Algorithm 1 CGSolver

```
Initialize $u_0$
$r_0 = b$ - A$u_0$
L2normr0 = L2norm($r_0$)
$p_0 = r_0$
niter = 0
while niter < nitermax do
    niter = niter + 1
    $alpha = (r_n^T\ r_n)\ /\ (p_n^T\ \mathrm{A}p_n)$
    $u_{n+1} = u_n + alpha_n\ \mathrm{A}\ p_n$
    $r_{n+1} = r_n$ - $alpha_n$ A $p_n$
    L2normr = L2norm($r_{n+1}$)
    if L2normr/L2normr0 < threshold then
        break
    end if
    $beta_n = (r_{n+1}^T\ r_{n+1})\ /\ (r_n^T\ r_n)$
    $p_{n+1} = r_{n+1} + beta_n p_n$
```

In matvecops.cpp, I defined a few functions for usual operations on vectors and matrices: sum of two vectors, difference of two vectors, dot product of two vectors, scalar product of a float and a vector, L2 norm of a vector and matrix-vector product. These functions are then called directly in the CGSolver algorithm so that the code is easier to read.