



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Final Report - Summer Visit 2011

R. Bank

August 1, 2012

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.



To: Rob Falgout  
From: Randy Bank  
Subject: Summer Visit  
Date: July 27, 2012

## Final Report—Summer Visit 2011

During my visit to LLNL during the summer of 2011, I worked on algebraic multilevel solvers for large sparse systems of linear equations arising from discretizations of partial differential equations. The particular emphasis this year was on finite elements matrices arising from  $hp$ -adaptive finite element computations. Such matrices are interesting in several respects. First, higher order polynomials give rise to dense blocks within the sparse matrix. For this reason I implemented a general sparse block data structure, in which the elements are blocks of arbitrary size and shape. The blocks are referenced via a  $ja$  array (pointers and block column indices) similar to a standard sparse matrix. However, *dense* matrix operations (matrix multiplies,  $ILLU$  factorizations) can now be done more efficiently using dense matrix techniques with no indirect addressing.

Within this new data structure, I implemented several solvers. The first was a block  $ILLU$  factorization using (block) minimum degree orderings, and a block version of drop tolerance. This was a generalization of a similar method I implemented for standard sparse matrices. The second basic method was a two-level iteration. The smoother in this scheme is a simple block symmetric Gauss-Seidel iteration. The coarse space is hierarchical in nature. For each vertex in the mesh, we associate the linear nodal basis function. For each edge in the mesh with degree  $p \geq 2$ , we associate a quadratic bump function. For each element with interior degree  $p \geq 3$ , we associate a cubic bubble function. If all basis functions are present, the maximum degree of this coarse space is approximately  $6N$ , where  $N$  is the number of vertices in the triangulation (2D). While this hierarchical coarse space cannot be sufficient to achieve convergence rates independent of both  $N$  and  $p$  as  $p \rightarrow \infty$ ,  $p \leq 9$  in my finite element program (due to availability of quadrature rules), so as a practical matter this is not an issue.

In addition to the work on sparse block solvers, we had several discussions related to the algebraic domain decomposition solver project. The idea here is to give each processor some portion of the fine matrix, and a coarse description of the remaining matrix. In this sense it is an algebraic version of the (geometric grid based) DD solver in *PLTMG*.