3 | The conjugate gradient method

BRIEF THEORY

Introduction

Recall that Richardson's method is defined by the formula

$$\mathbf{x}_{n+1} = (\mathbf{I} - \alpha\mathbf{A})\mathbf{x}_n + \alpha\mathbf{b} = \mathbf{x}_n + \alpha\mathbf{r}_n,$$

where $\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n$ is the residual at the n-th iteration, meaning that in each step, the current approximation is corrected by adding a fraction of the current residual. Here, we consider methods of the general form

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n\mathbf{p}_n,$$

where $\mathbf{p}_n$ is a search direction and $\alpha_n > 0$ is the step length, which may change along the iterations. Such methods are called non-stationary or dynamic methods. Different schemes provide different search directions and step lengths. The standard example of a nonstationary method is the conjugate gradient (CG) method, which only works for symmetric positive definite (SPD) matrices. The CG method belongs to the general class of Krylov space methods, see lecture 5. Here, we introduce the CG method as a modified version of the steepest descent algorithm for minimizing a function.

Remarks. A square matrix $\mathbf{A}$ is said to be symmetric, if it is equal to its transpose; that is, $\mathbf{A}(i,j) = \mathbf{A}(j,i)$ or $\mathbf{A} = \mathbf{A}^\top$. A symmetric matrix $\mathbf{A}$ is said to be positive definite, if $\mathbf{x}^\top\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$. When this is the case, all eigenvalues of $\mathbf{A}$ are greater than zero.

Minimization prelude

If $\mathbf{A}$ is an $m \times m$ SPD matrix, then, the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is equivalent to the problem of minimizing the quadratic function $\phi\colon \mathbb{R}^m \to \mathbb{R}$ with values

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{A}\mathbf{x} - \mathbf{b}^\top\mathbf{x},$$

that is, $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\min\{\phi(\mathbf{x})\colon \mathbf{x} \in \mathbb{R}^m\}$ have the same unique solution.

Remark. We use $\phi' = \nabla\phi$ and $\phi'' = \mathbf{H}_\phi$ to denote the gradient and the Hessian of a smooth function $\phi\colon \mathbb{R}^m \to \mathbb{R}$; see zeroth lecture.

For the quadratic function ϕ we obtain $\phi' = \mathbf{A}\mathbf{x} - \mathbf{b}$ and $\phi'' = \mathbf{A}$. Further, by Taylor's expansion theorem, the value $\phi(\mathbf{x}_{n+1})$ is

$$\phi(\mathbf{x}_{n+1}) = \phi(\mathbf{x}_n + \alpha_n \mathbf{p}_n) = \phi(\mathbf{x}_n) + \alpha_n \mathbf{p}_n^\top \phi'(\mathbf{x}_n) + \frac{1}{2}\alpha_n^2 \mathbf{p}_n^\top \phi''(\mathbf{v})\mathbf{p}_n,$$

where $\mathbf{v}$ lies on the segment between $\boldsymbol{u}_n$ and $\boldsymbol{u}_{n+1}$. If

i. the elements of ϕ'' are bounded in a neighbourhood of $\mathbf{x}_n$,
ii. α_n is sufficiently small, and
iii. $\mathbf{p}_n^\top \phi'(\mathbf{x}_n) < 0$,

then $\phi(\mathbf{x}_{n+1}) < \phi(\mathbf{x}_n)$, meaning that the pair $\mathbf{p}_n$, α_n decreases the value of ϕ; any direction $\mathbf{p}_n$ that leads to lower function values is said to be a descent direction. The descent direction condition $\mathbf{p}_n^\top \phi'(\mathbf{x}_n) < 0$ holds true, if we choose

$$\mathbf{p}_n = -\phi'(\mathbf{x}_n) \neq \mathbf{0},$$

since $\mathbf{p}_n^\top \phi'(\mathbf{x}_n) = -\|\phi'(\mathbf{x}_n)\|^2 < 0$; this choice corresponds to the so-called steepest descent method.

Remark. Recall that the total differential $\mathrm{d}\phi$ of a smooth function $\phi\colon \mathbb{R}^m \to \mathbb{R}$ that defines an m-dimensional manifold through the equation $\phi(\mathbf{x}) = c \in \mathbb{R}$ is

$$\mathrm{d}\phi = 0 \Leftrightarrow \frac{\partial \phi}{\partial x_1}\mathrm{d}x_1 + \frac{\partial \phi}{\partial x_2}\mathrm{d}x_2 + \cdots + \frac{\partial \phi}{\partial x_m}\mathrm{d}x_m = 0 \Leftrightarrow (\nabla\phi)^\top \mathrm{d}\mathbf{x} = 0$$

for all differential changes $\mathbf{dx} = (\mathrm{d}x_1, \mathrm{d}x_2, \dots \mathrm{d}x_m)^\top$. Hence, $\nabla\phi$ is orthogonal to all tangent vectors at a given point.

Moreover, the step length α_n may be determined as the solution to the one-dimensional line-search problem, which minimizes the value of ϕ; in particular,

$$\min_{\alpha \geq 0} \phi(\mathbf{x}_n + \alpha \mathbf{p}_n).$$

In this case, α_n is said to be optimal and $\mathrm{d}\phi(\mathbf{x}_n + \alpha\mathbf{p}_n)/\mathrm{d}\alpha = 0$ for $\alpha = \alpha_n$; hence,

$$\mathbf{p}_n^\top \phi'(\mathbf{x}_{n+1}) = 0,$$

which for the quadratic function ϕ results in the step length

$$\alpha_n = \frac{(\mathbf{b} - \mathbf{A}\mathbf{x}_n)^\top \mathbf{p}_n}{\mathbf{p}_n^\top \mathbf{A}\mathbf{p}_n} = \frac{\mathbf{r}_n^\top \mathbf{p}_n}{\mathbf{p}_n^\top \mathbf{A}\mathbf{p}_n}.$$

The conjugate gradient method

A set $\{\mathbf{p}_1, \dots, \mathbf{p}_\ell\}$ of ℓ nonzero vectors $\mathbf{p}_i$ is said to be conjugate with respect to the positive definite matrix $\mathbf{A}$ or $\mathbf{A}$-orthogonal, if

$$\langle \mathbf{p}_i, \mathbf{p}_j \rangle_{\mathbf{A}} = \left(\mathbf{p}_i, \mathbf{A}\mathbf{p}_j\right) = \mathbf{p}_i^\top \mathbf{A}\mathbf{p}_j = 0, \qquad \text{for all } i \neq j.$$

Further, $\langle \mathbf{p}_i, \mathbf{p}_i \rangle_{\mathbf{A}} > 0$, since $\mathbf{A}$ is positive definite. Thus, $\langle \cdot, \cdot \rangle_{\mathbf{A}}: \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a scalar product and we can define the so-called energy norm $\|\mathbf{p}_i\|_{\mathbf{A}} = \sqrt{\langle \mathbf{p}_i, \mathbf{p}_i \rangle_{\mathbf{A}}}$. The CG method chooses optimal step lengths and conjugate directions; that is,

$$\alpha_n = \frac{\mathbf{r}_n^\top \mathbf{p}_n}{\langle \mathbf{p}_n, \mathbf{p}_n \rangle_{\mathbf{A}}}$$

and if $\mathbf{p}_{n+1} = \mathbf{r}_{n+1} - \beta_n \mathbf{p}_n$, then

$$\langle \mathbf{p}_{n+1}, \mathbf{p}_n \rangle_{\mathbf{A}} = 0 \Leftrightarrow \langle \mathbf{r}_{n+1}, \mathbf{p}_n \rangle_{\mathbf{A}} - \beta_n \langle \mathbf{p}_n, \mathbf{p}_n \rangle_{\mathbf{A}} = 0 \Leftrightarrow \beta_n = \frac{\langle \mathbf{r}_{n+1}, \mathbf{p}_n \rangle_{\mathbf{A}}}{\langle \mathbf{p}_n, \mathbf{p}_n \rangle_{\mathbf{A}}}.$$

The CG method comprises the following sequential computations.

The conjugate gradient algorithm

Given $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_0$ compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ and $\mathbf{p}_0 = \mathbf{r}_0$; for $n = 0,1,2 \dots$ compute

$$\alpha_n = \frac{\mathbf{r}_n^\top \mathbf{p}_n}{\langle \mathbf{p}_n, \mathbf{p}_n \rangle_{\mathbf{A}}}, \qquad \mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n, \qquad \mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n \mathbf{A}\mathbf{p}_n,$$

$$\beta_n = \frac{\langle \mathbf{r}_{n+1}, \mathbf{p}_n \rangle_{\mathbf{A}}}{\langle \mathbf{p}_n, \mathbf{p}_n \rangle_{\mathbf{A}}}, \qquad \mathbf{p}_{n+1} = \mathbf{r}_{n+1} - \beta_n \mathbf{p}_n,$$

where the new direction $\mathbf{p}_{n+1}$.

The residual expression in the CG algorithm is obtained by multiplying the recursion formula $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n$ with $-\mathbf{A}$ and adding $\mathbf{b}$,

$$\mathbf{b} - \mathbf{A}\mathbf{x}_{n+1} = \mathbf{b} - \mathbf{A}\mathbf{x}_n - \alpha_n \mathbf{A}\mathbf{x}_n \Leftrightarrow \mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n \mathbf{A}\mathbf{p}_n.$$

Remarks

- It can be shown that each new direction $\mathbf{p}_{n+1}$ is also conjugate with respect to all other previous search directions; that is, $\langle \mathbf{p}_i, \mathbf{p}_j \rangle_{\mathbf{A}} = 0$, when $i \neq j$.
- In exact arithmetic, the CG method converges after at most m steps.
- The error $\mathbf{e}_n$ at the n-th iteration, with $n < m$, of the CG method is orthogonal to $\mathbf{p}_i$, $i = 0,1,\dots,n-1$, and

$$\|\mathbf{e}_n\|_{\mathbf{A}} \leq \frac{2c^n}{1 + c^{2n}} \|\mathbf{e}_0\|_{\mathbf{A}},$$

where $c = \left(\sqrt{\kappa(\mathbf{A})} - 1\right) / \left(\sqrt{\kappa(\mathbf{A})} + 1\right)$.

Example. In the following GNU Octave / Matlab snippet, we use the `pcg` function, which implements a preconditioned CG method, to solve a system with SPD coefficient matrix. The norm of the residual at each iteration is stored in `resvec` and is plotted with the `stem` function.

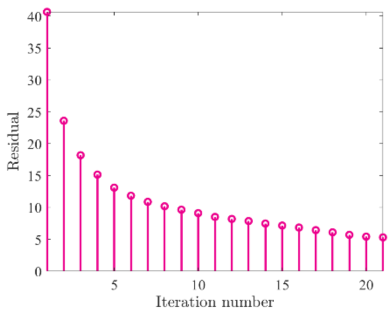
```
>> m = 5000;
>> A = diag(sparse(1:m));
>> f = rand(m, 1);
>> [x, flag, relres, iter, resvec] = pcg(A, f);
>> iter
iter =
    20
```

Preconditioning iterative methods

The convergence rate of iterative methods depends on the spectrum (the set of the eigenvalues of a matrix) of the coefficient matrix $\mathbf{A}$. Hence, we would like to transform the linear system $\mathbf{Ax} = \mathbf{b}$ into one that has better spectral properties. A preconditioner $\mathbf{P}$ (or $\mathbf{P}^{-1}$) is a non-singular matrix that approximates $\mathbf{A}$ (or $\mathbf{A}^{-1}$), in some sense, and it is easy to invert. Multiplication of $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{P}^{-1}$ from the left yields

$$\mathbf{P}^{-1}\mathbf{Ax} = \mathbf{P}^{-1}\mathbf{b},$$

which has the same solution as the original system, but its coefficient matrix $\mathbf{P}^{-1}\mathbf{A}$ may be numerically improved. If $\mathbf{P}$ is a good preconditioner, then $\mathbf{P}^{-1}\mathbf{A}$ is approximately the identity matrix. In this case, the eigenvalues of $\mathbf{P}^{-1}\mathbf{A}$ are clustered around one (or at least are away from zero) and the condition number is close to one.



i. A simple but generally poor preconditioner is the so-called Jacobi preconditioner, which consists of only the diagonal of the matrix $\mathbf{A}$; that is, $\mathbf{P} = \text{diag}(\mathbf{A})$. This preconditioner can be successful, if $\mathbf{A}$ is SPD.
ii. Choosing a sufficiently big number $\omega > 0$ so that the spectral radius of the matrix $\mathbf{B} = \mathbf{I} - \mathbf{A}/\omega$ is less than one, we obtain

$$\mathbf{A}^{-1} = \frac{1}{\omega}\left(\mathbf{I} - \left(\mathbf{I} - \frac{\mathbf{A}}{\omega}\right)\right)^{-1} = \frac{1}{\omega}(\mathbf{I} - \mathbf{B})^{-1} = \frac{1}{\omega}\sum_{k=0}^{\infty} \mathbf{B}^k.$$

Truncating the infinite sum, results in an approximation of the inverse of $\mathbf{A}$ that can be used as a preconditioner; this is a so-called polynomial preconditioner. Recall the following remark.
iii. An incomplete factorization is generated by discarding negligible values from the corresponding factorization. Two such factorizations are the incomplete LU factorization (ILU) and the incomplete Cholesky factorization (IC), which is commonly used in combination with the CG method. Further, a factorization known as modified incomplete Cholesky (MIC), dramatically speeds up the CG method. The basic idea behind MIC is that the row sums of $\mathbf{A}$ and $\mathbf{P}$ should be

the same. GNU Octave / Matlab have two built-in routines called `ilu` and `ichol` for computing the incomplete LU and Cholesky factorizations, respectively.

Remark. A Hermitian positive definite (HPD) matrix $\mathbf{A}$ admits the so-called Cholesky factorization, that is, $\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{H}}$, where $\mathbf{L}$ is a lower triangular matrix with positive (hence, real) diagonal entries, and $\mathbf{L}^{\mathrm{H}}$ denotes the conjugate transpose of $\mathbf{L}$.

COMPUTATIONAL LAB

Problem description

Here, we only provide the following naive version of the CG method. Make sure you understand each code line, before completing and using the function `cgm`. Note that the code is GNU Octave and has be transformed to Python code.

```
function [u, noi] = cgm(fun, u, f, tol)
if ( nargin == 3 ) ...; end
m = length(b);
r = f - feval(fun, ...); s = r;
for noi = 1:m
  u = feval(fun, s);
  alpha = ...;
  u = u + alpha*s;
  r = f - feval(fun, u);
  if ( sqrt(transpose(r)*r) < ... )
    ...;
  else
    beta = ...;
    s = r + beta*s;
  end
end
```

To test that the function is working as expected, solve the 3×3 symmetric positive definite system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

with pen and paper and compare the result with the one obtained, when using `cgm`, `pcg`, and `mldivide`. Then, repeat the same tests using the one- and two-dimensional Poisson BVP that has been introduced in the computational lab of the zeroth lecture.

COMPUTATIONAL LAB

Problem description

3.1 Prove that the directions generated by the CG method are indeed $\mathbf{A}$-orthogonal.

3.2 Verify by numerical experiments that for studying the convergence behaviour of the CG method we can restrict ourselves to diagonal matrices $\mathbf{A}$.

3.3 In view of problem 4.2, it is necessary to construct preconditioners in order to study the effect of clustering of eigenvalues. Do experiments with matrices that have clustered eigenvalues and compare the convergence histories with systems that have more uniformly distributed eigenvalue distributions. Does eigenvalue clustering make any difference?

3.4 Derive and implement the preconditioned CG method using the CG method applied to $\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}$.

3.5 Let $\Omega = (0,1) \times (0,1)$ and consider the BVP

$$\underbrace{\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)}_{\Delta\text{: Laplace operator}} u = -1 \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega,$$

where $\partial\Omega$ denotes the boundary of the unit square Ω. Solve this problem with all methods that you have been exposed and present in a common plot the residual history for each one of them.