REVIEW of Lecture 6

Material covered in class: Differential forms of conservation laws

- Material Derivative (substantial/total derivative)
- Conservation of Mass
  - Differential Approach
  - Integral (volume) Approach
  - Use of Gauss Theorem
  - Incompressibility
- Reynolds Transport Theorem
- Conservation of Momentum (Cauchy’s Momentum equations)
- The Navier-Stokes equations
  - Constitutive equations: Newtonian fluid
  - Navier-stokes, compressible and incompressible
Integral Conservation Law for a scalar $\phi$

\[
\begin{align*}
\frac{d}{dt} \int_{CM} \rho \phi dV &= \frac{d}{dt} \int_{CV_{fixed}} \rho \phi dV + \int_{CS} \rho \phi (\vec{v} \cdot \vec{n}) dA = -\int_{CS} q_\phi \cdot \vec{n} dA + \sum \int_{CV_{fixed}} s_\phi dV
\end{align*}
\]

Advecive fluxes ("convective" fluxes)

Other transports (diffusion, etc)

Sum of sources and sinks terms (reactions, etc)

Applying the Gauss Theorem, for any arbitrary CV gives:

\[
\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi \vec{v}) = -\nabla \cdot \vec{q}_\phi + s_\phi
\]

For a common diffusive flux model (Fick’s law, Fourier’s law):

\[
\vec{q}_\phi = -k \nabla \phi
\]

Conservative form of the PDE:

\[
\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi \vec{v}) = \nabla \cdot (k \nabla \phi) + s_\phi
\]
Strong-Conservative form of the Navier-Stokes Equations ($\phi \Rightarrow \mathbf{v}$)

Cons. of Momentum:

$$\frac{d}{dt} \int_{cv} \rho \mathbf{v} dV + \int_{cs} \rho \mathbf{v} \cdot \mathbf{n} dA = \int_{cs} -p \mathbf{n} dA + \int_{cs} \tau \cdot \mathbf{n} dA + \int_{cv} \rho \mathbf{g} dV$$

Applying the Gauss Theorem gives:

$$= \int_{cv} \left( -\nabla p + \nabla \cdot \mathbf{\tau} + \rho \mathbf{g} \right) dV$$

For any arbitrary CV gives:

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nabla \cdot \mathbf{\tau} + \rho \mathbf{g}$$

With Newtonian fluid + incompressible + constant $\mu$:

Momentum:

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{g}$$

Mass:

$$\nabla \cdot \mathbf{v} = 0$$

Equations are said to be in "strong conservative form" if all terms have the form of the divergence of a vector or a tensor. For the $i^{th}$ Cartesian component, in the general Newtonian fluid case:

With Newtonian fluid only:

$$\frac{\partial \rho v_i}{\partial t} + \nabla \cdot (\rho v_i \mathbf{v}) = \nabla \cdot \left( -p \mathbf{e}_i + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \mathbf{e}_j - \frac{2}{3} \mu \frac{\partial u_j}{\partial x_j} \mathbf{e}_i + \rho g, x_i \mathbf{e}_i \right)$$
Navier-Stokes Equations:
For an Incompressible Fluid with constant viscosity

\[ \mathbf{V}(x,y,z) \]

Fluid Velocity Field

\[ \mathbf{V} = iu + jv + kw \]

Conservation of Mass

\[ \text{div}\mathbf{V} = \nabla \cdot \mathbf{V} = 0 \]
\[ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \]

Navier-Stokes Equation

\[ \frac{D\mathbf{V}}{Dt} = \frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{V} \]

Hydrostatic Pressure:
\[ -\rho g z \text{ for } z \text{ positive upward} \]

Dynamic Pressure \( P = P_{\text{actual}} - \rho gz \)

Kinematic viscosity \( \nu \)

Density \( \rho \)

\[
\begin{align*}
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \\
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= -\frac{1}{\rho} \frac{\partial P}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \\
\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= -\frac{1}{\rho} \frac{\partial P}{\partial z} + \nu \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right)
\end{align*}
\]
Incompressible Fluid Pressure Equation

Navier-Stokes Equation

\[
\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{V}
\]

Conservation of Mass

\[ \nabla \cdot \mathbf{V} = 0 \]

Divergence of Navier-Stokes Equation

\[ \text{div}(\mathbf{V} \cdot \nabla \mathbf{V}) = -\frac{1}{\rho} \nabla^2 P \]

Dynamic Pressure Poisson Equation

\[
\Rightarrow \nabla^2 P = -\rho \left\{ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 + 2 \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} + 2 \frac{\partial w}{\partial x} \frac{\partial u}{\partial z} + 2 \frac{\partial w}{\partial y} \frac{\partial v}{\partial z} \right\}
\]

More general than Bernoulli – Valid for unsteady and rotational flow
Incompressive Fluid
Vorticity Equation

Vorticity

\[ \tilde{\omega} \equiv \text{curl} \mathbf{V} \equiv \nabla \times \mathbf{V} \]

Navier-Stokes Equation

\[ \frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{V} \]

curl of Navier-Stokes Equation

\[ \frac{D \tilde{\omega}}{Dt} = - (\tilde{\omega} \cdot \nabla) \mathbf{V} + \nu \nabla^2 \tilde{\omega} \]
Inviscid Fluid Mechanics

Euler's Equation

Navier-Stokes Equation: incompressible, constant viscosity

\[
\frac{\partial V}{\partial t} + (V \cdot \nabla)V = -\frac{1}{\rho}\nabla P + \nu \nabla^2 V
\]

If also inviscid fluid

\[\nu = 0\]

⇒ Euler's Equations

\[
\begin{align*}
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= -\frac{1}{\rho} \frac{\partial P}{\partial x} \\
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= -\frac{1}{\rho} \frac{\partial P}{\partial y} \\
\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= -\frac{1}{\rho} \frac{\partial P}{\partial z}
\end{align*}
\]
Inviscid Fluid Mechanics
Bernoulli Theorems

**Theorem 1**

Irrotational Flow, incompressible

\[ \nabla \times \mathbf{V} = 0 \]

Flow Potential

\[ \mathbf{V} = \nabla \phi \]

Define

\[ H = \frac{1}{2} |\mathbf{V}|^2 + \frac{P}{\rho} \]

\[ \frac{\partial \phi}{\partial t} + H = 0 \]

Introduce \( P_T \) = Thermodynamic pressure

\[ P_T = P - \rho g z \]

\[ \frac{\partial \phi}{\partial t} + \frac{1}{2} |\mathbf{V}|^2 + \frac{P_T}{\rho} + g z = 0 \]

**Theorem 2**

Steady, Incompressible, inviscid, no shaft work, no heat transfer

Navier-Stokes Equation

\[ \frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{1}{\rho} \nabla P \]

\[ \mathbf{V} \times \mathbf{\omega} = \nabla H \]

Along stream lines and vortex lines

\[ H = \frac{1}{2} |\mathbf{V}|^2 + \frac{P}{\rho} \]

\[ = \frac{1}{2} |\mathbf{V}|^2 + \frac{P_T}{\rho} + g z = \text{const} \]
Potential Flows
Integral Equations

Irrotational Flow
\[ \nabla \times \mathbf{V} = 0 \]
Flow Potential
\[ \mathbf{V} = \nabla \phi \]
Conservation of Mass
\[ \nabla \cdot \mathbf{V} = 0 \]
\[ \Rightarrow \]
\[ \nabla \cdot (\nabla \phi) = 0 \]
\[ \nabla^2 \phi = 0 \]

"Mostly" Potential Flows:
Only rotation occurs at boundaries due to viscous terms

In 2D:
Velocity potential:
\[ u = \frac{\partial \phi}{\partial x}, \quad v = \frac{\partial \phi}{\partial y} \]
Stream function:
\[ u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \]

• Since Laplace equation is linear, it can be solved by superposition of flows, called panel methods
• What distinguishes one flow from another are the boundary conditions and the geometry: there are no intrinsic parameters in the Laplace equation
Potential Flow
Boundary Integral Equations

Green’s Theorem

\[
\int_S \left[ G(x, x_0) \frac{\partial \phi(x_0)}{\partial n} - \phi(x_0) \frac{\partial G(x, x_0)}{\partial n} \right] dS_0 = \int_V \left[ \phi(x_0) \nabla^2 G(x, x_0) - G(x, x_0) \nabla^2 \phi(x_0) \right] dV_0
\]

Green’s Function

\[
G(x, x_0) = \frac{1}{r} = \frac{1}{\sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}} + \psi(x)
\]

Homogeneous Solution

\[
\nabla^2 \psi = 0
\]

\[
\nabla^2 G((x, x_0) = -\delta(x - x_0)
\]

Boundary Integral Equation

\[
\phi(x) = \int_S \left[ G(x, x_0) \frac{\partial \phi(x_0)}{\partial n} - \phi(x_0) \frac{\partial G(x, x_0)}{\partial n} \right] dS_0 - \int_V [G(x, x_0) \nabla^2 \phi(x_0)] dV_0
\]

Discretized Integral Equation

\[
\sum_{j=0}^{N-1} A_{ij} w_j = B_i
\]

Linear System of Equations

\[
\bar{A} \mathbf{u} = \mathbf{b}
\]

Panel Methods
Systems of Linear Equations

• Motivation and Plans

• Direct Methods for solving Linear Equation Systems
  – Cramer’s Rule (and other methods for a small number of equations)
  – Gaussian Elimination
  – Numerical implementation
    • Numerical stability
      – Partial Pivoting
      – Equilibration
      – Full Pivoting
  – Multiple right hand sides, Computation count
  – LU factorization
  – Error Analysis for Linear Systems
    – Condition Number
  – Special Matrices: Tri-diagonal systems

• Iterative Methods
  – Jacobi’s method
  – Gauss-Seidel iteration
  – Convergence
Motivations and Plans

- Fundamental equations in engineering are conservation laws (mass, momentum, energy, mass ratios/concentrations, etc)
  - Can be written as “System Behavior (state variables) = forcing”
- Result of the discretized (volume or differential form) of the Navier-Stokes equations (or most other differential equations):
  - System of (mostly coupled) algebraic equations which are linear or nonlinear, depending on the nature of the continuous equations
  - Often, resulting matrices are sparse (e.g. banded and/or block matrices)
- Lectures 3 and 4:
  - Methods for solving \( f(x) = 0 \) or \( f(x) = 0 \)
  - To be used for systems of equations: \( f(x) = b \), i.e: \( f = (f_1(x), f_2(x), ..., f_n(x)) = b \)
- Here we first deal with solving Linear Algebraic equations:

\[
A \mathbf{x} = b \quad \text{or} \quad A \mathbf{X} = \mathbf{B}
\]
Motivations and Plans

• Above 75% of engineering/scientific problems involve solving linear systems of equations
  – As soon as methods were used on computers => dramatic advances
• Main Goal: Learn methods to solve systems of linear algebraic equations and apply them to CFD applications
• Reading Assignment
  – For Matrix background, see Chapra and Canale (pg 219-227) and other linear algebra texts (e.g. Trefethen and Bau, 1997)
• Other References:
  – Any chapter on “Solving linear systems of equations” in CFD references provided.
Direct Numerical Methods
for Linear Equation Systems

\[ Ax = b \quad \text{or} \quad AX = B \]

- Main Direct Method is: Gauss Elimination
  Key idea is simply to “combine equations so as to eliminate unknowns”
- First, let’s consider systems with a small number of equations
  - Graphical Methods
    - Two equations: intersection of 2 lines
    - Three equations: intersection of 3 planes
    - Useful to illustrate issues:
      no solution / infinite solutions (singular) or ill-conditioned system

\[ \begin{align*}
- \frac{1}{2} x_1 + x_2 &= 1 \\
- \frac{1}{2} x_1 + x_2 &= \frac{1}{2} \\
- \frac{1}{2} x_1 + x_2 &= 1 \\
- x_1 + 2x_2 &= 2 \\
- \frac{2}{5} x_1 + x_2 &= 1.1 \\
- \frac{1}{2} x_1 + x_2 &= 1
\end{align*} \]
Direct Methods for Small Systems: Determinants and Cramer’s Rule

Linear System of Equations:

\[ a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \]
\[ \vdots \]
\[ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \]

Recall, for a 2 by 2 matrix, the determinant is:

\[ D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12} \]

Recall, for a 3 by 3 matrix, the determinant is:

\[ D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \]
Direct Methods for small systems: Determinants and Cramer’s Rule

Cramer’s rule:

“Each unknown $x_i$ in a system of linear algebraic equations can be expressed as a fraction of two determinants:

- Numerator is $D$ but with column $i$ replaced by $b$
- Denominator is $D$

\[
x_i = \frac{D_i}{D}
\]

Example: Cramer’s Rule, $n=2$

\[
A = \begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}
\]

\[
D = a_{11}a_{22} - a_{21}a_{12}
\]

\[
D_1 = b_{1}a_{22} - b_{2}a_{12}
\]

\[
D_2 = b_{2}a_{11} - b_{1}a_{21}
\]

\[
x_1 = \frac{D_1}{D} = \frac{b_{1}a_{22} - b_{2}a_{12}}{a_{11}a_{22} - a_{21}a_{12}}
\]

\[
x_2 = \frac{D_2}{D} = \frac{b_{2}a_{11} - b_{1}a_{21}}{a_{11}a_{22} - a_{21}a_{12}}
\]

Numerical case:

\[
\begin{bmatrix}
0.01 & -1.0 \\
1.0 & 0.01
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= \begin{bmatrix}
1.0 \\
1.0
\end{bmatrix}
\]

\[
x_1 = \frac{1.0 - 0.01 - 1.0(-1.0)}{0.01 - 0.01 - 1.0(-1.0)} = 1.0099
\]

\[
x_2 = \frac{1.0 - 0.01 - 1.0(1.0)}{0.01 - 0.01 - 1.0(-1.0)} = -0.9899
\]

Cramer’s rule becomes impractical for $n>3$:
The number of operations is of $O(n!)$
Direct Methods for large dense systems

**Gauss Elimination**

- **Main idea:** “combine equations so as to eliminate unknowns systematically”
  - Solve for each unknown one by one
  - Back-substitute result in the original equations
  - Continue with the remaining unknowns

**Linear System of Equations**

\[
\begin{align*}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
\vdots & \vdots \\
\vdots & \vdots \\
a_{n1}x_1 + \cdots + a_{nn}x_n &= b_n
\end{align*}
\]

**General Gauss Elimination Algorithm**

i. **Forward Elimination/Reduction to Upper Triangular Systems**

ii. **Back-Substitution**

**Comments:**

- Well suited for dense matrices
- Some modification of above simple algorithm needed to avoid division by zero and other pitfalls
Gauss Elimination

Linear System of Equations

\[ a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n = b_1 \]
\[ a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n = b_2 \]
\[ \vdots \quad \vdots \quad \cdots \quad = \quad \vdots \]
\[ a_{n1} x_1 + \cdots + a_{nn} x_n = b_n \]

If \( a_{11} \) is non zero, we can eliminate \( x_1 \) from the remaining equations 2 to \((n-1)\) by multiplying equation 1 with \( \frac{a_{i1}}{a_{11}} \) and subtracting the result from equation \( i \).

This leads to the following algorithm for “Step 1”:

\[ a_{i1}^{(1)} x_1 + a_{i2}^{(1)} x_2 + \cdots + a_{in}^{(1)} x_n = b_1^{(1)} \]
\[ a_{21}^{(1)} x_1 + a_{22}^{(1)} x_2 + \cdots + a_{2n}^{(1)} x_n = b_2^{(1)} \]
\[ \vdots \quad \vdots \quad \cdots \quad = \quad \vdots \]
\[ a_{n1}^{(1)} x_1 + \cdots + a_{nn}^{(1)} x_n = b_n^{(1)} \]

Reduction / Forward Elimination

Step 0

\[ a_{ij}^{(1)} = a_{ij}, \quad b_i^{(1)} = b_i \]
Gauss Elimination

Reduction / Forward Elimination

Step 1

\[
\begin{align*}
    m_{i1} &= \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} \\
    a_{ij}^{(2)} &= a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)}, \quad j = 1, \ldots, n \\
    b_i^{(2)} &= b_i^{(1)} - m_{i1}b_1^{(1)}
\end{align*}
\]

\(i = 2, \ldots, n\)

\(a_{11}\) is called pivot element:

\[
\begin{bmatrix}
    a_{11}^{(1)}x_1 \\
    a_{12}^{(1)}x_2 \\
    \vdots \\
    a_{1n}^{(1)}x_n
\end{bmatrix} = \begin{bmatrix}
    b_1^{(1)} \\
    b_2^{(1)} \\
    \vdots \\
    b_n^{(1)}
\end{bmatrix}
\]

(is called Pivot equation for step 1)

Notes:

- Result of step 1: last (n-1) equations have (n-1) unknowns
- Pivot \(a_{11}\) needs to be non-zero
Gauss Elimination

Reduction: Step k

Recursive repetition of step 1 for successively reduced set of (n-k) equations:

\[
\begin{align*}
  m_{ik} &= \frac{a^{(k)}_{ik}}{a^{(k)}_{kk}} \\
  a^{(k+1)}_{ij} &= a^{(k)}_{ij} - m_{ik}a^{(k)}_{kj}, \quad j = k, \cdots n \\
  b^{(k+1)}_i &= b^{(k)}_i - m_{ik}b^{(k)}_k 
\end{align*}
\]

\(i = k+1, \cdots n\)

The result after completion of step k is:

\[
\begin{bmatrix}
  a^{(1)}_{11} x_1 & a^{(1)}_{12} x_2 & \cdots & a^{(1)}_{1n} x_n = b^{(1)}_1 \\
  0 & a^{(2)}_{22} x_2 & \cdots & a^{(2)}_{2n} x_n = b^{(2)}_2 \\
  0 & 0 & a^{(k)}_{kk} x_k & \cdots = \cdot \\
  0 & 0 & 0 & \cdots = \cdot \\
  0 & 0 & 0 & \cdots = a^{(k+1)}_{nn} x_n = b^{(k+1)}_n
\end{bmatrix}
\]

First non-zero element on row n: \(a^{(k+1)}_{n,k+1} x_k\)
Gauss Elimination

Reduction/Elimination: Step k

\[
\begin{align*}
  m_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\
  a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad j = k, \ldots, n \\
  b_{i}^{(k+1)} &= b_{i}^{(k)} - m_{ik} b_{k}^{(k)} \\
\end{align*}
\]

\( i = k+1, \ldots, n \)

Reduction: Step (n-1)

\[
\begin{pmatrix}
  a_{11}^{(1)} x_1 & a_{12}^{(1)} x_2 & \cdots & a_{1n}^{(1)} x_n = b_1^{(1)} \\
  0 & a_{22}^{(2)} x_2 & \cdots & a_{2n}^{(2)} x_n = b_2^{(2)} \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & \cdots & 0 & a_{n-1,n-1}^{(n-1)} x_{n-1} + a_{n-1,n}^{(n-1)} x_n = b_{n-1}^{(n-1)} \\
  0 & \cdots & 0 & a_{nn}^{(n)} x_n = b_n^{(n)}
\end{pmatrix}
\]

Result after step (n-1) is an Upper triangular system!

Back-Substitution

\[
\begin{align*}
  x_n &= b_n^{(n)}/a_{nn}^{(n)} \\
  x_{n-1} &= \frac{b_{n-1}^{(n-1)} - a_{n-1,n}^{(n-1)} x_n}{a_{n-1,n-1}^{(n-1)}} \\
  &\quad \vdots \\
  x_k &= \frac{b_k^{(k)} - \sum_{j=k+1}^{n} a_{kj}^{(k)} x_j}{a_{kk}^{(k)}} \\
  x_1 &= \frac{b_1^{(1)} - \sum_{j=2}^{n} a_{1j}^{(1)} x_j}{a_{11}^{(1)}}
\end{align*}
\]
Gauss Elimination: Number of Operations

Reduction/Elimination: Step k

\[
\begin{align*}
    m_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\
    a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad j = k, \ldots n \tag{i = k+1, \ldots n} \\
    b_{i}^{(k+1)} &= b_{i}^{(k)} - m_{ik} b_{k}^{(k)}
\end{align*}
\]

: n-k divisions

: 2 (n-k) (n-k+1) additions/multiplications

: 2 (n-k) additions/multiplications

For reduction, total number of ops is:

\[
\sum_{k=1}^{n-1} 3(n-k) + 2(n-k) * (n-k+1) = 2\left(\frac{n^3}{3} + n^2 - \frac{n}{3}\right) = O(n^3)
\]

Use:

\[
\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \quad \text{and} \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}
\]

Back-Substitution

\[
x_{k} = \left(b_{k}^{(k)} - \sum_{j=k+1}^{n} a_{k}^{(k)} x_{j}\right) / a_{kk}^{(k)}
\]

: (n-k-1)+(n-k)+2=2(n-k)+1 additions/multiplications

Hence, total number of ops is:

\[
1 + \sum_{k=1}^{n-1} (2(n-k) + 1) = 1 + (n-1)(n+1) = n^2 \quad \left(\sum_{i=1}^{n} i = \frac{n(n+1)}{2}\right)
\]

Grand total number of ops is \(O\left(\frac{2}{3} n^3\right) = O(n^3)\)

- Grows rapidly with n

- Most ops occur in elimination step
Gauss Elimination:
Issues and Pitfalls to be addressed

• Division by zero:
  – Pivot elements $a_{k,k}^{(k)}$ must be non-zero and should not be close to zero

• Round-off errors
  – Due to recursive computations and so error propagation
  – Important when large number of equations are solved
  – Always substitute solution found back into original equations
  – Scaling of variables can be used

• Ill-conditioned systems
  – Occurs when one or more equations are nearly identical
  – If determinant of normalized system matrix $\mathbf{A}$ is close to zero, system will be ill-conditioned (in general, if $\mathbf{A}$ is not well conditioned)
  – Determinant can be computed using Gauss Elimination
    • Since forward-elimination consists of simple scaling and addition of equations, the determinant is the product of diagonal elements of the Upper Triangular System
Gauss Elimination: Pivoting

Step k

\[
\begin{align*}
\forall i, j \in \{1, \ldots, n\}, \quad & m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\
& a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)} \\
& b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}
\end{align*}
\]

\[i = 2, \ldots, n\]

Partial Pivoting by Columns

\[
\begin{bmatrix}
\times \\
0 & \times \\
\cdot & \cdot & \times \\
\cdot & 0 & \times \times \times \\
\cdot & \cdot & \times \\
\cdot & \cdot & \times \\
0 & 0 & \times
\end{bmatrix}
\]

\[
\bar{x} = \begin{bmatrix}
\times \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\times
\end{bmatrix}
\]

Pivot Elements

\[
a_1^{(1)}, a_2^{(2)}, \ldots, a_n^{(n)}
\]

\[
a_{kk}^{(k)} \neq 0
\]

Required at each step!
Gauss Elimination: Pivoting

**Reduction Step k**

\[
m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}
\]

\[
a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)} , \quad j = k, \ldots, n
\]

\[
b_{i}^{(k+1)} = b_{i}^{(k)} - m_{ik}b_{k}^{(k)}
\]

**Partial Pivoting by Columns**

\[
\begin{bmatrix}
  \times \\
  0 \times \\
  \vdots \times \\
  \cdot \cdot \times \\
  \cdot \cdot \cdot \times \\
  \cdot \cdot \cdot 0 \\
  \cdot \cdot \cdot \times \\
  0 \cdot 0 \times \\
\end{bmatrix}
\]

\[
\bar{x} = \begin{bmatrix}
\times \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\end{bmatrix}
\]

**Pivot Elements**

\[a_{11}^{(1)}, a_{22}^{(2)}, \ldots, a_{nn}^{(n)}\]

Required at each step!

### A. Partial Pivoting

i. Search for largest available coefficient in column below pivot element

ii. Switch rows k and i

### B. Complete Pivoting

i. As for Partial, but search both rows and columns

ii. Rarely done since column re-ordering changes order of x’s, hence more complex code

Two Solutions:
Gauss Elimination: Pivoting Example
(for division by zero but also reduces round-off errors)

Example, n=2

\[
\begin{bmatrix}
0.01 & -1.0 \\
1.0 & 0.01
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
1.0 \\
1.0
\end{bmatrix}
\]

Cramer's Rule - Exact

\[
x_1 = \frac{1.0 \cdot 0.01 - 1.0 \cdot (-1.0)}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = 1.0099
\]

\[
x_2 = \frac{1.0 \cdot 0.01 - 1.0 \cdot 1.0}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = -0.9899
\]

Direct Gaussian Elimination

\[
\begin{bmatrix}
0.01 & -1.0 \\
1.0 & 0.01
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
1.0 \\
1.0
\end{bmatrix} \Rightarrow
\begin{cases}
x_1 = 1.01 \\
x_2 = -0.99
\end{cases}
\]

2-digit Arithmetic

\[
m_{21} = 100
\]

\[
a_{21}^{(2)} = 0
\]

\[
a_{22}^{(2)} = 0.01 + 100 \approx 100
\]

\[
b_{2}^{(2)} = 1 - 100 \approx -100
\]

\[
x_2 = -1
\]

\[
x_1 = (1.0 - 1.0)/0.01 = 0
\]

```
n=2
a = [ [0.01 1.0]'; [-1.0 0.01]']
b= [1 1]'
r=a^(-1) * b
x=[0 0];
m21=a(2,1)/a(1,1);
a(2,1)=0;
a(2,2) = radd(a(2,2),-m21*a(1,2),n);
b(2) = radd(b(2),-m21*b(1),n);
x(2) = b(2)/a(2,2);
x(1) = (radd(b(1), -a(1,2)*x(2),n))/a(1,1);
x'
tbt.m
```

Relatively close to zero

100% error

1% error

100% error

2.29
Gauss Elimination: Pivoting Example
(for division by zero but also reduces round-off errors)

Partial Pivoting
Interchange Rows

Example, n=2

\[
\begin{bmatrix}
0.01 & -1.0 \\
1.0 & 0.01 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix}
= 
\begin{bmatrix}
1.0 \\
1.0 \\
\end{bmatrix}
\]

Cramer’s Rule - Exact

\[
x_1 = \frac{1.0 \cdot 0.01 - 1.0 \cdot (-1.0)}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = \frac{1.00099}{1.0001} \\
x_2 = \frac{1.0 \cdot 0.01 - 1.0 \cdot (-1.0)}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = \frac{-0.9899}{1.0001}
\]

2-digit Arithmetic

\[
m_{21} = 0.01 \\
a_{22}^{(2)} = -1 - 0.001 \approx -1.0 \\
b_2^{(2)} = 1 - 0.01 \approx 1.0 \\
x_2 = -1 \\
x_1 = 1 + 0.01 \approx 1.0
\]

Notes on coding:
- Pivoting can be done in function/subroutine
- Most codes don’t exchange rows, but rather keep track of pivot rows
  (store info in “pointer” vector)

See tbt2.m
Gauss Elimination: Equation Scaling Example
(normalizes determinant, also reduces round-off errors)

Example, n=2

\[ \begin{bmatrix} 0.01 & -1.0 \\ 1.0 & 0.01 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix} \]

Cramer’s Rule - Exact

\[ x_1 = \frac{1.0 \cdot 0.01 - 1.0 \cdot (-1.0)}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = 1.0099 \]
\[ x_2 = \frac{1.0 \cdot 0.01 - 1.0 \cdot 1.0}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = -0.9899 \]

Multiply Equation 1 by 200

\[ \begin{bmatrix} 2.0 & -200 \\ 1.0 & 0.01 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 200.0 \\ 1.0 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.01 \\ -0.99 \end{bmatrix} \]

2-digit Arithmetic

\[ m_{21} = 0.5 \]
\[ a_{21}^{(2)} = 0 \]
\[ a_{22}^{(2)} = 0.01 + 100 \approx 100 \]
\[ b_2^{(2)} = 1 - 0.5 \cdot 200 \approx -100 \]
\[ x_2 = -1 \]
\[ x_1 = \frac{(200 - 200)}{2} = 0 \]

Equations must be normalized for partial pivoting to ensure stability

This Equilibration is made by normalizing the matrix to unit norm

Element-based Infinity-Norm Normalization

\[ \|a_{ij}\|_\infty = \max_j |a_{ij}| \approx 1, \ i = 1, \ldots n \]

Element-based Two-Norm Normalization

\[ \|a_{ij}\|_2 = \sum_{j=1}^n a_{ij}^2 \approx 1, \ i = 1, \ldots n \]
Note: Examples of Matrix Norms

\[ \| A \|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}| \]  “Maximum Column Sum”

\[ \| A \|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}| \]  “Maximum Row Sum”

\[ \| A \|_F = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2} \]  “The Frobenius norm”

\[ \| A \|_2 = \sqrt{\lambda_{\max}(A^*A)} \]  “The l-2 norm”
Gauss Elimination: Full Pivoting Example
(normalizes determinant, also reduces round-off errors)

Example, n=2

\[
\begin{bmatrix}
2.0 & -200 \\
1.0 & 0.01
\end{bmatrix}
\begin{Bmatrix}
x_1 \\
x_2
\end{Bmatrix}
= 
\begin{Bmatrix}
200.0 \\
1.0
\end{Bmatrix}
\]

Cramer’s Rule - Exact

\[
x_1 = \frac{1.0 \cdot 0.01 - 1.0 \cdot (-1.0)}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = 1.0099
\]
\[
x_2 = \frac{1.0 \cdot 0.01 - 1.0 \cdot 1.0}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = -0.9899
\]

Interchange Unknowns

\[
x_1 = \tilde{x}_2
\]
\[
x_2 = \tilde{x}_1
\]

Pivoting by Rows

\[
\begin{bmatrix}
-200 & 2.0 \\
0.01 & 1.0
\end{bmatrix}
\begin{Bmatrix}
\tilde{x}_1 \\
\tilde{x}_2
\end{Bmatrix}
= 
\begin{Bmatrix}
200.0 \\
1.0
\end{Bmatrix}
\Rightarrow 
\begin{Bmatrix}
\tilde{x}_1 = -0.99 \\
\tilde{x}_2 = 1.01
\end{Bmatrix}
\]

2-digit Arithmetic

\[
m_{21} = -0.00005
\]
\[
a_{21}^{(2)} = 0
\]
\[
a_{22}^{(2)} = 0.01 + 1.0 \approx 1
\]
\[
b_2^{(2)} = 1 + 0.01 \approx 1
\]
\[
\tilde{x}_2 \approx 1
\]
\[
\tilde{x}_1 = (200 - 2)/(-200) \approx -1
\]

Full Pivoting
Find largest numerical value in same row and column and interchange
Affects ordering of unknowns (hence rarely done)
Gauss Elimination

Numerical Stability

• Partial Pivoting
  – Equilibrate system of equations
  – Pivoting by Columns
  – Simple book-keeping
    • Solution vector in original order

• Full Pivoting
  – Does not necessarily require equilibration
  – Pivoting by both row and columns
  – More complex book-keeping
    • Solution vector re-ordered

Partial Pivoting is simplest and most common

Neither method guarantees stability due to large number of recursive computations
Gauss Elimination:
Effect of variable transform (variable scaling)

Example, n=2

\[
\begin{bmatrix}
0.01 & -1.0 \\
1.0 & 0.01
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
1.0 \\
1.0
\end{bmatrix}
\]

Cramer’s Rule - Exact

\[
x_1 = \frac{1.0 \cdot 0.01 - 1.0 \cdot (-1.0)}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = 1.0099
\]

\[
x_2 = \frac{1.0 \cdot 0.01 - 1.0 \cdot 1.0}{0.01 \cdot 0.01 - 1.0 \cdot (-1.0)} = -0.9899
\]

Variable Transformation

\[
x_1 = \tilde{x}_1
\]

\[
x_2 = 0.01 \cdot \tilde{x}_2
\]

\[
\begin{bmatrix}
1.0 & -1.0 \\
1.0 & 0.0001
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2
\end{bmatrix}
= 
\begin{bmatrix}
100.0 \\
1.0
\end{bmatrix}
\Rightarrow 
\begin{bmatrix}
\tilde{x}_1 = 1.01 \\
\tilde{x}_2 = -99
\end{bmatrix}
\]

2-digit Arithmetic

\[
m_{21} = 1.0
\]

\[
a_{21} = 0
\]

\[
a_{22}^{(2)} = 0.0001 + 1.0 \approx 1.0
\]

\[
b_2^{(2)} = 1 - 100 = -100
\]

\[
\tilde{x}_2 = -100
\]

\[
\tilde{x}_1 = 100 - 100 = 0
\]

1% error

100% error
Systems of Linear Equations
Gauss Elimination

How to Ensure Numerical Stability

• System of equations must be well conditioned
  – Investigate condition number
    • Tricky, because it can require matrix inversion
  – Consistent with physics
    • E.g. don’t couple domains that are physically uncoupled
  – Consistent units
    • E.g. don’t mix meter and μm in unknowns
  – Dimensionless unknowns
    • Normalize all unknowns consistently

• Equilibration and Partial Pivoting, or Full Pivoting
For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.