

6.4 Krylov Subspaces and Conjugate Gradients

Our original equation is $Ax = b$. *The preconditioned equation is* $P^{-1}Ax = P^{-1}b$. When we write P^{-1}, we never intend that an inverse will be explicitly computed. P may come from Incomplete LU, or a few steps of a multigrid iteration, or "domain decomposition." Entirely new preconditioners are waiting to be invented.

The residual is $\boldsymbol{r_k = b - Ax_k}$**.** This is the error in $Ax = b$, not the error in x itself. An ordinary preconditioned iteration corrects x_k by the vector $P^{-1}r_k$:

$$Px_{k+1} = (P-A)x_k + b \quad \text{or} \quad Px_{k+1} = Px_k + r_k \quad \text{or} \quad \boldsymbol{x_{k+1} = x_k + P^{-1}r_k}\,. \tag{1}$$

In describing Krylov subspaces, I should work with $P^{-1}A$. **For simplicity I will only write** A! I am assuming that P has been chosen and used, and the preconditioned equation $P^{-1}Ax = P^{-1}b$ is given the notation $Ax = b$. The preconditioner is now $P = I$. Our new A is probably better than the original matrix with that name.

With $x_1 = b$, look first at two steps of the pure iteration $x_{j+1} = (I - A)x_j + b$:

$$x_2 = (I-A)b + b = \boldsymbol{2b - Ab} \qquad x_3 = (I-A)x_1 + b = \boldsymbol{3b - 3Ab + A^2b}. \tag{2}$$

My point is simple but important: $\boldsymbol{x_j}$ ***is a combination of*** $\boldsymbol{b, Ab, \ldots, A^{j-1}b}$**.** We can compute those vectors quickly, multiplying each time by a sparse A. Every iteration involves only one matrix-vector multiplication. Krylov gave a name to ***all*** combinations of those vectors, and he suggested that there might be better combinations than the particular choices x_j in (2).

Usually a different combination will come closer to the desired $x = A^{-1}b$.

Krylov subspaces

The linear combinations of $b, Ab, \ldots, A^{j-1}b$ form the jth Krylov subspace. This space depends on A and b. Following convention, I will write $\boldsymbol{\mathcal{K}_j}$ for that subspace and $\mathbf{K_j}$ for the matrix with those basis vectors in its columns:

$$\begin{aligned} &\textbf{Krylov matrix} \quad \mathbf{K_j} = [\ \boldsymbol{b} \quad \boldsymbol{Ab} \quad \boldsymbol{A^2b} \quad \ldots \quad \boldsymbol{A^{j-1}b}\ ]\,. \\ &\textbf{Krylov subspace} \quad \boldsymbol{\mathcal{K}_j} = \boldsymbol{\text{all combinations of } b, Ab, \ldots, A^{j-1}b}. \end{aligned} \tag{3}$$

Thus $\boldsymbol{\mathcal{K}_j}$ is the column space of $\mathbf{K_j}$. We want to choose the ***best combination*** as our improved x_j. Various definitions of "best" will give various x_j. Here are four different approaches to choosing a good x_j in $\boldsymbol{\mathcal{K}_j}$—this is the important decision:

1. The residual $r_j = b - Ax_j$ is orthogonal to $\boldsymbol{\mathcal{K}_j}$ (**Conjugate Gradients**).

2. The residual r_j has minimum norm for x_j in $\boldsymbol{\mathcal{K}_j}$ (**GMRES** and **MINRES**).

3. r_j is orthogonal to a different space $\boldsymbol{\mathcal{K}_j}(A^{\mathrm{T}})$ (**BiConjugate Gradients**).

4. The error e_j has minimum norm (**SYMMLQ**).

In every case we hope to compute the new x_j quickly from the earlier x's. If that step only involves x_{j-1} and x_{j-2} (**short recurrence**) it is especially fast. Short recurrences happen for conjugate gradients and symmetric positive definite A. The BiCG method is a natural extension of short recurrences to unsymmetric A (using two Krylov spaces). A stabilized version called **BiCGStab** chooses x_j in $A^{\mathrm{T}}\boldsymbol{\mathcal{K}_j}(A^{\mathrm{T}})$.

As always, computing x_j can be very unstable until we choose a decent basis.

Vandermonde Example

To follow each step of orthogonalizing the basis, and solving $Ax = b$ by conjugate gradients, we need a good example. It has to stay simple! I am happy with this one:

$$A = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 3 & \\ & & & 4 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad Ab = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \qquad A^{-1}b = \begin{bmatrix} 1/1 \\ 1/2 \\ 1/3 \\ 1/4 \end{bmatrix}. \tag{4}$$

That constant vector b spans the Krylov subspace $\boldsymbol{\mathcal{K}_1}$. Then Ab, A^2b, and A^3b are the other basis vectors in $\boldsymbol{\mathcal{K}_4}$. They are the columns of $\mathbf{K_4}$, which we will name V:

Vandermonde matrix $$\mathbf{K_4} = V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}. \tag{5}$$

Those columns are constant, linear, quadratic, and cubic. The column vectors are independent but not at all orthogonal. The best measure of non-orthogonality starts by computing *inner products of the columns* in the matrix $V^{\mathrm{T}}V$. When columns are orthonormal, their inner products are 0 or 1. (The matrix is called Q, and the inner products give $Q^{\mathrm{T}}Q = I$.) Here $V^{\mathrm{T}}V$ is far from the identity matrix!

$$V^{\mathrm{T}}V = \begin{bmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{bmatrix} \qquad \begin{aligned} 10 &= 1 + 2 + 3 + 4 \\ 30 &= 1^2 + 2^2 + 3^2 + 4^2 \\ 100 &= 1^3 + 2^3 + 3^3 + 4^3 \\ 1300 &= 1^5 + 2^5 + 3^5 + 4^5 \end{aligned}$$

The eigenvalues of this inner product matrix (*Gram matrix*) tell us something important. The extreme eigenvalues are $\lambda_{\max} \approx 5264$ and $\lambda_{\min} \approx .004$. Those are the squares of σ_4 and σ_1, the largest and smallest **singular values of $\boldsymbol{V}$**. The key measure is their ratio σ_4/σ_1, the **condition number of $\boldsymbol{V}$**:

$$\mathrm{cond}(V^{\mathrm{T}}V) \approx \frac{5264}{.004} \approx 10^6 \qquad \mathrm{cond}(V) = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \approx 1000\,.$$

For such a small example, 1000 is a poor condition number. For an orthonormal basis with $Q^{\mathrm{T}}Q = I$, all eigenvalues = singular values = condition number = 1.

We could improve the condition by rescaling the columns of V to unit vectors. Then $V^{\mathrm{T}}V$ has ones on the diagonal, and the condition number drops to 263. But when the matrix size is realistically large, that rescaling will not save us. In fact we could extend this Vandermonde model from constant, linear, quadratic, and cubic vectors to the functions $1, x, x^2, x^3$. (A multiplies by x.) Please look at what happens!

Continuous Vandermonde matrix
$$V_c = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix}. \tag{6}$$

Again, those four functions are far from orthogonal. The inner products in $V_c^{\mathrm{T}}V_c$ change from sums to *integrals*. Working on the interval from 0 to 1, the integrals are $\int_0^1 x^i x^j\,dx = 1/(i+j-1)$. They appear in the ***Hilbert matrix***:

Continuous inner products
$$V_c^{\mathrm{T}}V_c = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}. \tag{7}$$

The extreme eigenvalues of this Hilbert matrix are $\lambda_{\max} \approx 1.5$ and $\lambda_{\min} \approx 10^{-4}$. As always, those are the squares of the singular values $\sigma_{\max}$ and $\sigma_{\min}$ of V_c. *The condition number of the power basis* $1, x, x^2, x^3$ *is the ratio* $\sigma_{\max}/\sigma_{\min} \approx 125$. If you want a more impressive number (a numerical disaster), go up to x^9. The condition number of the 10 by 10 Hilbert matrix is $\lambda_{\max}/\lambda_{\min} \approx 10^{13}$, and $1, x, \ldots, x^9$ is a very poor basis.

To reduce that unacceptably large number, Legendre orthogonalized the basis. He chose the interval from -1 to 1, so that even powers would be automatically orthogonal to odd powers. The first ***Legendre polynomials*** are $1, x, x^2 - \frac{1}{3}, x^3 - \frac{3}{5}x$. Our point is that the Vandermonde matrix example (as we follow it below) will be completely parallel to the famous functions of Legendre.

In particular, the **three-term recurrence** in the Arnoldi-Lanczos orthogonalization is exactly like Legendre's classical three-term recurrence for his polynomials. They appear for the same reason—the symmetry of A.

Orthogonalizing the Krylov Basis

The best basis $q_1, \ldots, q_j$ for the Krylov subspace $\boldsymbol{\mathcal{K}_j}$ is orthonormal. Each new q_j comes from orthogonalizing $t = Aq_{j-1}$ to the basis vectors $q_1, \ldots, q_{j-1}$ that are already chosen. The iteration to compute these orthonormal q's is **Arnoldi's method**.

This method is essentially the Gram-Schmidt idea (called *modified* Gram-Schmidt when we subtract the projections of t onto the q's one at a time, for numerical stability). We display one Arnoldi cycle for the Vandermonde example that has $b = [1 \quad 1 \quad 1 \quad 1]'$ and $A = \texttt{diag}([1 \quad 2 \quad 3 \quad 4])$:

Arnoldi's orthogonalization of $\boldsymbol{b, Ab, \ldots, A^{n-1}b}$**:**

0	$\boldsymbol{q_1} = b/\|b\|$;	% Normalize b to $\|q_1\| = 1$	$\boldsymbol{q_1} = [1 \quad 1 \quad 1 \quad 1]'/2$
	for $j = 1, \ldots, n-1$	% Start computation of q_{j+1}	
1	$t = \boldsymbol{Aq_j}$;	% one matrix multiplication	$\boldsymbol{Aq_1} = [1 \quad 2 \quad 3 \quad 4]'/2$
	for $i = 1, \ldots, j$	% t is in the space $\boldsymbol{\mathcal{K}_{j+1}}$	
2	$\boldsymbol{h_{ij}} = q_i^{\mathrm{T}} t$;	% $h_{ij} q_i$ = projection of t on q_i	$\boldsymbol{h_{11}} = 5/2$
3	$\boldsymbol{t} = t - h_{ij} q_i$;	% Subtract that projection	$\boldsymbol{t} = [-3 \quad -1 \quad 1 \quad 3]'/4$
	end;	% t is orthogonal to $q_1, \ldots, q_j$	
4	$\boldsymbol{h_{j+1,j}} = \|t\|$;	% Compute the length of t	$\boldsymbol{h_{21}} = \sqrt{5}/2$
5	$\boldsymbol{q_{j+1}} = t/h_{j+1,j}$;	% Normalize t to $\|q_{j+1}\| = 1$	$\boldsymbol{q_2} = [-3 \quad -1 \quad 1 \quad 3]'/\sqrt{20}$
	end	% $q_1, \ldots, q_n$ are orthonormal	

You might like to see the four orthonormal vectors in the Vandermonde example. Those columns q_1, q_2, q_3, q_4 of Q are still constant, linear, quadratic, and cubic. I can also display the matrix H of numbers h_{ij} that produced the q's from the Krylov vectors b, Ab, A^2b, A^3b. (Since Arnoldi stops at $j = n-1$, the last column of H is not actually computed. It comes from a final command $H(:, n) = Q' * A * Q(:, n)$.) This H turns out to be *symmetric and tridiagonal*, and we will look for a reason.

Arnoldi's method for the Vandermonde example V gives Q and H:

$$Q = \begin{bmatrix} 1 & -3 & 1 & -1 \\ 1 & -1 & -1 & 3 \\ 1 & 1 & -1 & -3 \\ 1 & 3 & 1 & 1 \end{bmatrix} \begin{matrix} \frac{}{2} & \frac{}{\sqrt{20}} & \frac{}{2} & \frac{}{\sqrt{20}} \end{matrix} \qquad H = \begin{bmatrix} 5/2 & \sqrt{5}/2 & & \\ \sqrt{5}/2 & 5/2 & \sqrt{.80} & \\ & \sqrt{.80} & 5/2 & \sqrt{.45} \\ & & \sqrt{.45} & 5/2 \end{bmatrix}$$

Please notice that H is not upper triangular. The usual QR factorization of the original Krylov matrix $\mathbf{K}$ (which was V in our example) has this same Q, but Arnoldi does not compute R. Even though the underlying idea copies Gram-Schmidt (at every step q_{j+1} is a unit vector orthogonal to the previous j columns), there is a difference. The vector t that Arnoldi orthogonalizes against the previous $q_1, \ldots, q_j$ is $t = Aq_j$. This is not column $j+1$ of $\mathbf{K}$, as in Gram-Schmidt. Arnoldi is factoring AQ!

Arnoldi factorization $AQ = QH$ for the final subspace $\mathcal{K}_n$:

$$AQ = \begin{bmatrix} & & \\ Aq_1 & \cdots & Aq_n \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ q_1 & \cdots & q_n \\ & & \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & \cdot & h_{1n} \\ h_{21} & h_{22} & \cdot & h_{2n} \\ \mathbf{0} & h_{23} & \cdot & \cdot \\ \mathbf{0} & \mathbf{0} & \cdot & h_{nn} \end{bmatrix}. \tag{8}$$

This matrix H is upper triangular plus one lower diagonal, which makes it "*upper Hessenberg.*" The h_{ij} in step **2** go down column j as far as the diagonal. Then

$h_{j+1,j}$ in step **4** is below the diagonal. We check that the first column of $AQ = QH$ (multiplying by columns) is Arnoldi's first cycle that produces q_2:

$$\textbf{Column 1} \quad Aq_1 = h_{11}q_1 + h_{21}q_2 \qquad \text{which is} \qquad q_2 = (Aq_1 - h_{11}q_1)/h_{21}\,. \tag{9}$$

That subtraction is step **3** in Arnoldi's algorithm. The division by h_{21} is step **5**.

Unless more of the h_{ij} are zero, the cost is increasing at every iteration. The vector updates in step **3** for $j = 1, \dots, n-1$ give nearly $n^2/2$ updates and $n^3/2$ flops. A *short recurrence* means that most of these h_{ij} are zero, and the count of floating point operations drops to $O(n^2)$. That happens when $A = A^{\mathrm{T}}$.

Arnoldi Becomes Lanczos

The matrix H is symmetric and therefore tridiagonal when A is symmetric. This fact is the foundation of conjugate gradients. For a matrix proof, multiply $AQ = QH$ by Q^{T}. The left side $Q^{\mathrm{T}}AQ$ is always symmetric when A is symmetric. Since H has only one lower diagonal, it has only one upper diagonal. This tridiagonal H has only three nonzeros in its rows and columns. So computing q_{j+1} only involves q_j and q_{j-1}:

$$\textbf{Arnoldi when } \boldsymbol{A = A^{\mathrm{T}}} \qquad Aq_j = h_{j+1,j}\,q_{j+1} + h_{j,j}\,q_j + h_{j-1,j}\,q_{j-1}\,. \tag{10}$$

This is the ***Lanczos iteration***. Each new $q_{j+1} = (Aq_j - h_{j,j}q_j - h_{j-1,j}q_{j-1})/h_{j+1,j}$ involves one multiplication Aq_j, two dot products for h's, and two vector updates.

Allow me an important comment on the *symmetric eigenvalue problem* $Ax = \lambda x$. We have seen that $H = Q^{\mathrm{T}}AQ$ is tridiagonal, and $Q^{\mathrm{T}} = Q^{-1}$ from the orthogonality $Q^{\mathrm{T}}Q = I$. The matrix $H = Q^{-1}AQ$ has the same eigenvalues as A:

$$\textbf{Same } \boldsymbol{\lambda} \qquad Hy = Q^{-1}AQy = \lambda y \quad \text{gives} \quad Ax = \lambda x \;\; \text{with} \;\; x = Qy\,. \tag{11}$$

It is much easier to find the eigenvalues λ for a tridiagonal H than for the original A.

For a *large* symmetric matrix, we often stop the Arnoldi-Lanczos iteration at a tridiagonal H_k with $k < n$. The full n-step process to reach H_n is too expensive, and often we don't need all n eigenvalues. So we compute the k eigenvalues of H_k instead of the n eigenvalues of H. These computed $\lambda_{1k}, \dots, \lambda_{kk}$ (sometimes called Ritz values) can provide good approximations to the first k eigenvalues of A. And we have an excellent start on the eigenvalue problem for H_{k+1}, if we decide to take a further step.

This **Lanczos method** will find, approximately and iteratively and quickly, the leading eigenvalues of a large symmetric matrix. For its inner loop (the eigenvalues of the tridiagonal H_k) we use the "QR method" described in section ____.

The Conjugate Gradient Method

We return to iterative methods for $Ax = b$. The Arnoldi algorithm produced orthonormal basis vectors $q_1, q_2, \ldots$ for the growing Krylov subspaces $\boldsymbol{\mathcal{K}_1}, \boldsymbol{\mathcal{K}_2}, \ldots$. Now we select vectors x_k in $\boldsymbol{\mathcal{K}_k}$ that approach the exact solution to $Ax = b$.

We concentrate on the *conjugate gradient method when A is symmetric positive definite*. Symmetry gives a short recurrence. Definiteness prevents division by zero.

The rule for x_k in conjugate gradients is that the residual $r_k = b - Ax_k$ should be orthogonal to all vectors in $\boldsymbol{\mathcal{K}_k}$. Since r_k will be in $\boldsymbol{\mathcal{K}_{k+1}}$, it must be a multiple of Arnoldi's next vector q_{k+1}! Each residual is therefore orthogonal to all previous residuals (which are multiples of the previous q's):

Orthogonal residuals $$r_i^{\mathrm{T}} r_k = 0 \quad \text{for} \quad i < k\,. \tag{12}$$

The difference between r_k and q_{k+1} is that the q's are normalized, as in $q_1 = b/\|b\|$.

Similarly r_{k-1} is a multiple of q_k. Then the difference $r_k - r_{k-1}$ is orthogonal to each subspace $\boldsymbol{\mathcal{K}_i}$ with $i < k$. Certainly $x_i - x_{i-1}$ lies in that $\boldsymbol{\mathcal{K}_i}$. So Δr is orthogonal to earlier Δx's:

$$(x_i - x_{i+1})^{\mathrm{T}}(r_k - r_{k-1}) = 0 \quad \text{for} \quad i < k\,. \tag{13}$$

These differences Δx and Δr are directly connected, because the b's cancel in Δr:

$$r_k - r_{k-1} = (b - Ax_k) - (b - Ax_{k-1}) = -A(x_k - x_{k-1})\,. \tag{14}$$

Substituting (14) into (13), ***the updates Δx are "A-orthogonal" or conjugate***:

Conjugate directions $$(x_i - x_{i-1})^{\mathrm{T}} A(x_k - x_{k-1}) = 0 \quad \text{for } i < k\,. \tag{15}$$

Now we have all the requirements. Each conjugate gradient step ends with a "search direction" d_{k-1} for the next update $x_k - x_{k-1}$. Steps **1** and **2** compute the correct multiple $\alpha_k d_{k-1}$ to move to x_k. Using (14), step **3** finds the new r_k. Steps **4** and **5** orthogonalize r_k against the search direction just used, to find the next d_k.

The constants β_k in the search direction and α_k in the update come from (12) and (13) for $i = k-1$. For symmetric A, orthogonality will be automatic for $i < k-1$, as in Arnoldi. We have a "short recurrence" for the new x_k and r_k.

Here is one cycle of the algorithm, starting from $x_0 = 0$ and $r_0 = b$ and $d_0 = r_0$. Steps **1** and **3** involve the same matrix-vector multiplication Ad.

Conjugate Gradient Method for Symmetric Positive Definite A

Example: $\boldsymbol{A} = \texttt{diag}([1\ \ 2\ \ 3\ \ 4])$ **and** $\boldsymbol{b} = [1\ \ 1\ \ 1\ \ 1]'$

1	$\boldsymbol{\alpha_k} = r_{k-1}^{\mathrm{T}} r_{k-1} / d_{k-1}^{\mathrm{T}} A d_{k-1}$	% Step length to next x_k	$\boldsymbol{\alpha_1} = 4/10 = 2/5$
2	$\boldsymbol{x_k} = x_{k-1} + \alpha_k d_{k-1}$	% Approximate solution	$\boldsymbol{x_1} = [2\ \ 2\ \ 2\ \ 2]'/5$
3	$\boldsymbol{r_k} = r_{k-1} - \alpha_k A d_{k-1}$	% New residual from (14)	$\boldsymbol{r_1} = [3\ \ 1\ \ -1\ \ -3]'/5$
4	$\boldsymbol{\beta_k} = r_k^{\mathrm{T}} r_k / r_{k-1}^{\mathrm{T}} r_{k-1}$	% Improvement this step	$\boldsymbol{\beta_1} = 1/5$
5	$\boldsymbol{d_k} = r_k + \beta_k d_{k-1}$	% Next search direction	$\boldsymbol{d_1} = [4\ \ 2\ \ 0\ \ -2]'/5$

The formulas for α_k and β_k are explained briefly below—and fully by Trefethen-Bau [–] and Shewchuk [–] and many other good references.

When there is a preconditioner P (to use even fewer CG steps for an accurate x), step **3** uses $P^{-1}A$ and the inner products in steps **1** and **4** include an extra factor P^{-1}.

Different Viewpoints on Conjugate Gradients

I want to describe the (same!) conjugate gradient method in two different ways:

1. It solves a tridiagonal system $Hy = f$ recursively, for Arnoldi's H.

2. It minimizes the energy $\frac{1}{2}x^{\mathrm{T}}Ax - x^{\mathrm{T}}b$ recursively. This is important.

How does $Ax = b$ change to the tridiagonal $Hy = f$? Those are connected by Arnoldi's orthonormal columns $q_1, \dots, q_n$ in Q, with $Q^{\mathrm{T}} = Q^{-1}$ and $Q^{\mathrm{T}}AQ = H$:

$$Ax = b \quad \text{is} \quad (Q^{\mathrm{T}}AQ)(Q^{\mathrm{T}}x) = Q^{\mathrm{T}}b \quad \text{which is} \quad Hy = f = (\|b\|, 0, \dots, 0)\,. \tag{16}$$

Since q_1 is $b/\|b\|$, the first component of $f = Q^{\mathrm{T}}b$ is $q_1^{\mathrm{T}}b = \|b\|$. The other components of f are $q_i^{\mathrm{T}}b = 0$ because q_i is orthogonal to q_1. The conjugate gradient method is implicitly computing the symmetric tridiagonal H. When the method finds x_k, it also finds $y_k = Q_k^{\mathrm{T}}x_k$ (but it doesn't say so). Here is the third step:

Tridiagonal system $Hy = f$ Implicitly solved by CG

$$H_3y_3 = \begin{bmatrix} h_{11} & h_{12} & \\ h_{21} & h_{22} & h_{23} \\ & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \\ y_3 \\ \\ \end{bmatrix} = \begin{bmatrix} \|b\| \\ 0 \\ 0 \end{bmatrix}. \tag{17}$$

This is the equation $Ax = b$ *projected by* Q_3 onto the third Krylov subspace $\boldsymbol{\mathcal{K}_3}$.

These h's never appear in conjugate gradients. We don't want to do Arnoldi too! It is the LDL^{T} factors of H that CG is somehow computing—two new numbers α and β at each step. Those give a fast update from y_{k-1} to y_k. The iterates $x_k = Q_k y_k$ from conjugate gradients approach the exact solution $x_n = Q_n y_n$ which is $x = A^{-1}b$.

Energy By seeing conjugate gradients as an energy minimizing algorithm, we can extend it to nonlinear problems and use it in optimization. For our linear equation $Ax = b$, the energy is $E(x) = \frac{1}{2}x^{\mathrm{T}}Ax - x^{\mathrm{T}}b$. Minimizing $E(x)$ is the same as solving $Ax = b$, when A is positive definite—this was the main point of Section 1.6. **The CG iteration minimizes $\boldsymbol{E(x)}$ on the growing Krylov subspaces**.

The first subspace $\boldsymbol{\mathcal{K}_1}$ is the line in the direction $d_0 = r_0 = b$. Minimization of the energy $E(x)$ for the vectors $x = \alpha b$ produces the number α_1:

$$E(\alpha b) = \frac{1}{2}\alpha^2 b^{\mathrm{T}}Ab - \alpha b^{\mathrm{T}}b \quad \text{is minimized at} \quad \alpha_1 = \frac{b^{\mathrm{T}}b}{b^{\mathrm{T}}Ab}\,. \tag{18}$$

This α_1 is the constant chosen in step **1** of the first conjugate gradient cycle.

The gradient of $E(x) = \frac{1}{2}x^{\mathrm{T}}Ax - x^{\mathrm{T}}b$ is exactly $Ax - b$. *The steepest descent direction at x_1 is along the negative gradient, which is r_1!* This sounds like the perfect direction d_1 for the next move. But the great difficulty with steepest descent is that this r_1 can be too close to the first direction d_0. Little progress that way. So step **5** adds the right multiple $\beta_1 d_0$, in order that the new $d_1 = r_1 + \beta_1 d_0$ will be A-orthogonal to the first direction d_0.

Then we move in this conjugate direction d_1 to $x_2 = x_1 + \alpha_2 d_1$. This explains the name *conjugate gradients*. The pure gradients of steepest descent would be too nearly parallel, and we would take small steps across a valley instead of a good step to the bottom (the minimizing x in Figure 6.15). Every cycle of CG chooses α_k to minimize $E(x)$ in the new search direction $x = x_{k-1} + \alpha d_{k-1}$. The last cycle (if we go that far) gives the overall minimizer $x_n = x = A^{-1}b$.

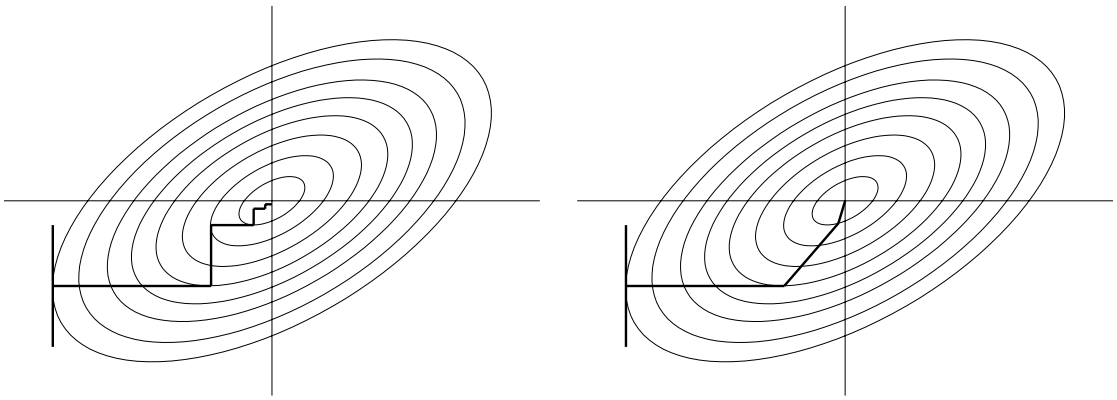

Figure 6.14: Steepest descent vs. conjugate gradient.

The main point is always this. **When you have orthogonality, projection and minimizations can be computed one direction at a time.**

Example

$$\text{Suppose} \quad Ax = b \quad \text{is} \quad \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}.$$

From $x_0 = (0,0,0)$ and $r_0 = d_0 = b$ the first cycle gives $\alpha_1 = \frac{1}{2}$ and $x_1 = \frac{1}{2}b = (2,0,0)$. The new residual is $r_1 = b - Ax_1 = (0,-2,-2)$. Then the CG algorithm yields

$$\beta_1 = \frac{8}{16} \quad d_1 = \begin{bmatrix} 2 \\ -2 \\ -2 \end{bmatrix} \quad \alpha_2 = \frac{8}{16} \quad x_2 = \begin{bmatrix} 3 \\ -1 \\ -1 \end{bmatrix} = A^{-1}b!$$

The correct solution is reached in two steps, where normally CG will take $n = 3$ steps. The reason is that this particular A has only two distinct eigenvalues 4 and 1. In that case $A^{-1}b$ is a combination of b and Ab, and this best combination x_2 is found at cycle 2. The residual r_2 is zero and the cycles stop early—very unusual.

Energy minimization leads in [ ] to an estimate of the convergence rate for the error $e = x - x_k$ in conjugate gradients, using the A-norm $\|e\|_A = \sqrt{e^{\mathrm{T}} A e}$:

$$\textbf{Error estimate} \qquad \|x - x_k\|_A \le 2\left(\frac{\sqrt{\lambda_{\max}} - \sqrt{\lambda_{\min}}}{\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}}}\right)^k \|x - x_0\|_A \,. \tag{19}$$

This is the best-known error estimate, although it doesn't account for any clustering of the eigenvalues of A. It involves only the condition number $\lambda_{\max}/\lambda_{\min}$. Problem ____ gives the "optimal" error estimate but it is not so easy to compute. That optimal estimate needs all the eigenvalues of A, while (19) uses the extreme eigenvalues $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$—which in practice we can bound above and below.

Minimum Residual Methods

When A is not symmetric positive definite, CG is not guaranteed to solve $Ax = b$. We lose control of $d^{\mathrm{T}} A d$ in computing α. We will follow van der Vorst [ ] in briefly describing the ***minimum norm residual*** approach, leading to MINRES and GMRES.

These methods choose x_j in the Krylov subspace $\boldsymbol{\mathcal{K}_j}$ so that $\|b - Ax_j\|$ is minimal. The first orthonormal vectors $q_1, \ldots, q_j$ go in the columns of Q_j, so $Q_j^{\mathrm{T}} Q_j = I$. As in (16) we set $x_j = Q_j y$, to express the solution as a combination of those q's: using (8) is

$$\textbf{Norm of residual} \qquad \|r_j\| = \|b - Ax_j\| = \|b - AQ_j y\| = \|b - Q_{j+1} H_{j+1,j} y\|. \tag{20}$$

Here I used the first j columns of Arnoldi's formula $AQ = QH$. Since the jth column of H is zero after entry $j+1$, we only need $j+1$ columns of Q on the right side:

$$\text{First } j \text{ columns of } QH = \begin{bmatrix} q_1 & \cdots & q_{j+1} \end{bmatrix} \begin{bmatrix} h_{11} & \cdots & h_{1j} \\ h_{12} & \ddots & \vdots \\ & \ddots & h_{jj} \\ & & h_{j+1,j} \end{bmatrix}. \tag{21}$$

The norm in (20) is not changed when we multiply by Q_{j+1}^{T}. Our problem becomes:

$$\textbf{Choose } y \textbf{ to minimize} \quad \|r_j\| = \|Q_{j+1}^{\mathrm{T}} b - H_{j+1,j}\, y\| \,. \tag{22}$$

This is an ordinary least squares problem with only $j+1$ equations and j unknowns. The right side $Q_{j+1}^{\mathrm{T}} b$ is $(\|r_0\|, 0, \ldots, 0)$ as in (16). The rectangular matrix $H_{j+1,j}$ is Hessenberg in (21). We face a completely typical problem of numerical linear algebra: *Use zeros in H and $Q_{j+1}^{\mathrm{T}} b$ to find a fast algorithm that computes y.* The two favorite algorithms for this least squares problem are closely related:

MINRES A is symmetric (likely indefinite, or we use CG) and H is tridiagonal.

GMRES A is *not* symmetric and the upper triangular part of H can be full.

In both cases we want to clear out that nonzero diagonal below the main diagonal of H. The natural way to do that, one entry at a time, is by "*Givens rotations*." These plane rotations are so useful and simple (the essential part is only 2 by 2) that we complete this section by explaining them.

Givens Rotations

The direct approach to the least squares solution of $Hy = f$ constructs the normal equations $H^{\mathrm{T}}H\widehat{y} = H^{\mathrm{T}}f$. That was the central idea in Chapter 1, but you see what we lose. If H is Hessenberg, with many good zeros, $H^{\mathrm{T}}H$ is full. Those zeros in H should simplify and shorten the computations, so we don't want the normal equations.

The other approach to least squares is by Gram-Schmidt. **We factor H into orthogonal times upper triangular**. Since the letter Q is already used, the orthogonal matrix will be called G (after Givens). The upper triangular matrix is $G^{-1}H$. The 3 by 2 case shows how a rotation in the 1–2 plane can clear out h_{21}:

$$G_{21}^{-1}H = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ 0 & h_{32} \end{bmatrix} = \begin{bmatrix} * & * \\ \mathbf{0} & * \\ 0 & * \end{bmatrix}. \tag{23}$$

That bold zero entry requires $h_{11}\sin\theta = h_{21}\cos\theta$, which determines the rotation angle θ. A second rotation G_{32}^{-1}, in the 2-3 plane, will zero out the $3,2$ entry. Then $G_{32}^{-1}G_{21}^{-1}H$ is a square upper triangular matrix U above a row of zeros!

The Givens orthogonal matrix is $G_{21}G_{32}$ but there is no reason to do this multiplication. We use each G_{ij} as it is constructed, to simplify the least squares problem. Rotations (and all orthogonal matrices) leave the lengths of vectors unchanged:

$$\|Hy - f\| = \|G_{32}^{-1}G_{21}^{-1}Hy - G_{32}^{-1}G_{21}^{-1}f\| = \left\| \begin{bmatrix} U \\ 0 \end{bmatrix} y - \begin{bmatrix} F \\ e \end{bmatrix} \right\|. \tag{24}$$

This length is what MINRES and GMRES minimize. The row of zeros below U means that the last entry e is the error—we can't reduce it. But we get all the other entries exactly right by solving the j by j system $Uy = F$ (here $j = 2$). This gives the best least squares solution y. Going back to the original problem of minimizing the residual $\|r\| = \|b - Ax_j\|$, the best x_j in the jth Krylov space is $Q_j y$.

For non-symmetric A (GMRES rather than MINRES) the recurrence is not short. The upper triangle in H can be full, and *step j becomes expensive*. Possibly it is inaccurate as j increases. So we may change "full GMRES" to GMRES(m), which restarts the algorithm every m steps. It is not so easy to choose a good m. But GMRES is an important algorithm for unsymmetric A.

Problem Set 6.4

1 When the tridiagonal K is preconditioned by $P = T$ (second difference matrix with $T_{11} = 1$) show that $\boldsymbol{T^{-1}K = I + \ell e_1^{\mathrm{T}}}$ with $e_1^{\mathrm{T}} = [\,1\ 0\ \ldots\ 0\,]$. Start from

$K = T + e_1 e_1^{\mathrm{T}}$. Then $T^{-1}K = I + (T^{-1}e_1)e_1^{\mathrm{T}}$. Verify that $T^{-1}e_1 = \ell$ from:

$$\boldsymbol{T\ell = e_1} = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & \cdot & \cdot & \cdot \\ & & -1 & 2 \end{bmatrix} \begin{bmatrix} N \\ N-1 \\ \cdot \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ 0 \end{bmatrix}.$$

Second differences of this linear vector ℓ are zero. Multiply $I + \ell e_1^{\mathrm{T}}$ times $I - (\ell e_1^{\mathrm{T}})/(N+1)$ to establish that this is the inverse matrix $K^{-1}T$.

2 For the model of a square grid with separator down the middle, create the reordered matrix K in equation (4). Use `spy`(K) to print its pattern of nonzeros.

3 Arnoldi expresses each Aq_j as $h_{j+1,j}q_{j+1} + h_{j,j}q_j + \cdots + h_{1,j}q_1$. Multiply by q_i^{T} to find $h_{i,j} = q_i^{\mathrm{T}}Aq_j$. *If A is symmetric* this is $(Aq_i)^{\mathrm{T}}q_j$. Explain why $(Aq_i)^{\mathrm{T}}q_j = 0$ for $i < j-1$ by expanding Aq_i into $h_{i+1,i}q_{i+1} + \cdots + h_{1,i}q_1$. We have a *short recurrence if* $A = A^{\mathrm{T}}$ (only $h_{j+1,j}$ and $h_{j,j}$ and $h_{j-1,j}$ are nonzero).

4 (This is Problem 3 at the matrix level) The Arnoldi equation $AQ = QH$ gives $H = Q^{-1}AQ = Q^{\mathrm{T}}AQ$. Therefore the entries of H are $h_{ij} = q_i^{\mathrm{T}}Aq_j$.

(a) Which Krylov space contains Aq_j? What orthogonality gives $h_{ij} = 0$ when $i > j+1$? Then H is upper Hessenberg.

(b) If $A^{\mathrm{T}} = A$ then $h_{ij} = (Aq_i)^{\mathrm{T}}q_j$. Which Krylov space contains Aq_i? What orthogonality gives $h_{ij} = 0$ when $j > i+1$? Now H is tridiagonal.

5 Test the `pcg`$(A,\ \ )$ MATLAB command on the $-1, 2, -1$ second difference matrix $A = K$. As preconditioner use $P = T$, when $T_{11} = 1$.

6 If $K = [\,b \quad Ab \quad \ldots \quad A^{n-1}b\,]$ is a Krylov matrix with $A = A^{\mathrm{T}}$, why is the inner product matrix $K^{\mathrm{T}}K$ a **Hankel matrix**? This means constant entries down each *antidiagonal* (the opposite of Toeplitz). Show that $(K^{\mathrm{T}}K)_{ij}$ depends on $i + j$.

7 These are famous names associated with linear algebra (and a lot of other mathematics too). All dead. Write one sentence on what they are known for.

Arnoldi	Gram	Jacobi	Schur
Cholesky	Hadamard	Jordan	Schwartz
Fourier	Hankel	Kronecker	Seidel
Frobenius	Hessenberg	Krylov	Toeplitz
Gauss	Hestenes-Stiefel	Lanczos	Vandermonde
Gershgorin	Hilbert	Markov	Wilkinson
Givens	Householder	Schmidt	Woodbury

`solution: Jacobi (matrices), Gauss (elimination, numerical integration), Lanczos [-1,2,-1 with q_1=(1,0,-)], zeros of cosines/convergence rate?, Another q_1? Berresford.`