Hard Optimization Problems: Practical Approach

Final Project

1. **Algebraic MultiGrid (AMG) - Derivation of the coarse level system of equations.** Let A^f be symmetric positive definite and assume we want to solve $A^f x^f = b^f$. This is equivalent to minimizing $1/2 (x^f)^T A^f x^f - (x^f)^T b^f + c$, where c is a constant. Assume a subset x^c of the variables of x^f has been chosen to represent the set of the coarse variables. Define the interpolation from the coarse level variables to the fine level variables to be $x^f \leftarrow x^f + I_c^f x^c$, where I_c^f is a linear interpolation operator. The coarse level system of equations can be derived by substituting the interpolation into the minimization formulation above to obtain the minimization formulation of the coarse level which has the same structure. Find A^c and b^c.

2. **Geometric Multigrid – Correction Scheme.** Write a multigrid program to solve the Poisson equation in 2 dimensions: $u_{xx}+u_{yy}=F(x,y)$, $F(x,y)=\sin(x+2y)$, given $G(x,y)=\cos(2x+y)$ on the boundary which is the rectangle $0<=x<=3$,

$0 \le y \le 2$. Use G(x,y) also as the initial solution for the finest level. Employ 6 levels where the mesh size for the coarsest level is h=1, thus the coarsest grid is of size 3x4 with only 2 variables and can be solved directly. The same 5-point discretization to the Laplace operator is used on all levels. The residual transfer is the trivial one - injection, and the interpolation is linear. Run the program for 3 V(2,1) cycles and print out after each relaxation sweep a line showing the level number k=1,…,6 and the "dynamic" L_2 norm of the residuals. The dynamic residual at a grid point (i,j) is defined as $r_{ij}^k = f_{ij}^k - L^k u_{ij}^k$, where u^k is the approximate solution just before relaxing at (i,j), L^k is the discrete 5-point Laplace operator and f_{ij}^k is the right hand side at (i,j). The "dynamic" L_2 norm of the residual is defined to be $h^k \, SQRT(\Sigma(r_{ij}^k)^2)$, where h^k is the mesh size of level k. This norm is supposed to decrease by a factor of ~10 at each additional cycle. Hand in these printouts along with the code itself.