CV5100 – MUDE
Modeling, Uncertainty, and Data for Engineers
Ch4 – Linear Algebra

Course instructors:

Prof. Phanisri Pradeep Pratapa

Prof. Prakash Singh Badal

Prof. Sudheendra Herkal

Department of Civil Engineering

Indian Institute of Technology Madras

27 August 2025

Numerical Modelling, Linear Algebra, Optimization

Date	Day	Week	Lecture#	Topic	Topic summary
28-Jul-25	Mon	1			NO CLASS
29-Jul-25	Tue	1			NO CLASS
30-Jul-25	Wed	1	1	Introduction lecture	motivation, course content, logistics, etc.
31-Jul-25	Thu	1	2	Modelling Concepts	sudheendra's part, course logistics; model classification
4-Aug-25	Mon	2	Asst-0		**Software installation & github**
5-Aug-25	Tue	2	3	Numerical Modelling	model decisions, verification vs. validation; differential equations in structural engg, examples, ODE types (linear, nonlinear, order)
6-Aug-25	Wed	2	4	Numerical Modelling	Analytical vs. Numerical solutions, code, algorithm
7-Aug-25	Thu	2	5	Numerical Modelling	Numerical derivative/Finite Differences, Taylor series, error, convergence, forward, backward difference
11-Aug-25	Mon	3	Asst-1		**Finite difference, convergence rate, newton-raphson for finding roots (solving an equation)**
12-Aug-25	Tue	3	6	Numerical Modelling	Central finite diff (1st &2nd order); Numerical integration; IVP example, accuracy/error, convergence,
13-Aug-25	Wed	3	7	Numerical Modelling	stability, explicit vs. implicit; BVP -- e.g. analytical, numerical/FDM, code implementation
14-Aug-25	Thu	3	8	Numerical Modelling	BVP matrix eqn; PDEs - types, gradient and laplacian operators,
18-Aug-25	Mon	4	Asst-2		**IVP & BVP numerical solutions - explicit vs implicit and implementing matrix eq solving using inverse**
19-Aug-25	Tue	4	9	Linear Algebra	Vector spaces, span, linear dependence, basis, dimension, examples, tensor vs. matrix
20-Aug-25	Wed	4	10	Linear Algebra	System of linear eqns, matrix form, solution approach-direct
21-Aug-25	Thu	4	11	Linear Algebra	matrix eqns solutions approach - iterative methods
25-Aug-25	Mon	5	Asst-3		**--**
26-Aug-25	Tue	5	-	Linear Algebra	**NO CLASS**
28-Aug-25	Thu	5	12	Linear Algebra	Eigenvalueproblem, solution approaches, compliexity/scaling; parallel computing?
29-Aug-25	Fri	5	13		Quiz-1 @ 8am
1-Sep-25	Mon	6	Asst-4		**Direct vs Iterative method implementation and comparison; convergence, error plot etc.; Eig vs Eigs**
2-Sep-25	Tue	6	14	Optimization	Classification, Mathematical formulations, standard form, key concepts
3-Sep-25	Wed	6	15	Optimization	Gradient based approaches
4-Sep-25	Thu	6	16	Optimization	Non-gradient approaches
8-Sep-25	Mon	7	Asst-5		
9-Sep-25	Tue	7	17	Uncertainty and Estimation	random variables (rv), covariance, correlation; goodness of fit concepts

Outline

- Vector space concepts
- System of linear equations
- Solution using direct methods
- Solution using indirect (iterative) methods
- Eigenvalue problem
- Orthogonality and orthogonalization

Vector space concepts

- Vector (Linear) space and subspace
- Span
- Linear dependence/independence
- Basis
- Dimension
- Orthogonal vs. non-orthogonal basis
- Column space, null space, rank

System of Linear Equations

- Algebraic to Matrix form

$$
\begin{aligned}
y_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\
y_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\
\vdots & \qquad\qquad\qquad \vdots \\
y_m &= a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n
\end{aligned}
\qquad
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
$$

$$\mathrm{y} = \mathrm{Ax}$$

$$\mathrm{y} = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} x_2 + \cdots + \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} x_n$$

- Give an example of linear equation in structural engineering

y is an element of the range space of A: $\mathrm{y} \in \mathcal{R}(\mathrm{A})$

Solution using direct methods

- Gaussian Elimination

Elementary row operations to get row echelon form:
1. Interchanging two rows.
2. Multiplying a row by a non-zero scalar.
3. Adding a scalar multiple of one row to another.

$$\begin{aligned} 2x + y - z &= 8 \qquad (L_1) \\ -3x - y + 2z &= -11 \qquad (L_2) \\ -2x + y + 2z &= -3 \qquad (L_3) \end{aligned}$$

<table>
<tr><th>System of equations</th><th>Row operations</th><th>Augmented matrix</th></tr>
<tr><td>$\begin{aligned} 2x + y - z &= 8 \\ -3x - y + 2z &= -11 \\ -2x + y + 2z &= -3 \end{aligned}$</td><td></td><td>$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right]$</td></tr>
<tr><td>$\begin{aligned} 2x + y - z &= 8 \\ \tfrac{1}{2}y + \tfrac{1}{2}z &= 1 \\ 2y + z &= 5 \end{aligned}$</td><td>$L_2 + \tfrac{3}{2}L_1 \to L_2$
$L_3 + L_1 \to L_3$</td><td>$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & \tfrac{1}{2} & \tfrac{1}{2} & 1 \\ 0 & 2 & 1 & 5 \end{array}\right]$</td></tr>
<tr><td>$\begin{aligned} 2x + y - z &= 8 \\ \tfrac{1}{2}y + \tfrac{1}{2}z &= 1 \\ -z &= 1 \end{aligned}$</td><td>$L_3 + -4L_2 \to L_3$</td><td>$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & \tfrac{1}{2} & \tfrac{1}{2} & 1 \\ 0 & 0 & -1 & 1 \end{array}\right]$</td></tr>
<tr><td>$\begin{aligned} 2x + y &= 7 \\ \tfrac{1}{2}y &= \tfrac{3}{2} \\ -z &= 1 \end{aligned}$</td><td>$L_1 - L_3 \to L_1$
$L_2 + \tfrac{1}{2}L_3 \to L_2$</td><td>$\left[\begin{array}{ccc|c} 2 & 1 & 0 & 7 \\ 0 & \tfrac{1}{2} & 0 & \tfrac{3}{2} \\ 0 & 0 & -1 & 1 \end{array}\right]$</td></tr>
<tr><td>$\begin{aligned} 2x + y &= 7 \\ y &= 3 \\ z &= -1 \end{aligned}$</td><td>$2L_2 \to L_2$
$-L_3 \to L_3$</td><td>$\left[\begin{array}{ccc|c} 2 & 1 & 0 & 7 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$</td></tr>
<tr><td>$\begin{aligned} x &= 2 \\ y &= 3 \\ z &= -1 \end{aligned}$</td><td>$L_1 - L_2 \to L_1$
$\tfrac{1}{2}L_1 \to L_1$</td><td>$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$</td></tr>
</table>

https://en.wikipedia.org/wiki/Gaussian_elimination

Solution using direct methods

- LU decomposition

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

- Decomposes matrix A into:
  - L: Lower triangular matrix
  - U: Upper triangular matrix
- Solves $AX = B$ by solving $LY = B$ and then $UX = Y$.

- Useful for multiple RHS
  e.g.?

- What if A is symmetric?
  LU → LL^T Cholesky

Solution using direct methods

- Gaussian Elimination – Computational efficiency
  - Eliminate elements of first column – nxn ops
  - For every subsequent column reduce n by 1
  - Sum of squares of n natural number – O(n^3)
- Run time scaling
- Memory scaling (storing the matrix); 8 bytes per number in 64-bit (double precision) – 15 decimals
- Triangular matrix – solved sequentially (not efficient for parallel computing)
- Finds the exact solution in a finite number of steps

Eigenvalue problem

- Definition and details
- Computational complexity/scaling
- *eig* vs. *eigs* eigensolvers (numpy vs scipy) – full vs sparse matrices – direct vs indirect/iterative methods
- Inverse and pseudoinverse
- Energy and eigenvalues

- Where do we come across eigenvalues in structural engineering?

Orthogonalization

- Gram-Schmidt process

Given k nonzero linearly-independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ the Gram–Schmidt process defines the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as follows:

$$
\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1, \\
\mathbf{u}_2 &= \mathbf{v}_2 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_2), \\
\mathbf{u}_3 &= \mathbf{v}_3 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_3) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_3), \\
\mathbf{u}_4 &= \mathbf{v}_4 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_3}(\mathbf{v}_4), \\
&\ \ \vdots \\
\mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \mathrm{proj}_{\mathbf{u}_j}(\mathbf{v}_k),
\end{aligned}
$$

The vector projection of a vector $\mathbf{v}$ on a nonzero vector $\mathbf{u}$ is defined

$$\mathrm{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}$$

https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process

Solution using indirect methods

- Jacobi iteration

Let $A\mathbf{x} = \mathbf{b}$ be a square system of n linear equations, where:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

$$\mathbf{A} = \mathbf{D} + \mathbf{R}$$

$$\mathbf{g}(\mathbf{x}) = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x})$$

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \mathbf{x}$$

$$\mathbf{e}(\mathbf{x}) = \mathbf{x} - \mathbf{x}^*$$

$$\mathbf{e}(\mathbf{x}_{k+1}) = \left(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\right)\mathbf{e}(\mathbf{x}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k)$$

$$\|\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\| < 1$$

$$\mathbf{x}_{k+1} = (1-\omega)\mathbf{x}_k + \omega\mathbf{g}(\mathbf{x}_k)$$

$$\|\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{A}\| < 1$$

- Fixed-point iteration
- Residual vs. Error
- L^2 norm
- Condition number
- Spectral radius

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$$

$$\rho(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$$

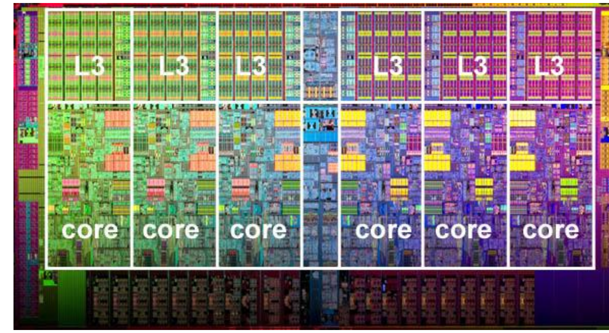
Solution using indirect methods

- Gauss-Seidel Method
- Conjugate Gradient Method
- GMRES
- Preconditioned Conjugate Gradient (PCG)
- Multigrid methods

- Anderson-Jacobi
- Alternating Anderson Jacobi

Pratapa, P. P., Suryanarayana, P., & Pask, J. E. (2016). Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems. *Journal of Computational Physics*, *306*, 43-54.

High Performance & Parallel Computing



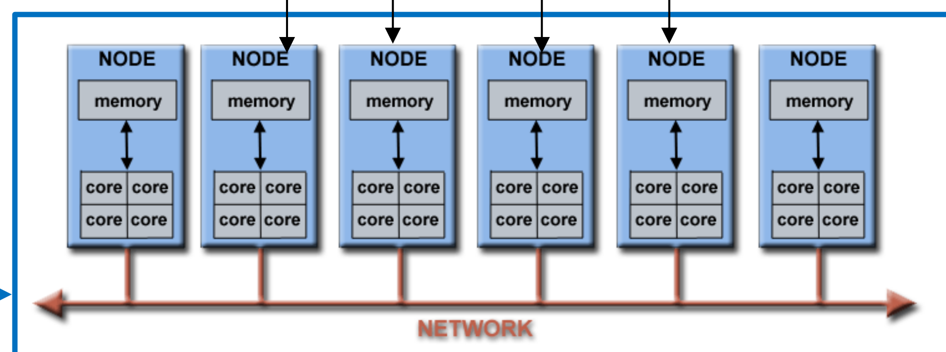
$$b_i = A_{i1}x_1 + A_{i2}x_2 + \ldots + A_{in}x_n = \Sigma_{j=1}^{n} A_{ij}x_j$$

$$\begin{bmatrix} A_{11} & \ldots & A_{1n} \\ A_{21} & \ldots & A_{2n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \ldots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

```
% Given, Matrix A of size n x n.
% Given, Vector x of size n x 1.
% Find, Vector b = A*x.
for i=1:n
    S = 0;
    for j=1:n
        S = S + A(i,j)*x(j);
    end
    b(i) = S;
end
```

Run on multiple computers

Computer Cluster



Ref: https://computing.llnl.gov/tutorials/parallel_comp/

Other things if you are curious

- Scientific/High-performance computing
- Profiling
- BLAS
- LAPACK
- MPI
- PETSc
- Threading
- Parallel scaling, parallel computing
- Cache, retrieval
- Communication
- Petaflops, Exaflops, Supercomputers