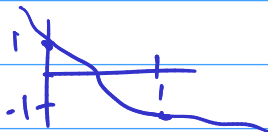
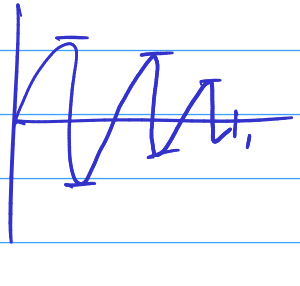
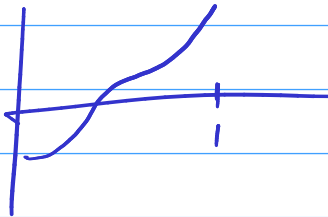
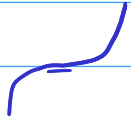
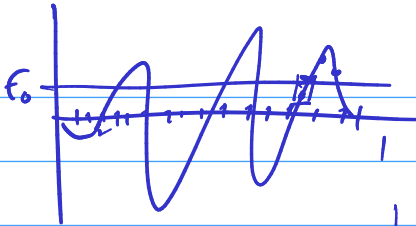


$$K_{\text{abs}} = \max_{x \in [0,1]} \frac{\text{absolute error of output}}{\text{absolute error in input}}$$

$$= \max_{x \in [0,1]} \frac{|f(x+h) - f(x)|}{|h|}$$

$$\approx \max_{x \in [0,1]} |f'(x)|$$





# Modelling the World with Matrices

---

Predicting Movie

vectors:  $x^{(i)}$  - preferences of  
friend  $i$

$$A = \begin{bmatrix} \text{mov. 1 ctr} & \text{ctr} \\ \text{mov. 2 ctr} & \text{ctr} \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}$$

Popularity

Goal:

$$y = Ax$$

$$y = \sum_{i=1}^n A x^{(i)}$$

$$Y = \begin{bmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{bmatrix} = A \cdot \begin{bmatrix} x^{(1)} & x^{(2)} & \dots & x^{(n)} \end{bmatrix}$$

$Y = A \cdot X$  Part 1: given  $A, X$

$y_i = \sum_{j=1}^n A_{ij} x_j$  Part 2: given  $A, Y$

$A_{ij}$  =  $j$ th attribute of the  $i$ th movie

$X_{jk}$  = preference of friend  $k$  with respect to attribute  $j$

1a. solve

# Outline

Python, Numpy, and Matplotlib  
Making Models with Polynomials  
Making Models with Monte Carlo

Error, Accuracy and Convergence  
Floating Point

## Modeling the World with Arrays

The World in a Vector  
What can Matrices Do?  
Graphs  
Sparsity

Norms and Errors  
The 'Undo' Button for Linear Operations: LU

LU: Applications  
Linear Algebra Applications  
Interpolation

Repeating Linear Operations:  
Eigenvalues and Steady States  
Eigenvalues: Applications  
Approximate Undo: SVD and Least Squares

## SVD: Applications

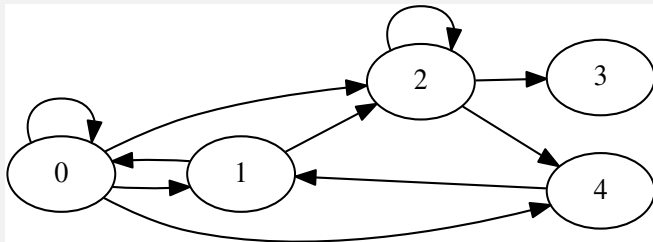
Solving Funny-Shaped Linear Systems  
Data Fitting  
Norms and Condition Numbers  
Low-Rank Approximation

## Iteration and Convergence

Solving One Equation  
Solving Many Equations  
Finding the Best: Optimization in 1D  
Optimization in  $n$  Dimensions

# Graphs as Matrices

How could this (directed) graph be written as a matrix?



$$A = \begin{matrix} \text{\#Vertex} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}^T \quad A_{ji} = 1 \text{ if } \exists (i,j) \in E$$

$G = (V, E)$

## Matrices for Graph Traversal: Technicalities

What is the general rule for turning a graph into a matrix?

$A_{ji} = 1$  if there is an edge from  $i$  to  $j$

What does the matrix for an *undirected* graph look like?

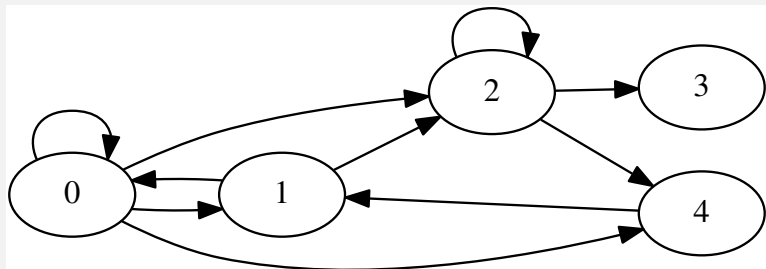
symmetric

How could we turn a *weighted graph* (i.e. one where the edges have weights—maybe 'pipe widths') into a matrix?

$A_{ji}$  is the weight of edge  $i \rightarrow j$

## Graph Matrices and Matrix-Vector Multiplication

If we multiply a graph matrix by the  $i$ th unit vector, what happens?



$$A \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = ? = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



## Demo: Matrices for Graph Traversal

# Outline

Python, Numpy, and Matplotlib  
Making Models with Polynomials  
Making Models with Monte Carlo

Error, Accuracy and Convergence  
Floating Point

## Modeling the World with Arrays

The World in a Vector  
What can Matrices Do?  
Graphs  
Sparsity

Norms and Errors  
The 'Undo' Button for Linear Operations: LU

LU: Applications  
Linear Algebra Applications  
Interpolation

Repeating Linear Operations:  
Eigenvalues and Steady States  
Eigenvalues: Applications  
Approximate Undo: SVD and Least Squares

## SVD: Applications

Solving Funny-Shaped Linear Systems  
Data Fitting  
Norms and Condition Numbers  
Low-Rank Approximation

## Iteration and Convergence

Solving One Equation  
Solving Many Equations  
Finding the Best: Optimization in 1D  
Optimization in  $n$  Dimensions

## Storing Sparse Matrices

Some types of matrices (including graph matrices) contain many zeros.

Storing all those zero entries is wasteful.

How can we store them so that we avoid storing tons of zeros?

## Storing Sparse Matrices Using Arrays

How can we store a sparse matrix using just arrays? For example:

$$\begin{pmatrix} 0 & 2 & 0 & 3 \\ 1 & 4 & & \\ & & 5 & \\ 6 & & & 7 \end{pmatrix}$$