$$P_n(x) = a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0,$$

$$E = \sum_{i=1}^{m}(y_i - P_n(x_i))^2$$

$$= \sum_{i=1}^{m} y_i^2 - 2\sum_{i=1}^{m} P_n(x_i)y_i + \sum_{i=1}^{m}(P_n(x_i))^2$$

$$\sum_{k=0}^{n} a_k \sum_{i=1}^{m} x_i^{j+k} = \sum_{i=1}^{m} y_i x_i^j, \quad \text{for each } j = 0, 1, \ldots, n.$$

$$a_0\sum_{i=1}^{m} x_i^0 + a_1\sum_{i=1}^{m} x_i^1 + a_2\sum_{i=1}^{m} x_i^2 + \cdots + a_n\sum_{i=1}^{m} x_i^n = \sum_{i=1}^{m} y_i x_i^0,$$

$$a_0\sum_{i=1}^{m} x_i^1 + a_1\sum_{i=1}^{m} x_i^2 + a_2\sum_{i=1}^{m} x_i^3 + \cdots + a_n\sum_{i=1}^{m} x_i^{n+1} = \sum_{i=1}^{m} y_i x_i^1,$$

$$a_0\sum_{i=1}^{m} x_i^n + a_1\sum_{i=1}^{m} x_i^{n+1} + a_2\sum_{i=1}^{m} x_i^{n+2} + \cdots + a_n\sum_{i=1}^{m} x_i^{2n} = \sum_{i=1}^{m} y_i x_i^n.$$

$$\int_a^b [f(x) - P_n(x)]^2\,dx.$$

$$E \equiv E_2(a_0, a_1, \ldots, a_n) = \int_a^b \left(f(x) - \sum_{k=0}^{n} a_k x^k\right)^2 dx.$$

$$\sum_{k=0}^{n} a_k \int_a^b x^{j+k}\,dx = \int_a^b x^j f(x)\,dx,$$

The set of functions $\{\phi_0, \ldots, \phi_n\}$ is said to be **linearly independent** on $[a, b]$ if, whenever

$$c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_n\phi_n(x) = 0, \quad \text{for all } x \in [a, b],$$

we have $c_0 = c_1 = \cdots = c_n = 0$. Otherwise the set of functions is said to be **linearly dependent**. ■

$\{\phi_0, \phi_1, \ldots, \phi_n\}$ is said to be an **orthogonal set of functions** for the interval $[a, b]$ respect to the weight function w if

$$\int_a^b w(x)\phi_k(x)\phi_j(x)\,dx = \begin{cases} 0, & \text{when } j \neq k, \\ \alpha_j > 0, & \text{when } j = k. \end{cases}$$

If, in addition, $\alpha_j = 1$ for each $j = 0, 1, \ldots, n$, the set is said to be **orthonormal**.

If $\{\phi_0, \ldots, \phi_n\}$ is an orthogonal set of functions on an interval $[a, b]$ with respect to the weight function w, then the least squares approximation to f on $[a, b]$ with respect to w is

Theorem 8.6 $$P(x) = \sum_{j=0}^{n} a_j\phi_j(x),$$

$$a_j = \frac{\int_a^b w(x)\phi_j(x)f(x)\,dx}{\int_a^b w(x)[\phi_j(x)]^2\,dx} = \frac{1}{\alpha_j}\int_a^b w(x)\phi_j(x)f(x)\,dx.$$

The set of polynomial functions $\{\phi_0, \phi_1, \ldots, \phi_n\}$ defined in the following way is orthogonal on $[a, b]$ with respect to the weight function w.

Theorem 8.7 $\phi_0(x) \equiv 1, \quad \phi_1(x) = x - B_1, \quad$ for each x in $[a, b]$,

where $$B_1 = \frac{\int_a^b xw(x)[\phi_0(x)]^2\,dx}{\int_a^b w(x)[\phi_0(x)]^2\,dx},$$

and when $k \geq 2$, $\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x), \quad$ for each x in $[a, b]$,

where $$B_k = \frac{\int_a^b xw(x)[\phi_{k-1}(x)]^2\,dx}{\int_a^b w(x)[\phi_{k-1}(x)]^2\,dx}$$

and $$C_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-2}(x)\,dx}{\int_a^b w(x)[\phi_{k-2}(x)]^2\,dx}.$$

A **vector norm** on $\mathbb{R}^n$ is a function, $\|\cdot\|$, from $\mathbb{R}^n$ into $\mathbb{R}$ with the following properties:

(i) $\|\mathbf{x}\| \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$,

(ii) $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$,

(iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ for all $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$,

(iv) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

The l_2 and l_∞ norms for the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ are defined by

$$\|\mathbf{x}\|_2 = \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} \quad \text{and} \quad \|\mathbf{x}\|_\infty = \max_{1\leq i\leq n}|x_i|.$$

$$\|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2, \quad \text{for each } \mathbf{x}, \mathbf{y} \in \mathbb{R}_n,$$

(Cauchy-Bunyakovsky-Schwarz Inequality for Sums)

For each $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$ in $\mathbb{R}^n$,

$$\mathbf{x}^t\mathbf{y} = \sum_{i=1}^{n} x_iy_i \leq \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2}\left\{\sum_{i=1}^{n} y_i^2\right\}^{1/2} = \|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2.$$

A sequence $\{\mathbf{x}^{(k)}\}_{k=1}^{\infty}$ of vectors in $\mathbb{R}^n$ is said to **converge** to $\mathbf{x}$ with respect to the norm $\|\cdot\|$ if, given any $\varepsilon > 0$, there exists an integer $N(\varepsilon)$ such that

Definition 7.5 $\|\mathbf{x}^{(k)} - \mathbf{x}\| < \varepsilon, \quad$ for all $k \geq N(\varepsilon)$. ■

For each $\mathbf{x} \in \mathbb{R}^n$,

Theorem 7.7 $$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty.$$

A **matrix norm** on the set of all $n \times n$ matrices is a real-valued function, $\|\cdot\|$, defined on this set, satisfying for all $n \times n$ matrices A and B and all real numbers α:

(i) $\|A\| \geq 0$;

(ii) $\|A\| = 0$, if and only if A is O, the matrix with all 0 entries;

(iii) $\|\alpha A\| = |\alpha|\|A\|$;

(iv) $\|A + B\| \leq \|A\| + \|B\|$;

(v) $\|AB\| \leq \|A\|\|B\|$. ■

If $||\cdot||$ is a vector norm on $\mathbb{R}^n$, then

Theorem 7.9 $$\|A\| = \max_{\|\mathbf{x}\|=1}\|A\mathbf{x}\|$$

is a matrix norm.

If $A = (a_{ij})$ is an $n \times n$ matrix, then

Theorem 7.11 $$\|A\|_\infty = \max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|.$$

If A is a square matrix, the **characteristic polynomial** of A is defined by

Definition 7.12 $$p(\lambda) = \det(A - \lambda I).$$

Spectral Radius

The **spectral radius** $\rho(A)$ of a matrix A is defined l theorem.

Definition 7.14 $\rho(A) = \max|\lambda|, \quad$ where λ is an eigenvalue of A.

Theorem 7.15 If A is an $n \times n$ matrix, then

(i) $\|A\|_2 = [\rho(A^tA)]^{1/2}$,

(ii) $\rho(A) \leq \|A\|$, for any natural norm $\|\cdot\|$.

We call an $n \times n$ matrix A **convergent** if

Definition 7.16 $\lim_{k\to\infty}(A^k)_{ij} = 0$, for each $i = 1, 2, \ldots, n$ $j = 1, 2, \ldots, n$.

Theorem 7.17 The following statements are equivalent.

(i) A is a convergent matrix.

(ii) $\lim_{n\to\infty}\|A^n\| = 0$, for some natural norm.

(iii) $\lim_{n\to\infty}\|A^n\| = 0$, for all natural norms.

(iv) $\rho(A) < 1$.

(v) $\lim_{n\to\infty}A^n\mathbf{x} = \mathbf{0}$, for every $\mathbf{x}$.

7.3 J and GS

$$A\mathbf{x} = \mathbf{b}, \text{ or } (D - L - U)\mathbf{x} = \mathbf{b},$$
$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b},$$
$$\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}.$$
$$\mathbf{x}^{(k)} = D^{-1}(L + U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b},$$
$$\mathbf{x}^{(k)} = T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j.$$

Gauss-Seidel method

$$(D - L)\mathbf{x}^{(k)} = U\mathbf{x}^{(k-1)} + \mathbf{b}$$
$$\mathbf{x}^{(k)} = (D - L)^{-1}U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b},$$
$$T_g = (D - L)^{-1}U \text{ and } \mathbf{c}_g = (D - L)^{-1}\mathbf{b},$$
$$\mathbf{x}^{(k)} = T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g.$$

Corollary 7.20

If $\|T\| < 1$ for any natural matrix norm and $\mathbf{c}$ is a given vector, then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$ converges, for any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to a vector $\mathbf{x} \in \mathbb{R}^n$, with $\mathbf{x} = T\mathbf{x} + \mathbf{c}$, and the following error bounds hold:

(i) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|T\|^k\|\mathbf{x}^{(0)} - \mathbf{x}\|$; (ii) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T\|^k}{1-\|T\|}\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$. ■

Theorem 7.22 (Stein-Rosenberg)

If $a_{ij} \leq 0$, for each $i \neq j$ and $a_{ii} > 0$, for each $i = 1, 2, \ldots, n$, then one and only one of the following statements holds:

(i) $0 \leq \rho(T_g) < \rho(T_j) < 1$; (ii) $1 < \rho(T_j) < \rho(T_g)$;

(iii) $\rho(T_j) = \rho(T_g) = 0$; (iv) $\rho(T_j) = \rho(T_g) = 1$. ■

SOR method

$$x_i^{(k)} = (1 - \omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1}a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}\right]$$

$$(D - \omega L)\mathbf{x}^{(k)} = [(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega\mathbf{b}.$$
$$\mathbf{x}^{(k)} = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega(D - \omega L)^{-1}\mathbf{b}.$$
$$\mathbf{x}^{(k)} = T_\omega\mathbf{x}^{(k-1)} + \mathbf{c}_\omega.$$

Suppose that $\tilde{\mathbf{x}}$ is an approximation to the solution of $A\mathbf{x} = \mathbf{b}$, A is a nonsingular matrix and $\mathbf{r}$ is the residual vector for $\tilde{\mathbf{x}}$. Then for any natural norm,

Theorem 7.27 $$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{r}\| \cdot \|A^{-1}\|$$

and if $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$, $$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|A\| \cdot \|A^{-1}\|\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

The **condition number** of the nonsingular matrix A relative to a norm $\|\cdot\|$ is

Definition 7.28 $K(A) = \|A\| \cdot \|A^{-1}\|$.

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq K(A)\frac{\|\mathbf{r}\|}{\|A\|}$$

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

A is **well-conditioned** if $K(A)$ is close to 1,

Suppose A is nonsingular and

Theorem 7.29 $$\|\delta A\| < \frac{1}{\|A^{-1}\|}.$$

The solution $\tilde{\mathbf{x}}$ to $(A + \delta A)\tilde{\mathbf{x}} = \mathbf{b} + \delta\mathbf{b}$ approximates the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ with the error estimate

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{K(A)\|A\|}{\|A\| - K(A)\|\delta A\|}\left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\delta A\|}{\|A\|}\right). \tag{7.25}$$

7.6 The Conjugate Gradient Method

$$\langle\mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^t\mathbf{y},$$

Theorem 7.30

For any vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ and any real number α, we have

(a) $\langle\mathbf{x}, \mathbf{y}\rangle = \langle\mathbf{y}, \mathbf{x}\rangle$; (b) $\langle\alpha\mathbf{x}, \mathbf{y}\rangle = \langle\mathbf{x}, \alpha\mathbf{y}\rangle = \alpha\langle\mathbf{x}, \mathbf{y}\rangle$;

(c) $\langle\mathbf{x} + \mathbf{z}, \mathbf{y}\rangle = \langle\mathbf{x}, \mathbf{y}\rangle + \langle\mathbf{z}, \mathbf{y}\rangle$; (d) $\langle\mathbf{x}, \mathbf{x}\rangle \geq 0$;

(e) $\langle\mathbf{x}, \mathbf{x}\rangle = 0$ if and only if $\mathbf{x} = \mathbf{0}$.

When A is positive definite, $\langle\mathbf{x}, A\mathbf{x}\rangle = \mathbf{x}^tA\mathbf{x} > 0$ unless $\mathbf{x} = \mathbf{0}$. Also, since A is symmetric, we have $\mathbf{x}^tA\mathbf{y} = \mathbf{x}^tA^t\mathbf{y} = (A\mathbf{x})^t\mathbf{y}$, so in addition to the results in Theorem 7.30, we have for each $\mathbf{x}$ and $\mathbf{y}$,

$$\langle\mathbf{x}, A\mathbf{y}\rangle = (A\mathbf{x})^t\mathbf{y} = \mathbf{x}^tA^t\mathbf{y} = \mathbf{x}^tA\mathbf{y} = \langle A\mathbf{x}, \mathbf{y}\rangle. \tag{7.27}$$

The vector $\mathbf{x}^*$ is a solution to the positive definite linear system $A\mathbf{x} = \mathbf{b}$ if and only if $\mathbf{x}^*$ produces the minimal value of

Theorem 7.31 $$g(\mathbf{x}) = \langle\mathbf{x}, A\mathbf{x}\rangle - 2\langle\mathbf{x}, \mathbf{b}\rangle.$$ ■

Let $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\}$ be an A-orthogonal set of nonzero vectors associated with the positive definite matrix A, and let $\mathbf{x}^{(0)}$ be arbitrary. Define

Theorem 7.32 $$t_k = \frac{\langle\mathbf{v}^{(k)}, \mathbf{b} - A\mathbf{x}^{(k-1)}\rangle}{\langle\mathbf{v}^{(k)}, A\mathbf{v}^{(k)}\rangle} \quad \text{and} \quad \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)},$$

for $k = 1, 2, \ldots, n$. Then, assuming exact arithmetic, $A\mathbf{x}^{(n)} = \mathbf{b}$. ■

The residual vectors $\mathbf{r}^{(k)}$, where $k = 1, 2, \ldots, n$, for a conjugate direction method, satisfy the equations

Theorem 7.33 $\langle\mathbf{r}^{(k)}, \mathbf{v}^{(j)}\rangle = 0, \quad$ for each $j = 1, 2, \ldots, k$. ■

CG

In summary, we have

$$\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}; \quad \mathbf{v}^{(1)} = \mathbf{r}^{(0)};$$

and, for $k = 1, 2, \ldots, n$,

$$t_k = \frac{\langle\mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)}\rangle}{\langle\mathbf{v}^{(k)}, A\mathbf{v}^{(k)}\rangle}, \quad \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)},$$

$$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - t_kA\mathbf{v}^{(k)}, \quad s_k = \frac{\langle\mathbf{r}^{(k)}, \mathbf{r}^{(k)}\rangle}{\langle\mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)}\rangle},$$

$$\mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} + s_k\mathbf{v}^{(k)}.$$

Positive Definite Matrices

Definition 6.22

A matrix A is **positive definite** if it is symmetric and if $\mathbf{x}^tA\mathbf{x} > 0$ for every n-dimensional vector $\mathbf{x} \neq \mathbf{0}$.

heorem 6.14

he following operations involving the transpose of a matrix

i) $(A^t)^t = A$, (iii) $(AB)^t = B^tA^t$,

i) $(A + B)^t = A^t + B^t$ (iv) if A^{-1} exists, then $(A^{-1})^t = (A^t)^{-1}$.

Jacobi Iterative

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}}\left[-\sum_{\substack{j=1\\ j\ne i}}^{n} (a_{ij}XO_j) + b_i\right].$$

Step 4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT ($x_1, \ldots, x_n$);
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP.

Gauss-Seidel Iterative

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i\right].$$

Step 4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT ($x_1, \ldots, x_n$);
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP.

SOR

To solve $A\mathbf{x} = \mathbf{b}$ given the parameter ω and an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$, of the matrix A; the entries b_i, $1 \le i \le n$, of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$, of $\mathbf{XO} = \mathbf{x}^{(0)}$; the parameter ω; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = (1-\omega)XO_i + \frac{1}{a_{ii}}\left[\omega\left(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i\right)\right].$$

Step 4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT ($x_1, \ldots, x_n$);
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP.

Iterative Refinement

To approximate the solution to the linear system $A\mathbf{x} = \mathbf{b}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the maximum number of iterations N; tolerance TOL; number of digits of precision t.

OUTPUT the approximation $\mathbf{xx} = (xx_i, \ldots, xx_n)^t$ or a message that the number of iterations was exceeded, and an approximation $COND$ to $K_\infty(A)$.

Step 0 Solve the system $A\mathbf{x} = \mathbf{b}$ for $x_1, \ldots, x_n$ by Gaussian elimination saving the multipliers m_{ji}, $j = i+1, i+2, \ldots, n$, $i = 1, 2, \ldots, n-1$ and noting row interchanges.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–9.

Step 3 For $i = 1, 2, \ldots, n$ (*Calculate* $\mathbf{r}$.)

$$\text{set } r_i = b_i - \sum_{j=1}^{n} a_{ij}x_j.$$

(*Perform the computations in double-precision arithmetic.*)

Step 4 Solve the linear system $A\mathbf{y} = \mathbf{r}$ by using Gaussian elimination in the same order as in Step 0.

Step 5 For $i = 1, \ldots, n$ set $xx_i = x_i + y_i$.

Step 6 If $k = 1$ then set $COND = \dfrac{\|\mathbf{y}\|_\infty}{\|\mathbf{xx}\|_\infty}10^t$.

Step 7 If $\|\mathbf{x} - \mathbf{xx}\|_\infty < TOL$ then OUTPUT ($\mathbf{xx}$);
OUTPUT ($COND$);
(*The procedure was successful.*)
STOP.

Step 8 Set $k = k + 1$.

Step 9 For $i = 1, \ldots, n$ set $x_i = xx_i$.

Step 10 OUTPUT ('Maximum number of iterations exceeded');
OUTPUT ($COND$);
(*The procedure was unsuccessful.*)
STOP.

Preconditioned Conjugate Gradient Method

To solve $A\mathbf{x} = \mathbf{b}$ given the preconditioning matrix C^{-1} and the initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_j, $1 \le j \le n$ of the vector $\mathbf{b}$; the entries γ_{ij}, $1 \le i, j \le n$ of the preconditioning matrix C^{-1}, the entries x_i, $1 \le i \le n$ of the initial approximation $\mathbf{x} = \mathbf{x}^{(0)}$, the maximum number of iterations N; tolerance TOL.

OUTPUT the approximate solution $x_1, \ldots x_n$ and the residual $r_1, \ldots r_n$ or a message that the number of iterations was exceeded.

Step 1 Set $\mathbf{r} = \mathbf{b} - A\mathbf{x}$; (*Compute* $\mathbf{r}^{(0)}$.)
$\mathbf{w} = C^{-1}\mathbf{r}$; (*Note:* $\mathbf{w} = \mathbf{w}^{(0)}$)
$\mathbf{v} = C^{-t}\mathbf{w}$; (*Note:* $\mathbf{v} = \mathbf{v}^{(1)}$)
$\alpha = \sum_{j=1}^{n} w_j^2$.

Step 2 Set $k = 1$.

Step 3 While ($k \le N$) do Steps 4–7.

Step 4 If $\|\mathbf{v}\| < TOL$, then
OUTPUT ('Solution vector'; $x_1, \ldots, x_n$);
OUTPUT ('with residual'; $r_1, \ldots, r_n$);
(*The procedure was successful.*)
STOP

Step 5 Set $\mathbf{u} = A\mathbf{v}$; (*Note:* $\mathbf{u} = A\mathbf{v}^{(k)}$)
$t = \dfrac{\alpha}{\sum_{j=1}^{n} v_j u_j}$; (*Note:* $t = t_k$)
$\mathbf{x} = \mathbf{x} + t\mathbf{v}$; (*Note:* $\mathbf{x} = \mathbf{x}^{(k)}$)
$\mathbf{r} = \mathbf{r} - t\mathbf{u}$; (*Note:* $\mathbf{r} = \mathbf{r}^{(k)}$)
$\mathbf{w} = C^{-1}\mathbf{r}$; (*Note:* $\mathbf{w} = \mathbf{w}^{(k)}$)
$\beta = \sum_{j=1}^{n} w_j^2$. (*Note:* $\beta = \langle \mathbf{w}^{(k)}, \mathbf{w}^{(k)} \rangle$)

Step 6 If $|\beta| < TOL$ then
if $\|\mathbf{r}\| < TOL$ then
OUTPUT('Solution vector'; $x_1, \ldots, x_n$);
OUTPUT('with residual'; $r_1, \ldots, r_n$);
(*The procedure was successful.*)
STOP

Step 7 Set $s = \beta/\alpha$; ($s = s_k$)
$\mathbf{v} = C^{-t}\mathbf{w} + s\mathbf{v}$; (*Note:* $\mathbf{v} = \mathbf{v}^{(k+1)}$)
$\alpha = \beta$; (*Update* α.)
$k = k + 1$.

Step 8 If ($k > n$) then
OUTPUT ('The maximum number of iterations was exceeded.');
(*The procedure was unsuccessful.*)
STOP.

```
state gs(const Matrix& A, const Vector& b, Vector& x,
             int maxIter, double tol) {

  // CHECK DATA
  int n = A.n(0);
  if(A.n(1) != n || b.n() != n || x.n() != n) return BAD_DATA;
  if(tol <= 0) return BAD_DATA;
  if(maxIter <= 0) maxIter = 1;
  for(int i=0; i<n; i++) {if(A(i,i) == 0) return BAD_DIAGONAL;}

  // APPLY GS
  Vector xOld(x);
  for(int iter=0; iter<maxIter; iter++) {
    // Get new x
    for(int i=0; i<n; i++) {
      double sum = 0;
      for(int j=0; j<n; j++) {
        if(j < i) sum += A(i,j)*x(j);
        if(j==i) continue;
        if(j > i) sum += A(i,j)*xOld(j); }
      x(i) = ( -sum + b(i) ) / A(i,i); }

    // Check error tolerance
    xOld -= x;
    double maxerror = maxNorm(xOld) / maxNorm(x);
#ifdef MONITOR
    std::cout << "Iter " << iter+1 << ", max-error "<<
    maxerror << std::endl;
#endif
    if( maxerror <= tol) return SUCCESS;
    xOld = x;}
    return WONT_STOP; }
int main(){ //poly-LS
   int n;
   cout<<"Enter polynomial degree: "<<flush;
   cin >> n;
   cout << "y = ";
   for(int i =0; i <=n;i++){
      if(i == 0) {
       cout << "a_0 + "; }
      else if (i == n) {
       cout << "a_" << i << "x^" << i << endl; }
      else {
       cout << "a_" << i << "x^" << i << " + ";}}
   int m;
   cout<<"Enter num of data points: "<<flush;cin>>m;
   Vector x(m), y(m);
   cout<<"Enter x values: "<<flush;cin>>x;
   cout<<"Enter y values: "<<flush; cin>>y;
   Matrix A(n+1,n+1);
   for (int i = 0; i <n+1; i++) {
      for(int j = 0; j<n+1; j++) {
          double sumpow = 0;
          for(int k = 0; k < m; k++){
               sumpow += pow(x(k), i+j); }
          A(i,j) = sumpow;}}
  Vector B(n+1);
  for (int i = 0;i < n+1; i++) {
     double sumypow = 0;
     for( int k =0; k <m;k++){
       sumypow += pow(x(k), i) * y(k); }
     B(i) = sumypow; }
  Permutation p(n+1);
  solve(A, p, B);
  for(int i = 0; i < n+1; i++){
      cout << "a_" << i << " = " << B(i) << endl; }
  cout << endl;
  return 0;}
```

```
state sor(const Matrix& A, const Vector& b, Vector& x,
          int maxIter, double tol, double w) {

  // CHECK DATA ...
  // APPLY SOR
  Vector xOld(x);
  for(int iter=0; iter<maxIter; iter++) {
    // Get new x
    for(int i=0; i<n; i++) {
      double sum = 0;
      for(int j=0; j<n; j++) {
        if(j < i) sum += A(i,j)*x(j);
        if(j==i) continue;
        if(j > i) sum += A(i,j)*xOld(j);}
      x(i) = (1-w) *  xOld(i) + w * (-sum+b(i))/A(i,i); }

    xOld -= x;
    double maxerror = maxNorm(xOld) / maxNorm(x);
std::cout<<"Iter"<<iter+1<<",max-error"<<maxerror<<std::endl;
    if( maxerror <= tol) return SUCCESS;
    xOld = x; }
  return WONT_STOP; }
```

```
//CG with precondi
// Set initial residual r = b - Ax
  Vector r(n);
  for(int i=0; i<n; i++) {
    r(i) = b(i);
    for(int j=0; j<n; j++) {
      r(i) -= A(i,j)*x(j); } }
  Vector w(n), v(n), vv(n);
  matVecMult(Cinv, r, w);
  matTVecMult(Cinv, w, v);
  double alpha = scDot(w,w);
  double tolSq = tol*tol;
  // CONJUGATE GRADIENT LOOP
  for(int iter=0; iter<maxIter; iter++) {
    if(scDot(v,v) <= tolSq) return SUCCESS;
    // Set u = Av
    Vector u(n);
    for(int i=0; i<n; i++) {
      u(i) = 0;
      for(int j=0; j<n; j++) {
        u(i) += A(i,j)*v(j); }}
    // Update x = x + tv and r = r - tu
    double t = alpha / scDot(v,u);
    for(int i=0; i<n; i++) {
      x(i) += t*v(i);
      r(i) -= t*u(i); }
    matVecMult(Cinv, r, w);
    // Get new search direction v = r + s*v;
    double beta = scDot(w,w);
    if(beta <= tolSq) return SUCCESS;
    double s = beta / alpha;
    for(int i=0; i<n; i++) {
      matTVecMult(Cinv, w, vv);
      v(i) = vv(i) + s*v(i); }
```