Write-up

Yu Gu

November 21, 2018

1. **Pseudo-code for Conjugate Gradient (CG) algorithm**

Algorithm 1 CG algorithm

Initialize: u_0
$r_0 = b - Au_0$
$p_0 = r_0$
$niter = 0$
while $niter < nitermax$ **do**
  $niter = niter + 1$
  $\alpha_n = {r_n}^T r_n / {p_n}^T A p_n$
  $u_{n+1} = u_n + \alpha_n p_n$
  $r_{n+1} = r_n - \alpha_n A p_n$
  if $\|r_{n+1}\|_2 / \|r_0\|_2 < threshold$ **then**
    break
  end if
  $\beta_n = {r_{n+1}}^T r_{n+1} / {r_n}^T r_n$
  $p_{n+1} = r_{n+1} + \beta_n p_n$
end while

2. **Function design**
   Noticing that in this CG algorithm, we need to deal with some computations between constant, vectors and matrices. Thus, I design six methods below to help us solve these problems.

   - *matvecDot*: dot product of matrix with CSR format matrix and vector
     Input 4 double type std::vector val, row_ptr, col_idx represents CSR format matrix, and vec. Output is a double type std::vector.
   - *vecAdd*: add two vectors
     Input 2 double type std::vector vec1, vec2. Output is a double type std::vector.
   - *vecSubtract*: subtract two vectors
     Input 2 double type std::vector vec1, vec2. Output is a double type std::vector.
   - *vecMul*: multiply constant to vector
     Input 1 double type number a, 1 double type std::vector vec. Output is a double type std::vector.
   - *vecDot*: dot product of two vectors
     Input 2 double type std::vector vec1, vec2. Output is a double type number.
   - *vecNorm*: 2-norm of a vector
     Input 1 double type std::vector vec. Output is a double type number.