Methods in Computational Science – Iterative methods for linear equations (ch.7)

Johan Hoffman

Iterative methods

- The algorithm does not have a fixed number of steps.
- In each step an approximation is computed.
- The goal is for the approximations improve in each step.
- The algorithm terminates when stopping criterion satisfied.

Iterative methods for linear equations

For a given nonsingular *system matrix* $A \in R^{n\times n}$ and *data vector* $b \in R^n$, we address the problem to find a solution vector $x \in R^n$, such that

$$Ax = b, \tag{7.1}$$

where n is large and the matrix A is sparse. In contrast to direct methods, we will not try to construct the exact solution $x = A^{-1}b$ by matrix factorization. Instead we will develop *iterative methods* which generate a sequence of approximations $\{x^{(k)}\}_{k\geq 0} \subset R^n$ such that $x^{(k)} \approx x$. We define the *approximation error* in each iteration k as

$$e^{(k)} = x - x^{(k)},$$

and we seek to design iterative methods that produce a sequence of approximations that *converge*, in the sense that

$$\|e^{(k)}\| \to 0,$$

as $k \to \infty$. Further, we would like the error to be reduced as much as possible in each iteration.

Convergence of iterative methods

Assuming that the sequence of approximations $\{x^{(k)}\}_{k\geq 0}$ converges to the exact solution x, we say that its *rate of convergence* is of *order* q with *asymptotic error constant* $C > 0$, if

$$\lim_{k\to\infty} \frac{\|x - x^{(k+1)}\|}{\|x - x^{(k)}\|^q} = C.$$

If $q = 1$ the method is said to be of *linear order of convergence*, $q = 2$ *quadratic order of convergence*, and $q = 3$ *cubic order of convergence*. In the asymptotic region of large k,

$$\|x - x^{(k+1)}\| \approx C\|x - x^{(k)}\|^q,$$

and by forming the ratio of two successive approximations the constant C is eliminated,

$$\frac{\|x - x^{(k+1)}\|}{\|x - x^{(k)}\|} \approx \left(\frac{\|x - x^{(k)}\|}{\|x - x^{(k-1)}\|}\right)^q.$$

Estimation of convergence order

$$q \approx \frac{\log \dfrac{\|x - x^{(k)}\|}{\|x - x^{(k-1)}\|}}{\log \dfrac{\|x - x^{(k-1)}\|}{\|x - x^{(k-2)}\|}}$$

$$q \approx \frac{\log \dfrac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)} - x^{(k-1)}\|}}{\log \dfrac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|}}$$

Residual and error equation

Assume now that we have an iterative method that converges and an algorithm by which the method can be implemented. To determine when the algorithm should be terminated we need to construct a *stopping criterion*. It is natural to seek to construct a stopping criterion based on the size of the error, but since the exact solution x is unknown the approximation error is not directly computable. On the other hand, the error can be expressed in terms of a computable *residual*

$$r^{(k)} = b - Ax^{(k)},$$

which measures to what degree the approximation $x^{(k)}$ satisfies equation (7.1). The residual is related to the error through the system matrix A, since

$$r^{(k)} = b - Ax^{(k)} = Ax - Ax^{(k)} = Ae^{(k)},$$

which leads to the *error equation*

$$Ae^{(k)} = r^{(k)}. \tag{7.5}$$

Error estimation and condition number

Theorem 7.1 (Error estimate). *For $\{x^{(k)}\}_{k\geq 0}$ a sequence of approximate solutions to the system of linear equations (7.1), we have the following error estimate,*

$$\frac{\|e^{(k)}\|}{\|e^{(0)}\|} \leq \kappa(A) \frac{\|r^{(k)}\|}{\|r^{(0)}\|}. \tag{7.7}$$

Proof. By the error equation (7.5),

$$\|e^{(k)}\| = \|A^{-1} r^{(k)}\| \leq \|A^{-1}\| \|r^{(k)}\|,$$

and, analogously,

$$\|r^{(0)}\| = \|A e^{(0)}\| \leq \|A\| \|e^{(0)}\|,$$

from which the result follows by the definition of the condition number $\kappa(A) = \|A^{-1}\| \|A\|$.

Stopping criterion for iterative methods

The error estimate (7.7) may be used as a stopping criterion for an algorithm used to implement the iterative method, since we know that the relative error is bounded by the computable residual. That is, we terminate the algorithm if the following condition is satisfied,

$$\|r^{(k)}\|/\|r^{(0)}\| < TOL,$$

with $TOL > 0$ a chosen tolerance. In practice, it is often problematic to use the relative error with respect to the initial approximation, since the choice of $x^{(0)}$ may be arbitrary, without significance for the problem at hand. Hence, it is more suitable to formulate a stopping criterion based on $x^{(0)} = 0$, which leads instead to the following condition,

$$\|r^{(k)}\|/\|b\| < TOL.$$

Sensitivity wrt perturbed data $\tilde{b} = b + \delta b$

$$\tilde{x} = A^{-1}\tilde{b} = A^{-1}(b + \delta b) = A^{-1}b + A^{-1}\delta b \equiv x + \delta x,$$

and the sensitivity can be expressed as the ratio between the relative perturbations x and b,

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta b\|}{\|b\|} = \frac{\|A^{-1}\delta b\|}{\|A^{-1}b\|} \Big/ \frac{\|\delta b\|}{\|b\|} = \frac{\|A^{-1}\delta b\|}{\|\delta b\|} \frac{\|b\|}{\|A^{-1}b\|}.$$

We find that the maximal sensitivity is equal to the condition number, which follows from the definition of an induced matrix norm

$$\max_{b,\delta b \neq 0} \left(\frac{\|A^{-1}\delta b\|}{\|\delta b\|} \frac{\|b\|}{\|A^{-1}b\|} \right) = \max_{\delta b \neq 0} \frac{\|A^{-1}\delta b\|}{\|\delta b\|} \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \|A^{-1}\|\|A\| = \kappa(A).$$

Sensitivity and the condition number

Further, by using an SVD of the system matrix $A = U\Sigma V^T$,

$$\|A\| = \max_{\|x\|=1} \|Ax\| = \max_{\|x\|=1} \|U\Sigma V^T x\| = \max_{\|V^T x\|=1} \|\Sigma V^T x\| = \max_{\|y\|=1} \|\Sigma y\| = \sigma_{max}(A),$$

with $\sigma_{max}(A)$ the maximal singular value of A. The last equality follows since $y = (1, 0, 0, ...)^T$ is the $y \in R^n$ on the unit sphere which maximizes $\|\Sigma y\|$, as the singular values are numbered in descending order. Analogously, we have that $\|A^{-1}\| = \sigma_{min}(A)^{-1}$, with $\sigma_{min}(A)$ the minimal singular value of A. Therefore, the condition number can be expresses as

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}.$$

Sensitivity and the condition number

For a normal matrix A, we also have that

$$\kappa(A) = \frac{|\lambda_{max}(A)|}{|\lambda_{min}(A)|}, \tag{7.9}$$

where $|\lambda_{max}(A)|$ and $|\lambda_{min}(A)|$ are the maximal and minimal (in magnitude) of the (possibly complex) eigenvalues of A, which is a consequence of Theorem 6.7 and equation (6.18),

$$\begin{aligned}\sigma_{max}(A) &= \sqrt{\lambda_{max}(A^T A)} = \sqrt{\lambda_{max}((U\Lambda U^T)^T U\Lambda U^T)} \\ &= \sqrt{\lambda_{max}(U\Lambda^2 U^T)} = \sqrt{\lambda_{max}(\Lambda^2)} = \sqrt{|\lambda_{max}(A)|^2} = |\lambda_{max}(A)|,\end{aligned}$$

and in the same way $\sigma_{min}(A) = |\lambda_{min}(A)|$. If A is symmetric positive definite, the eigenvalues are real and positive, and it follows that $\sigma_{max}(A) = \lambda_{max}(A)$ and $\sigma_{min}(A) = \lambda_{min}(A)$.

Sensitivity and the condition number

Example 7.2. Consider the symmetric positive definite matrix

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

To compute the singular values we form the matrix

$$A^T A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix},$$

for which the characteristic equation is

$$\det(A^T A - \lambda I) = (5-\lambda)^2 - 16 = 0 \Leftrightarrow 5 - \lambda = \pm 4 \Leftrightarrow \lambda = 5 \pm 4.$$

The singular values of A are then the square roots of the eigenvalues of $A^T A$,

$$\sigma_1 = \sqrt{9} = 3, \quad \sigma_2 = \sqrt{1} = 1.$$

The linear system $Ax = b$ is well-conditioned, since the condition number of A is

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} = \frac{3}{1} = 3.$$

We can also verify that since A is symmetric, $\lambda_{max}(A) = \sigma_{max}(A)$ and $\lambda_{min}(A) = \sigma_{min}(A)$.

Sensitivity and the condition number

Example 7.3. Now instead consider the diagonal matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix},$$

with $0 < \epsilon < 1$, from which we get that

$$A^T A = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2 \end{bmatrix}.$$

The singular values of A are then

$$\sigma_1 = \sqrt{1} = 1, \quad \sigma_2 = \sqrt{\epsilon^2} = \epsilon,$$

which leads to the condition number

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} = \frac{1}{\epsilon}.$$

If ϵ is small then $\kappa(A)$ is very large, which leads to an ill-conditioned system of linear equations $Ax = b$. Since A is symmetric, we have that $\lambda_{max}(A) = \sigma_{max}(A)$ and $\lambda_{min}(A) = \sigma_{min}(A)$.

Sensitivity and the condition number

Example 7.4. The skew-symmetric normal matrix

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

has a condition number $\kappa(A) = 1$, hence, the system $Ax = b$ is well-conditioned. This follows from the fact that $\lambda_1 = i$ and $\lambda_2 = -i$, and therefore $\sigma_{max}(A) = \sigma_{min}(A) = |\lambda_1| = |\lambda_2| = 1$, which we can verify from

$$A^T A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Fixed point iteration

The first family of iterative methods that we develop is based on the idea of *fixed point iteration*,

$$x^{(k+1)} = g(x^{(k)}), \tag{7.10}$$

to solve the equation $x = g(x)$, where the function $g : X \to X$ may be a linear operator in the form of a matrix, or a general nonlinear function. By the *Banach fixed point theorem*, if X is a Banach space and the map $x \mapsto g(x)$ satisfies a certain stability condition, the fixed point iteration (7.10) generates a Cauchy sequence $\{x^{(k)}\}_{k\geq 0}$ which converges to the solution $x \in X$,

$$\lim_{n\to\infty} \|x - x^{(n)}\| = 0.$$

Recall that in a Cauchy sequence $\{x^{(k)}\}_{k\geq 0}$ the approximations $x^{(k)}$ approach each other,

$$\lim_{n\to\infty} \|x^{(m)} - x^{(n)}\| = 0, \quad m > n,$$

and since R^n is a Banach space, the Banach fixed point theorem applies to systems of linear equations.

Stationary iterative methods

Stationary iterative methods are formulated as a linear fixed point iteration of the form

$$x^{(k+1)} = Mx^{(k)} + c, \tag{7.11}$$

corresponding to the function $g(x) = Mx + c$, with $M \in R^{n\times n}$ the *iteration matrix*, $\{x^{(k)}\}_{k\geq 0}$ a sequence of approximations $x^{(k)} \in R^n$, and $c \in R^n$ a constant vector.

Theorem 7.5 (Banach fixed point theorem for matrices). *If $M \in R^{n\times n}$ and $\|M\| < 1$, the fixed point iteration (7.11) converges to the unique solution of the equation $x = Mx + c$.*

By Banach fixed point theorem, if $\|M\| < 1$ the iteration (7.11) converges to the unique solution, and since $\|M\| = \sigma_{max}$(M), an equivalent convergence criterion is $\sigma_{max}(M) < 1$. For a normal matrix A, the induced norm is also equal to the spectral radius $\|M\| = \rho(M)$, so that we can use the convergence criterion $\rho(M) < 1$.

Richardson iteration for solving $Ax = b$

For the system of linear equations (7.1), we can formulate an iterative method in the form of a fixed point iteration (7.11) with $M = I - \alpha A$ and $c = \alpha b$,

$$x^{(k+1)} = (I - \alpha A)x^{(k)} + \alpha b = x^{(k)} + \alpha r^{(k)}, \tag{7.12}$$

where $\alpha \in R$ is an arbitrary parameter of the method, and

$$r^{(k)} = b - Ax^{(k)}$$

We note that the Richardson iteration (7.12) also takes the form of a *search method*, meaning that the new approximation $x^{(k+1)}$ is obtained from the previous approximation $x^{(k)}$ by taking a step in a *search direction* a distance determined by a *step length*. Here, the search direction is equal to the residual and the step length is the parameter α.

Richardson iteration for solving $Ax = b$

ALGORITHM 7.1. x = richardson_iteration(A, b, alpha).
Input: a nxn system matrix **A**, an n vector **b**, a parameter **alpha**.
Output: solution vector **x**.

```
1: x = 0
2: while norm(r) > TOL do
3:     r = matrix_vector_product(A, x)
4:     r[:] = b[:] - r[:]
5:     x[:] = x[:] + alpha*r[:]
6: end while
7: return x
```

Preconditioned Richardson iteration

Estimated convergence: $$\frac{\|x - x^{(k+1)}\|}{\|x - x^{(k)}\|} \approx \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)} - x^{(k-1)}\|} \leq \|I - \alpha A\|$$

To maximize the rate of convergence one may choose the parameter α such that $\|I - \alpha A\|$ is minimized. More generally, we can seek to modify, or *precondition*, the linear system of equations (7.1) in such a way that the rate of convergence is improved while the solution stays the same. By multiplication of both sides of the system (7.1) from the left by a matrix $B \approx A^{-1}$, referred as an *approximate inverse* of A, we get new system with the same solution x,

$$BAx = Bb.$$

for which the preconditioned Richardson iteration takes the form

$$x^{(k+1)} = (I - \alpha BA)x^{(k)} + \alpha Bb = x^{(k)} + \alpha Br^{(k)}$$

Preconditioned Richardson iteration

Analogously, we can precondition the system by multiplication of the system matrix A from the right by an approximate inverse B,

$$AB(B^{-1}x) = b,$$

to get a preconditioned Richardson iteration of the form

$$y^{(k+1)} = (I - \alpha AB)y^{(k)} + \alpha b = y^{(k)} + \alpha r^{(k)},$$

with $y^{(k)} = B^{-1}x^{(k)}$ and $x^{(k)} = By^{(k)}$. The error reduction in each step is estimated to be $\|I - \alpha AB\|$, and both the search direction and the stopping criterion are based on the original residual, since

$$r^{(k)} = b - ABy^{(k)} = b - Ax^{(k)}.$$

Jacobi preconditioner

Example 7.6 (Jacobi preconditioner). The diagonal preconditioner, or *Jacobi preconditioner*, is based on a diagonal scaling of the system matrix A, expressed through the approximate inverse

$$B = \text{diag}(A)^{-1},$$

where $\text{diag}(A)$ is the diagonal matrix with identical diagonal elements as the matrix A. Used as a left preconditioner, the error reduction in each step of a Richardson iteration with $\alpha = 1$ can then be estimated to be

$$\|I - BA\| = \|I - \text{diag}(A)^{-1}A\| = \|A - \text{diag}(A)\|.$$

Jacobi preconditioning can be efficient if the matrix A is *diagonally dominant*, meaning that the magnitude of the diagonal component on each row is dominating,

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad \forall i.$$

Matrix splitting

Matrix splitting is a technique to formulate stationary iterative methods (7.11) based on splitting the system matrix into a sum

$$A = A_1 + A_2.$$

Here A_1 is chosen to be a nonsingular matrix which is easy to invert, such as a diagonal matrix D, a lower triangular matrix L or an upper triangular matrix U.

Jacobi iteration

Example 7.8 (Jacobi iteration). *Jacobi iteration* is based on the splitting

$$A_1 = D, \quad A_2 = A - D,$$

where $D = \text{diag}(A)$. This gives the iteration matrix

$$M_J = -D^{-1}(A - D) = (I - D^{-1}A)$$

and the vector

$$c = D^{-1}b,$$

for which the convergence criterion is $\|I - D^{-1}A\| < 1$. In terms of the components of the matrix $A = (a_{ij})$, Jacobi iteration takes the form

$$x_i^{(k+1)} = a_{ii}^{-1}\left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}\right), \quad \forall i.$$

Jacobi iteration is equivalent to Jacobi (left) preconditioned Richardson iteration with $\alpha = 1$.

Gauss-Seidel iteration

Example 7.9 (Gauss-Seidel iteration). *Gauss-Seidel iteration* is based on the splitting

$$A_1 = L, \quad A_2 = A - L,$$

where L is the lower triangular matrix obtained from the matrix A by zeroing out all entries above the diagonal. The iteration matrix is

$$M_{GS} = -L^{-1}(A - L) = (I - L^{-1}A)$$

and the vector

$$c = L^{-1}b.$$

Hence, the iteration is equivalent to a left preconditioned Richardson iteration with $B = L^{-1}$ and $\alpha = 1$, for which the convergence criterion $\|I - L^{-1}A\| < 1$. The triangular matrix L is inverted by forward substitution, so that

$$x_i^{(k+1)} = a_{ii}^{-1}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right), \quad \forall i.$$

Damped Jacobi and SOR iteration

To improve convergence of an iterative method we can use *relaxation* to set the degree of diagonal dominance by a parameter $\omega > 0$, in the form of a left preconditioned Richardson iteration with $B = D^{-1}$ and $\alpha = \omega$, also referred to as the *damped Jacobi iteration*,

$$x^{(k+1)} = x^{(k)} + \omega D^{-1}(b - Ax^{(k)}).$$

Convergence can be further enhanced by a matrix splitting technique $A = A_1 + A_2$, that suggests the following iteration

$$x^{(k+1)} = (D + \omega A_1)^{-1}(\omega b - (\omega A_2 - D)x^{(k)}),$$

which for $A_1 = L$ is referred to as *successive over-relaxation* (SOR).

Krylov subspace methods

A *Krylov subspace method*, or *Krylov method*, is a family of iterative methods for the solution of an $n \times n$ system of linear equations $Ax = b$, where in each iteration k we construct an approximation

$$x^{(k)} \approx x$$

in the *Krylov subspace* of R^n,

$$\mathcal{K}_k = \langle b, Ab, ..., A^{k-1}b \rangle.$$

The Krylov subspace is spanned by the vectors $b, Ab, ..., A^{k-1}b$, which we can store as column vectors in an $n \times k$ matrix, the *Krylov matrix*

$$K_k = \left[\begin{array}{c|c|c|c} & & & \\ b & Ab & \cdots & A^{k-1}b \\ & & & \end{array} \right],$$

so that

$$\mathcal{K}_k = \text{range}(K_k).$$

GMRES

The idea of *GMRES* (generalized minimal residuals) is that, at each step k of the iteration, we construct the vector $x^{(k)} \in \mathcal{K}_k$ that minimizes the norm of the residual $r^{(k)} = b - Ax^{(k)}$, which corresponds to the least squares problem

$$\min_{x^{(k)} \in \mathcal{K}_k} \|b - Ax^{(k)}\|.$$

But instead of expressing the approximation as a linear combination of the Krylov vectors $b, Ab, ..., A^{k-1}b$, which may lead to an unstable algorithm, we construct an orthonormal basis $\{q_j\}_{j=1}^{k}$ for $\mathcal{K}_k$, such that

$$\mathcal{K}_k = \langle q_1, q_2, ..., q_k \rangle.$$

GMRES

With Q_k the $n \times k$ matrix with column vectors $q_{:j} = q_j$, we can express the approximation as $x^{(k)} = Q_k y$, with $y \in R^k$ a vector with the coordinates of $x^{(k)}$ in the orthonormal basis. The least squares problem then takes the form,

$$\min_{y \in R^k} \|b - AQ_k y\|.$$

GMRES

To compute the orthonormal basis we use a modified Gram-Schmidt iteration with the Krylov vectors $A^k b$ replaced by $Aq_{:k}$, starting from $q_{:1} = b/\|b\|$, referred to as *Arnoldi iteration*. In matrix form, the update formula can be expressed by a Hessenberg matrix $\tilde{H}_k \in R^{k+1\times k}$ as

$$AQ_k = Q_{k+1}\tilde{H}_k,$$

or in component form,

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \left[\begin{array}{c|c|c} & & \\ q_1 & \cdots & q_k \\ & & \end{array}\right] = \left[\begin{array}{c|c|c} & & \\ q_1 & \cdots & q_{k+1} \\ & & \end{array}\right] \begin{bmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & \ddots & \vdots \\ & \ddots & \vdots \\ & & h_{kk} \\ & & h_{k+1k} \end{bmatrix}$$

from which it follows that $b = \|b\|q_{:1} = \|b\|Q_{k+1}e_1$, with e_1 the first standard basis vector in R^{k+1}.

GMRES

$$\min_{y \in R^k} \|b - AQ_k y\| = \min_{y \in R^k} \|\|b\|Q_{k+1}e_1 - Q_{k+1}\tilde{H}_k y\| = \min_{y \in R^k} \|Q_{k+1}(\|b\|e_1 - \tilde{H}_k y)\|$$
$$= \min_{y \in R^k} \|\|b\|e_1 - \tilde{H}_k y\|,$$

where the last equality is a consequence of the fact that $Q_{k+1}^T Q_{k+1} = I$, since

$$\|Q_{k+1}(\|b\|e_1 - \tilde{H}_k y)\| = ((\|b\|e_1 - \tilde{H}_k y)^T Q_{k+1}^T Q_{k+1}(\|b\|e_1 - \tilde{H}_k y))^{1/2} = \|\|b\|e_1 - \tilde{H}_k y\|.$$

The solution to the least squares problem is $y \in R^k$, from which we obtain the approximate solution to the system of linear equations

$$x^{(k)} = Q_k y.$$

GMRES

ALGORITHM 7.2. x = gmres(A, b).
Input: nxn matrix **A** and n vector **b**.
Output: solution vector **x**.

```
1:  Q[:,0] = b[:]/norm(b)
2:  k = 0
3:  while norm(r)/norm(b) > TOL do
4:      k = k+1
5:      e1 = standard_basis(k+1, 0)
6:      (Q, H) = arnoldi_iteration(A, b, k)
7:      y = least_squares_problem(H, norm(b)*e1)
8:      r = matrix_vector_product(H, y)
9:      r[:] = norm(b)*e1[:] - r[:]
10: end while
11: x = matrix_vector_product(Q[:,0:k-1], y)
12: return x
```

GMRES least squares problem

To implement the function **y = least_squares_problem(H, norm(b)*e1)**, we can exploit the structure of the $k+1 \times k$ Hessenberg matrix $\tilde{H}_k$, which can be transformed into an upper triangular matrix $\tilde{R}$ by a sequence of k Givens rotations, represented by the matrices G_i for $i = 1, ..., k$. The product

$$\bar{G}_k = G_k \cdots G_1$$

is an orthogonal $(k+1) \times (k+1)$ matrix, which if multiplied to $\tilde{H}_k$ results in

$$\bar{G}_k \tilde{H}_k = \tilde{R}_k = (R_k, 0)^T,$$

with R_k a square $k \times k$ upper triangular matrix, and

$$\tilde{R}_k = \begin{bmatrix} r_{11} & \cdots & r_{1k} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{kk} \\ 0 & \cdots & 0 \end{bmatrix}$$

GMRES least squares problem

$$\min_{y \in R^k} \| \|b\| e_1 - \tilde{H}_k y \| = \min_{y \in R^k} \| \bar{G}_k (\|b\| e_1 - \tilde{H}_k y) \|$$
$$= \min_{y \in R^k} \| \|b\| \bar{G}_k e_1 - \tilde{R}_k y \|$$

with solution

$$y = R_k^{-1} (\|b\| \bar{G}_k e_1)_{1:k},$$

computed by backward substitution, where we use the notation $(\|b\| \bar{G}_k e_1)_{1:k} \in R^k$ for the first k components of the $k+1$ vector.

Arnoldi iteration

ALGORITHM 7.3. (Q, H) = arnoldi_iteration(A, b, k).
Input: nxn matrix **A**, n vector **b**, and transformation size **k**.
Output: nx(k+1) transformation matrix **Q**, (k+1)xk Hessenberg matrix **H** similar to A.

```
1:  Q[:,0] = b/norm(b)
2:  for j=0:k do
3:      v[:] = matrix_vector_product(A, Q[:,j])
4:      for i=0:j do
5:          h[i,j] = scalar_product(Q[:,i], v[:])
6:          v[:] = v[:] - h[i,k]*Q[:,i]
7:      end for
8:      h[j+1,j] = norm(v)
9:      q[:,j+1] = v[:]/h[j+1,j]
10: end for
11: return  Q, H
```

Eigenvalue computation by Arnoldi iteration

Arnoldi iteration (referred to as Lanczos iteration for symmetric matrices) can also be used to approximate the eigenpair of the matrix A, since $\mathcal{K}_k$ is spanned by the first k vectors of a power iteration based on the initial vector b, with $A^k b / \|A^k b\|$ an approximation of the dominant eigenvector of A. By orthogonalization of the Krylov vectors using Algorithm 7.3, we generate an orthonormal basis for $\mathcal{K}_k$, but also an approximate unitary similarity transformation

$$H_k \approx Q_k^T A Q_k,$$

where H_k denotes the $k \times k$ Hessenberg matrix we obtain by removing the last row $(0, \ldots, h_{k+1k})$ from $\tilde{H}_k$. By computing H_k eigenpairs (θ_i, z_i), e.g. by the implicit QR algorithm, we obtain approximate eigenpairs of A as the *Ritz eigenvalues* θ and *Ritz eigenvectors* $Q_k z_i$, with residual

$$r = AQ_k z_i - \theta_i Q_k z_i = (AQ_k - Q_k H_k) z_i.$$

Conjugate gradient method

For a real symmetric positive definite $n \times n$ matrix A, we can define the inner product

$$(x, y)_A = x^T A y, \quad x, y \in R^n,$$

for which symmetry follows by

$$(x, y)_A = x^T A y = (Ay)^T x = y^T A^T x = y^T A x = (y, x)_A,$$

and which generates the *energy norm*, or the A *norm*,

$$\|x\|_A = (x, x)_A^{1/2} = \sqrt{x^T A x}.$$

The *Conjugate Gradient method* (CG) is based on minimization of the error $e^{(k)} = x - x^{(k)}$ in the energy norm over the Krylov subspace $\mathcal{K}_k$, or equivalently by equation (7.5), minimization of the residual $r^{(k)} = b - Ax^{(k)}$ in the A^{-1} norm,

$$\|e^{(k)}\|_A = (e^{(k)}, Ae^{(k)})^{1/2} = (e^{(k)}, r^{(k)})^{1/2} = (A^{-1}r^{(k)}, r^{(k)})^{1/2} = \|r^{(k)}\|_{A^{-1}}.$$

Conjugate gradient method

We can formulate CG as a search method:

$$
\begin{aligned}
x^{(k+1)} &= x^{(k)} + \alpha^{(k)} p^{(k)}, \\
r^{(k+1)} &= r^{(k)} - \alpha^{(k)} A p^{(k)}, \\
p^{(k+1)} &= r^{(k+1)} + \beta^{(k)} p^{(k)},
\end{aligned}
$$

with

$$
\alpha^{(k)} = \frac{\|r^{(k)}\|^2}{\|p^{(k)}\|_A^2}, \quad \beta^{(k)} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2},
$$

and the search method is initialized by

$$
p_0 = r_0 = b - A x_0,
$$

Conjugate gradient method

ALGORITHM 7.4. x = conjugate_gradient_method(A, b).
Input: nxn matrix **A**, and n vector **b**.
Output: solution vector **x**.

```
1:  x[:] = 0
2:  r[:] = b[:]
3:  p[:] = r[:]
4:  while norm(r)/norm(b) > TOL do
5:      Ap = matrix_vector_product(A, p)
6:      alpha = scalar_product(r, r)/scalar_product(p, Ap)
7:      x[:] = x[:] + alpha*p[:]
8:      beta = 1/ scalar_product(r, r)
9:      r[:] = r[:] - alpha*matrix_vector_product(A, p)
10:     beta = scalar_product(r, r)*beta
11:     p[:] = r[:] + beta*p[:]
12: end while
13: return  x
```

Conjugate gradient method

Theorem 7.11 (CG characteristics). *For the Conjugate Gradient (CG) method applied to the equation*

$$Ax = b,$$

with A an $n \times n$ symmetric positive definite matrix, the orthogonality relations (7.16) and (7.17) are true, and

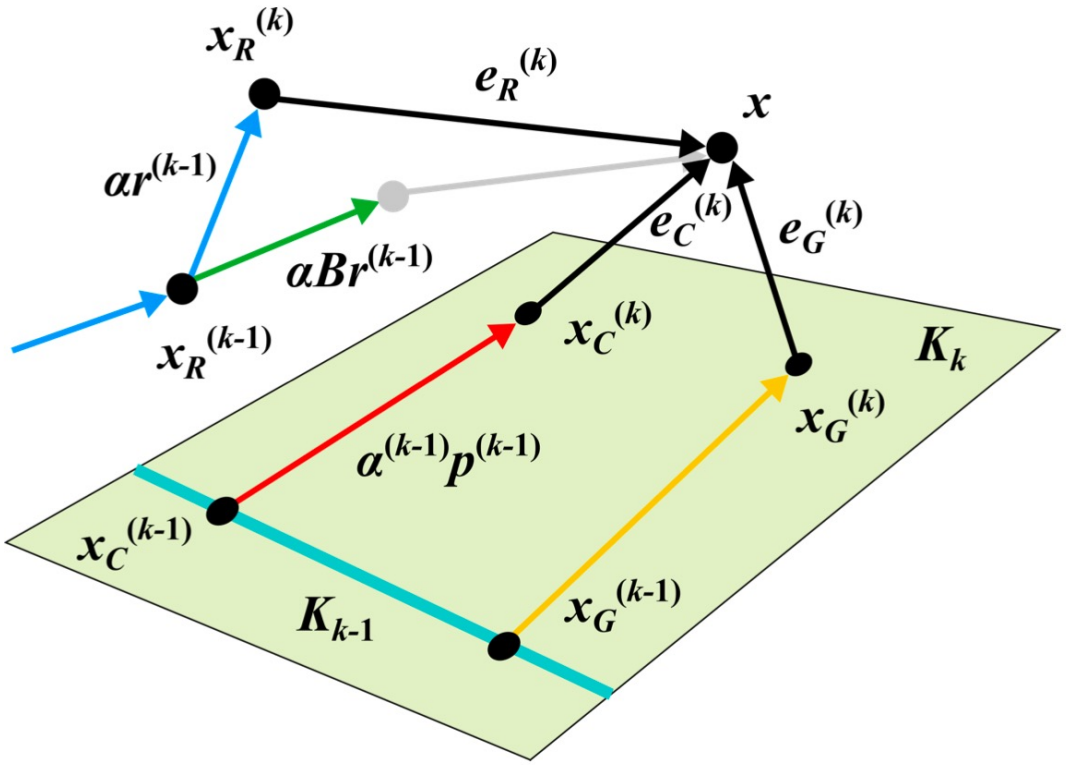
$$\begin{aligned}\mathcal{K}_k &= \langle b, Ab, \ldots, A^{k-1}b \rangle = \langle x^{(1)}, x^{(2)}, \ldots, x^{(k)} \rangle \\ &= \langle p^{(0)}, p^{(1)}, \ldots, p^{(k-1)} \rangle = \langle r^{(0)}, r^{(1)}, \ldots, r^{(k-1)} \rangle,\end{aligned}$$

with the approximate solutions $x^{(k)}$, search directions $p^{(k)}$ and residuals $r^{(k)}$ constructed from Algorithm 7.4. Further, $x^{(k)}$ is the unique point in $\mathcal{K}_k$ that minimizes $\|e^{(k)}\|_A$, and convergence is monotonic,

$$\|e^{(k)}\|_A \leq \|e^{(k-1)}\|_A,$$

with $e^{(k)} = 0$ for some $k \leq n$.

Richardson iteration (R), GMRES (G), CG (C)



Low rank matrix approximations

The fundamental building block of iterative methods is the matrix-vector multiplication Ax, where $A \in R^{n \times n}$ and $x \in R^n$. If n is large this may be an expensive computation. To reduce the computational cost we can construct a *low rank approximation*

$$A_k = Q_k Q_k^T A \approx A,$$

given by an ortogonal projector $Q_k Q_k^T$, where Q_k is an $n \times k$ matrix with orthonormal column vectors that approximately span the columns space of A,

$$\text{range}(Q_k) \approx \text{range}(A),$$

but with $k << n$. By the SVD, with $Q_k = U_k$ we get

$$A_k = U_k U_k^T A = U_k U_k^T U \Sigma V^T = U_k \Sigma_k V_k^T.$$

Low rank matrix approximations

Alternatively, a randomized algorithm can be used, based on the generation of a set of k random vectors $\{g_i\}_{i=1}^{k}$ which are multiplied by the matrix A to approximate the range of A,

$$\text{span}(\{Ag_i\}_{i=1}^{k}) \approx \text{range}(A).$$

The vectors $\{Ag_i\}_{i=1}^{k}$ are then orthogonalized to form the column vectors of the matrix Q_k.

Dynamical systems

A *dynamical system* describes the time evolution over a time interval $I = [0, T]$ of a state vector $x^{(k)} \in R^n$, where the index k corresponds to a sequence of snapshots in time $\{t_k\}_{k=0}^N$. The state vector may represent the temperature measured at n positions, the concentration of n chemical species in a chemical reactor, or the dynamics of a mechanical system, for example. We can express the evolution of a linear dynamical system by the update formula

$$x^{(k+1)} = (I - \alpha A)x^{(k)} + \alpha b,$$

where $b \in R^n$ is a vector that represents data, and $(I - \alpha A)$ is a state transition matrix which describes the evolution of the system, with $\alpha \in R$ and $I, A \in R^{n \times n}$.

Dynamical systems

If we partition the time interval $I = [0, T]$ into N subintervals,

$$[0, \alpha], [\alpha, 2\alpha], [2\alpha, 3\alpha], ..., [(N-1)\alpha, N\alpha],$$

each of length $\alpha = T/N$, we can rewrite the update formula as

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha} + Ax^{(k)} = b,$$

which takes the form of a discretized differential equation with the approximate time derivative

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha} \approx \frac{dx}{dt}.$$

Dynamical systems

Now, if we assume that $b = 0$, then the solution at time $t = T$ is given by

$$x^{(N)} = (I - \alpha A)^N x^{(0)} = \left(I - \frac{T}{N}A\right)^N x^{(0)}.$$

In the limit $N \to \infty$, the solution vector converges to a state

$$\bar{x}(T) = \lim_{N\to\infty} x^{(N)} = \lim_{N\to\infty} \left(I - \frac{T}{N}A\right)^N x^{(0)} = \exp(-AT)x^{(0)},$$

expressed in terms of the *matrix exponential*, which is defined for a general $n \times n$ matrix B by

$$\exp(B) = \sum_{k=0}^{\infty} \frac{1}{k!}B^k = \lim_{N\to\infty} \left(I + \frac{1}{N}B\right)^N.$$

Stability of dynamical systems

If the matrix A is non-defective, then

$$A = X\Lambda X^{-1},$$

where Λ is a diagonal matrix with the eigenvales $\lambda_j \in C$ on the diagonal, and X is an invertible matrix which holds the corresponding eigenvectors as columns. If A is normal then X is a unitary matrix, else the eigenvectors are linearly independent but not mutually orthogonal.

By the power series definition of the matrix exponential (7.21), and the property of inverse matrices, the exponential acts directly on the diagonal matrix Λ,

$$\exp(-AT) = X \exp(-\Lambda T) X^{-1} = X \begin{bmatrix} \exp(-\lambda_1 T) & & 0 \\ & \ddots & \\ 0 & & \exp(-\lambda_n T) \end{bmatrix} X^{-1}.$$

Stability of dynamical systems

$$\exp(-AT) = Xe^{-\Lambda T}X^{-1} = X\begin{bmatrix} \exp(-\lambda_1 T) & & 0 \\ & \ddots & \\ 0 & & \exp(-\lambda_n T) \end{bmatrix} X^{-1}.$$

$$\begin{aligned} \exp(-\lambda_j T) &= \exp(-(\mathrm{Re}(\lambda_j) + i\mathrm{Im}(\lambda_j))T) \\ &= \exp(\mathrm{Re}(-\lambda_j T))\exp(i\mathrm{Im}(-\lambda_j T)). \end{aligned}$$

Here, the positive real parts $\mathrm{Re}(\lambda_j)$ correspond to decay, whereas the negative real parts represent growth. The imaginary parts of the eigenvalues $\mathrm{Im}(\lambda_j)$ do not change the size of the initial data since

$$|\exp(i\mathrm{Im}(\lambda_j))| = 1.$$

Symmetric positive definite matrix

Example 7.12. Consider the dynamical system (7.19) with $b = 0$, and a real symmetric positive definite matrix

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix},$$

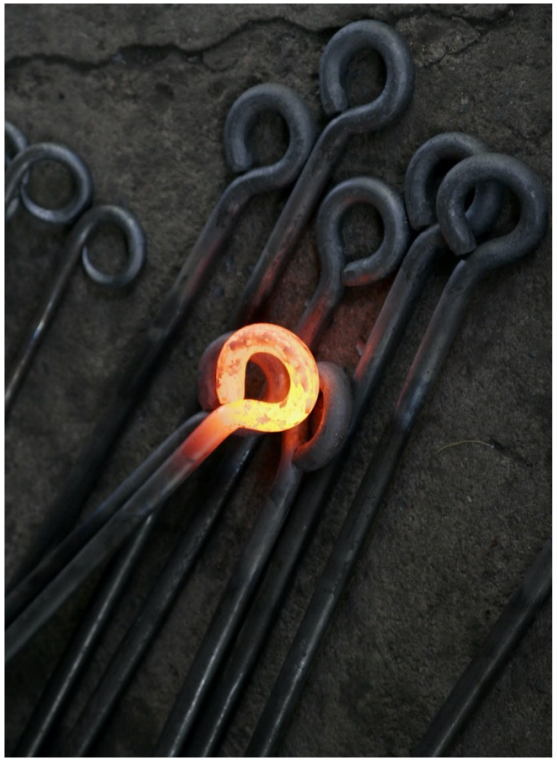
which models diffusion processes, such as heat conduction. Compare A to a discretization of the Poisson equation (5.27), see Example 5.10. The matrix A has two real positive eigenvalues $\lambda_1 = 3$ and $\lambda_2 = 1$, with associated eigenvectors

$$x^{(1)} = \frac{1}{\sqrt{2}}(1, -1)^T, \quad x^{(2)} = \frac{1}{\sqrt{2}}(1, 1)^T,$$

which implies that the initial data will dissipate with time at an exponential rate,

$$\begin{aligned} \bar{x}(T) &= \lim_{N \to \infty} x^{(N)} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \exp(-3T) & 0 \\ 0 & \exp(-1T) \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} x^{(0)} \\ &= \frac{1}{2} \begin{bmatrix} \exp(-3T) + \exp(-T) & -\exp(-3T) + \exp(-T) \\ -\exp(-3T) + \exp(-T) & \exp(-3T) + \exp(-T) \end{bmatrix} x^{(0)}. \end{aligned}$$

Diffusion and heat conduction



Skew-symmetric normal matrix

Example 7.13. The skew-symmetric normal matrix

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

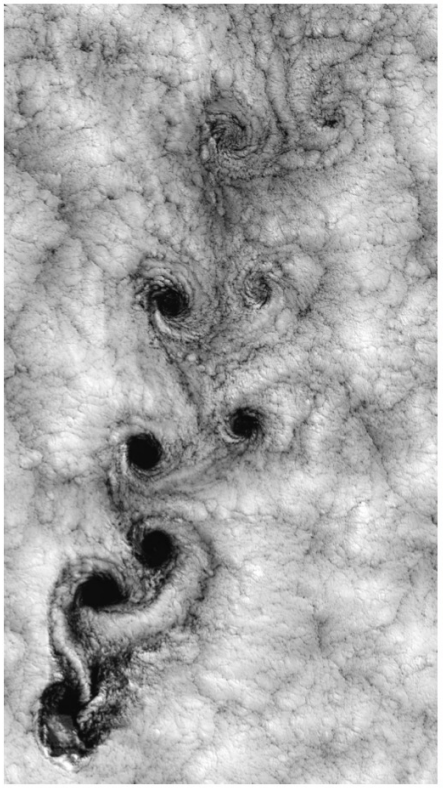
has the complex conjugate eigenvalues $\lambda_1 = i$ and $\lambda_2 = -i$, with associated eigenvectors

$$x^{(1)} = \frac{1}{\sqrt{2}}(1, i)^T, \quad x^{(2)} = \frac{1}{\sqrt{2}}(1, -i)^T.$$

Therefore, the state vector at time $t = T$ is given by

$$\begin{aligned} \bar{x}(T) &= \lim_{N \to \infty} x^{(N)} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} \exp(-iT) & 0 \\ 0 & \exp(iT) \end{bmatrix} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} x^{(0)} \\ &= \begin{bmatrix} \exp(-iT) + \exp(iT) & i(\exp(iT) - \exp(-iT)) \\ i(\exp(-iT) - \exp(iT)) & \exp(-iT) + \exp(iT) \end{bmatrix} x^{(0)}. \end{aligned}$$

Vortex and wave propagation



Defective matrix

Example 7.14. To compute the matrix exponential of the following defective matrix

$$A = \begin{bmatrix} 1 & \kappa \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & \kappa \\ 0 & 0 \end{bmatrix} = I + N,$$

we use the property that $\exp(A) = \exp(I + N) = \exp(I)\exp(N)$. The matrix N is *nilpotent*, meaning that $N^q = 0$ for all integers $q > 1$. Hence, by the power series definition of a matrix exponential (7.21),

$$\exp(N) = I + N,$$

which implies that the asymptotic state of the system $\bar{x}(T) = \lim_{N\to\infty} x^{(N)}$ is given by

$$\begin{aligned} \bar{x}(T) &= \exp(-AT)x^{(0)} = \exp(-IT)\exp(-NT)x^{(0)} = \exp(-IT)(I - NT)x^{(0)} \\ &= \left(\begin{bmatrix} \exp(-T) & 0 \\ 0 & \exp(-T) \end{bmatrix} + \begin{bmatrix} \exp(-T) & 0 \\ 0 & \exp(-T) \end{bmatrix} \begin{bmatrix} 0 & -\kappa T \\ 0 & 0 \end{bmatrix} \right) x^{(0)} \\ &= \begin{bmatrix} \exp(-T) & -\kappa T \exp(-T) \\ 0 & \exp(-T) \end{bmatrix} x^{(0)}. \end{aligned}$$

Transition to turbulence