New Approaches

to

Large Scale Eigen-analysis

$$\boxed{\mathbf{Ax} = \mathbf{Bx}\lambda}$$

D.C. Sorensen
Rice University

CRPC Annual Meeting 1997

Past and Recent Research Accomplishments

Development of Implicitly Restarted Methods

Development and dissemination of ARPACK and P_ARPACK

ARPACK++

New algorithms – TRQ

Regularization of Discrete Ill-Posed Problems

- Trust Region Subproblem

Applications and Outreach

Goals of Current and Future Research

Eigenvalues: Construct Projection Methods with

- No Matrix Factorizations
- Pre-Conditioning (Spectral Enhancement)
- Admit Inaccurate Solves for Shift-Invert

Reduced basis methods for Molecular Dynamics

- Extend SVD reduction and analysis techniques
- Develop reduced basis solver for (molecular) dynamical systems
- Visualize millisecond trajectories using spatial vs temporal reduced basis decomposition

Public Domain Software - ARPACK and P_ARPACK

R. Lehoucq, D.C. Sorensen, P.A. Vu, C. Yang
K. Maschhoff - Parallel

DESCRIPTION

ARPACK is a package of Fortran77 subroutines which implement the Implicitly Restarted Arnoldi Method for solving large sparse eigenvalue problems.

- Used to compute a Few (k) Eigenvalues - Specified Properties
- Symmetric, Non-symmetric, Standard or Generalized
- Fixed Storage Requirement O(nk)
- Shift Invert Mode and SVD Templates
- Reverse Communication Interface
- Users Guide

Builds on Existing Lin. Alg. Software and MP Interfaces

BLAS , LAPACK, BLACS, MPI

Distributed on Request

ftp://ftp.caam.rice.edu/pub/software/ARPACK
Also available from Netlib@ornl.gov under ScaLAPACK

COMING SOON – ARPACK++

C++ Interface to the ARPACK Fortran codes
Francisco Gomes - CRPC

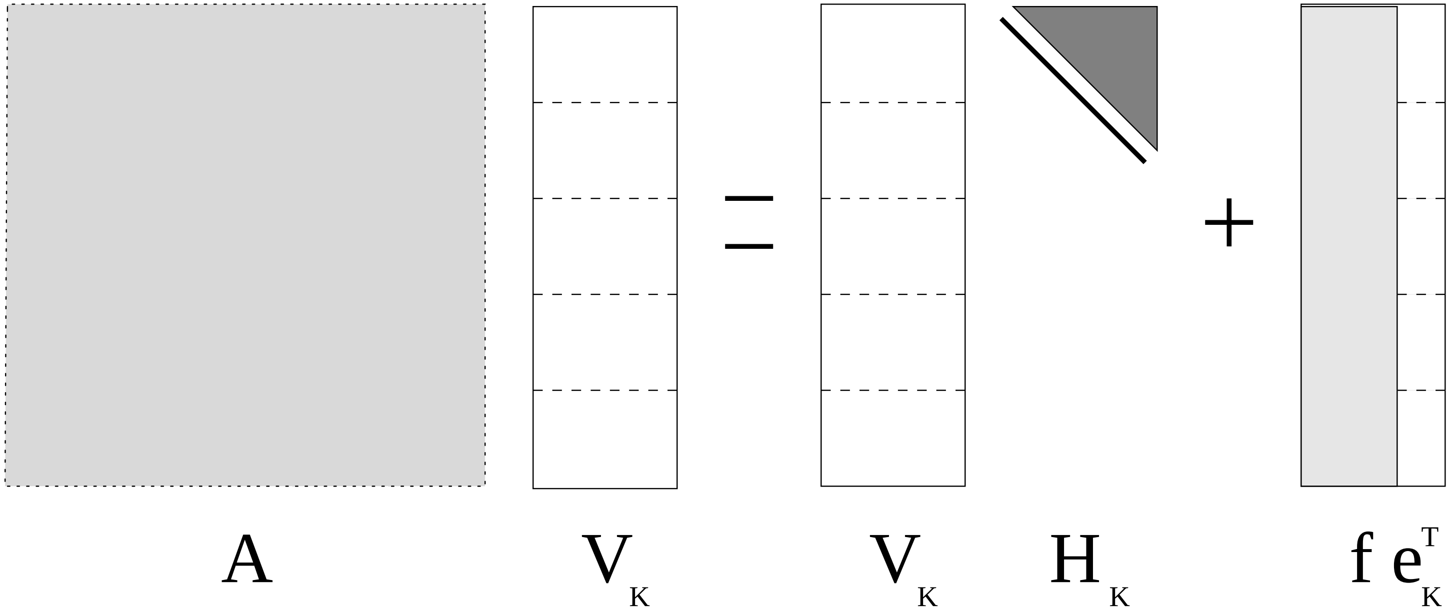
- User-Provided Member Functions
  - MatrixClass::MultMv
- Sparse Matrices
  - SuperLU
  - UMFPACK
- Reverse Communication
  - Same flavor as Fortran code

Also on the Horizon – Release 3.1

- Block extensions to IRA and IRL
- Addition of new deflation techniques

- H_k replicated on every processor
- V_k is distributed across a 1-D processor grid
- f_k and workspace distributed accordingly

Arnoldi Factorization



Parallel ARPACK - Platforms Used (So Far)

- SUN4 (SPARC 20)
  - MPI - MPICH 1.1.10
  - MPI-BLACS - MPICH 1.1.10
- SGI Power Challenge Cluster - Rice University
  - PVM-BLACS
  - MPI
- IBM SP2 - Maui HPCC
  - MPL-BLACS
  - MPI - MPICH 1.0.10
  - MPI-BLACS - MPICH 1.0.10
- CRAY T3D - Ohio Supercomputer Center
  - MPI
  - BLACS
- Intel Paragon - Sandia Nat. Labs
  - MPI
  - NX-BLACS

Parallel Performance - Non-symmetric Code

Nodes	Problem Size	Total Time (s)	Efficiency
1	100,000 * 1	40.53	
4	100,000 * 4	40.97	0.98
8	100,000 * 8	42.48	0.95
12	100,000 * 12	42.53	0.95
16	100,000 * 16	42.13	0.96
32	100,000 * 32	46.59	0.87
64	100,000 * 64	54.47	0.74
128	100,000 * 128	57.69	0.70

Internal Scalability of Parallel ARPACK

Platform: IBM SP2 - MPL-BLACS (Maui HPCC)
Maui HPCC

Test Matrix: Block Diagonal
- No Communication for MV Product

ARPACK Parameters: NEV = 4, NCV = 20

Objective of Test: Show that if a user supplies an efficient Matrix-Vector product, the internal performance of ARPACK will scale with the problem size.