Parallel Lanczos Method

for Symmetric Eigenvalue Problems

Kesheng Wu and Horst Simon

Lawrence Berkeley Lab/NERSC

PLANSO source code:

http://www.nersc.gov/research/SIMON/planso.tar.gz

Introduction

- eigenvalue λ and eigenvector x of a matrix A satisfy

$$Ax = \lambda x$$

- Rayleigh-Ritz projection: given a basis V, let $T = V^T A V$, (λ, y) be an eigenpair of T, $Ty = \lambda y$, then (λ, Vy) is an approximate eigenpair of A.
- the Lanczos iteration for computing V and T: starting with v_1,
  - $p = Av_i$, $\alpha_i = v_i^T p$
  - $r_i = p - \alpha_i v_i - \beta_{i-1} v_{i-1}$
  - $\beta_i = \|r_i\|$, $v_{i+1} = \dfrac{r_i}{\beta_i}$

Omega recurrence

- Lanczos recurrence

$$Av_i = \alpha_i v_i + \beta_{i-1} v_{i-1} + \beta_i v_{i+1}$$

- loss of orthogonality among v_i,

$$\omega_{i,j} = v_i^T v_j$$

- ω-recurrence

$$\beta_i \omega_{i+1,j} = (\alpha_j - \alpha_i)\omega_{i,j} + \beta_{j-1}\omega_{i,j-1} - \beta_{i-1}\omega_{i-1,j}$$

- partial re-orthogonalization
  - orthogonalize against previous basis vector
  - orthogonalize against all basis vectors only if $\|\omega_{i+1,\circ}\| \geq \sqrt{\varepsilon_u}$

PLANSO

a parallel implementation of Lanczos

with partial re-orthogonalization

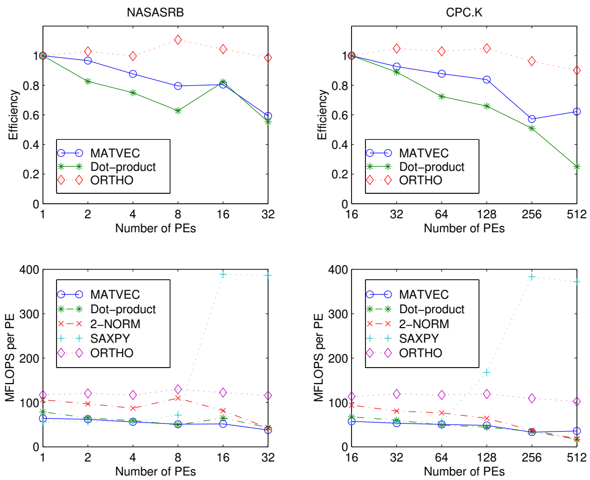
- ✔ less arithmetic operations than all similar methods
- ✔ effective in finding extreme eigenvalues and corresponding eigenvectors
- ✔ simple to implement
- ✔ no spurious solution
- ✘ no control on storage requirement
- ✘ ω-recurrence is derived for symmetric problem
- ✘ can not compute interior eigenvalues

Questions

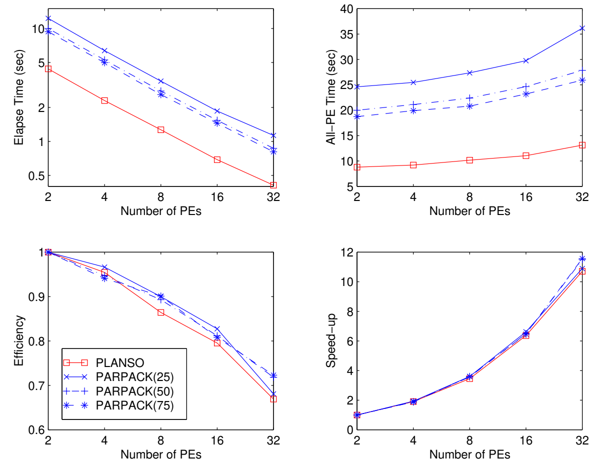
- parallel efficiency?
- compare to other parallel eigenvalue code?

Performance of major components

MPI,T3E-900



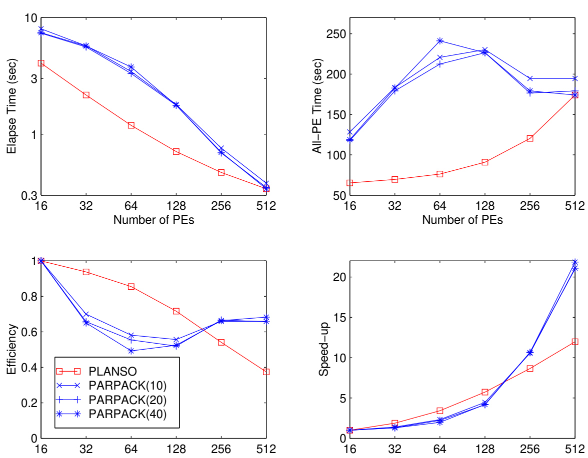
Performance of PLANSO



NASASRB 54,870x54,870, NNZ 2,677,324

5 largest eigenvalues, 75 Lanczos steps

Performance of PLANSO



CPC.K 637,244x637,244, NNZ 30,817,002

5 largest eigenvalues, 40 Lanczos steps

PLANSO -- summary

- good parallel efficiency
- less time than other parallel eigenvalue package
- portable -- MPI communication interface
- easy to interface with other parallel packages
- free code at

  http://www.nersc.gov/research/SIMON/planso.tar.gz

Thick-Restart Lanczos iterations

To control the maximum memory usage

- before restart $A\bar{V} = \bar{V}\bar{T} + \beta_m \overline{v_{m+1}} e_m^T$
- restarting with arbitrary number of Ritz pairs, say, k, after restart
- $Av_i = \alpha_i v_i + \beta_i v_{k+1}$, where $\alpha_i = \lambda_i$, v_i is the Ritz vector, β_i is the residual norm, $v_{k+1} = \overline{v_{m+1}}$
- define v_{k+2} by the following equation

$$Av_{k+1} = \alpha_{k+1} v_{k+1} + \beta_{k+1} v_{k+2} + \sum_{i=1}^{k} \beta_i v_i$$

- v_{i+1}, $i > k+1$, can be computed from the following equation

$$Av_i = \alpha_i v_i + \beta_{i-1} v_{i-1} + \beta_i v_{i+1}$$

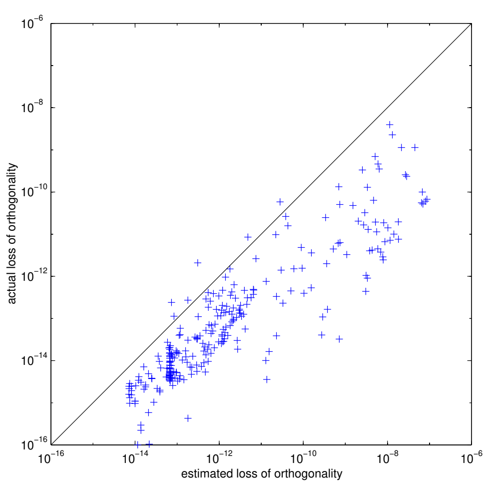
Partial re-orthogonalization after restart

- ω-recurrence can be extended to restarted Lanczos
- orthogonalize v_{m+1} to the current basis before restart
- use ω-recurrence to control global re-orthogonalization
- use ω-recurrence to control local re-orthogonalization

Attributes of thick-restart Lanczos algorithm

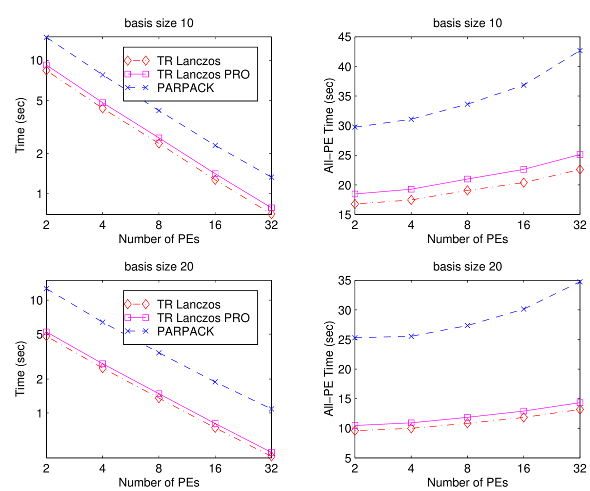
- ✔ user specify maximum basis size
- ✔ dynamic thick-restart
- ✔ minimal amount of arithmetic for orthogonalization

Accuracy of estimated loss of orthogonality



actual loss of orthogonality measured from 15 test problems

Performance of restarted Lanczos



NASASRB on Cray T3E-900