Gauss Elimination and LU with pivoting

Friday, November 15, 2024 10:03 AM

Class 35: November 15, 2024

Recall: Last time, we introduced two algorithms involved in the direct solution of linear systems Ax = b (for A n x n, invertible mat rix):

- **Backward / Forward substitution:** If A is upper (or lower) triangular, then we can solve the last (first) equation, which involves only one unknown, and then substitute backwards (forwards) to find the solution. We counted the number of sums and multiplications, and concluded this involves **O(n^2) work**.
- **Gaussian elimination:** If A is not triangular, then this algorithm uses **elementary row operations** to produce an equivalent linear system (one with the same solution) of the form U x = c. We wrote a pseudocode for it and concluded it involves **O(n^3) work**.

So, the typical way to solve Ax=b is to first perform Gaussian elimination and then do one back substitution. Now, if we are faced with many linear systems of the form Ax = b_k, for k=1,…,N, this is very wasteful, as we do the same work to reduce A over and over again. We can, instead, s tore this work in a lower triangular factor L , which means we:

1. **LU decomposition:** Compute L and U to decompose A = LU. This is done by storing L_k encoding the elimination for each pivot, and then multiplying to find L. ***This involves O(n^3) work, with the constant being about 2x that of regular Gaussian elimination.***
2. **Two triangular solves:** Then, for each right-hand-side, the solve consists of two triangular solves. That is:

   y = forward_substitution(L, b);
   x = backward_substitution(U,y);

   This allows us to solve for each b_k in O(n^2) work. Over many rhs, the cost of computing LU is amortized.

Today, we will deal with the elephant in the numerical methods room: these algorithms, as presented, ***are unstable***. We need to modify them to fix this, using something known as **pivoting strategies**.

EXAMPLE:

$$A_\varepsilon = \begin{bmatrix} \varepsilon & 1 \\ 1 & -\varepsilon \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$A^{-1} = \frac{1}{1+\varepsilon^2} A, \quad x_\varepsilon = \frac{1}{1+\varepsilon^2}\begin{bmatrix} 2+\varepsilon \\ 1-2\varepsilon \end{bmatrix}$$

FLOP
"Floating Pt Operation"

$LU \approx \frac{4}{3}n^3 + \ldots$
(2 GE)

$\varepsilon = 10^{-6}$

$$\left[\begin{array}{cc|c} 10^{-6} & 1 & 1 \\ 1 & -10^{-6} & 2 \end{array}\right]$$

5 digit prec.

$R_2 \leftarrow R_2 - 10^6 R_1$

$$\sim \left[\begin{array}{cc|c} 10^{-6} & 1 & 1 \\ 0 & \underbrace{-10^{-6} - 10^6}_{-10^6} & \underbrace{2 - 10^6}_{-10^6} \end{array}\right]$$

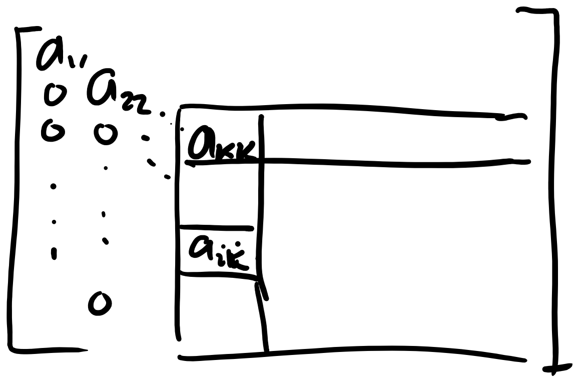
$x_2 = 1$, $x_1 = 10^6(1 - 1) = 0$
✓ ✗

More prec: $x_2 \approx 1 \quad x_1 = 10^6(1 - x_2)$

IDEA: permuting entries of A

IDEA: permuting entries of A to get the biggest pivot (in abs value)

↳ $\boxed{\text{PIVOTING}}$



2 strategies:

(i) **Row / Partial Pivot:**

argmax $(|A(k:n, k)|)$

↳ $+k \to \boxed{\text{index}}$

- Permute rows k, index of A.
- Permute $b(k)$, $b(\text{index})$.

(ii) Complete / Total Pivoting "(Row and Column)"

↳ Best row and best column

↳ argmax $(A[k:n, k:n])$

↳ $(i, j) + k \to (i_p, j_p)$

* Permute rows & columns of A and rows of b.

Store permutation.

↳ Permutation matrix (scrambling of I)

→ **Permutation vector**

→ Permutation vector

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & \cdots & n \end{array}$$
$$p = [10 \;\; 8 \;\; 3 \;\; 5 \;\; \cdots \;\;]$$

Partial pivoting → P_{row}.
Complete " → P_{row}, q_{col}.

$(U_A, c, p_A) = \text{GaussElimPP}(A, b);$
$x = \text{backsolve}(U_A, c);$

→ $\log_{10} \|Ax - b\|$ (not the same as $\log_{10} \|x - x_{true}\|$).

LU w/ partial pivoting:

$$A \neq \mathbb{L}\mathbb{U}$$

1. $$\boxed{A(p,:) = \mathbb{L}\mathbb{U}}$$

2. Solve: $\underbrace{A(p,:)}_{\mathbb{L}\mathbb{U}}\, x = b(p)$

   - $y = \text{forward\_sub}(\mathbb{L}, b(p));$
   - → $x = \text{back\_sub}(\mathbb{U}, y);$

How stable is Gauss Elim w/ PP?

How stable is Gauss Elim w/ PP?

↳ Almost solves all the issues

- Examples where PP is unstable.
  (Isolated, perturb → no problem)

Complete pivoting - iron clad.

↳ (p, q)

Solve: → $x[q]$. unscramble it.

(inv. perm, $x[q] = x;$)

LU: $\boxed{A(p,q) = LU}$

Sparse A: