Chapter 7

Iterative methods for large sparse linear systems

In this chapter we revisit the problem of solving linear systems of equations, but now in the context of large sparse systems. The price to pay for the direct methods based on matrix factorization is that the factors of a sparse matrix may not be sparse, so that for large sparse systems the memory cost make direct methods too expensive, in memory and in execution time.

Instead we introduce iterative methods, for which matrix sparsity is exploited to develop fast algorithms with a low memory footprint.

7.1 Sparse matrix algebra

Large sparse matrices

We say that the matrix $A \in \mathbb{R}^n$ is large if n is large, and that A is *sparse* if most of the elements are zero. If a matrix is not sparse, we say that the matrix is *dense*. Whereas for a dense matrix the number of nonzero elements is $\mathcal{O}(n^2)$, for a sparse matrix it is only $\mathcal{O}(n)$, which has obvious implications for the memory footprint and efficiency for algorithms that exploit the sparsity of a matrix.

A diagonal matrix is a sparse matrix $A = (a_{ij})$, for which $a_{ij} = 0$ for all $i \neq j$, and a diagonal matrix can be generalized to a *banded* matrix, for which there exists a number p, the *bandwidth*, such that $a_{ij} = 0$ for all $i < j - p$ or $i > j + p$. For example, a *tridiagonal* matrix A is a banded

matrix with $p = 1$,

$$A = \begin{bmatrix} \mathbf{x} & \mathbf{x} & 0 & 0 & 0 & 0 \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & 0 & 0 & 0 \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & 0 & 0 \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & 0 \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix}, \tag{7.1}$$

where $\mathbf{x}$ represents a nonzero element.

Compressed row storage

The compressed row storage (CRS) format is a data structure for efficient represention of a sparse matrix by three arrays, containing the nonzero values, the respective column indices, and the extents of the rows.

For example, the following sparse matrix

$$A = \begin{bmatrix} 3 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 2 & 4 & 0 \\ 0 & 4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 \end{bmatrix}, \tag{7.2}$$

is represented as

$$\begin{aligned} val &= [3 \;\; 2 \;\; 2 \;\; 1 \;\; 2 \;\; 2 \;\; 1 \;\; 3 \;\; 2 \;\; 4 \;\; 4 \;\; 1 \;\; 2 \;\; 3] \\ col\_idx &= [1 \;\; 2 \;\; 2 \;\; 3 \;\; 6 \;\; 2 \;\; 3 \;\; 3 \;\; 4 \;\; 5 \;\; 2 \;\; 5 \;\; 5 \;\; 6] \\ row\_ptr &= [1 \;\; 3 \;\; 6 \;\; 8 \;\; 11 \;\; 13] \end{aligned}$$

where *val* contains the nonzero matrix elements, *col_idx* their column indices, and *row_ptr* the indices in the other two arrays corresponding to the start of each row.

Sparse matrix-vector product

For a sparse matrix A, algorithms can be constructed for efficient matrix-vector multiplication $b = Ax$, exploiting the sparsity of A by avoiding multiplications by the zero elements of A.

For example, the CRS data structure implies an efficient algorithm for *sparse matrix-vector* multiplication, for which both the memory footprint and the number of floating point operations are of the order $\mathcal{O}(n)$, rather than $\mathcal{O}(n^2)$ as in the case of a dense matrix.

Algorithm 7: Sparse matrix-vector multiplication

```
for i = 1 : n do
    b_i = 0
    for j = row_ptr(i) : row_ptr(i + 1) − 1 do
        b_i = b_i + val(j)x(col_idx(j))
    end
end
```

7.2 Iterative methods

Iterative methods for large sparse linear systems

For a given nonsingular matrix $A \in \mathbb{R}^{n\times n}$ and vector $b \in \mathbb{R}^n$, we consider the problem of finding a vector $x \in \mathbb{R}^n$, such that

$$Ax = b, \tag{7.3}$$

where n is large, and the matrix A is sparse.

Now, in contrast to direct methods, we do not seek to construct the exact solution $x = A^{-1}b$ by matrix factorization, which is too expensive. Instead we develop *iterative methods* based on multiplication by a (sparse) iteration matrix, which generates a sequence of approximations $\{x^{(k)}\}_{k\geq 0}$ that converges towards x, with the *error* at iteration k given as

$$e^{(k)} = x - x^{(k)}. \tag{7.4}$$

Error estimation and stopping criterion

Since the exact solution is unknown, the error is not directly computable, but can be expressed in terms of a computable *residual* $r^{(k)} = b - Ax^{(k)}$,

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

The relative error can be estimated in terms of the relative residual and the *condition number* of A with respect to the norm $\|\cdot\|$, defined as

$$\kappa(A) = \|A\|\|A^{-1}\|. \tag{7.6}$$

Theorem 10 (Error estimate)**.** *For $\{x^{(k)}\}_{k\geq 0}$ a sequence of approximate solutions to the linear system of equations $Ax = b$, the relative error can be estimated as*

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

Proof. By (7.5), we have that

$$\|e^{(k)}\| = \|A^{-1}r^{(k)}\| \leq \|A^{-1}\|\|r^{(k)}\|, \tag{7.8}$$

and similarly

$$\|r^{(0)}\| = \|Ae^{(0)}\| \leq \|A\|\|e^{(0)}\|, \tag{7.9}$$

from which the result follows by the definition of the condition number. □

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

$$\frac{\|r^{(k)}\|}{\|r^{(0)}\|} < TOL, \tag{7.10}$$

with $TOL > 0$ the chosen tolerance.

Although, to use the relative error with respect to the initial approximation is problematic, since the choice of $x^{(0)}$ may be arbitrary, without significance for the problem at hand. It is often more suitable to formulate a stopping criterion based on the following condition,

$$\frac{\|r^{(k)}\|}{\|b\|} < TOL, \tag{7.11}$$

corresponding to $x^{(0)} = 0$.

Convergence of iterative methods

The iterative methods that we will develop are all based on the idea of *fixed point iteration*,

$$x^{(k+1)} = g(x^{(k)}), \tag{7.12}$$

where the map $x \mapsto g(x)$ may be a linear transformation in the form of a matrix, or a general nonlinear function. By *Banach fixed point theorem*, if the map satisfies certain stability conditions, the fixed point iteration (7.12) generates a *Cauchy sequence*, that is, a sequence for which the approximations $x^{(k)}$ become closer and closer as k increases.

A Cauchy sequence in a normed vector space X is defined as a sequence $\{x^{(k)}\}_{k=1}^{\infty}$, for which each element $x^{(k)} \in X$, and

$$\lim_{n\to\infty} \|x^{(m)} - x^{(n)}\| = 0, \tag{7.13}$$

for $m > n$. If all Cauchy sequences in X converges to an element $x \in X$, that is,

$$\lim_{n\to\infty} \|x - x^{(n)}\| = 0, \quad x \in X, \tag{7.14}$$

we refer to X as a *Banach space*.

The vector spaces $\mathbb{R}^n$ and $\mathbb{C}^n$ are both Banach spaces, whereas, for example, the vector space of rational numbers $\mathbb{Q}$ is not. To see this, recall that $\sqrt{2}$ is a real number that is the limit of a Cauchy sequence of rational numbers, for example, constructed by iterated bisection of the interval $[1, 2]$.

Further, a Banach space that is also an inner product space is referred to as a *Hilbert space*, which is central for the theory of differential equations.

Rate of convergence

We are not only interested in *if* an iterative method converges, but also *how fast*, that is the *rate of convergence*. We say that a sequence of approximate solutions $\{x^{(k)}\}_{k=1}^{\infty}$ converges with *order* p to the exact solution x, if

$$\lim_{k\to\infty} \frac{|x - x^{(k+1)}|}{|x - x^{(k)}|^p} = C, \quad C > 0, \tag{7.15}$$

where $p = 1$ corresponds to a *linear order of convergence*, and $p = 2$ a *quadratic order of convergence*.

We can approximate the rate of convergence by extrapolation,

$$p \approx \frac{\log \dfrac{|x^{(k+1)} - x^{(k)}|}{|x^{(k)} - x^{(k-1)}|}}{\log \dfrac{|x^{(k)} - x^{(k-1)}|}{|x^{(k-1)} - x^{(k-2)}|}}, \tag{7.16}$$

which is useful in practice when the exact solution is not available.

7.3 Stationary iterative methods

Stationary iterative methods

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

$$x^{(k+1)} = Mx^{(k)} + c, \tag{7.17}$$

with $M \in \mathbb{R}^{n\times n}$ the *iteration matrix*, $\{x^{(k)}\}_{k\geq 0} \subset \mathbb{R}^n$ a sequence of approximations, and $c \in \mathbb{R}^n$ a vector.

Theorem 11 (Banach fixed point theorem for matrices)**.** *If* $\|M\| < 1$*, the fixed point iteration (7.17) converges to the solution of the equation* $x = Mx + c$*.*

Proof. For any $k > 1$, we have that

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &= \|Mx^{(k)} - Mx^{(k-1)}\| = \|M(x^{(k)} - x^{(k-1)})\| \\ &\leq \|M\|\|x^{(k)} - x^{(k-1)}\| \leq \|M\|^k \|x^{(1)} - x^{(0)}\|. \end{aligned}$$

Further, for $m > n$,

$$\begin{aligned} \|x^{(m)} - x^{(n)}\| &= \|x^{(m)} - x^{(m-1)}\| + ... + \|x^{(n+1)} - x^{(n)}\| \\ &\leq \left(\|M\|^{m-1} + ... + \|M\|^n\right) \|x^{(1)} - x^{(0)}\|, \end{aligned}$$

so that with $\|M\| < 1$. We thus have that

$$\lim_{n\to\infty} \|x^{(m)} - x^{(n)}\| = 0, \tag{7.18}$$

that is $\{x^{(n)}\}_{n=1}^{\infty}$ is a Cauchy sequence, and since the vector space $\mathbb{R}^n$ is complete, all Cauchy sequences converge, so there exists an $x \in \mathbb{R}^n$ such that

$$x = \lim_{n\to\infty} x^{(n)}. \tag{7.19}$$

By taking the limit of both sides of (7.17) we find that x satisfies the equation $x = Mx + c$. □

An equivalent condition for convergence of (7.17) is that the spectral radius $\rho(M) < 1$. In particular, for a real symmetric matrix A, the spectral radius is identical to the induced 2-norm, that is $\rho(A) = \|A\|$.

Richardson iteration

The linear system $Ax = b$ can be formulated as a fixed point iteration through the *Richardson iteration*, with an iteration matrix $M = I - A$,

$$x^{(k+1)} = (I - A)x^{(k)} + b, \tag{7.20}$$

which will converge if $\|I - A\| < 1$, or $\rho(A) < 1$. We note that for an initial approximation $x^{(0)} = 0$, we obtain for $k = 0$,

$$x^{(1)} = (I - A)x^{(0)} + b = b$$

for $k = 1$,

$$x^{(2)} = (I - A)x^{(1)} + b = (I - A)b + b = 2b - Ab,$$

for $k = 2$,

$$x^{(3)} = (I - A)x^{(2)} + b = (I - A)(2b - Ab) + b = 3b - 3Ab + A^2b,$$

and more generally, that the iterate $x^{(k)}$ is a linear combination of powers of the matrix A acting on b, that is

$$x^{(k)} = \sum_{i=0}^{k-1} \alpha_i A^i b, \tag{7.21}$$

with $\alpha_i \in \mathbb{R}$.

Preconditioned Richardson iteration

To improve convergence of Richardson iteration we can *precondition* the system $Ax = b$ by multiplication of both sides of the equation by a matrix B, so that we get the new system

$$BAx = Bb, \tag{7.22}$$

for which Richardson iteration will converge if $\|I - BA\| < 1$, or equivalently $\rho(BA) < 1$, and we then refer to B as an *approximate inverse* of A. The preconditioned Richardson iteration takes the form

$$x^{(k+1)} = (I - BA)x^{(k)} + Bb, \tag{7.23}$$

and the preconditioned residual $Bb - BAx^{(k)}$ is used as basis for a stopping criterion.

Iterative methods based on matrix splitting

An alternative to Richardson iteration is *matrix splitting*, where stationary iterative methods are formulated based on splitting the matrix into a sum

$$A = A_1 + A_2, \tag{7.24}$$

where A_1 is chosen as a nonsingular matrix easy to invert, such as a diagonal matrix D, a (strict) lower triangular matrix L or (strict) upper triangular matrix U, where L and U have zeros on the diagonal.

Jacobi iteration

Jacobi iteration is based on the splitting

$$A_1 = D, \quad A_2 = R = A - D, \tag{7.25}$$

which gives the iteration matrix $M_J = -D^{-1}R$ and $c = D^{-1}b$, or in terms of the elements of $A = (a_{ij})$,

$$x_i^{(k+1)} = a_{ii}^{-1}(b - \sum_{j \neq i} a_{ij} x_j^{(k)}), \tag{7.26}$$

where the diagonal matrix D is trivial to invert. To use Jacobi iteration as a preconditioner, we choose $B = D^{-1}$.

Gauss-Seidel iteration

Gauss-Seidel iteration is based on the splitting

$$A_1 = L, \quad A_2 = R = A - L, \tag{7.27}$$

which gives the iteration matrix $M_{GS} = -L^{-1}R$ and $c = L^{-1}b$, or

$$x_i^{(k+1)} = a_{ii}^{-1}(b - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)}), \tag{7.28}$$

where the lower triangular matrix L is inverted by forward substitution. Gauss-Seidel iteration as a preconditioner leads to the choice of $B = L^{-1}$, where the inversion corresponds to a forward substitution.

7.4 Krylov methods

Krylov subspace

A *Krylov method* is an iterative method for the solution of the system $Ax = b$ based on, for each iteration, finding an approximation $x^{(k)} \approx x = A^{-1}b$ in a *Krylov subspace* $\mathcal{K}_k$, spanned by the vectors $b, Ab, ..., A^{k-1}b$, that is

$$\mathcal{K}_k = \langle b, Ab, ..., A^{k-1}b \rangle. \tag{7.29}$$

The basis for Krylov methods is that, by the *Cayley-Hamilton theorem*, the inverse of a matrix A^{-1} is a linear combination of its powers A^k, which is also expressed in (7.21).

GMRES

The idea of *GMRES* (generalized minimal residuals) is that, at each step k of the iteration, find 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

$$\min_{x^{(k)} \in \mathcal{K}_k} \|b - Ax^{(k)}\|. \tag{7.30}$$

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

$$\mathcal{K}_k = \langle q_1, q_2, ..., q_k \rangle, \tag{7.31}$$

with Q_k the $n \times k$ matrix with the basis vectors q_j as columns.

Thus we can express the approximation as $x^{(k)} = Q_k y$, with $y \in \mathbb{R}^k$ a vector with the coordinates of $x^{(k)}$, so that the least squares problem take the form

$$\min_{y \in \mathbb{R}^k} \|b - AQ_k y\|. \tag{7.32}$$

Algorithm 8: Arnoldi iteration

```
q_1 = b/‖b‖
for k = 1, 2, 3, ... do
    v = A q_k
    for j = 1 : k do
        h_jk = q_j^T v
        v = v − h_jk q_j
    end
    h_{n+1n} = ‖v‖
    q_{n+1} = v/h_{n+1n}
end
```

The *Arnoldi iteration* is just the modified Gram-Schmidt iteration (Algorithm 1) that constructs a partial similarity transformation of A into an *Hessenberg matrix* $\tilde{H}_k \in \mathbb{R}^{k+1 \times k}$,

$$AQ_k = Q_{k+1}\tilde{H}_k, \tag{7.33}$$

that is

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

Multiplication of (7.32) by Q_{k+1}^T does not change the norm, so that the least squares problem takes the form,

$$\min_{y \in \mathbb{R}^k} \|Q_{k+1}^T b - \tilde{H}_k y\|, \tag{7.34}$$

where we note that since $q_1 = b/\|b\|$, we have that $Q_{k+1}^T b = \|b\| e_1$ with $e_1 = (1, 0, ..., 0)^T$ the first vector in the standard basis in $\mathbb{R}^{k+1}$, so that we can write (7.34) as

$$\min_{y \in \mathbb{R}^k} \|\|b\| e_1 - \tilde{H}_k y\|, \tag{7.35}$$

which is a $(k+1) \times k$ least squares problem that we solve for $y \in \mathbb{R}^k$ at each iteration k, to get $x^{(k)} = Q_k y$.

Algorithm 9: GMRES

$q_1 = b/\|b\|$
while $\|r^{(k)}\|/\|r^{(0)}\| \geq TOL$ **do**
  Arnoldi iteration step $k \rightarrow Q_k, \tilde{H}_k$ ▷ `orthogonalize`
  $\min_{y \in \mathbb{R}^k} \|\|b\| e_1 - \tilde{H}_k y\|$ ▷ `least squares problem`
  $x^{(k)} = Q_k y$ ▷ `construct solution`
end

Conjugate Gradient method

For a symmetric positive definite matrix A, we can define the *A-norm* of a vector $x \in \mathbb{R}^n$, as

$$\|x\|_A = (x, Ax)^{1/2}, \tag{7.36}$$

with $(\cdot, \cdot)$ the l_2-norm. The *Conjugate Gradient method* (CG) is based on minimization of the error $e^{(k)} = x - x^{(k)}$ in the A-norm, or equivalently, by (7.5), minimization of the residual $r^{(k)} = b - Ax^{(k)}$ in the A^{-1}-norm,

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

to compare to GMRES where the residual is minimized in the l_2-norm.

Further, to solve the minimization problem in CG we do not solve a least squares problem over the Krylov subspace $\mathcal{K}_k$, but instead we iteratively construct a *search direction* $p^{(k)}$ and a *step length* $\alpha^{(k)}$ to find the new approximate solution $x^{(k)}$ from the previous iterate $x^{(k-1)}$. In particular, this means that we do not have to store the full Krylov basis.

Algorithm 10: Conjugate Gradient method

$x^{(0)} = 0$, $r^{(0)} = b$, $p^{(k)} = r^{(0)}$
while $\|r^{(k)}\|/\|r^{(0)}\| \geq TOL$ **do**
  $\alpha^{(k)} = \|r^{(k-1)}\|/\|p^{(k-1)}\|_A$ ▷ `step length`
  $x^{(k)} = x^{(k-1)} + \alpha^{(k)} p^{(k-1)}$ ▷ `approximate solution`
  $r^{(k)} = r^{(k-1)} - \alpha^{(k)} A p^{(k-1)}$ ▷ `residual`
  $\beta^{(k)} = \|r^{(k)}\|/\|r^{(k-1)}\|$ ▷ `improvement`
  $p^{(k)} = r^{(k)} + \beta^{(k)} p^{(k-1)}$ ▷ `search direction`
end

The key to the success of the CG method is that the residuals are mutually orthogonal,

$$(r^{(k)}, r^{(j)}) = 0, \quad \forall j < k, \tag{7.37}$$

and that the search directions are *A-conjugate*,

$$(p^{(k)}, p^{(j)})_A = 0, \quad \forall j < k, \tag{7.38}$$

where $(\cdot,\cdot)_A$ is the *weighted inner product*, defined for symmetric positive definite matrices as

$$(x, y)_A = x^T A y = (Ay)^T x = y^T A^T x = y^T A x = (y, x)_A, \tag{7.39}$$

where we note that $(\cdot,\cdot)_A$ induces the *A-norm*,

$$\|x\|_A = (x, x)_A^{1/2}, \tag{7.40}$$

which is also referred to as the *energy norm* for the equation $Ax = b$.

Theorem 12 (CG characteristics)**.** *For the CG method applied to the equation $Ax = b$, with A an $n \times n$ symmetric positive definite matrix, the orthogonality relations (7.37) and (7.38) are true, and*

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

with the approximate solutions $x^{(k)}$, search directions $p^{(k)}$ and residuals $r^{(k)}$ constructed from Algorithm 10. Further, $x^{(k)}$ is the unique point in $\mathcal{K}_k$ that minimizes $\|e^{(k)}\|_A$, and the convergence is monotonic, that is

$$\|e^{(k)}\|_A \leq \|e^{(k-1)}\|_A, \tag{7.41}$$

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