# Conjugate direction methods:

- a set of vectors, $p_0 \cdots p_n$ is conjugate for A positive definite if

$$p_i' A p_j = 0 \qquad i \neq j$$

- Assume we wish to min

$$\frac{x'Ax}{2} - b'x$$

- useful because:

a) solution to $Ax = b$ for A p.d.

b) $\min_x \|Ax - b\|^2$ is like this

- now write

$$x = \alpha_0 p_0 + \alpha_1 p_1 + \cdots$$

We then have

$$x'Ux + v'x = \alpha_0^2 p_0'Ap_0 + \alpha_1^2 p_1'Ap_1 + \cdots + \alpha_0 v'p_0 + \alpha_1 v'p_1 + \cdots$$

- we have no $p_i'Ap_j$ terms, the basis elements do not interact
- so we could minimize each dim'n independently!
- Conjugate dirns come from Gram-Schmidt

$$u_0 = \frac{e_0}{(e_0'Ae_0)^{1/2}}$$

$$w_1 = e_1 - (u_0'Ae_1)u_0$$

$$u_1 = \frac{w_1}{(w_1'Aw_1)^{1/2}}$$

$$w_2 = e_2 - (w_0'Ae_2)u_{20} - (u_p'Ae_2)u_1$$

$$u_2 = \frac{w_2}{(w_2'Aw_2)^2}$$

Conjugate direction in incremental form

Start with $x_0$, $p_0$

$$x_1 = x_0 + \alpha_0 p_0$$

now min wrt $\alpha_0$

to get

$$\frac{(Ax_0 - b)' p_0}{p_0' A p_0} = \alpha_0$$

write

$$r_k = (Ax_k - b)$$

and get

$$x_{k+1} = x_k + \alpha_k p_k$$

$$\alpha_k = \frac{r_k' p_k}{p_k' A p_k}$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

## properties of conjugate direction

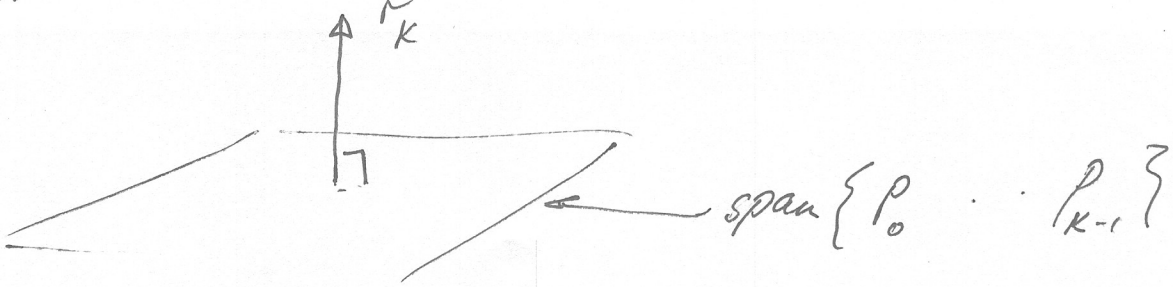
(1)

$$r_k^T \cdot p_i = 0 \qquad \text{for } i < k$$

Proof: easy induction, p 16a.

Significance:

- $r_k$ is $\perp$ to space spanned by $p$'s we have seen
- this means that ~~we have minimized~~ $r_k$ might be a good source of a ~~new~~ direction conj to all other $p$'s.



## props of C.D.

(2) at $k$, we have minimized the quadratic form in span $\{p_0 \dots p_{k-1}\}$.

→ Obvious from Defn of conj. dirs., but v. important.

→ options: - go some steps, then keep $x_k$; this has reduced the Q form value; if we are minimizing, this gives as a descent dir

→ What happens if A is not PD?

⇒ there is some $k+1$ so that the $\alpha_k$ is a max in that dir (or where the g form is constant!)

i.e. we choose

$$x_{k+1} = x_k + \alpha_k p_k$$

$$(x_k + \alpha_k p_k)^T A (x_k + \alpha_k p_k) = \alpha_k^2 (p_k' A p_k) + 2\alpha_k (x_k^T A p_k) + (x_k^T A x_k)$$

now if $p_k' A p_k > 0$, then this has a min otherwise, doesn't.

Strategy: go till we find a bad $\alpha_k$; keep the $x_k$, which is a better point than $x_0$

Minimizing: regard $x_k$ as: offset to current point.

No help in current form because we need to know all the $p_i$'s for a new direction.

But special choices of initial dir, etc can make this unnecessary

Conjugate Gradient:

- want to solve

$$A x = b$$

$A$: symmetric, positive definite

equiv:

$$x = \arg\min \; \frac{x' A x}{2} - b' x$$

We will do this by generating special conj directions.

Start with

$x_0$ estimate of sol'n

$r_0 = Ax_0 - b$ residual

$p_0 = -r_0$ first dir

we will update by:

$$x_{k+1} = x_k + \alpha_k p_k$$

and get $\alpha_k$ by minimizing along $p_k$ alone (OK, because $p_k$'s will be conjugate

Gonin
This gives

$$\frac{1}{2}\left[ (x_k + \alpha_k p_k)' A (x_k + \alpha_k p_k) \right] - b_k (x_k + \alpha_k p_k)$$

min is at:

$$\frac{-(Ax_k - b)' p_k}{p_k' A p_k}$$

write

$$r_k = Ax_k - b$$

so $$\alpha_k = \frac{-r_k' p_k}{p_k' A p_k}$$

# Conjugate gradient (simple form)

Start $x_0$, $r_0 = Ax_0 - b$, $p_0 = -r_0$

Step:

$$x_{k+1} = x_k + \alpha_k p_k$$

$$\alpha_k = \frac{-r_k' p_k}{p_k' A p_k}$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

This is conj. direction ~~trick~~ method

The clever bit →

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$$

$$\beta_{k+1} = \frac{r_{k+1}' A p_k}{p_k^T A p_k}$$

from $p_{k+1}' A p_k = 0$

# Properties of conjugate gradient



(1) $r_k' p_i = 0$ for $i < k$. | assuming $p$ conj.

Proof: easy induction; notice

$r_1' p_0 = 0$ (by defn of $\alpha_0$)

$(= r_0' r_0 + \alpha_0 p_0' A p_0)$

$r_2' p_1 = 0$ ; $r_{k+1} p_k = 0$

then $r_2' p_0 = (r_1 + \alpha_1 A p_1)' p_0 = 0$ (because $p$ are conj.)

(2): $r_k^T r_i = 0$ for $i < k$

Proof: See 5.3 attached

(3) span $\{r_0 \cdots r_k\}$ = span $\{r_0, Ar_0 \cdots A^k r_0\}$

(4) span $\{p_0 \cdots p_k\}$ = Span $\{r_0, Ar_0 \cdots A^k r_0\}$

(5) $\boxed{p_k^T A p_i = 0}$

Conjugate gradient.

Cleaner form:

- By properties, we have

$$\alpha_{k+1} = \frac{r_k' r_k}{p_k' A p_k}$$

- Now $\alpha_k A p_k = r_{k+1} - r_k$

So $$\beta_{k+1} = r_{k+1}' \frac{(r_{k+1} - r_k)}{\alpha_k} \cdot \frac{1}{p_k' A p_k}$$

$$= \frac{r_{k+1}' (r_{k+1} - r_k)}{r_k' r_k}$$

$$= \frac{r_{k+1}' r_{k+1}}{r_k' r_k}$$ (By properties)

# C. G., clean form:

**Start:** $x_0$, $r_0 = Ax_0 - b$, $p_0 = -r$

**Step:**

$$x_{k+1} = x_k + \alpha_k p_k$$

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$$

C.G. , properties.

- if $A$ is $n \times n$, $n$ steps give $A^{-1}$.

- in practice, can get to a good point rather fast; we might successfully stop well before $n$-steps; we might have a small residual early, too.

- if $A$ has $r$ distinct eigenvalues, takes only $r$ steps!

- if evalues form $r$ "clusters", "about" $r$ steps are enough to get v. good estimate.

Potential applications

- Newton, Quasi Newton methods to solve linear alg.
- ~~Newton, where~~
- Large scale Newton, where we get

  $Hp$ with a Directional Derivative

- Direct min. (Polak-Ribiere, etc).
  - Keep doing CG until $x_k$ is a min
  - Some issues to handle here

of $\hat{x}$. Because of the relation (5.9), however, the $i$th coordinate direction in $\hat{x}$-space corresponds to the direction $p_i$ in $x$-space. Hence, the coordinate search strategy applied to $\hat{\phi}$ is equivalent to the conjugate direction algorithm (5.6), (5.7). We conclude, as in Theorem 5.1, that the conjugate direction algorithm terminates in at most $n$ steps.

Returning to Figure 5.1, we note another interesting property: When the Hessian matrix is diagonal, each coordinate minimization correctly determines one of the components of the solution $x^*$. In other words, after $k$ one-dimensional minimizations, the quadratic has been minimized on the subspace spanned by $e_1, e_2, \ldots, e_k$. The following theorem proves this important result for the general case in which the Hessian of the quadratic is not necessarily diagonal. (Here and later, we use the notation $\text{span}\{p_0, p_1, \ldots, p_k\}$ to denote the set of all linear combinations of the vectors $p_0, p_1, \ldots, p_k$.) In proving the result we will make use of the following expression, which is easily verified from the relations (5.4) and (5.6):

$$r_{k+1} = r_k + \alpha_k A p_k. \tag{5.10}$$

**Theorem 5.2** (Expanding Subspace Minimization).

*Let $x_0 \in \mathbb{R}^n$ be any starting point and suppose that the sequence $\{x_k\}$ is generated by the conjugate direction algorithm (5.6), (5.7). Then*

$$r_k^T p_i = 0, \qquad \text{for } i = 0, 1, \ldots, k-1, \tag{5.11}$$

*and $x_k$ is the minimizer of $\phi(x) = \frac{1}{2} x^T A x - b^T x$ over the set*

$$\{x \mid x = x_0 + \text{span}\{p_0, p_1, \ldots, p_{k-1}\}\}. \tag{5.12}$$

PROOF. We begin by showing that a point $\tilde{x}$ minimizes $\phi$ over the set (5.12) if and only if $r(\tilde{x})^T p_i = 0$, for each $i = 0, 1, \ldots, k-1$. Let us define $h(\sigma) = \phi(x_0 + \sigma_0 p_0 + \cdots + \sigma_{k-1} p_{k-1})$, where $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{k-1})^T$. Since $h(\sigma)$ is a strictly convex quadratic, it has a unique minimizer $\sigma^*$ that satisfies

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = 0, \qquad i = 0, 1, \ldots, k-1.$$

By the chain rule, this equation implies that

$$\nabla \phi(x_0 + \sigma_0^* p_0 + \cdots + \sigma_{k-1}^* p_{k-1})^T p_i = 0, \qquad i = 0, 1, \ldots, k-1.$$

By recalling the definition (5.3), we have for the minimizer $\tilde{x} = x_0 + \sigma_0^* p_0 + \sigma_1^* p_2 + \cdots + \sigma_{k-1}^* p_{k-1}$ on the set (5.12) that $r(\tilde{x})^T p_i = 0$, as claimed.

We now use induction to show that $x_k$ satisfies (5.11). For the case $k = 1$, we have from the fact that $x_1 = x_0 + \alpha_0 p_0$ minimizes $\phi$ along $p_0$ that $r_1^T p_0 = 0$. Let us now make

the induction hypothesis, namely, that $r_{k-1}^T p_i = 0$ for $i = 0, 1, \ldots, k-2$. By (5.10), we have

$$r_k = r_{k-1} + \alpha_{k-1} A p_{k-1},$$

so that

$$p_{k-1}^T r_k = p_{k-1}^T r_{k-1} + \alpha_{k-1} p_{k-1}^T A p_{k-1} = 0,$$

by the definition (5.7) of $\alpha_{k-1}$. Meanwhile, for the other vectors $p_i$, $i = 0, 1, \ldots, k-2$, we have

$$p_i^T r_k = p_i^T r_{k-1} + \alpha_{k-1} p_i^T A p_{k-1} = 0,$$

where $p_i^T r_{k-1} = 0$ because of the induction hypothesis and $p_i^T A p_{k-1} = 0$ because of conjugacy of the vectors $p_i$. We have shown that $r_k^T p_i = 0$, for $i = 0, 1, \ldots, k-1$, so the proof is complete. □

The fact that the current residual $r_k$ is orthogonal to all previous search directions, as expressed in (5.11), is a property that will be used extensively in this chapter.

The discussion so far has been general, in that it applies to a conjugate direction method (5.6), (5.7) based on *any* choice of the conjugate direction set $\{p_0, p_1, \ldots, p_{n-1}\}$. There are many ways to choose the set of conjugate directions. For instance, the eigenvectors $v_1, v_2, \ldots, v_n$ of $A$ are mutually orthogonal as well as conjugate with respect to $A$, so these could be used as the vectors $\{p_0, p_1, \ldots, p_{n-1}\}$. For large-scale applications, however, computation of the complete set of eigenvectors requires an excessive amount of computation. An alternative approach is to modify the Gram–Schmidt orthogonalization process to produce a set of conjugate directions rather than a set of orthogonal directions. (This modification is easy to produce, since the properties of conjugacy and orthogonality are closely related in spirit.) However, the Gram–Schmidt approach is also expensive, since it requires us to store the entire direction set.

### BASIC PROPERTIES OF THE CONJUGATE GRADIENT METHOD

The conjugate gradient method is a conjugate direction method with a very special property: In generating its set of conjugate vectors, it can compute a new vector $p_k$ by using only the previous vector $p_{k-1}$. It does *not* need to know all the previous elements $p_0, p_1, \ldots, p_{k-2}$ of the conjugate set; $p_k$ is automatically conjugate to these vectors. This remarkable property implies that the method requires little storage and computation.

In the conjugate gradient method, each direction $p_k$ is chosen to be a linear combination of the negative residual $-r_k$ (which, by (5.3), is the steepest descent direction for the

function $\phi$) and the previous direction $p_{k-1}$. We write

$$p_k = -r_k + \beta_k p_{k-1}, \tag{5.13}$$

where the scalar $\beta_k$ is to be determined by the requirement that $p_{k-1}$ and $p_k$ must be conjugate with respect to $A$. By premultiplying (5.13) by $p_{k-1}^T A$ and imposing the condition $p_{k-1}^T A p_k = 0$, we find that

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

We choose the first search direction $p_0$ to be the steepest descent direction at the initial point $x_0$. As in the general conjugate direction method, we perform successive one-dimensional minimizations along each of the search directions. We have thus specified a complete algorithm, which we express formally as follows:

**Algorithm 5.1** (CG–Preliminary Version).
Given $x_0$;
Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;
**while** $r_k \neq 0$

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k}; \tag{5.14}$$
$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{5.15}$$
$$r_{k+1} \leftarrow Ax_{k+1} - b; \tag{5.16}$$
$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k}; \tag{5.17}$$
$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{5.18}$$
$$k \leftarrow k + 1; \tag{5.19}$$

**end (while)**

This version is useful for studying the essential properties of the conjugate gradient method, but we present a more efficient version later. We show first that the direction $p_0, p_1, \ldots, p_{n-1}$ are indeed conjugate, which by Theorem 5.1 implies termination in $n$ steps. The theorem below establishes this property and two other important properties. First, the residuals $r_i$ are mutually orthogonal. Second, each search direction $p_k$ and residual $r_k$ is contained in the *Krylov subspace of degree k for* $r_0$, defined as

$$\mathcal{K}(r_0; k) \stackrel{\text{def}}{=} \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}. \tag{5.20}$$

**Theorem 5.3.**

*Suppose that the k th iterate generated by the conjugate gradient method is not the solution point $x^*$. The following four properties hold:*

$$r_k^T r_i = 0, \qquad \text{for } i = 0, 1, \ldots, k-1, \tag{5.16}$$

$$\text{span}\{r_0, r_1, \ldots, r_k\} = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}, \tag{5.17}$$

$$\text{span}\{p_0, p_1, \ldots, p_k\} = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}, \tag{5.18}$$

$$p_k^T A p_i = 0, \qquad \text{for } i = 0, 1, \ldots, k-1. \tag{5.19}$$

*Therefore, the sequence $\{x_k\}$ converges to $x^*$ in at most n steps.*

PROOF. The proof is by induction. The expressions (5.17) and (5.18) hold trivially for $k = 0$, while (5.19) holds by construction for $k = 1$. Assuming now that these three expressions are true for some $k$ (the induction hypothesis), we show that they continue to hold for $k + 1$.

To prove (5.17), we show first that the set on the left-hand side is contained in the set on the right-hand side. Because of the induction hypothesis, we have from (5.17) and (5.18) that

$$r_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}, \qquad p_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\},$$

while by multiplying the second of these expressions by $A$, we obtain

$$Ap_k \in \text{span}\{Ar_0, \ldots, A^{k+1} r_0\}. \tag{5.20}$$

By applying (5.10), we find that

$$r_{k+1} \in \text{span}\{r_0, Ar_0, \ldots, A^{k+1} r_0\}.$$

By combining this expression with the induction hypothesis for (5.17), we conclude that

$$\text{span}\{r_0, r_1, \ldots, r_k, r_{k+1}\} \subset \text{span}\{r_0, Ar_0, \ldots, A^{k+1} r_0\}.$$

To prove that the reverse inclusion holds as well, we use the induction hypothesis on (5.18) to deduce that

$$A^{k+1} r_0 = A(A^k r_0) \in \text{span}\{Ap_0, Ap_1, \ldots, Ap_k\}.$$

Since by (5.10) we have $Ap_i = (r_{i+1} - r_i)/\alpha_i$ for $i = 0, 1, \ldots, k$, it follows that

$$A^{k+1} r_0 \in \text{span}\{r_0, r_1, \ldots, r_{k+1}\}.$$

By combining this expression with the induction hypothesis for (5.17), we find that

$$\text{span}\{r_0, Ar_0, \ldots, A^{k+1}r_0\} \subset \text{span}\{r_0, r_1, \ldots, r_k, r_{k+1}\}.$$

Therefore, the relation (5.17) continues to hold when $k$ is replaced by $k+1$, as claimed.

We show that (5.18) continues to hold when $k$ is replaced by $k+1$ by the following argument:

$$\begin{aligned}
&\text{span}\{p_0, p_1, \ldots, p_k, p_{k+1}\} \\
&\quad = \text{span}\{p_0, p_1, \ldots, p_k, r_{k+1}\} && \text{by (5.14e)} \\
&\quad = \text{span}\{r_0, Ar_0, \ldots, A^k r_0, r_{k+1}\} && \text{by induction hypothesis for (5.18)} \\
&\quad = \text{span}\{r_0, r_1, \ldots, r_k, r_{k+1}\} && \text{by (5.17)} \\
&\quad = \text{span}\{r_0, Ar_0, \ldots, A^{k+1}r_0\} && \text{by (5.17) for } k+1.
\end{aligned}$$

Next, we prove the conjugacy condition (5.19) with $k$ replaced by $k+1$. By multiplying (5.14e) by $Ap_i$, $i = 0, 1, \ldots, k$, we obtain

$$p_{k+1}^T A p_i = -r_{k+1}^T A p_i + \beta_{k+1} p_k^T A p_i. \tag{5.21}$$

By the definition (5.14d) of $\beta_k$, the right-hand-side of (5.21) vanishes when $i = k$. For $i \leq k-1$ we need to collect a number of observations. Note first that our induction hypothesis for (5.19) implies that the directions $p_0, p_1, \ldots, p_k$ are conjugate, so we can apply Theorem 5.2 to deduce that

$$r_{k+1}^T p_i = 0, \qquad \text{for } i = 0, 1, \ldots, k. \tag{5.22}$$

Second, by repeatedly applying (5.18), we find that for $i = 0, 1, \ldots, k-1$, the following inclusion holds:

$$\begin{aligned}
Ap_i \in A\,\text{span}\{r_0, Ar_0, \ldots, A^i r_0\} &= \text{span}\{Ar_0, A^2 r_0, \ldots, A^{i+1} r_0\} \\
&\subset \text{span}\{p_0, p_1, \ldots, p_{i+1}\}.
\end{aligned} \tag{5.23}$$

By combining (5.22) and (5.23), we deduce that

$$r_{k+1}^T A p_i = 0, \qquad \text{for } i = 0, 1, \ldots, k-1,$$

so the first term in the right-hand-side of (5.21) vanishes for $i = 0, 1, \ldots, k-1$. Because of the induction hypothesis for (5.19), the second term vanishes as well, and we

conclude that $p_{k+1}^T A p_i = 0, i = 0, 1, \ldots, k$. Hence, the induction argument holds for (5.19) also.

It follows that the direction set generated by the conjugate gradient method is indeed a conjugate direction set, so Theorem 5.1 tells us that the algorithm terminates in at most $n$ iterations.

Finally, we prove (5.16) by a noninductive argument. Because the direction set is conjugate, we have from (5.11) that $r_k^T p_i = 0$ for all $i = 0, 1, \ldots, k-1$ and any $k = 1, 2, \ldots, n-1$. By rearranging (5.14e), we find that

$$p_i = -r_i + \beta_i p_{i-1},$$

so that $r_i \in \text{span}\{p_i, p_{i-1}\}$ for all $i = 1, \ldots, k-1$. We conclude that $r_k^T r_i = 0$ for all $i = 1, \ldots, k-1$. To complete the proof, we note that $r_k^T r_0 = -r_k^T p_0 = 0$, by definition of $p_0$ in Algorithm 5.1 and by (5.11). □

The proof of this theorem relies on the fact that the first direction $p_0$ is the steepest descent direction $-r_0$; in fact, the result does not hold for other choices of $p_0$. Since the gradients $r_k$ are mutually orthogonal, the term "conjugate gradient method" is actually a misnomer. It is the search directions, not the gradients, that are conjugate with respect to $A$.

### A PRACTICAL FORM OF THE CONJUGATE GRADIENT METHOD

We can derive a slightly more economical form of the conjugate gradient method by using the results of Theorems 5.2 and 5.3. First, we can use (5.14e) and (5.11) to replace the formula (5.14a) for $\alpha_k$ by

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}.$$

Second, we have from (5.10) that $\alpha_k A p_k = r_{k+1} - r_k$, so by applying (5.14e) and (5.11) once again we can simplify the formula for $\beta_{k+1}$ to

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

By using these formulae together with (5.10), we obtain the following standard form of the conjugate gradient method.

**Algorithm 5.2** (CG).
Given $x_0$;
Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;
**while** $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}; \tag{5.24a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{5.24b}$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \tag{5.24c}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}; \tag{5.24d}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{5.24e}$$

$$k \leftarrow k + 1; \tag{5.24f}$$

**end (while)**

At any given point in Algorithm 5.2 we never need to know the vectors $x$, $r$, and $p$ for more than the last two iterations. Accordingly, implementations of this algorithm overwrite old values of these vectors to save on storage. The major computational tasks to be performed at each step are computation of the matrix–vector product $Ap_k$, calculation of the inner products $p_k^T(Ap_k)$ and $r_{k+1}^T r_{k+1}$, and calculation of three vector sums. The inner product and vector sum operations can be performed in a small multiple of $n$ floating-point operations, while the cost of the matrix–vector product is, of course, dependent on the problem. The CG method is recommended only for large problems; otherwise, Gaussian elimination or other factorization algorithms such as the singular value decomposition are to be preferred, since they are less sensitive to rounding errors. For large problems, the CG method has the advantage that it does not alter the coefficient matrix and (in contrast to factorization techniques) does not produce fill in the arrays holding the matrix. Another key property is that the CG method sometimes approaches the solution quickly, as we discuss next.

## RATE OF CONVERGENCE

We have seen that in exact arithmetic the conjugate gradient method will terminate at the solution in at most $n$ iterations. What is more remarkable is that when the distribution of the eigenvalues of $A$ has certain favorable features, the algorithm will identify the solution in many fewer than $n$ iterations. To explain this property, we begin by viewing the expanding subspace minimization property proved in Theorem 5.2 in a slightly different way, using it to show that Algorithm 5.2 is optimal in a certain important sense.