TECHNICAL REPORT NO. 328

Undulant-Block Pivoting and $(L + U)$, D' Decomposition

by

David S. Wise

Revised: September 1991

COMPUTER SCIENCE DEPARTMENT
INDIANA UNIVERSITY
Bloomington, Indiana 47405-4101

Undulant-Block Pivoting and $(L + U),\ D'$ Decomposition

David S. Wise[1]
Computer Science Department
Indiana University
Bloomington, Indiana 47405–4101
dswise@iuvax.cs.indiana.edu

revised: September, 1991

[1]Research reported herein was sponsored, in part, by the National Science Foundation under Grant Number DCR 90-027092.

Abstract

A new formulation for LU decomposition allows efficient representation of intermediate matrices when pivoting on blocks of various sizes, *i.e.* during "undulant-block" pivoting. Its efficiency arises from block encapsulization that is implicit in the data structure or that is used by the process scheduler, and defers row/column permutations that destroy such encapsulizations. It is useful, therefore, for parallel or distributed processing on matrix representations that encapsulate submatrices as substructures.

A given matrix, A is decomposed into two matrices, plus two permutations. The permutations, P and Q, are the row/column rearrangements usual to full pivoting. The principal results are $(L+U)$ and $(QDP)^{-1}$, where L and U are proper lower—and respectively upper—triangular, D is quasi-diagonal following the zero blocks in $L+U$, and $PAQ = (I+L)D(U+I)$.

An example of a motivating data structure, the quadtree representation for matrices, is reviewed. Candidate pivots for Gaussian elimination under that structure are subtrees, both constraining and assisting the pivot search; block operations there decompose accordingly. Finally, an integer-preserving version is presented.

1991 Mathematics Subject Classification: 65F05 primary; 68Q22, 65F50 secondary.
CR categories and Subject Descriptors:
G.1.3 [**Numerical Linear Algebra**]: Linear systems, Matrix inversion, Sparse and very large systems; C.1.2 [**Multiple Data Stream Architectures (Multiprocessors)**]: Parallel processors. E.1 [**Data Structures**]: Trees; D.1.1 [**Applicative (Functional) Programming Techniques**]; I.1.2 [**Algebraic Manipulation**]: Algebraic algorithms.
General Term: Algorithms.
Additional Key Words and Phrases: Gaussian elimination, block algorithm, LU factorization, quadtree matrices, quaternary trees, exact arithmetic.