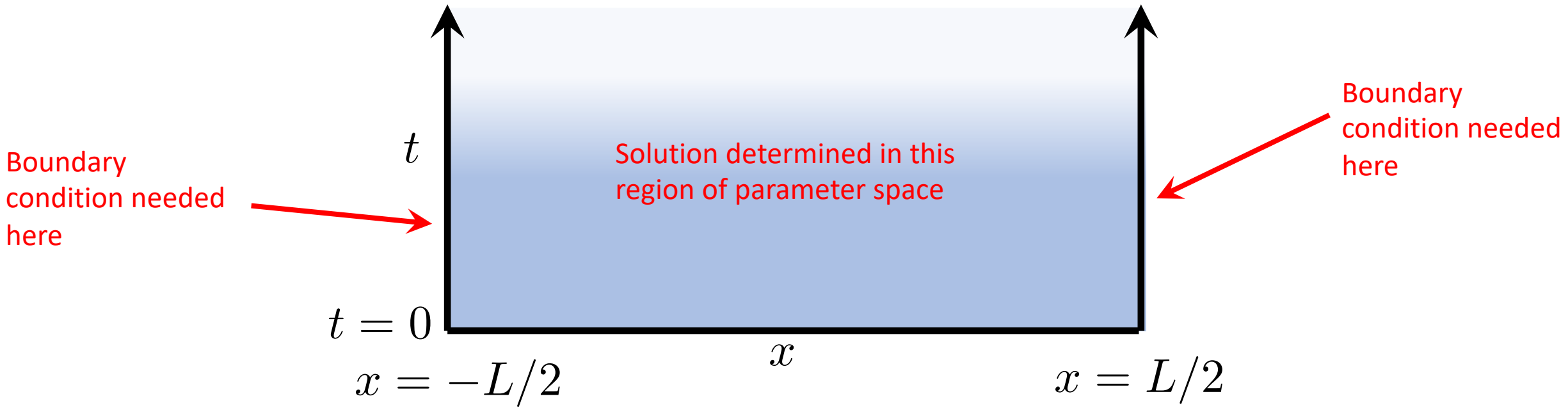
- Consider a general PDE of two independent variables:

$$a\frac{\partial^2 A}{\partial x^2} + b\frac{\partial^2 A}{\partial x \partial y} + c\frac{\partial^2 A}{\partial y^2} + d\frac{\partial A}{\partial x} + e\frac{\partial A}{\partial y} + f\,A(x,y) + g = 0$$

- Hyperbolic if: $b^2 - 4ac > 0$
- Parabolic if: $b^2 - 4ac = 0$
- Elliptic if: $b^2 - 4ac < 0$
- Most problems involve hybrid systems, including multiple types

Initial value problems

- Diffusion and wave equations usually solved as initial value problems
  - Diffusion: Given initial temperature distribution, find temperature distribution at a later time
  - Wave: Start with initial amplitude and velocity of wave pulse and find the wave pulse at a later time
- Need to specify initial conditions as well as boundary conditions



Types of boundary conditions

- Dirichlet boundary conditions: Specify the solution on boundary
  - E.g., fix the temperature at the boundaries:

$$T(x = -L/2, t) = T_a, \quad T(x = L/2, t) = T_b$$

- Neumann boundary conditions: Specify the derivative on the boundary
  - E.g., "insulated" boundaries

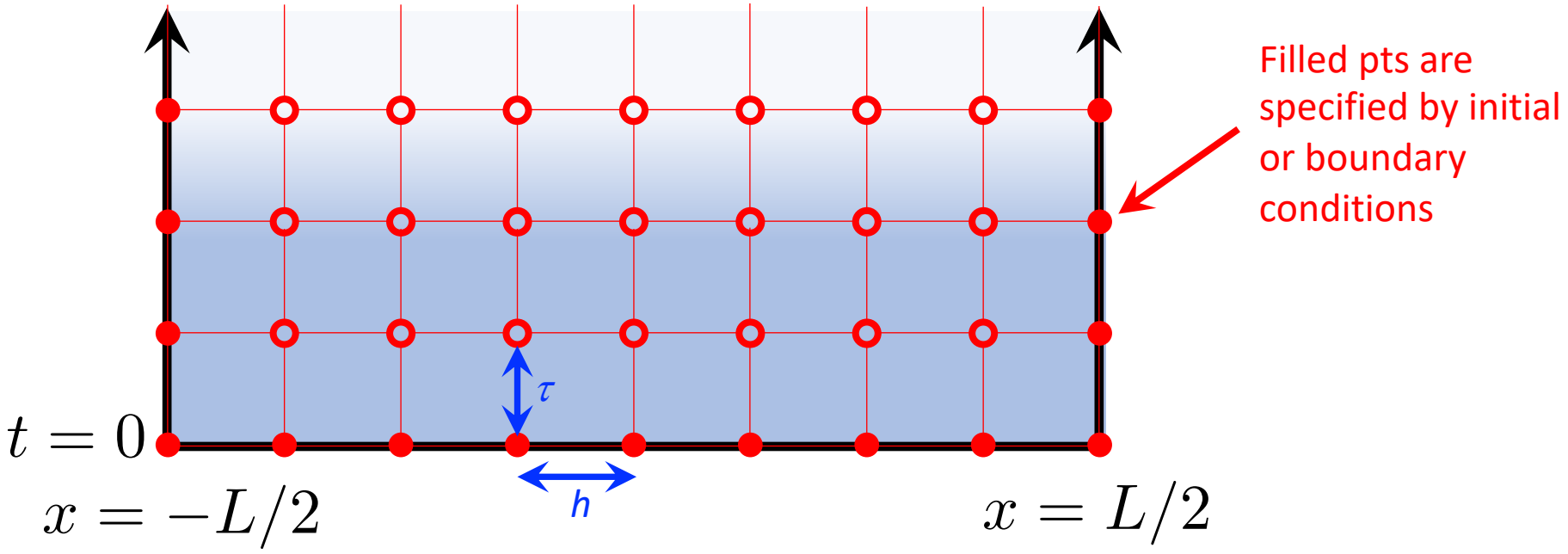
$$-\kappa \frac{dT}{dx}\bigg|_{x=-L/2} = F_a = 0, \qquad -\kappa \frac{dT}{dx}\bigg|_{x=L/2} = F_b = 0$$

- Periodic boundary conditions: Equate the functions at both ends

$$T(x = -L/2, t) = T(x = L/2, t), \qquad \frac{dT}{dx}\bigg|_{x=-L/2} = \frac{dT}{dx}\bigg|_{x=L/2}$$

Marching methods for initial value problems

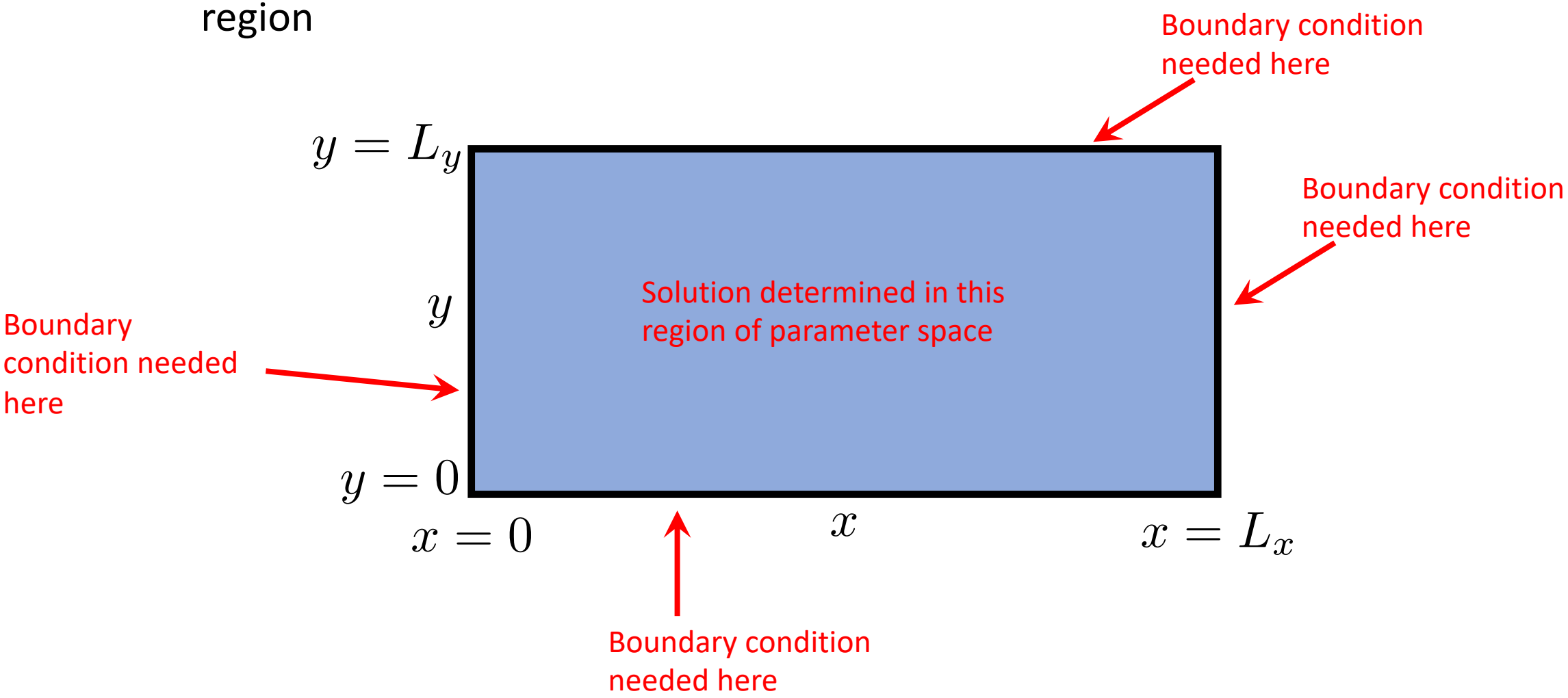
- We first must discretize in time and space:



- Start from the initial condition, move forward in time one timestep at a time

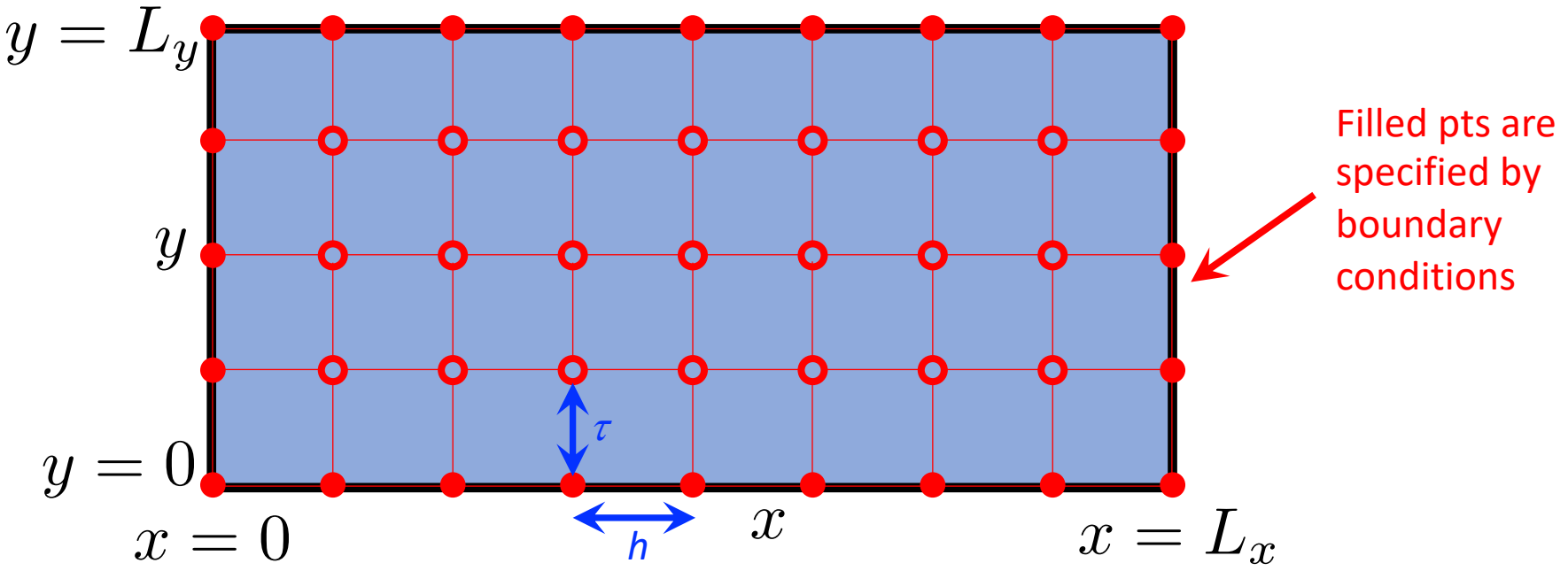
Boundary value problems

- All boundary values are specified at the outset
  - E.g., Laplace's equation in electrostatics, potential fixed on for sides of spatial region



Jury methods for boundary value problems

- Discretize in space:



- Potential in interior is influenced by all the boundary points, reconciles all the constraints imposed by boundaries

Diffusion equation with FTCS

- Diffusion equation: $\frac{\partial T(x,t)}{\partial t} = \kappa \frac{\partial^2 T(x,t)}{\partial x^2}$
- Discretize in space and time: $T_i^n = T(x_i, t_n)$
  - $x_i = i\,h - L/2$ and $t_n = n\,\tau$
  - Take spatial boundary points as $i = 0$ and $i = N\text{-}1$, so h = $L/(N\text{-}1)$

- Discretize time derivative with forward difference:

$$\frac{\partial T(x,t)}{\partial t} \rightarrow \frac{T_i^{n+1} - T_i^n}{\tau}$$

- Discretize spatial derivative using central difference:

$$\frac{\partial^2 T(x,t)}{\partial x^2} \rightarrow \frac{T_{i+1}^n + T_{i-1}^n - 2T_i^n}{h^2}$$

Diffusion equation with FTCS

- Now the discretized PDE is:

$$\frac{T_i^{n+1} - T_i^n}{\tau} = \kappa \frac{T_{i+1}^n + T_{i-1}^n - 2T_i^n}{h^2}$$

- And temperature at future time is:

$$T_i^{n+1} = T_i^n + \frac{\kappa\tau}{h^2}(T_{i+1}^n + T_{i-1}^n - 2T_i^n)$$

- Explicit: Everything that depends on previous timestep *n* is on RHS

- Discretization is reminiscent of Euler's method for ODEs

Numerical stability of FTCS method

- The numerical stability of the solution depends on the timestep
- Consider initial conditions of a delta-function peak in *T* located at *N*/2
  - Discrete approximation, $T(x=N/2,t_0) = 1/h$
- Can show from analytical solutions to this problem that the delta function will spread into a Gaussian with width:

$$\sigma(t) = \sqrt{2\kappa t}$$

- Thus, if t_σ is the time it takes for σ to increase by one grid spacing:

$$t_\sigma = \frac{h^2}{2\kappa}$$

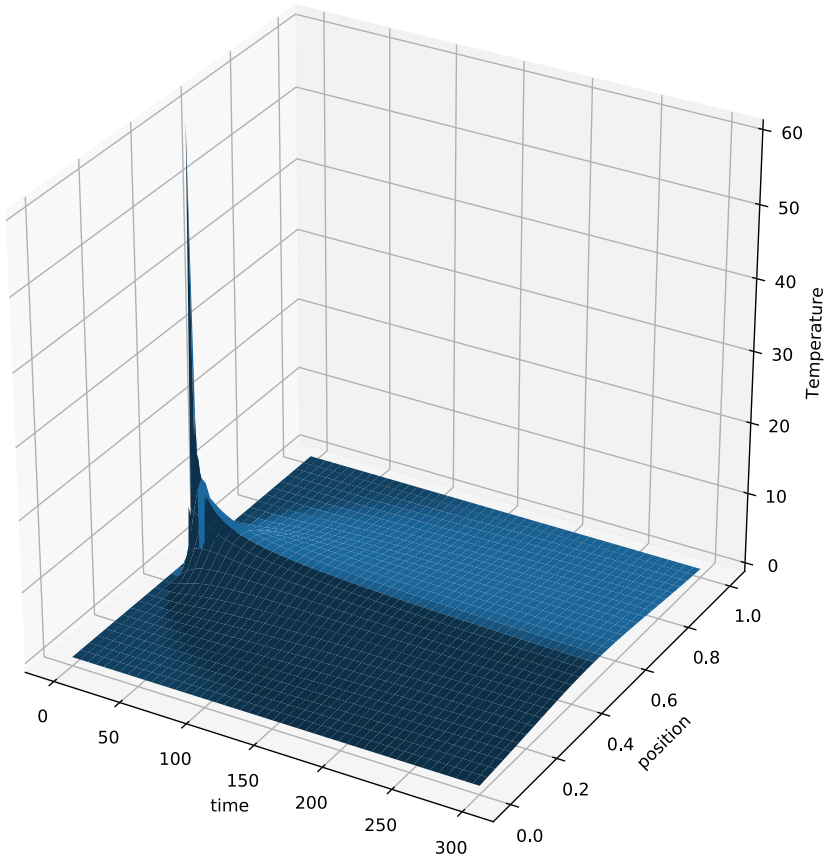
- Then:

$$T_i^{n+1} = T_i^n + \frac{\tau}{2t_\sigma}(T_{i+1}^n + T_{i-1}^n - 2T_i^n)$$

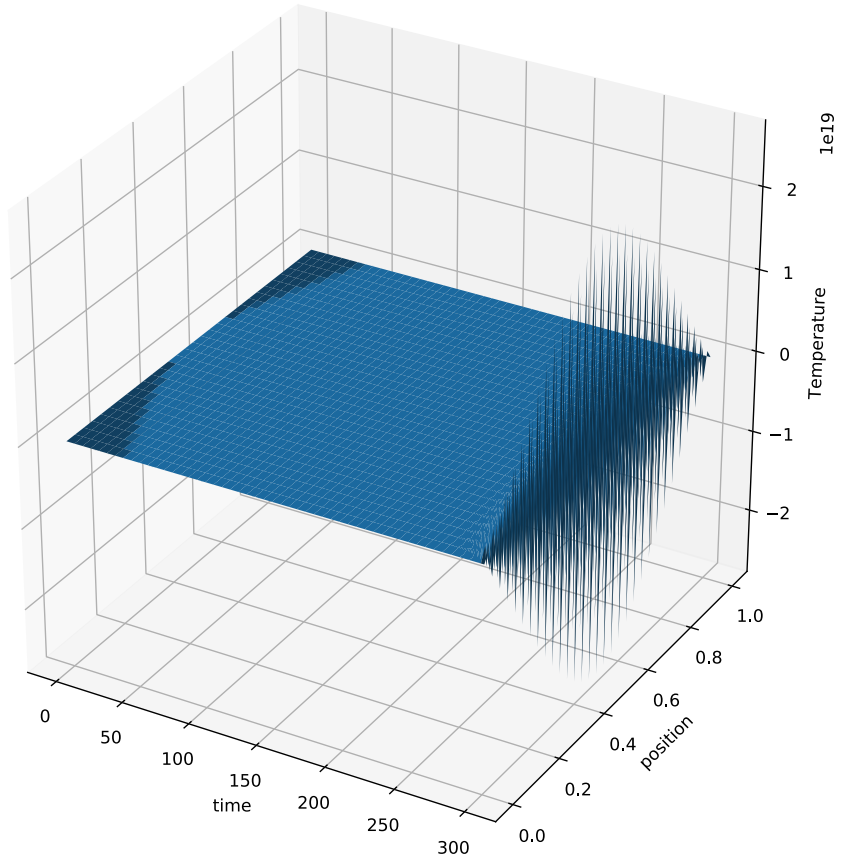
- Should not use a timestep much larger than t_σ

FCTS method on diffusion equation

Numerically stable: τ = 1e-4



Numerically stable: τ = 1.5e-4



Wave and advection equations

- Wave equation: $$\frac{\partial^2 A(x,t)}{\partial t^2} = c^2 \frac{\partial^2 A(x,t)}{\partial x^2}$$
- When we discussed ODEs we used the trick to turn 2nd order equations into systems of 1D equations with auxiliary variables
- Use a similar trick for wave PDE:

$$P \equiv \frac{\partial A}{\partial t}, \qquad Q \equiv \frac{\partial A}{\partial x}$$

- So, we have the pair of equations:

$$\frac{\partial P}{\partial t} = c\frac{\partial Q}{\partial x}, \qquad \frac{\partial Q}{\partial t} = c\frac{\partial P}{\partial x}$$

- Or:

$$\frac{\partial \mathbf{a}}{\partial t} = c\mathbf{B}\frac{\partial \mathbf{a}}{\partial x}, \qquad \mathbf{a} = \begin{bmatrix} P \\ Q \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Advection equation

- Thus, we see that there is a simpler hyperbolic equation, the advection equation:

$$\frac{\partial a}{\partial t} = -c\frac{\partial a}{\partial x}$$

- Describes the evolution of some scalar field *a* carried by a flow of velocity *c*
  - Also known as linear convection equation
  - Waves move only in one direction (to the right if $c > 0$), unlike the wave equation
- "Flux conservation" equation
  - E.g., continuity equation in electrodynamics/quantum mechanics:

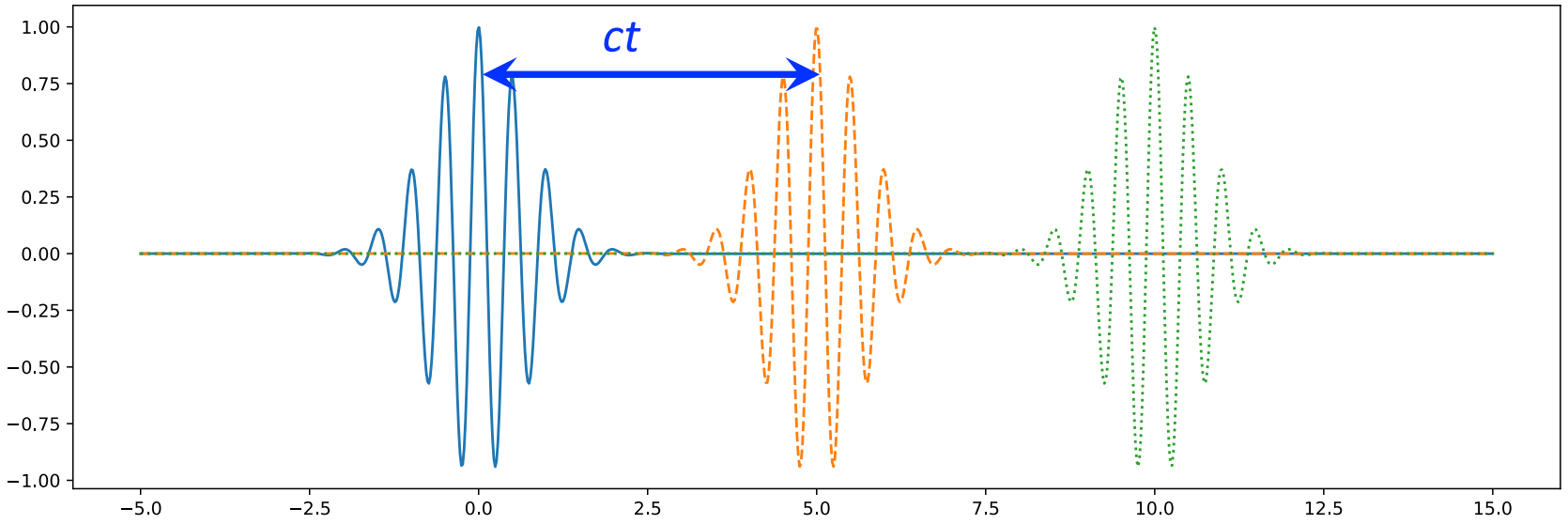
$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{J}(p)$$

Advection equation is easy to solve analytically

- For initial condition: $a(x, t = 0) = f_0(x)$
- Solution is: $a(x, t) = f_0(x - ct)$
- Consider a wavepacket of the form:

$$a(x, t = 0) = \cos[k(x - x_0)] \exp\left[-\frac{(x - x_0)^2}{2\sigma^2}\right]$$

- Solution: $a(x, t) = \cos[k((x - ct) - x_0)] \exp\left[-\frac{((x - ct) - x_0)^2}{2\sigma^2}\right]$



Why study such a simple equation?

- Excellent test case for numerical methods since we know exactly what we should get

- Let's start with the FTCS methods we used for the diffusion equation:

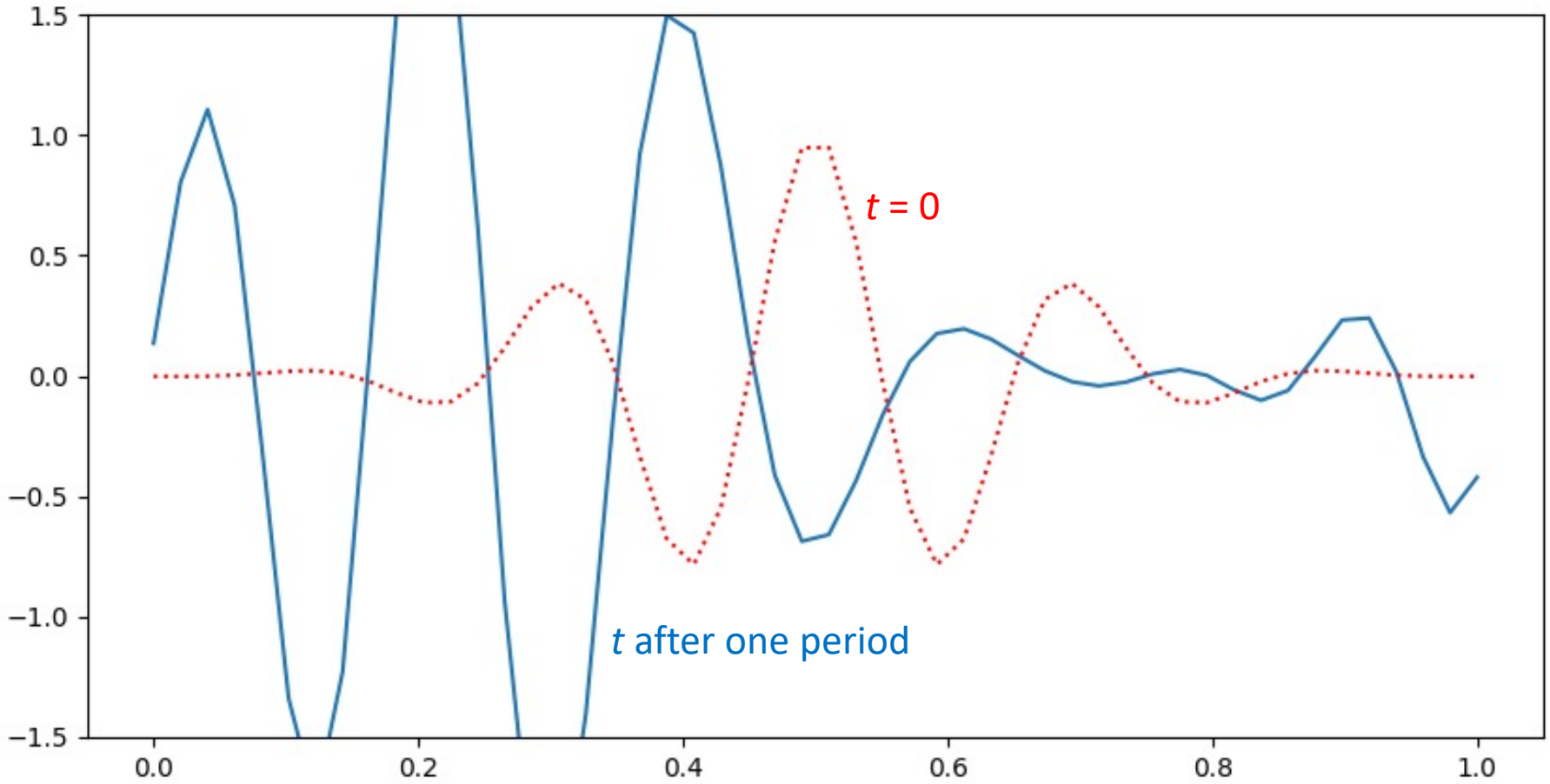
$$\frac{\partial a}{\partial t} \rightarrow \frac{a_i^{n+1} - a_i^n}{\tau}, \qquad \frac{\partial a}{\partial x} \rightarrow \frac{a_{i+1}^n - a_{i-1}^n}{2h}$$

- So, the FTCS equation is:

$$a_i^{n+1} = a_i^n - \frac{c\tau}{2h}(a_{i+1}^n - a_{i-1}^n)$$

- We will use periodic boundary conditions for this case

FTCS method clearly fails for the advection equation using this timestep



How can we do a better job?

- We could try to adjust numerical parameters, but it will not work!
  - FTCS is unstable for any τ ! (will come back to this later on)
  - Can delay the problems but not get rid of them

- Stability problem can be fixed with a simple modification: The Lax method:

$$a_i^{n+1} = \frac{1}{2}(a_{i+1}^n + a_{i-1}^n) - \frac{c\tau}{2h}(a_{i+1}^n - a_{i-1}^n)$$

- Simply replacing the first term with the average of the left and right neighbors

Stability of the Lax method

- It can be shown that the Lax method is numerically stable (i.e., does not diverge) if:

$$\frac{c\tau}{h} \leq 1$$

- So:

$$\tau_{\max} = \frac{h}{c}$$

- Courant-Friedrighs-Lewy (CFL) condition
  - We will discuss more on stability conditions later
  - If we want a finer grid in space, we need a finer timestep

Stability of the Lax method

- It can be shown that the Lax method is numerically stable (i.e., does not diverge) if:

$$\frac{c\tau}{h} \leq 1$$

- So: $\tau_{\max} = \frac{h}{c}$

- Courant-Friedrighs-Lewy (CFL) condition
  - We will discuss more on stability conditions later on
  - If we want a finer grid in space, we need a finer timestep

- Too large of a timestep: Numerically unstable
- Too small of a timestep: Amplitude suppressed
  - Averaging term introduces and artificial viscosity, proportional to τ

Lax-Wendroff scheme for hyperbolic PDEs

- Lax-Wedroff is second-order finite difference scheme
- Take the Taylor expansion in time:

$$a(x, t+\tau) = a(x,t) + \tau\frac{\partial a}{\partial t} + \frac{\tau^2}{2}\frac{\partial^2 a}{\partial t^2} + \mathcal{O}(\tau^3)$$

- Generally, for a flux-conserving equations: $\frac{\partial a}{\partial t} = -\frac{\partial}{\partial x}F(a)$
  - $F(a) = ca$ for advection equations
- Differentiate both sides: $\frac{\partial^2 a}{\partial t^2} = -\frac{\partial}{\partial x}\frac{\partial F(a)}{\partial t}$

- Chain rule: $\frac{\partial F}{\partial t} = \frac{dF}{da}\frac{\partial a}{\partial t} = F'(a)\frac{\partial a}{\partial t} = -F'(a)\frac{\partial F}{\partial x}$

Second order expansion

- So, we have: $a(x,t+\tau) \simeq a(x,t) - \tau \frac{\partial F(a)}{\partial x} + \frac{\tau^2}{2} \frac{\partial F'(a)}{\partial x} \frac{\partial F(a)}{\partial x}$

- Now we discretize derivatives:

$$a_i^{n+1} = a_i^n - \tau \frac{F_{i+1} - F_{i-1}}{2h} + \frac{\tau^2}{2h} \left(F'_{i+1/2} \frac{F_{i+1} - F_i}{h} - F'_{i-1/2} \frac{F_i - F_{i-1}}{h} \right)$$

- Where: $F_i \equiv F(a_i^n), \quad F'_{i\pm 1/2} \equiv F'[(a_{i\pm 1}^n + a_i^n)/2]$

- For advection equations, $F_i = c a_i^n, \quad F'_{i\pm 1/2} = c$

Discretized second derivative of a

$$a_i^{n+1} = a_i^n - \frac{c\tau}{2h}(a_{i+1}^n - a_{i-1}^n) + \frac{c^2\tau^2}{2h^2}(a_{i+1}^n + a_{i-1}^n - 2a_i^n)$$

After class tasks

- Homework 2 due today
- Homework 3 posted

- Readings
  - Garcia Chapters 6 and 7