AMSC 600 /CMSC 760 Advanced Linear Numerical Analysis

Fall 2007

Krylov Minimization and Projection (KMP)

Dianne P. O'Leary

This unit:

So far:

- A survey of iterative methods for solving linear systems:
  - fixed point iterations (stationary iterative methods)
    - Jacobi
    - Gauss-Seidel
    - SOR
  - an example of a non-stationary method: Chebyshev

All of these work in a Krylov subspace, but they are not usually called Krylov methods.

Next:

- An introduction to Krylov projection/minimization (KMP) methods
  - conjugate gradients
  - GMRES
  - preconditioning

Notational Note: I seem to be mixing superscripts, and subscripts to denote an iteration number. Apologies. But vectors and matrices are still boldface, while scalars are not.

KMP iterative methods

Define the **Krylov subspace** $\mathcal{K}_k(\mathbf{G}, \mathbf{c})$ to be the span of the vectors $\mathbf{c}, \mathbf{Gc}, \ldots, \mathbf{G}^{k-1}\mathbf{c}$.

KMP (Krylov Projection/Minimization) methods solve linear systems in one of two ways:

- Choose $\mathbf{x}^{(k)}$ in a Krylov subspace so that some function of the error $\mathbf{x}^{(k)} - \mathbf{x}^*$ is minimized, or

- Choose $\mathbf{x}^{(k)}$ in a Krylov subspace so that the residual $\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$ is orthogonal to a Krylov space.

Two ingredients to each algorithm:

- what to minimize or project
- how to generate a convenient basis for the Krylov subspace

For convenience, we assume that $\mathbf{x}^{(0)} = \mathbf{0}$, and some of the expressions in these notes take advantage of this assumption.

In this set of notes, we consider two KMP methods – CG and GMRES – as examples. We'll develop the family more systematically later.

Some Early History

Myth: Hestenes and Stiefel stole the CG method from Lanczos.

Cornelius LANCZOS

Magnus HESTENES

Eduard STIEFEL

J. Barkley Rosser
George Forsythe, William Karush, T. Motzkin, L. J. Paige
Leslie Fox, H. D. Huskey, Jim Wilkinson (1948): conjugate directions
Forsythe, Hestenes, Rosser 1951 abstract

The 1950s

Myth: CG was viewed as a direct method, and the "modern" view only developed within the last 20 years.

- Turing (1948): Preconditioning
- Arnoldi (1951): Nonsymmetric problems

- Hestenes and Stiefel (1952): Conjugate Gradients
  - direct method: finite termination.
  - use as iterative method: solves 106 "difference equations" in 90 iterations. (By 1958: 10x10 grid Laplace equation in 11 Chebyshev iterations + 2 cg.)
  - monotonicity properties.
  - round-off error analysis.
  - smoothing initial residual.
  - remedy for loss of orthogonality.
  - solution if A is rank deficient.
  - algebraic formulation of preconditioning.
  - relation to Lanczos algorithm and continued fractions.
- Lanczos (1952): Lanczos biorthogonalization
  - credits Hestenes and Stiefel with independent development.
  - advocates complete reorthogonalization or periodic restarts.
  - preconditioning by diagonal matrices.
  - initial Chebyshev smoothing (if eigenvalues nonnegative).
  - regularization of ill-posed problems.
- Hayes (1954): Hilbert spaces
  - linear convergence for general operators
  - superlinear convergence for I + completely continuous operator.
- Engeli, Ginsberg, Rutishauser, Stiefel (1958) "the monograph"
  - combine Chebyshev and cg
  - provide numerical evidence for use of cg as iterative method
  - provide the basis for many practical ideas rediscovered later

---

The 1960s

Myth: CG was forgotten in the 60's

Successes:

- Bothner-By et al (1962): spectral analysis
- Feder (1962): lens design
- Wachspress (1963): pde's, with adi preconditioner
- Dufour (1964): geodesy
- Campbell (1965): polar circulation

- Pitha and Jones (1967): infrared spectral analysis
- Sinnott and Luenberger (1967): optimal control
- Eu (1968): collision theory
- Fox and Stanton (1968): structural analysis
- G. Nagy (1968): pattern recognition
- Wallach (1968): power system load flow
- Bierson (1969): optimal flight paths
- Fried (1969): finite element analysis
- Garibotti and Villani (1969): nonrelativistic scattering
- Kawamura and Volz (1969): network analysis
- Sebe and Nachamkin (1969): nuclear shell computation

Failures:

- Livesley (1960): structural analysis
- P.C. Young (1966): noisy parameter estimation

Extensions:

- Fletcher and Reeves (1964): function minimization
- (Davidon (1959): Quasi-Newton)

The 1970s: Computer architecture starts to catch up

John Reid: "On the method of cg for the solution of large, sparse systems of linear equations" (1971)

Key issue: preconditioning

Gene Golub: Lanczos/SVD (1965).
Students in early 1970's: John Palmer, John Lewis, Richard Underwood, Franklin Luk, etc.

Henk van der Vorst and J. A. Meijerink: Incomplete LU preconditioning (1977). (cf Dick Varga (1960))

Some other activity 1970-76:

Lanczos:
– Chris Paige
– Beresford Parlett and W. Kahan
– Jane Cullum

CG:
– Jim Douglas and Todd Dupont
– Owe Axelsson
– Pete Stewart

The KMP Family

Connections with:

- matrix polynomials
- continued fractions
- Pade' table
- matrix decompositions
- ...

Case 1: $\mathbf{A}$ is Symmetric

$\mathcal{K}_k(\mathbf{A}, \mathbf{b}) \equiv \text{span}\{\mathbf{b}, \mathbf{Ab}, \ldots, \mathbf{A}^{k-1}\mathbf{b}\}$

$\mathbf{x}_k$ is chosen so that $(\mathbf{b} - \mathbf{Ax}_k, \mathbf{v}) = 0$ for $\mathbf{v} \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})$.

If $\mathbf{A}$ is positive definite, then this is called the conjugate gradient algorithm (cg).

If $\mathbf{A}$ is positive definite, then in addition, $\mathbf{x}_k$ minimizes $\|\mathbf{x} - \mathbf{x}^*\|_A$ for all $\mathbf{x} \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})$.

Case 2: $\mathbf{A}$ is not Symmetric

Convenient choice of basis:

symmetric Lanczos:

$$\mathbf{AV} = \mathbf{VH}, \quad \mathbf{V}^T\mathbf{V} = \mathbf{I},$$

or nonsymmetric Lanczos:

$$\begin{aligned} \mathbf{AV} &= \mathbf{VT}, \\ \mathbf{WA} &= \mathbf{TW}, \\ \mathbf{V}^H\mathbf{W} &= \mathbf{I}. \end{aligned}$$

Minimization and Galerkin give distinct algorithms.

	$\mathbf{AV = VH}$	$\mathbf{AV = VT}$, $\mathbf{WA = TW}$
minimization	GMRES	(Quasi) Minimum residual
Galerkin	Arnoldi	Lanczos
		Bi-CG

Let's look at two examples: cg and GMRES.

Choice of a basis

Important note: The vectors $\mathbf{c}, \mathbf{Gc}, \ldots, \mathbf{G}^{k-1}\mathbf{c}$ are a particularly bad basis since $\mathbf{G}^k\mathbf{c} \rightarrow$ the eigenvector of $\mathbf{G}$ corresponding to the largest-magnitude eigenvalue, so the vectors tend to become almost linearly dependent.

Best conditioned basis: an orthogonal one.

Best way to orthogonalize: (modified) Gram-Schmidt algorithm.

(modified) Gram-Schmidt algorithm

Given an orthonormal basis $\mathbf{v}_1, \ldots \mathbf{v}_k$ and a linearly independent direction $\mathbf{v}_{k+1}$, we subtract off a multiple of each of the orthonormal vectors in order to make the resulting vector orthogonal to all. Then we normalize it to length 1.

for $i = 1, \ldots k$,

$$\gamma_{i,k+1} = \mathbf{v}_{k+1}^T \mathbf{v}_i$$
$$\mathbf{v}_{k+1} = \mathbf{v}_{k+1} - \gamma_{i,k+1}\mathbf{v}_i$$

end for
$\mathbf{v}_{k+1} = \mathbf{v}_{k+1}/\|\mathbf{v}_{k+1}\|$

Note: The order of operations is important here, because of round-off error. Computing $\gamma_{i,k+1}$ without updating $\mathbf{v}_{k+1}$ using $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$ does not produce orthogonal vectors.

Notes:

- In certain cases (e.g., conjugate gradients, QMR) it can be proven that most of the γ's are zero, so that, for example, we may only need one or two old $\mathbf{v}$'s in order to form a new one.

  This is a significant savings in time and storage!

- In other cases (e.g., GMRES), we need to save all the old vectors, so we won't let k get too big.
- Sometimes we want orthogonality in a norm other than the Euclidean one. In that case, we just compute the γ's using the appropriate inner product. For example, if we want $\mathbf{v}_i^T \mathbf{A} \mathbf{v}_j = 0$ instead of $\mathbf{v}_i^T \mathbf{v}_j = 0$, then choose $\gamma_{i,k+1} = \mathbf{v}_{k+1}^T \mathbf{A} \mathbf{v}_i$.

Minimization: CG as an example of a KMP method

Let's look at an example of the minimization problems that arise in KMP algorithms. We'll consider the conjugate gradient algorithm.

Suppose $\mathbf{A}$ is symmetric and positive definite, and we choose to minimize

$$E(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T \mathbf{A} (\mathbf{x} - \mathbf{x}^*)$$

Now $\mathbf{x}^{(k)} = \alpha_1 \mathbf{v}_1 + \ldots \alpha_k \mathbf{v}_k = \mathbf{V}_k \boldsymbol{\alpha}$, where the vectors $\mathbf{v}_i$ form the columns of the matrix $\mathbf{V}_k$.

So the vector $\mathbf{x}^{(k)}$ should be chosen to minimize $E(\mathbf{x})$ over all choices of the vector $\boldsymbol{\alpha}$:

$$\begin{aligned} E(\mathbf{x}^{(k)}) &= (\mathbf{x}^{(k)} - \mathbf{x}^*)^T \mathbf{A} (\mathbf{x}^{(k)} - \mathbf{x}^*) \\ &= (\mathbf{V}_k \boldsymbol{\alpha} - \mathbf{x}^*)^T \mathbf{A} (\mathbf{V}_k \boldsymbol{\alpha} - \mathbf{x}^*) \end{aligned}$$

We minimize by setting the derivative equal to zero:

$$2\mathbf{V}_k^T \mathbf{A} \mathbf{V}_k \boldsymbol{\alpha} - 2\mathbf{V}_k^T \mathbf{A} \mathbf{x}^* = \mathbf{0}$$

If the columns of $\mathbf{V}_k$ are orthogonal in the $\mathbf{A}$ inner product, then $\mathbf{V}_k^T \mathbf{A} \mathbf{V}_k = \mathbf{D}_k$, a diagonal matrix, so

$$\alpha_i = \frac{\mathbf{v}_i^T \mathbf{b}}{\mathbf{v}_i^T \mathbf{A} \mathbf{v}_i}, i = 1, \ldots, k.$$

Notes:

- The early entries of $\boldsymbol{\alpha}$ do not change as more columns are added to $\mathbf{V}_k$, so old directions $\mathbf{v}_k$ can be discarded.
- The formulas we derived for the coefficients $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ are mathematically correct, but not the most convenient for computation, so the choices used in the programs are somewhat different but equivalent.

The conjugate gradient algorithm

Given an initial guess $\mathbf{x}_0$,

Let $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ and $\mathbf{p}_0 = \mathbf{M}^{-1}\mathbf{r}_0$. (For now, take $\mathbf{M} = \mathbf{I}$.)

For $k = 0, 1, 2, \ldots$, until convergence,

Compute the search parameter α_k and the new iterate and residual

$$\begin{aligned} \alpha_k &= \frac{\mathbf{r}_k^T \mathbf{M}^{-1} \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k, \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k. \end{aligned}$$

Compute the new search direction

$$\begin{aligned} \beta_k &= \frac{\mathbf{r}_{k+1}^T \mathbf{M}^{-1} \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{M}^{-1} \mathbf{r}_k}, \\ \mathbf{p}_{k+1} &= \mathbf{M}^{-1}\mathbf{r}_{k+1} + \beta_k \mathbf{p}_k. \end{aligned}$$

End for.

See `cg1.m`.

Work per iteration: One matrix-vector product with $\mathbf{A}$ plus some vector operations (2 inner products, 3 saxpys, plus termination test). Comparable to that for SIMs unless $\mathbf{A}$ has very few nonzeros.

Convergence theory

- Any vector $\mathbf{x}$ that is in the space $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ can be expressed as $\mathbf{x} = \delta_0 \mathbf{r}_0 + \delta_1 \mathbf{A}\mathbf{r}_0 + \ldots + \delta_{k-1}\mathbf{A}^{k-1}\mathbf{r}_0$. In other words, $\mathbf{x}$ is the product of $\mathbf{r}_0$ with a polynomial $\mathcal{P}_{k-1}(\mathbf{A}) = \delta_0 \mathbf{I} + \delta_1 \mathbf{A} + \ldots + \delta_{k-1}\mathbf{A}^{k-1}$.
- We choose the cg iterate in order to minimize the error function over all choices of $\mathbf{x}$, and this is equivalent to choosing the coefficients of the polynomial $\mathcal{P}_{k-1}$.

A polynomial representation for the error function

Now, $\mathbf{x}_k = \mathcal{P}_{k-1}(\mathbf{A})\mathbf{r}_0$, so we have a nice expression for the residual, too: $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$, so $\mathbf{r}_k$ is a polynomial in $\mathbf{A}$ of degree k times $\mathbf{r}_0$, with the constraint that the constant term is 1. For notation, let's say $\mathbf{r}_k = \mathcal{Q}_k(\mathbf{A})\mathbf{r}_0$.

Since $\mathbf{x}^* - \mathbf{x}_k = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_k) = \mathbf{A}^{-1}\mathbf{r}_k$, we know that $\mathbf{x}^* - \mathbf{x}_k = \mathbf{A}^{-1}\mathcal{Q}_k(\mathbf{A})\mathbf{r}_0 = \mathcal{Q}_k(\mathbf{A})\mathbf{A}^{-1}\mathbf{r}_0 = \mathcal{Q}_k(\mathbf{A})\mathbf{e}_0$.

So, conjugate gradient minimizes
$E(\mathbf{x}_k) \equiv (\mathbf{x}_k - \mathbf{x}^*)^T \mathbf{A}(\mathbf{x}_k - \mathbf{x}^*) = \mathbf{e}_0^T (\mathcal{Q}_k(\mathbf{A}))^2 \mathbf{A}\mathbf{e}_0$ over all choices of polynomial $\mathcal{Q}_k$ of degree less than or equal to k with constant coefficient 1.

Therefore, we need to choose this polynomial optimally.

Some properties of powers of matrices

Suppose that the eigenvalues of $\mathbf{A}$ are λ_i, and the corresponding eigenvectors are $\mathbf{u}_i$, $i = 1, \ldots, n$, where we have normalized so that $\|\mathbf{u}_i\| = 1$. Then

1. $\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i$,
2. The vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ form a basis for n dimensional space (since a theorem in linear algebra says that they are all linearly independent and, for symmetric matrices, orthogonal).
3. If we let $\mathbf{U}$ be the matrix with columns equal to $\mathbf{u}_i$, and $\mathbf{\Lambda}$ be the diagonal matrix with entries λ_i, then $\mathbf{A}[\mathbf{u}_1, \ldots, \mathbf{u}_n] = [\mathbf{u}_1, \ldots, \mathbf{u}_n]\mathbf{\Lambda}$, and thus $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$.
4. Since the eigenvectors are orthogonal and have norm 1, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.
5. Therefore, $\mathbf{A}^2 = (\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T) = \mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^T$ and, in general, $\mathbf{A}^i = \mathbf{U}\mathbf{\Lambda}^i\mathbf{U}^T$.

Therefore, since

$$\mathbf{A}^i = \mathbf{U}\mathbf{\Lambda}^i\mathbf{U}^T,$$

we have

$$\begin{aligned} \mathcal{P}_{k-1}(\mathbf{A}) &= \delta_0 + \delta_1\mathbf{A} + \ldots + \delta_{k-1}\mathbf{A}^{k-1} \\ &= \mathbf{U}(\delta_0 + \delta_1\mathbf{\Lambda} + \ldots + \delta_{k-1}\mathbf{\Lambda}^{k-1})\mathbf{U}^T \\ &= \mathbf{U}\mathcal{P}_{k-1}(\mathbf{\Lambda})\mathbf{U}^T \end{aligned}$$

We make use of an important property:

$$\mathcal{P}_{k-1}(\mathbf{\Lambda}) = (\mathcal{P}_{k-1}(\lambda_1), \ldots, \mathcal{P}_{k-1}(\lambda_n)).$$

Two theorems on convergence of cg

- Suppose that the matrix $\mathbf{A}$ has m distinct eigenvalues. Then we can find a polynomial of degree m that has those eigenvalues as roots, and therefore $p_m(\mathbf{A}) = \mathbf{0}$. Thus cg must terminate with the exact solution in at most m iterations.

  (It behaves almost as well if there are m small clusters of eigenvalues.)

- If the eigenvalues of $\mathbf{A}$ lie in the interval $[\lambda_{\min}, \lambda_{\max}]$, then we can bound the error expression by minimizing the maximum value that the polynomial attains on this interval. The solution to this min-max problem is related to a Chebyshev polynomial, and the construction yields the error bound

$$E(\mathbf{x}^{(k)}) \leq \left(\frac{1 - \sqrt{\kappa^{-1}}}{1 + \sqrt{\kappa^{-1}}} \right)^{2k} E(\mathbf{x}^{(0)}) \,,$$

  where $\kappa = \lambda_{\max}/\lambda_{\min}$.

Preconditioning

Now we let $\mathbf{M} \neq \mathbf{I}$.

If life hands us an ill-conditioned matrix (large κ) with no clustering of eigenvalues, then we can use *preconditioning* to try to cluster its eigenvalues or make it better conditioned.

If we do this, then κ depends on eigenvalues of $\mathbf{M}^{-1}\mathbf{A}$, not eigenvalues of $\mathbf{A}$.

We choose a symmetric positive definite matrix $\mathbf{M}$ so that $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$ has better eigenvalue properties, and so that it is easy to apply the operator $\mathbf{M}^{-1}$.

- For fast iterations, we want to be able to apply $\mathbf{M}^{-1}$ very quickly.
- To make the number of iterations small, we want $\mathbf{M}^{-1}$ to be an approximate inverse of $\mathbf{A}$.

Some common choices of $\mathbf{M}$:

- $\mathbf{M}$ = the diagonal of $\mathbf{A}$.
- $\mathbf{M}$ = a banded piece of $\mathbf{A}$.
- $\mathbf{M}$ = an incomplete factorization of $\mathbf{A}$: omit inconvenient elements.
- $\mathbf{M}$ = a related matrix; e.g., if $\mathbf{A}$ is a discretization of a differential operator, $\mathbf{M}$ might be a discretization of a related operator that is easier to solve.
- $\mathbf{M}$ might be the matrix from a SIM:

  Consider your favorite stationary iterative method (Jacobi, Gauss-Seidel, SOR, etc.) Derive it by taking the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$, splitting $\mathbf{A}$ into two pieces $\mathbf{A} = \mathbf{M} - \mathbf{N}$, and writing $\mathbf{M}\mathbf{x} = \mathbf{N}\mathbf{x} + \mathbf{b}$. The iteration then becomes

$$\mathbf{M}\mathbf{x}^{(k+1)} = \mathbf{N}\mathbf{x}^{(k)} + \mathbf{b}$$

  or

$$\mathbf{x}^{(k+1)} = \mathbf{M}^{-1}\mathbf{N}\mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{b}.$$

Manipulating this a bit, we get

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + (\mathbf{M}^{-1}\mathbf{N} - \mathbf{I})\mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{b} \\ &= \mathbf{x}^{(k)} + \mathbf{M}^{-1}(\mathbf{N} - \mathbf{M})\mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{b} \\ &= \mathbf{x}^{(k)} + \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}) \\ &= \mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{r}^{(k)}\,.\end{aligned}$$

Therefore, the matrix $\mathbf{M}^{-1}$ determines the multiple of the residual that we add on to $\mathbf{x}$. The matrix $\mathbf{M}$ becomes the conjugate gradient preconditioner.

The program `cg1.m` uses preconditioning.

We'll discuss a couple of these preconditioning ideas in more detail later.

When A is not symmetric...

... then we need to use KMP methods such as GMRES or QMR. Each has its disadvantages:

- GMRES needs to do the complete Gram-Schmidt process to form the basis, so all of the old vectors must be stored.
- QMR needs only a few vectors of storage, but it does not monotonically reduce any error measure, and it can break down before the solution is reached.

But these are the state-of-the-art methods.

GMRES as an example of a KMP method

GMRES = Generalized Minimum Residual.

Recall: CG minimizes the error function $E(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T\mathbf{A}(\mathbf{x} - \mathbf{x}^*)$ over vectors in the Krylov subspace $\mathcal{K}_k(\mathbf{G}, \mathbf{c})$.

- When $\mathbf{A}$ is symmetric but not positive definite, the error function has no minimum. We need to use a related algorithm called Symmlq instead of cg.
- When $\mathbf{A}$ fails to be symmetric – we solve the wrong linear system, one involving the symmetric part of $\mathbf{A}$. What should we do?

GMRES: minimize $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|$ over vectors in the Krylov subspace $\mathcal{K}_k(\mathbf{G}, \mathbf{c})$.

A useful decomposition: The Arnoldi Iteration

Given a matrix $\mathbf{A}$ and a vector $\mathbf{v}_1$ with $\|\mathbf{v}_1\|_2 = 1$, let's try to construct a decomposition

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\mathbf{H}_k$$

where $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, $\mathbf{V}_{k+1}^T\mathbf{V}_{k+1} = \mathbf{I}_{k+1}$, and $\mathbf{H}_k$ is a $(k+1) \times k$ matrix that is **upper Hessenberg**, i.e., zero below its first subdiagonal.

In other words, for $k = 3$,

$$\mathbf{A}[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4] \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \\ 0 & 0 & h_{43} \end{bmatrix}.$$

Now, to make this work, let's consider one column at a time.

Column 1: We need $\mathbf{A}\mathbf{v}_1 = \mathbf{v}_1 h_{11} + \mathbf{v}_2 h_{21}$, or $h_{21}\mathbf{v}_2 = \mathbf{A}\mathbf{v}_1 - h_{11}\mathbf{v}_1$. And we need $\mathbf{v}_1^T\mathbf{v}_2 = 0$. In other words, $\mathbf{v}_2$ is the result of the Gram-Schmidt process applied to the vector $\mathbf{A}\mathbf{v}_1$ using the vector $\mathbf{v}_1$, and the numbers h_{i1} are just the scalars γ_{i1} we computed there:

$$\begin{aligned} h_{11} &= \mathbf{v}_1^T(\mathbf{A}\mathbf{v}_1) \\ \hat{\mathbf{v}}_2 &= \mathbf{A}\mathbf{v}_1 - h_{11}\mathbf{v}_1 \\ h_{21} &= \|\hat{\mathbf{v}}_2\| \\ \mathbf{v}_2 &= \hat{\mathbf{v}}_2/h_{21}\,. \end{aligned}$$

Column 2: We need $\mathbf{A}\mathbf{v}_2 = \mathbf{v}_1 h_{12} + \mathbf{v}_2 h_{22} + \mathbf{v}_3 h_{32}$. We recognize this again as a Gram-Schmidt process: orthogonalize $\mathbf{A}\mathbf{v}_2$ against $\mathbf{v}_1$ and $\mathbf{v}_3$ to compute $\mathbf{v}_3$ and the h coefficients.

Column 3: Same story.

Conclusion: We can continue this process until we produce a vector $\mathbf{A}\mathbf{v}_j$ that is linearly dependent on the previous vectors $\mathbf{v}_1, \ldots, \mathbf{v}_j$. At that point, the vector $\hat{\mathbf{v}}_{j+1}$ will be zero, and the process terminates.

Properties

1. The vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ are an orthogonal basis for $\mathcal{K}(\mathbf{A}, \mathbf{v}_1)$.

2. After a given number of steps k, the GMRES process is stopped, and we seek the solution vector $\mathbf{x}_k$ that minimizes the residual. In other words, we let $\mathbf{x}_k = \mathbf{V}_k\mathbf{s}$ for some vector $\mathbf{s}$, and we want to minimize

$$\|\mathbf{r}_k\| = \|\mathbf{b} - \mathbf{A}\mathbf{x}_k\| = \|\mathbf{b} - \mathbf{A}\mathbf{V}_k\mathbf{s}\|$$

over all choices of the vector $\mathbf{s}$. Since $\mathbf{b} = \|\mathbf{b}\|\mathbf{v}_1$, and, for any matrix $\mathbf{W}$ with orthogonal columns, $\|\mathbf{W}\mathbf{z}\| = \|\mathbf{z}\|$, we want to minimize

$$\begin{aligned} \|\mathbf{b} - \mathbf{A}\mathbf{V}_k\mathbf{s}\| &= \|\|\mathbf{b}\|\mathbf{v}_1 - \mathbf{V}_{k+1}\mathbf{H}_k\mathbf{s}\| \\ &= \|\mathbf{V}_n^T(\|\mathbf{b}\|\mathbf{v}_1 - \mathbf{V}_{k+1}\mathbf{H}_k\mathbf{s})\| \\ &= \|\|\mathbf{b}\|\mathbf{e}_1 - \mathbf{H}_k\mathbf{s}\|\,, \end{aligned}$$

and this is a least squares problem of dimension $(k+1) \times k$ which can be solved in $O(k^2)$ operations since $\mathbf{H}_k$ is upper Hessenberg.

3. $\mathbf{V}_k^T\mathbf{A}\mathbf{V}_k = \mathbf{V}_k^T\mathbf{V}_{k+1}\mathbf{H}_k =$ the first k rows of $\mathbf{H}_k$. Call this matrix $\bar{\mathbf{H}}_k$. So, if $\mathbf{A}$ is symmetric, then $\bar{\mathbf{H}}_k$ must also be symmetric, and therefore it is tridiagonal! This means that the Gram-Schmidt process shortens to only 3 terms! In this case the Arnoldi iteration is called the (symmetric) Lanczos process, and $h_{k+1,k}\mathbf{v}_{k+1}$ is $\mathbf{r}_{k+1}$, the "minres" residual after $k+1$ steps.

4. If the iteration goes a full n steps, then $\mathbf{v}_{n+1} = \mathbf{0}$ (since we can't have more than n basis vectors to span n dimensional space), so we have $\mathbf{A}\mathbf{V}_n = \mathbf{V}_n\bar{\mathbf{H}}_n$, and since $\mathbf{V}_n^{-1} = \mathbf{V}_n^T$, we have $\mathbf{A} = \mathbf{V}_n\bar{\mathbf{H}}_n\mathbf{V}_\mathbf{n}^\mathbf{T}$. Thus $\mathbf{A}$ is **similar** to $\bar{\mathbf{H}}_n$ (i.e., has the same eigenvalues). We'll make use of this in the eigenvalue section of the course.

5. If the matrix is symmetric, then the iteration simplifies, since $\mathbf{H}_k$ is symmetric, too.

Unquiz: Write a Matlab program for GMRES.

So, ...

SIM's are quick to program. They look like ideal algorithms for computation!

KMP methods are more complicated.

But don't forget to count number of iterations

Example (which is actually rather typical):

Consider the Laplace problem with Dirichlet boundary conditions, discretized by finite differences on the unit square.

This gives us the 5-point operator equation that you have seen before. (main diagonal entries = 4, off diagonal entries = -1)

How many iterations does each of our algorithms take?

n	Jacobi	G-S	CG
10^2	273	138	24
20^2	1001	501	47
40^2	3803	1903	93

Notes:

- For the smallest mesh, CG takes only 0.17 times as many iterations as Gauss-Seidel and 0.08 times as many as Jacobi.
- When n is increased by a factor of 4, the number of CG iterations approximately doubles, while the number of iterations for Jacobi or Gauss-Seidel is multipled by about 4! Note that for a class of matrices including this one, $\rho(G_{GS}) = (\rho(\mathbf{G}_J))^2$.
- SOR would give fewer iterations than Gauss-Seidel, but, in general, determining the parameter ω is difficult.
- With preconditioning, we could cut the growth rate for the CG iterations.

So, even though Jacobi and Gauss-Seidel are much easier, CG will probably beat them on all but the easiest problems.

Summary

- We have set up a framework for Krylov methods:
  - they either minimize a measure of the error or
  - they project the residual

  using a Krylov subspace.
- We looked at two minimization algorithms: cg and GMRES.
- There are many other iterative methods, including projection methods, to choose among, and we'll look at more algorithms.
- Fast matrix-vector product $\rightarrow$ fast iterations.
- Don't be fooled by high gigaflop rates or utilization; make sure that the algorithm you choose also requires only a small number of iterations.

Additional References

Main reference: Chapter 6 of Saad.

The history is drawn from Gene H. Golub and Dianne P. O'Leary, "Some history of the conjugate gradient and Lanczos algorithms: 1948-1976," *SIAM Review* 31 (1989) 50-102.

There is a good discussion of SIMs, Krylov methods, and preconditioning in Chapter 10 of Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1989.

It is also a standard topic in other advanced numerical analysis textbooks.

There is a good, but lengthy, set of notes on conjugate gradients in "An Introduction to the Conjugate Gradient Method without the Agonizing Pain," Jonathan R. Shewchuk, `http://www.cs.cmu.edu/People/jrs/`.

And there is a shorter set of notes on cg posted on the course's webpage.