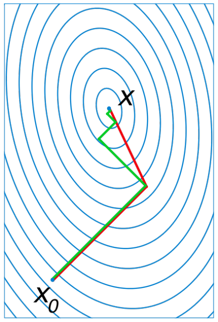
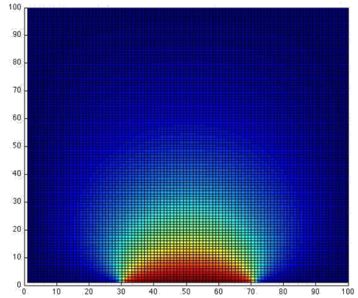


Lecture 08:

# Iterative Solvers of Ax = b

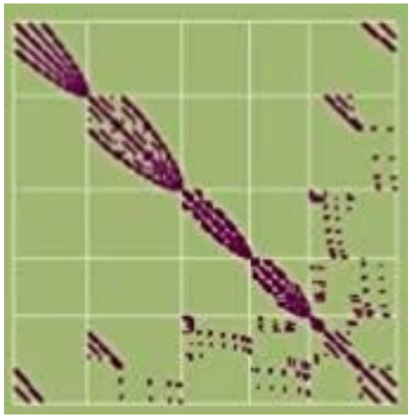
# The Jacobi and CG Methods 1

CS 111: Intro to Computational Science

Spring 2023


Ziad Matni, Ph.D.

Dept. of Computer Science, UCSB

Carl Gustav Jacob Jacobi (1804 – 1851)

# Administrative

- New homework…!
- Lab tomorrow

- Quiz 2 results will be out on Canvas today
  - Median grade = 8/10 Average = 7.9/10

- Preliminary slides for this lecture available on Canvas

# $Ax = b$ Solvers We've Learned So Far

- Solving $\mathbf{Ax = b}$ can be easier to do if we first "decompose" **A**:
  - LU Factorization  P.**A** = L.U  *Useful for most matrices*
  - Cholesky  **A** = R.$R^T$  *Useful for SPD matrices*
  - QR Factorization  **A** = Q.R  *Useful for real matrices*

New **iterative methods** we'll learn:

- *Jacobi Method*
- *Conjugate Gradient Method*

# Jacobi Method

An **iterative** algorithm for determining
the solutions of a ***diagonally dominant*** system of linear equations.

- **Diagonally dominant:**
  - for every row of the matrix, the *absolute value* of the diagonal entry in a row **is greater than or equal to** the *sum* of *absolute values* of all the other (non-diagonal) entries in that row

# Diagonally Dominant Example

For example,

$$\mathbf{A} = \begin{bmatrix} -4 & 2 & 1 \\ 1 & 6 & -2 \\ -1 & -3 & -5 \end{bmatrix}$$

is a ***strictly*** diagonally dominant matrix

because:

$|a_{00}| > |a_{01}| + |a_{02}|$ , that is, $4 > (2 + 1)$

$|a_{11}| > |a_{10}| + |a_{12}|$ , that is, $6 > (1 + 2)$

$|a_{22}| > |a_{20}| + |a_{21}|$ , that is, $5 > (1 + 3)$

# Jacobi Method

- Simple, programmable, efficient
  - Not always the most *accurate* results (but we can get them *fast*)
  - Also, not the most efficient iterative algorithm out there

- Given a system: A**x** = b, solve for **x**
- We can start with a *guess* for **x**
  - Good starting point is the **zero vector** (i.e. x = [0, 0, 0, …] )
  - We'll call it the **$0^{th}$ approximation**, or $\mathbf{x}^{(0)}$
  - We'll ***iterate*** some process to keep guessing for a better and better solution
  - Our *iteration* will lead to a ***convergence*** of $\mathbf{x}^{(n)}$ to an approximate correct answer
    - But matrix **A** really needs to be "strictly diagonally dominant"

# Jacobi Method

**CALCULATION DEMO**
**Coming up!!!**

## Given a current ($k^{th}$) approximation for **x** as:

$$x^{(k)} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \end{bmatrix}$$

- Find the $(k+1)^{th}$ solution.
- Keep repeating (iterating) until your "error" is "small enough"
  - Would indicate a *convergence* to the correct value
  - You want to see the error decrease as you keep iterating
  - In a real application, you don't always know if you're close to an answer, but in class, we'll "cheat" and compare with the "ideal" **x**.
- We'll do an "error analysis" with each iteration
  - To see the difference of the norms w.r.t. the ideal answer
  - We'll want to define a "threshold" of an acceptable stop-point for convergence

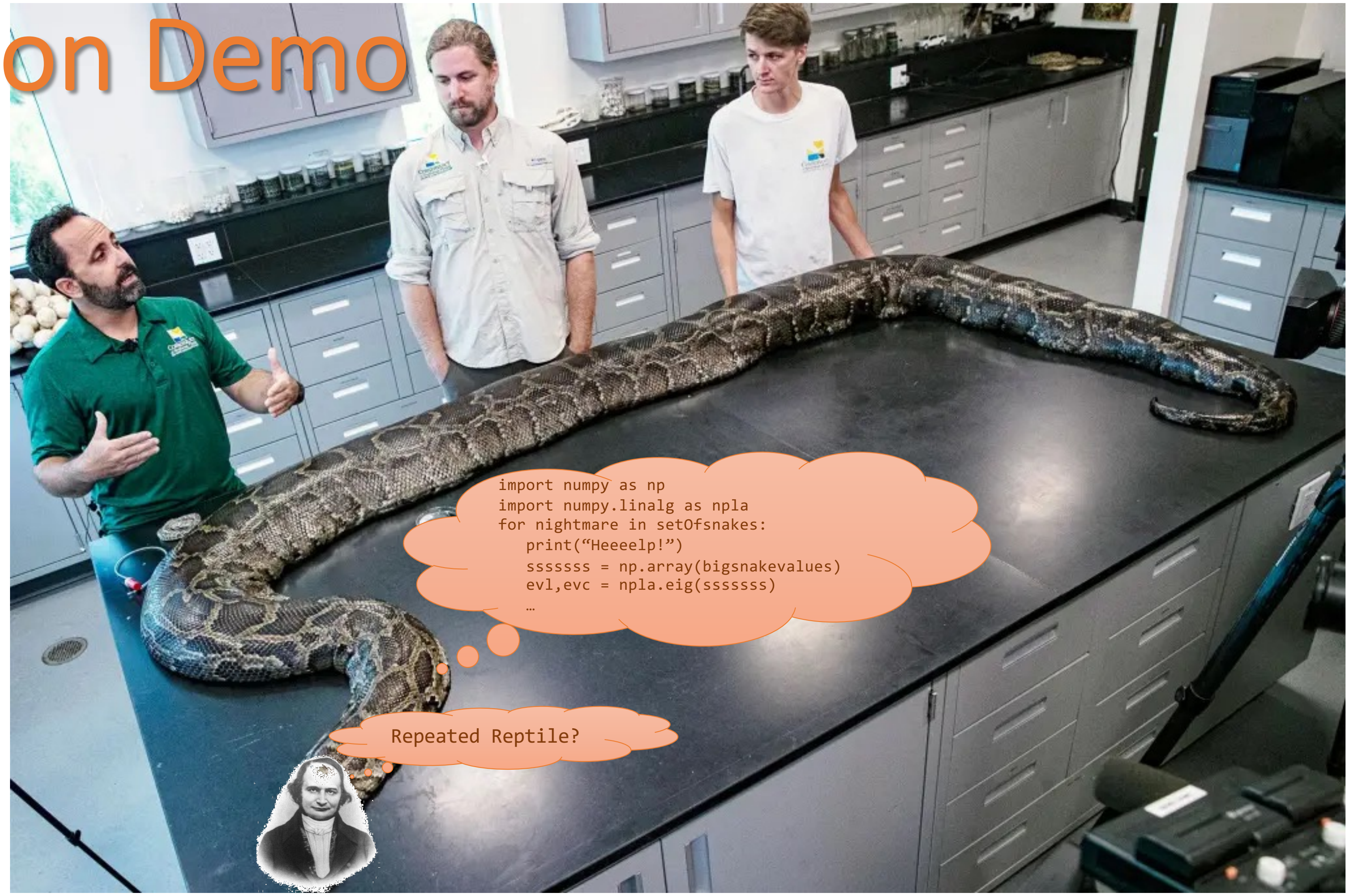
# Jacobi Method

- Strict row diagonal dominance is a **sufficient** *but not necessary* condition for Jacobi's Method to converge
  - It *sometimes* converges even if this condition is not satisfied (though it'll do so much slower)

- The standard convergence condition (for any iterative method) is when the ***spectral radius*** of the iteration matrix is less than 1
  - Written as: $\rho(\mathbf{D}^{-1}.\mathbf{C}) < 1$
  - It means that the **max. eigenvalue** of $\mathbf{D}^{-1}.\mathbf{C}$ should be $< 1$

- A nice write-up to read (optional) on Wikipedia: *https://en.wikipedia.org/wiki/Jacobi_method*

# Python Demo






*Credit: The News-Press-USA TODAY NETWORK*

# Your TO DOs!

- Work on your homework!

# </LECTURE>