5.5.11 The method of Conjugate Gradients

In this section, we will derive the Conjugate Gradients (CG) method, which is a specific implementation of the FOM algorithm. In particular, it has pleasant computational properties in the case of an SPD matrix A.

The CG method takes as its basic form the coupled recurrences formulation described above, and the coefficients are defined by demanding that the sequence of residuals $r_0, r_1, r_2, \ldots$ satisfy

$$r_i^t K^{-1} r_j = 0 \quad \text{if } i \neq j.$$

We start by deriving the CG method for nonsymmetric systems, and then show how it simplifies in the symmetric case. (The approach here is taken from [58]).

The basic equations are

$$\begin{cases} x_{i+1} = x_i - \delta_i p_i \\ r_{i+1} = r_i - \delta_i A p_i \\ p_{i+1} = K^{-1} r_{i+1} + \sum_{j \leq i} \gamma_{ji+1} p_j, \end{cases} \tag{5.25}$$

where the first and third equation were introduced above, and the second can be found by multiplying the first by A (check this!).

We will now derive the coefficients in this method by induction. In essence, we assume that we have current residual r_{cur}, a residuals to be computed r_{new}, and a collection of known residuals R_{old}. Rather than using subscripts 'old, cur, new', we use the following convention:

- x_1, r_1, p_1 are the current iterate, residual, and search direction. Note that the subscript 1 does not denote the iteration number here.
- x_2, r_2, p_2 are the iterate, residual, and search direction that we are about to compute. Again, the subscript does not equal the iteration number.
- X_0, R_0, P_0 are all previous iterates, residuals, and search directions bundled together in a block of vectors.

In terms of these quantities, the update equations are then

$$\begin{cases} x_2 = x_1 - \delta_1 p_1 \\ r_2 = r_1 - \delta_i A p_1 \\ p_2 = K^{-1} r_2 + v_{12} p_1 + P_0 u_{02} \end{cases} \tag{5.26}$$

where δ_1, v_{12} are scalars, and u_{02} is a vector with length the number of iterations before the current. We now derive δ_1, v_{12}, u_{02} from the orthogonality of the residuals. To be specific, the residuals have to be orthogonal under the K^{-1} inner product: we want to have

$$r_2^t K^{-1} r_1 = 0, \qquad r_2^t K^{-1} R_0 = 0.$$

Combining these relations gives us, for instance,

$$\left.\begin{array}{l} r_1^t K^{-1} r_2 = 0 \\ r_2 = r_1 - \delta_i A K^{-1} p_1 \end{array}\right\} \Rightarrow \delta_1 = \frac{r_1^t K^{-1} r_1}{r_1^t K^{-1} A p_1}.$$

Finding v_{12}, u_{02} is a little harder. For this, we start by summarizing the relations for the residuals and search directions in equation (5.25) in block form as

$$(R_0, r_1, r_2)\left(\begin{array}{ccc|c|c} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ \hline & & -1 & 1 & \\ \hline & & & -1 & 1 \end{array}\right) = A(P_0, p_1, p_2)\,\mathrm{diag}(D_0, d_1, d_2)$$

$$(P_0, p_1, p_2)\begin{pmatrix} I - U_{00} & -u_{01} & -u_{02} \\ & 1 & -v_{12} \\ & & 1 \end{pmatrix} = K^{-1}(R_0, r_1, r_2)$$

or abbreviated $RJ = APD$, $P(I-U) = R$ where J is the matrix with identity diagonal and minus identity subdiagonal. We then observe that

- $R^t K^{-1} R$ is diagonal, expressing the orthogonality of the residuals.
- Combining that $R^t K^{-1} R$ is diagonal and $P(I-U) = R$ gives that $R^t P = R^t K^{-1} R (I-U)^{-1}$. We now reason that $(I-U)^{-1}$ is upper diagonal, so $R^t P$ is upper triangular. This tells us quantities such as $r_2^t p_1$ are zero.
- Combining the relations for R and P, we get first that

  $$R^t K^{-t} AP = R^t K^{-t} RJD^{-1}$$

  which tells us that $R^t K^{-t} AP$ is lower bidiagonal. Expanding R in this equation gives

  $$P^t AP = (I-U)^{-t} R^t R J D^{-1}.$$

  Here D and $R^t K^{-1} R$ are diagonal, and $(I-U)^{-t}$ and J are lower triangular, so $P^t AP$ is lower triangular.
- This tells us that $P_0^t A p_2 = 0$ and $p_1^t A p_2 = 0$.
- Taking the product of $P_0^t A$, $p_1^t A$ with the definition of p_2 in equation (5.26) gives

  $$u_{02} = -(P_0^t A P_0)^{-1} P_0^t A K^{-1} r_2, \qquad v_{12} = -(p_1^t A p_1)^{-1} p_1^t A K^{-1} r_2.$$

- If A is symmetric, $P^t AP$ is lower triangular (see above) and symmetric, so it is in fact diagonal. Also, $R^t K^{-t} AP$ is lower bidiagonal, so, using $A = A^t$, $P^t A K^{-1} R$ is upper bidiagonal. Since $P^t A K^{-1} R = P^t AP(I-U)$, we conclude that $I-U$ is upper bidiagonal, so, only in the symmetric case, $u_{02} = 0$.

Some observations about this derivation.

- Strictly speaking we are only proving necessary relations here. It can be shown that these are sufficient too.
- There are different formulas that wind up computing the same vectors, in exact arithmetic. For instance, it is easy to derive that $p_1^t r_1 = r_1^t r_1$, so this can be substituted in the formulas just derived. The implementation of the CG method as it is typically implemented, is given in figure 5.11.

Compute $r^{(0)} = Ax^{(0)} - b$ for some initial guess $x^{(0)}$
for $i = 1, 2, \ldots$
  solve $Kz^{(i-1)} = r^{(i-1)}$
  $\rho_{i-1} = r^{(i-1)^T} z^{(i-1)}$
  if $i = 1$
    $p^{(1)} = z^{(0)}$
  else
    $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
    $p^{(i)} = z^{(i-1)} + \beta_{i-1} p^{(i-1)}$
  endif
  $q^{(i)} = Ap^{(i)}$
  $\delta_i = \rho_{i-1}/p^{(i)^T} q^{(i)}$
  $x^{(i)} = x^{(i-1)} - \delta_i p^{(i)}$
  $r^{(i)} = r^{(i-1)} - \delta_i q^{(i)}$
  check convergence; continue if necessary
end

Figure 5.11: The Preconditioned Conjugate Gradient Method.

- In the k-th iteration, computing $P_0^t A r_2$ (which is needed for u_{02}) takes k inner products. First of all, inner products are disadvantageous in a parallel context. Secondly, this requires us to store all search directions indefinitely. This second point implies that both work and storage go up with the number of iterations. Contrast this with the stationary iteration scheme, where storage was limited to the matrix and a few vectors, and work in each iteration was the same.
- The objections just raised disappear in the symmetric case. Since u_{02} is zero, the dependence on P_0 disappears, and only the dependence on p_1 remains. Thus, storage is constant, and the amount of work per iteration is constant. The number of inner products per iteration can be shown to be just two.

Exercise 5.57. Do a flop count of the various operations in one iteration of the CG method. Assume that A is the *matrix of a five-point stencil* and that the preconditioner M is an incomplete factorization of A (section 5.5.6.1). Let N be the matrix size.

5.5.12 Derivation from minimization

The above derivation of the CG method is not often found in the literature. The typical derivation starts with a minimization problem with a *symmetric positive definite (SPD)* matrix A:

$$\text{For which vector } x \text{ with } \|x\| = 1 \text{ is } f(x) = 1/2\, x^t Ax - b^t x \text{ minimal?} \tag{5.27}$$

If we accept the fact that the function f has a minimum, which follows from the positive definiteness, we find the minimum by computing the derivative

$$f'(x) = Ax - b.$$

and asking where $f'(x) = 0$. And, presto, there we have the original linear system.