Intro to Analysis of Algorithms
Linera Algebra / Parallel
Chapter 7

Michael Soltys

CSU Channel Islands

[Ed: 4th, last updated: September 16, 2025]

Row-echelon form

$$\begin{bmatrix}
1 & *\dots* & * & *\dots* & * & *\dots* & * & \\
 & & 1 & *\dots* & * & *\dots* & * & \\
 & \ddots & & & 1 & *\dots* & * & \\
 & & 0 & & & & 1 & \dots \\
 & & & \ddots & & & \vdots & \ddots
\end{bmatrix}$$

Elementary matrices

one of the following three forms:

$$
\begin{aligned}
&I + aT_{ij} \quad i \neq j && \text{(elementary of type 1)} \\
&I + T_{ij} + T_{ji} - T_{ii} - T_{jj} && \text{(elementary of type 2)} \\
&I + (c-1)T_{ii} \quad c \neq 0 && \text{(elementary of type 3)}
\end{aligned}
$$

Gaussian Elimination is a divide and conquer algorithm, with a recursive call to smaller matrices.

If A is a $1 \times m$ matrix, $A = [a_{11} a_{12} \dots a_{1m}]$, then:

$$GE(A) = \begin{cases} [1/a_{1i}] & \text{where } i = \min\{1, 2, \dots, m\} \text{ such that } a_{i1} \neq 0 \\ [1] & \text{if } a_{11} = a_{12} = \dots = a_{1m} = 0 \end{cases}$$

Suppose now that $n > 1$. If $A = 0$, let $GE(A) = I$. Otherwise, let:

$$GE(A) = \begin{bmatrix} 1 & 0 \\ 0 & GE((EA)[1|1]) \end{bmatrix} E$$

where E is a product of at most $n + 1$ elementary matrices. Note that $C[i|j]$ denotes the matrix C with row i and j.

if $n = 1$ **then**
    if $a_{11} = a_{12} = \cdots = a_{1m} = 0$ **then**
        return $[1]$
    else
        return $[1/a_{1\ell}]$ where $\ell = \min_{i \in [n]}\{a_{1i} \neq 0\}$
    end if
else
    if $A = 0$ **then**
        return I
    else
        if first column of A is zero **then**
            Compute E as in Case 1.
        else
            Compute E as in Case 2.
        end if
        return $\left[\begin{array}{cc} 1 & 0 \\ 0 & GE((EA)[1|1]) \end{array}\right] E$
    end if
end if

Gram-Schmidt

Pre-condition: $\{v_1, \ldots, v_n\}$ a basis for $\mathbb{R}^n$

$v_1^* \longleftarrow v_1$
for $i = 2, 3, \ldots, n$ **do**
  for $j = 1, 2, \ldots, (i-1)$ **do**
    $\mu_{ij} \longleftarrow (v_i \cdot v_j^*)/\|v_j^*\|^2$
  end for
  $v_i^* \longleftarrow v_i - \sum_{j=1}^{i-1} \mu_{ij} v_j^*$
end for

Post-condition: $\{v_1^*, \ldots, v_n^*\}$ an orthogonal basis for $\mathbb{R}^n$

Gauss lattice reduction

Pre-condition: $\{v_1, v_2\}$ are linearly independent in $\mathbb{R}^2$

loop
    if $\|v_2\| < \|v_1\|$ **then**
        swap v_1 and v_2
    end if
    $m \longleftarrow \lfloor v_1 \cdot v_2 / \|v_1\|^2 \rceil$ (note that $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$)
    if $m = 0$ **then**
        return v_1, v_2
    else
        $v_2 \longleftarrow v_2 - m v_1$
    end if
end loop

Csanky

Given a matrix A, its *trace* is defined as the sum of the diagonal entries, i.e., $\mathrm{tr}(A) = \sum_i a_{ii}$. Using traces we can compute the *Newton's symmetric polynomials* which are defined as follows: $s_0 = 1$, and for $1 \leq k \leq n$, by:

$$s_k = \frac{1}{k} \sum_{i=1}^{k} (-1)^{i-1} s_{k-i} \mathrm{tr}(A^i).$$

Then, it turns out that
$p_A(x) = s_0 x^n - s_1 x^{n-1} + s_2 x^{n-2} - \cdots \pm s_n x^0$, that is, Newton's symmetric polynomials compute the coefficients of the characteristic polynomial, $p_A(x) = \det(xI - A)$.

$$
\begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}, \quad
\begin{pmatrix}
0 & 0 & 0 & \dots \\
\frac{1}{2}\mathrm{tr}(A) & 0 & 0 & \dots \\
\frac{1}{3}\mathrm{tr}(A^2) & \frac{1}{3}\mathrm{tr}(A) & 0 & \dots \\
\frac{1}{4}\mathrm{tr}(A^3) & \frac{1}{4}\mathrm{tr}(A^2) & \frac{1}{4}\mathrm{tr}(A) & \dots \\
\vdots & \vdots & \vdots & \ddots
\end{pmatrix}, \quad
\begin{pmatrix} \mathrm{tr}(A) \\ \frac{1}{2}\mathrm{tr}(A^2) \\ \vdots \\ \frac{1}{n}\mathrm{tr}(A^n) \end{pmatrix}
$$

Berkowitz

Berkowitz's algorithm is also Divide and Conquer, and it computes the characteristic polynomial of A from the characteristic polynomial of its *principal minor*, i.e., the matrix M obtained from deleting the first row and column of A:

$$A = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix},$$

R is an $1 \times (n-1)$ row matrix and S is a $(n-1) \times 1$ column matrix and M is $(n-1) \times (n-1)$. Let $p(x)$ and $q(x)$ be the characteristic polynomials of A and M respectively. Suppose that the coefficients of p form the column vector:

$$p = \begin{pmatrix} p_n & p_{n-1} & \dots & p_0 \end{pmatrix}^t,$$

where p_i is the coefficient of x^i in $\det(xI - A)$, and similarly for q. Then:

$$p = C_1 q,$$

where C_1 is an $(n+1) \times n$ *Toeplitz* lower triangular matrix (Toeplitz means that the values on each diagonal are constant) where the entries in the first column are defined as follows: $c_{i1} = 1$ if $i = 1$, $c_{i1} = -a_{11}$ if $i = 2$, and $c_{i1} = -(RM^{i-3}S)$ if $i \geq 3$. Berkowitz's algorithm consists in repeating this for q, and continuing so that p is expressed as a product of matrices. Thus:

$$p_A^{\text{BERK}} = C_1 C_2 \cdots C_n,$$

where C_i is an $(n+2-i) \times (n+1-i)$ Toeplitz matrix defined as above except A is replaced by its i-th principal sub-matrix. Note that $C_n = (1 \quad -a_{nn})^t$.