Institut für Theoretische Physik — Universität Heidelberg
Dr. Alexander Rothkopf,Dr. Dénes Sexty
Quantum fields on the lattice: Excercise sheet Nr. 7 — Sommersemester 2015

Conjugate Gradient algorithm

The conjugate gradient (CG) algorithm can efficiently solve linear equations numerically. We use the notation $A\mathbf{x} = \mathbf{b}$ where A is a matrix, $\mathbf{b}$ is a given vector, and we are searching for the $\mathbf{x}$ vector. CG is an iterative method, thus it is especially efficient for sparse A matrices, such as the Dirac operator in lattice QCD.
Consider the following matrix $A = 1 + nH$, where n is a normalization and H is a sparse matrix with 1 on the offdiagonals and antiperiodic boundary conditions:

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & ... & -1 \\ 1 & 0 & 1 & 0 & ... & 0 \\ 0 & 1 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & & \ddots & \\ -1 & 0 & 0 & ... & 1 & 0 \end{pmatrix} \tag{1}$$

This matrix should have dimension $N = 64$ for the purposes of this excercise. Implement a function of the signature *void Amul(double *in,double *out);* which carries out the multiplication with A (using global variables N and n, use the fact that A is sparse).
Now implement the CG algorithm in a function with signature *int CG(double *x,double *b,void (*matmul)(double*,double*),double prec,int iterlim);* . This way you can easily reuse the algorithm by implementing a new matrix multiplication (and possibly new vector operations if you use complex vectors and (hermitian) matrices).
Generate a random vector b and solve for x. Check that you got the right solution by plugging the result back in $A\mathbf{x} = \mathbf{b}$. Investigate the number of iterations needed as a function of n.

The pseudo-code of the CG algorithm
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0; \mathbf{p}_0 = \mathbf{r}_0; k = 0$
repeat
$\alpha_k = (\mathbf{r}_k, \mathbf{r}_k)/(\mathbf{p}_k, A\mathbf{p}_k)$
$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$
if $\mathbf{r}_{k+1}$ is small enough then exit with result $\mathbf{x}_{k+1}$
$\beta_k = (\mathbf{r}_{k+1}, \mathbf{r}_{k+1})/(\mathbf{r}_k, \mathbf{r}_k)$
$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p_k}$
k=k+1
goto repeat