github.com/lkeegan/blockCG

Liam Keegan
liam@keegan.ch

June 28, 2018

1 Introduction

This document contains a description of the implemented algorithms. They are all variants of CG inverters, where the inversion may include N_{shifts} shifts and act simultaneously on n_{RHS} vectors. Notation conventions used:

- $\mathbf{A}$ is the hermitian positive definite $L \times L$ matrix to be inverted
- x, r, p, etc. are L–component complex vectors
- X, R, P, etc. are $L \times n_{\text{RHS}}$ complex block–vectors
- α, β, ζ, etc. are $n_{\text{RHS}} \times n_{\text{RHS}}$ complex matrices
- σ is a positive real scalar shift

The norm used here is the L2 norm, defined for a vector x as

$$|x| \equiv \left(x^\dagger x\right)^{1/2} \tag{1.1}$$

2 Solvers

2.1 CG

The standard CG solver for $\mathbf{A}x = b$, with initial guess $x_0 = 0$ and criterion for convergence that the relative norm of the residual is smaller than some number ϵ, i.e. $|\mathbf{A}x_k - b|/|b| < \epsilon$ is given in Alg. 1.

Algorithm 1 CG: Solve $\mathbf{A}x = b$

$x, p, r, t \in \mathcal{C}^L$; $\alpha, \beta \in \mathcal{R}$
$x_0 = t_0 = 0$, $r_0 = p_0 = b$
for $k = 1, 2, \ldots$ until $|r_k|/|r_0| < \epsilon$ **do**
    $t_k \leftarrow \mathbf{A}p_k$
    $\beta_k \leftarrow (p_k^\dagger t_k)^{-1}(r_{k-1}^\dagger r_{k-1})$
    $r_k \leftarrow r_{k-1} - t_k\beta_k$
    $\alpha_k \leftarrow (r_{k-1}^\dagger r_{k-1})^{-1}(r_k^\dagger r_k)$
    $x_k \leftarrow x_{k-1} + p_k\beta_k$
    $p_k \leftarrow r_k + p_k\alpha_k$
end for

2.2 SCG

Since the Krylov basis is shift–invariant, the residual r_k^σ of the solution of the shifted equation $(\mathbf{A}+\sigma)x_k^\sigma = b$, where σ is a real and positive scalar, can be related to the residual of the unshifted equation, $r_k^\sigma = r_k \zeta_k$. This allows the construction of a shifted CG (SCG) solver [1], which can solve multiple shifts wihout additional aplications of the matrix $\mathbf{A}$, as described in Alg. 2. For the case $N_{\text{shifts}} = 1$ this reduces to the CG solver of Alg. 1.

Algorithm 2 SCG: Solve $(\mathbf{A}+\sigma_j)x^{\sigma_j} = b$, where $j = 0, \ldots, N_{\text{shifts}} - 1$

$x^{\sigma_j}, p^{\sigma_j}, r, t \in \mathcal{C}^L$; $\alpha, \beta, \zeta^{\sigma_j} \in \mathcal{R}$
$x_0^{\sigma_j} = t_0 = 0$, $r_0 = p_0^{\sigma_j} = b$; $\alpha_0 = 0$, $\beta_0 = \zeta_{-1}^{\sigma_j} = \zeta_0^{\sigma_j} = 1$
for $k = 1, 2, \ldots$ until $|r_k|/|r_0| < \epsilon$ **do**
    $t_k \leftarrow (\mathbf{A}+\sigma_0)p_k^{\sigma_0}$
    $\beta_k \leftarrow ((p_k^{\sigma_0})^\dagger t_k)^{-1}(r_{k-1}^\dagger r_{k-1})$
    $r_k \leftarrow r_{k-1} - t_k\beta_k$
    $\alpha_k \leftarrow (r_{k-1}^\dagger r_{k-1})^{-1}(r_k^\dagger r_k)$
    $x_k^{\sigma_0} \leftarrow x_{k-1}^{\sigma_0} + p_k^{\sigma_0}\beta_k$
    $p_k^{\sigma_0} \leftarrow r_k + p_k^{\sigma_0}\alpha_k$
    for $j = 1, \ldots, N_{\text{shifts}} - 1$ **do**
        $\zeta_k^{\sigma_j} \leftarrow \zeta_{k-1}^{\sigma_j}\left[1 + (\sigma_j - \sigma_0)\beta_k + \alpha_{k-1}(\beta_k/\beta_{k-1})(1 - \zeta_{k-1}^{\sigma_j}/\zeta_{k-2}^{\sigma_j})\right]^{-1}$
        $x_k^{\sigma_j} \leftarrow x_{k-1}^{\sigma_j} + p_k^{\sigma_j}\beta_k(\zeta_k^{\sigma_j}/\zeta_{k-1}^{\sigma_j})$
        $p_k^{\sigma_j} \leftarrow r_k\zeta_k^{\sigma_j} + p_k^{\sigma_j}\alpha_k(\zeta_k^{\sigma_j}/\zeta_{k-1}^{\sigma_j})^2$
    end for
end for

3 Block Solvers

3.1 BCG

The BlockCG (BCG) [2] algorithm is a straightforward extension of the CG algorithm to multiple RHS vectors, described in Alg. 3.

Algorithm 3 [IN PROGRESS] BCG: Solve $AX = B$

$X, P, R, T \in \mathcal{C}^{L \times n_{\text{RHS}}}$; $\alpha, \beta \in \mathcal{C}^{n_{\text{RHS}} \times n_{\text{RHS}}}$
$X_0 = T_0 = P_0 = 0$, $R_0 = B$, $\alpha_0 = B^\dagger B$
for $k = 1, 2, \ldots$ until $|AX_k - B| \,/\, |B| < \epsilon$ **do**
    $P_k \leftarrow R_{k-1} + P_{k-1}\alpha_{k-1}$
    $T_k \leftarrow AP_k$
    $\beta_k \leftarrow (P_k^\dagger T_k)^{-1}(R_{k-1}^\dagger R_{k-1})$
    $R_k \leftarrow R_{k-1} - T_k\beta_k$
    $X_k \leftarrow X_{k-1} + P_k\beta_k$
    $\alpha_k \leftarrow (R_{k-1}^\dagger R_{k-1})^{-1}(R_k^\dagger R_k)$
end for

3.2 BCGrQ

Algorithm 4 BCGrQ: Solve $AX = B$

$X, P, R, T \in \mathcal{C}^{L\times n_{\mathrm{RHS}}}$; $\alpha, \beta, \delta \in \mathcal{C}^{n_{\mathrm{RHS}}\times n_{\mathrm{RHS}}}$
$X_0 = T_0 = P_0 = 0$, $\{R_0, \delta_0\} = \mathrm{qr}(B), \alpha_0 = 1$
for $k = 1, 2, \ldots$ until $\max_i \sqrt{\delta_k(i,i)/\delta_0(i,i)} < \epsilon$ **do**
    $P_k \leftarrow R_{k-1} + P_{k-1}\alpha^\dagger_{k-1}$
    $T_k \leftarrow AP_k$
    $\beta_k \leftarrow (P_k^\dagger T_k)^{-1}$
    $\{R_k, \alpha_k\} \leftarrow \mathrm{qr}(R_{k-1} - T_k\beta_k)$
    $X_k \leftarrow X_{k-1} + P_k\beta_k\delta_{k-1}$
    $\delta_k \leftarrow \alpha_k\delta_{k-1}$
end for

References

[1] B. Jegerlehner, *Krylov space solvers for shifted linear systems*, hep-lat/9612014.

[2] D. P. O'Leary, *The block conjugate gradient algorithm and related methods*, *Linear Algebra and its Applications* **29** (1980) 293 – 322. Special Volume Dedicated to Alson S. Householder.