1 Symmetric positive definite matrices

In this chapter we look at some iterative methods for solving $Ax = b$, where A is a symmetric positive definite n by n matrix. Although the algorithms work fine even for nonreal (but hermitian) matrices, we will restrict our attention to the real case.

A symmetric matrix $A = A^T$ is positive definite (*spd*), if

$$x^T Ax > 0 \qquad \forall x \neq 0.$$

A sufficient condition for A to be *spd* is that A is diagonally dominant, i.e.,

$$a_{ii} > \sum_{j \neq i} |a_{ij}|.$$

It is well known and easy to show that any *spd* matrix A has strictly positive real eigenvalues λ_j, and the corresponding eigenvectors v_j can be chosen to form an orthonormal basis of $\mathbb{R}^n$. Thus, the matrix V which has the v_j as its columns,

$$V = [v_1| \ldots |v_n],$$

is an orthogonal matrix. The system $Ax = b$ has a unique solution z which is the minimizer of the quadratic function

$$f(x) = \frac{1}{2}x^T Ax - b^T x.$$

1.1 Steepest descent

Choose $p_k = -\nabla f(x_k)$, i.e., the direction of steepest descent.

For the quadratic case,

$$f(x) = \frac{1}{2}x^T Ax - x^T b, \tag{1.1}$$

with A a positive definite matrix, we have

$$\nabla f(x) = Ax - b, \qquad g_k := \nabla f(x_k) = Ax_k - b \tag{1.2}$$

$$\nabla^2 f(x) = A, \qquad H_k := \nabla^2 f(x_k) = A. \tag{1.3}$$

A convenient way to measure the error:

$$E(x) = \frac{1}{2}||x - z||_A^2 = \frac{1}{2}(x - z)^T A(x - z) = f(x) + \frac{1}{2}z^T Az, \tag{1.4}$$

where z is the minimizer of f, i.e, $z = A^{-1}b$. Note, that $(x^T Ax)^{1/2}$ is a **norm** for any positive definite matrix A.

The parameter α_k can be explicitly determined in the quadratic case.

$$\begin{aligned} f(x_k - \alpha g_k) &= \frac{1}{2}(x_k - \alpha g_k)^T A(x_k - \alpha g_k) - (x_k - \alpha g_k)^T b \\ &= \underbrace{\frac{1}{2}x_k^T Ax_k - x_k^T b}_{= f(x_k)} - \alpha g_k^T Ax_k + \frac{1}{2}\alpha^2 g_k^T Ag_k + \alpha g_k^T b \\ &= \text{const} + \frac{1}{2}\alpha^2 g_k^T Ag_k - \alpha g_k^T \underbrace{(Ax_k - b)}_{= g_k}; \end{aligned}$$

the minimum on the line $x_k - \alpha g_k$ is attained for

$$\alpha = \alpha_k = \frac{g_k^T g_k}{g_k^T Ag_k}. \tag{1.5}$$

Proof of convergence:

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= -\frac{1}{2}\alpha_k^2 g_k^T Ag_k + \alpha_k g_k^T g_k \\ &= \frac{1}{2}\alpha_k g_k^T g_k = \frac{1}{2}\frac{(g_k^T g_k)^2}{g_k^T Ag_k}. \end{aligned}$$

Setting

$$e_k = x_k - z \qquad \Rightarrow \quad g_k = Ae_k,$$

and using that

$$f(x_k) - f(x_{k+1}) = E(x_k) - E(x_{k+1}),$$

and

$$E(x_k) = \frac{1}{2}e_k^T Ae_k = \frac{1}{2}g_k^T A^{-1}g_k,$$

we obtain

$$E(x_{k+1}) = \left\{1 - \frac{(g_k^T g_k)^2}{(g_k^T Ag_k)(g_k^T A^{-1}g_k)}\right\} E(x_k). \tag{1.6}$$

This shows $E(x_{k+1}) < E(x_k)$, but does not yet prove $E(x_k) \to 0$. Since A is a positive definite matrix, its eigenvalues $\lambda_1, \ldots, \lambda_n$ are all positive,

$$0 < m = \lambda_1 \leq \ldots \leq \lambda_n = M,$$

and

$$m = \min_{||x||=1} x^T Ax, \qquad M = \max_{||x||=1} x^T Ax = ||A||,$$

and therefore

$$\frac{1}{m} = \max_{||x||=1} x^T A^{-1} x = ||A^{-1}||.$$

The condition number $\kappa(A)$ of A is defined as

$$\kappa(A) = \frac{M}{m} = ||A||||A^{-1}||.$$

To proceed, we need a famous result by Kantorovich (without proof), namely

Lemma 1.1 *With m and M defined as the smallest and largest eigenvalue of A, we have.*

$$\frac{(x^T x)^2}{(x^T Ax)(x^T A^{-1} x)} \geq \frac{4mM}{(m+M)^2}. \tag{1.7}$$

Proof. In the coordinate system with the eigenvectors of A as basis vectors ($x = \sum c_j v_j$), we get

$$\frac{(x^T x)^2}{(x^T Ax)(x^T A^{-1} x)} = \frac{\left(\sum_{j=1}^n c_j^2\right)^2}{\left(\sum_{j=1}^n \lambda_j c_j^2\right)\left(\sum_{j=1}^n c_j^2/\lambda_j\right)} = \frac{1/\sum_{j=1}^n \gamma_j \lambda_j}{\sum_{j=1}^n \gamma_j/\lambda_j},$$

where $\gamma_i = c_i^2/\sum_{j=1}^n c_j^2$. Thus, we can write our original expression as the ratio of two functions involving **convex** combinations,

$$\frac{(x^T x)^2}{(x^T Ax)(x^T A^{-1} x)} = \frac{1/\sum_{j=1}^n \gamma_j \lambda_j}{\sum_{j=1}^n \gamma_j/\lambda_j} = \frac{\phi(\gamma)}{\psi(\gamma)}.$$

In the numerator we have a combination of the λ_i's, in the denominator a combination of the $1/\lambda_i$'s. See Figure 1.1: $\sum \gamma_j \lambda_j$ is a point between λ_1 and λ_n, the value $\phi(\gamma)$ is a point on the curve $1/\lambda$. On the other hand, $\psi(\gamma)$ is a convex combination of curves on the curve, and its value corresponds to a point in the shaded area. It is not difficult to see, that the minimum value for this quotient is attained at the point $\lambda = (\lambda_1 + \lambda_n)/2$, yielding

$$\frac{\phi(\gamma)}{\psi(\gamma)} \geq \frac{4\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2}.$$

$\square$

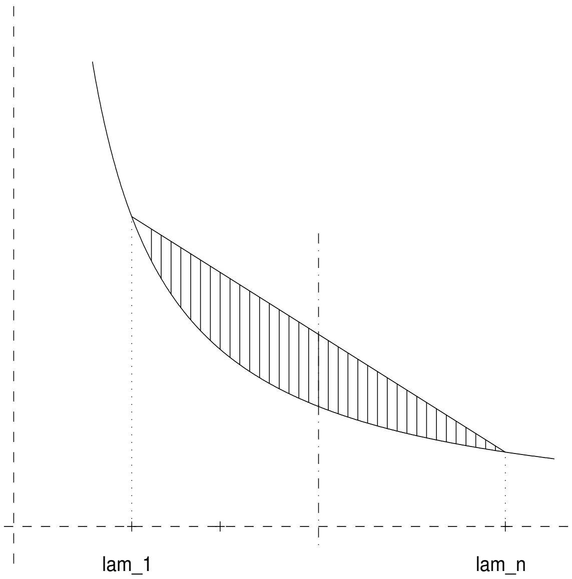

Figure 1.1: Convex combinations and $1/\lambda$

Substituting this into (1.6) and using

$$1 - \frac{4mM}{(m+M)^2} = \frac{(m+M)^2 - 4mM}{(m+M)^2} = \frac{(M-m)^2}{(M+m)^2} = \left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^2,$$

finally yields

$$E(x_{k+1}) \leq \left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^2 E(x_k),$$

showing convergence for the quadratic case.

1.2 Conjugate gradient method

Conjugate gradient methods are increasingly important in the field of iterative methods for solving linear equations. We consider the quadratic function

$$\min \ \frac{1}{2}x^T Ax - b^T x, \tag{1.8}$$

whose minimizer is the solution to the linear system of equations

$$Ax = b.$$

We assume A is symmetric positive definite.

Definition 1.2 *Let B be a symmetric matrix. Two vectors x and y are called B-orthogonal, if $x^T By = y^T Bx = 0$.*

If A is symmetric positive definite, than a system of A-orthogonal vectors is linearly independent. A-orthogonal directions are called *conjugate directions*.

A basis which is A-orthogonal is preferable when one tries to minimize the quadratic function (1.8); note that steepest descent works great, when A is an orthogonal matrix, i.e., when orthogonality and A-orthogonality are the same.

From now, let $d_0, \ldots, d_{n-1}$ denote a set of n nonzero A-orthogonal vectors, and let z be the exact solution to $Az = b$, i.e., $z = A^{-1}b$. The (d_k) form a basis for $\mathbb{R}^n$, therefore

$$z = \sum_{k=0}^{n-1} \alpha_k d_k.$$

Thus,

$$b = Az = \sum_{k=0}^{n-1} \alpha_k A d_k,$$

and, since the (d_k) are A-orthogonal, we obtain

$$\alpha_j = \frac{d_j^T Az}{d_j^T A d_j} = \frac{d_j^T b}{d_j^T A d_j}. \tag{1.9}$$

Formulated as an iterative algorithm with start vector $x_0 \in \mathbb{R}^n$, we obtain

$$\begin{aligned} \alpha_k &= -\frac{g_k^T d_k}{d_k^T A d_k}, \qquad g_k = Ax_k - b & (1.10)\\ x_{k+1} &= x_k + \alpha_k d_k, \qquad k = 0, 1, \ldots, n-1. & (1.11) \end{aligned}$$

We have $x_n = z$, i.e., the algorithm converges to the exact solution in no more than n steps. Note that the α_k in (1.10) are not the same as the ones in (1.9); they are the expansion coefficients of $z - x_0$, i.e.,

$$z - x_0 = \sum_{k=0}^{n-1} \alpha_k d_k,$$

and as above, we have

$$\alpha_j = \frac{d_j^T A(z - x_0)}{d_j^T A d_j} = \frac{d_j^T (b - Ax_0)}{d_j^T A d_j} = -\frac{d_j^T g_0}{d_j^T A d_j}.$$

Noting that

$$x_k - x_0 = \sum_{j=0}^{k-1} \alpha_j d_j$$

is A-orthogonal to d_k, $d_k^T A(x_k - x_0) = 0$, we have

$$\begin{aligned} d_k^T A(z - x_0) &= d_k^T A(z - x_0) - \underbrace{d_k^T A(x_k - x_0)}_{=0} \\ &= d_k^T A(z - x_k) = -d_k^T g_k, \end{aligned}$$

establishing (1.10).

Let us define the k-dimensional subspaces $\mathcal{B}_k := \operatorname{span}\{d_0, \ldots, d_{k-1}\}$. It is clear, that $\mathcal{B}_k \subset \mathcal{B}_{k+1}$.

Theorem 1.3 (Expanding Subspace Theorem) *Let $x_0 \in \mathbb{R}^n$, and let the sequence x_k be defined by (1.10) and (1.11). The vector x_k minimizes $f(x) = \frac{1}{2}x^T Ax - b^T x$ on the line $x_{k-1} + \alpha d_{k-1}$; however, x_k also minimizes $f(x)$ over the whole affine space $x_0 + \mathcal{B}_k$.*

Proof. f is strictly convex, so h, the restriction of f to the affine subspace $x_0 + \mathcal{B}_k$ is also strictly convex. For x_k to be the unique global minimizer of h it is sufficient that $\nabla h(x_k) = 0$ which is equivalent to

$$\nabla f(x_k) \perp \mathcal{B}_k.$$

We prove that $g_k \perp \mathcal{B}_k$ by induction. Since $\mathcal{B}_0$ is empty, the statement is true for $k = 0$. Now we assume that $g_k \perp \mathcal{B}_k$, i.e.,

$$d_i^T g_k = 0 \qquad \text{for } i < k. \tag{1.12}$$

Obviously,

$$g_{k+1} = g_k + \alpha_k A d_k. \tag{1.13}$$

Thus,

$$d_k^T g_{k+1} = d_k^T g_k + \alpha_k d_k^T A d_k = 0,$$

because of (1.10), and, for $i < k$,

$$d_i^T g_{k+1} = \underbrace{d_i^T g_k}_{=0\ (1.12)} + \alpha_k \underbrace{d_i^T A d_k}_{=0\ \text{A-orthogonality}} = 0.$$

Therefore, $g_{k+1} \perp \mathcal{B}_{k+1}$. $\square$

Usually, the conjugate directions d_0, …, d_{n-1} are not given at the outset. We determine them as we go along. We start with

$$d_0 = -g_0;$$

in the absence of any further information it is best to head into the direction of steepest descent. At the next step, we choose d_1 to be the direction of the component of $-g_1$ orthogonal to d_0:

$$d_1 = -g_1 + \beta_0 d_0,$$

with β_0 chosen such that $d_1^T A d_0 = 0$. Then we take for d_2 the component of $-g_2$ which orthogonal to the space spanned by d_0 and d_1, and so on. Note that the gradient $g_k \neq 0$, unless the minimum is reached, and because of (1.12), or $g_k \perp \mathcal{B}_k$, g_k is not in the span of the previous directions, and therefore $d_k \neq 0$.

CONJUGATE GRADIENT ALGORITHM. Start with $x_0 \in \mathbb{R}^n$, set $d_0 = -g_0 = b - Ax_0$. For $k = 0, 1, \ldots, n-1$,

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.14}$$

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k}, \tag{1.15}$$

$$g_{k+1} = Ax_{k+1} - b \tag{1.16}$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{1.17}$$

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}. \tag{1.18}$$

Verification of the algorithm.

(a)

$$\begin{aligned} \operatorname{span}(g_0, g_1, \ldots, g_k) &= \operatorname{span}(d_0, d_1, \ldots, d_k) = \\ \operatorname{span}(g_0, Ag_0, \ldots, A^k g_0) &= \mathcal{K}_k(A, g_0), \end{aligned}$$

where $\mathcal{K}_k(A, g_0)$ denotes the $(k+1)$-dimensional *Krylov* subspace spanned by g_0. This can be shown by induction.

(b) $d_k^T A d_j = 0$ for $j < k$, the d_j are A-orthogonal. This is true for $k = 0$;

$$d_{k+1}^T A d_j = -g_{k+1}^T A d_j + \beta_k d_k^T A d_j.$$

For $j = k$ we obtain zero because of (1.18). For $j < k$, we have

$$Ad_j \in A\mathcal{B}_{j+1} = \mathcal{B}_{j+2} \subseteq \mathcal{B}_{k+1},$$

and $g_{k+1} \perp \mathcal{B}_{k+1}$, and $d_k^T A d_j = 0$ because of the induction hypothesis.

(c)

$$\alpha_k = \frac{g_k^T g_k}{d_k^T A d_k},$$

as

$$-g_k^T d_k = -g_k^T(-g_k + \beta_{k-1} d_{k-1}) = g_k^T g_k.$$

(d)

$$g_{k+1} = g_k + \alpha_k A d_k \qquad \Rightarrow \qquad A d_k = \frac{1}{\alpha_k}\left(g_{k+1} - g_k\right),$$

and therefore

$$g_{k+1}^T A d_k = \frac{1}{\alpha_k} g_{k+1}^T g_{k+1},$$

as $g_{k+1}^T g_k = 0$ ($g_k \in \mathcal{B}_{k+1}$). So we obtain

$$\beta_k = \frac{1}{\alpha_k}\frac{g_{k+1}^T g_{k+1}}{d_k^T A d_k} = -\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

1.3 Convergence of Conjugate Gradient

In the Conjugate gradient algorithm we have

$$x_{k+1} = x_k + \alpha_k d_k,$$

with $d_k \in \mathcal{B}_{k+1} = \mathcal{K}_k(A, g_0)$, therefore

$$x_{k+1} - x_0 = p_k(A) g_0, \qquad g_0 = A x_0 - b,$$

where p_k is a polynomial of degree k. Denoting by z again the exact solution $Az = b$, we obtain

$$\begin{aligned} x_{k+1} - z &= x_0 - z - (x_{k+1} - x_0) \\ &= x_0 - z - p_k(A) A (x_0 - z) \\ &= [I + A p_k(A)]\,(x_0 - z). \end{aligned}$$

Denoting by λ_i and v_i, $i = 1, \ldots, n$, the eigenvalues and (orthonormal) eigenvectors of A, we have

$$x_0 - z = \sum_{i=1}^n \xi_i v_i, \qquad \Rightarrow \qquad A(x_0 - z) = \sum_{i=1}^n \lambda_i \xi_i v_i,$$

and therefore

$$\begin{aligned} E(x_0) &= \frac{1}{2}(x_0 - z)^T A(x_0 - z) = \frac{1}{2}\sum_{i=1}^{n} \lambda_i \xi_i^2 \\ E(x_{k+1}) &= \frac{1}{2}\sum_{i=1}^{n} \left[1 + \lambda_i p_k(\lambda_i)^2\right]^2 \lambda_i \xi_i^2; \end{aligned}$$

since conjugate gradient picks the p_k to minimize the function over the whole subspace $x_0 + \mathcal{B}_k$, we obtain

$$E(x_{k+1}) \leq \max_{1\leq i\leq n} \left[p(\lambda_i)\right]^2 E(x_0), \tag{1.19}$$

for **any** polynomial p of degree $k+1$ with $p(0) = 1$ (see Figures 1.2 and 1.3).

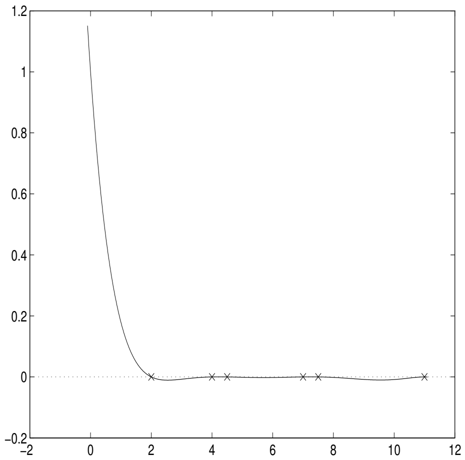

Figure 1.2: Polynomial p_k

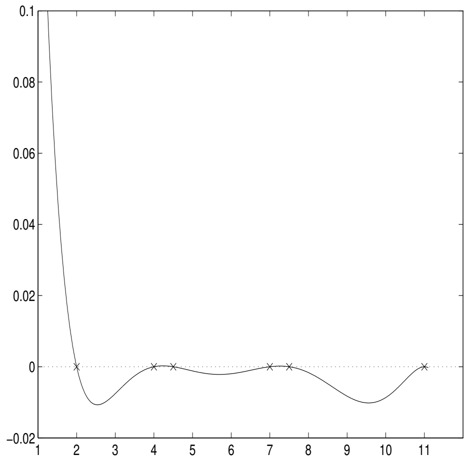

Figure 1.3: Polynomial p_k, blow-up

In particular, we can find a polynomial of degree n which vanishes at all the n eigenvalues λ_i, showing convergence in at most n steps. If A has multiple eigenvalues, we can find a polynomial p of degree less than n which is zero at the eigenvalues, and the algorithm finds the solution sooner. Furthermore, if the eigenvalues occur in just a few, say ℓ, clusters, there is a polynomial of (low) degree $\ell << n$ which takes small values at the eigenvalues, and the corresponding approximation x_ℓ will be close to the exact solution.

By taking the polynomial p in (1.19) to be a suitably scaled Chebyshev polynomial, we arrive at the error estimate

$$E(x_k) \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^{2k} E(x_0).$$

1.4 CG as a polynomial recurrence

Let us quickly restate the Conjugate Gradient Algorithm, slightly rearranging the counters and order of instructions:

$k = 0$; $r_0 = b - Ax_0$; $\rho_{-1} = 1$; $p_{-1} = 0$;
while " $r_k \neq 0$"
    $\rho_k = r_k^T r_k = ||r_k||^2$; $\beta_k = \frac{\rho_k}{\rho_{k-1}}$
    $p_k = r_k + \beta_k p_{k-1}$
    $\sigma_k = p_k^T A p_k$; $\alpha_k = \frac{\rho_k}{\sigma k}$
    $r_{k+1} = r_k - \alpha_k A p_k$
    $x_{k+1} = x_k + \alpha_k p_k$
    $k = k + 1$
end while

The algorithm minimizes the quantity $E_k = (x_k - z)^T A(x_k - z) = r_k^T A^{-1} r_k$. As we have seen in the previous section, we have

$$r_j^T r_k = \rho_j \delta_{jk}, \qquad p_j^T A p_k = \sigma_j \delta_{jk},$$

where δ_{jk} denotes the *Kronecker delta*[1]. We also have found that

$$r_k = \phi_k(A) r_0, \qquad p_k = \psi_k(A) r_0,$$

where ϕ_k and ψ_k are polynomials of degree less than or equal to k. From the algorithm we can obtain recurrences for the polynomials ϕ_k and ψ_k. We initialize

$$\phi_0(t) = 1, \qquad \psi_{-1}(t) = 0;$$

From

$$\begin{aligned} \psi_k(A) r_0 &= p_k = r_k + \beta_k p_{k-1} = \phi_k(A) r_0 + \beta_k \psi_{k-1}(A) r_0 \\ \phi_{k+1}(A) r_0 &= r_{k+1} = r_k - \alpha_k A p_k = \phi_k(A) r_0 - \alpha_k A \psi_k(A) r_0 \end{aligned}$$

[1] $\delta_{jk} = 0$ for $j \neq k$, $\delta_{jj} = 1$

we obtain the recurrrence

$$\begin{aligned} \psi_k(t) &= \phi_k(t) + \beta_k \psi_{k-1}(t) \\ \phi_{k+1}(t) &= \phi_k(t) - \alpha_k t \psi_k(t). \end{aligned}$$

Define the (symmetric) bilinear form

$$(\phi, \psi) = [\phi(A) r_0]^T \, \psi(A) r_0 = r_0^T \phi(A) \psi(A) r_0,$$

(because A is symmetric), and denote by θ the polynomial $\theta(t) = t$. We note that

$$(\phi_j, \phi_k) = \rho_j \delta_{jk}, \qquad (\psi_j, \theta\psi_k) = \sigma_j \delta_{jk}. \tag{1.20}$$

So in this sense, CG generates sequences of orthogonal polynomials. Note, that the bilinear form we have introduced satisfies

$$(\phi\chi, \psi) = (\phi, \chi\psi) \tag{1.21}$$

for any polynomial χ. The bilinear form $(\cdot, \cdot)$ is positive semidefinte.

Now, the polynomial counter part of the conjugate gradient algorithm reads

Polynomial CG ALgorithm

$k = 0$; $\rho_{-1} = 1$; $\phi_0 = 1$; $\psi_{-1} = 0$;
while " $\rho_k > 0$"
  $\rho_k = (\phi_k, \phi_k)$; $\beta_k = \frac{\rho_k}{\rho_{k-1}}$
  $\psi_k = \phi_k + \beta_k \psi_{k-1}$
  $\sigma_k = (\psi_k, \theta\psi_k)$; $\alpha_k = \frac{\rho_k}{\sigma k}$
  $\phi_{k+1} = \phi_k - \alpha_k \theta \psi_k$
  $k = k + 1$
end while

This algorithm minimizes $E_k = (\phi_k, \theta^{-1}\phi_k)$, among all polynomials with $\phi(0) = 1$.

We have the following

Theorem 1.4 *Let $(\cdot, \cdot)$ be any symmetric bilinear form defined on the space of polynomials of degree $\leq n$, satisfying the relation (1.21). Provided the Polynomial CG algorithm does not break down (because of zero division in the computation of the quantities α_k and/or β_k), it will produce "sequences" of "orthogonal" polynomials, i.e., polynomials ϕ_k and ψ_k satisfying (1.20).*

We will later generalize this polynomial conjugate gradient algorithm to obtain the Bi-Conjugate Gradient Algorithm and CGS for nonsymmetric systems.