NUMERICAL LINEAR ALGEBRA
ACADEMIC YEAR 2008-2009

MATLAB assignment day 13

In this assignment you will develop a simple multigrid code for solving the 1D Poisson equation.

1. Generate a 1D Poisson system using the following commands:
```
level = input('Level = ')
n = 2**level-1
h = 1/(n+1);
e = ones(n,1);
A = (1/h**2)*spdiags([-e 2*e -e], -1:1, n, n);
b = ones(n,1);
```
   The parameter `Level` determines the size of the system. Take `Level = 10`. Write a code that performs 10 Gauss-Seidel iteration on the system, starting with a random initial guess. Plot the residual after every iteration and verify that the Gauss-Seidel iterations smooth the residual.

2. Write subroutines for the prolongation and the restriction operation. The restriction operation is such that a vector x_c on the courser level takes as values in the gridpoints
$$x_c(i) = 0.25x_f(2*i-1) + 0.5*x_f(2*i) + 0.25*x_f(2*i+1)$$
   where x_f is the vector on the finer grid. The prolongation operation is such that
$$x_f(2*i) = x_c(i)$$
   and
$$x_f(2*i+1) = 0.5*(x_c(i) + x_c(i+1))\ .$$
   Note that $x(0) = x(n+1) = 0$. Test your subroutines, for example on the solution of the system.

3. Write a two grid method. A cycle must consist of the following steps:
   - Perform one Gauss-Seidel iteration on the approximate solution x_f (pre-smoothing);
   - Compute the residual r_f (stop if the norm of the residual is small enough).

- Transfer the residual to the courser grid (one level courser), using your restriction routine.
- Solve the system $A_c u_c = r_c$, where all vectors are defined at the courser level.
- Prolongate u_c to the finer level, add the resulting u_f to x_f.
- Perform one Gauss-Seidel iteration (post smoothing).
- Repeat the above steps until convergence.

Test your program for different problem sizes. How does the number of iterations depends on the problem size?

4. Make a recursive version of your program such that your program preforms a complete V-cycle. Take `level = 2` the lowest, on which you solve the system with a direct solver. How does the number of iterations depend on the problem size?