Parallel Robust Computation of Generalized Eigenvectors of Matrix Pencils

Carl Christian Kjelgaard Mikkelsen   Mirko Myllykoski

Department of Computing Science
Umeå University

PPAM-2019

Generalized eigenvalue problem

Matrices

$$\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times m}$$

Generalized eigenvalues

$$\lambda(\boldsymbol{A}, \boldsymbol{B}) = \{\lambda \in \mathbb{C} \ : \ \det(\boldsymbol{A} - \lambda \boldsymbol{B}) = 0\}$$

Generalized eigenvectors

$$\boldsymbol{A}\boldsymbol{z} = \lambda \boldsymbol{B}\boldsymbol{z}, \quad \boldsymbol{z} \in \mathbb{C}^m, \quad \boldsymbol{z} \neq 0$$

What is a robust algorithm?

We say that an algorithm is robust iff

all intermediate and final results are in the representable range

What is a robust algorithm?

We say that an algorithm is robust iff

all intermediate and final results are in the representable range

A robust algorithm returns a result which

is free of `Inf` and `NaN`

and can be evaluated in the context of the user's application.

Are there any non-robust algorithms?

Are there any non-robust algorithms?

$\mathbb{F}$	set of all finite machine numbers
$\mathbb{F}^*$	set of all machine number numbers

Addition,

$$f : \mathbb{F}^2 \to \mathbb{F}^*, \quad f(x, y) = x + y$$

is not robust.

Are there any non-robust algorithms?

$\mathbb{F}$	set of all finite machine numbers
$\mathbb{F}^*$	set of all machine number numbers

Addition,

$$f : \mathbb{F}^2 \to \mathbb{F}^*, \quad f(x, y) = x + y$$

is not robust.

However, if

$$V = \{(x, y) \in \mathbb{F}^2 \ : \ |x| \leq \Omega - |y|\}$$

then

$$f\Big|_V \to \mathbb{F},$$

is robust.

Eigenvectors can overflow

Matrices

$$\boldsymbol{S} = \begin{bmatrix} a & & & \\ -c & a+b & & \\ \vdots & \ddots & \ddots & \\ -c & \dots & -c & a+mb \end{bmatrix}, \quad \boldsymbol{T} = \boldsymbol{I}.$$

Generalized eigenvectors

$$\boldsymbol{X} = [x_{ij}], \quad x_{ij} = \begin{cases} 0 & i < j \\ z_{i-j} & i \geq j \end{cases}, \quad z_k = \binom{\gamma + k + 1}{k}, \quad \gamma = \frac{c}{b}.$$

Lower bound for x_{ij}

$$\gamma \geq m \quad \Rightarrow \quad x_{ij} \geq 2^{i-j}, \quad i \geq j$$

Is overflow a real concern?

	not robust	robust
triangular system (single RHS)	xTRSV	xLATRS
triangular system (multiple RHS)	xTRSM	
standard eigenvectors		xTREVC(3)
generalized eigenvectors		xTGEVC

The problem has been the development of algorithms which are

blocked, parallel and robust

Real arithmetic

The complex equation

$$\boldsymbol{S}(\boldsymbol{x} + i\boldsymbol{y}) = (a + ib)\boldsymbol{T}(\boldsymbol{x} + i\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m, \quad a, b \in \mathbb{R}$$

Real arithmetic

The complex equation

$$\boldsymbol{S}(\boldsymbol{x} + i\boldsymbol{y}) = (a + ib)\boldsymbol{T}(\boldsymbol{x} + i\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m, \quad a, b \in \mathbb{R}$$

is equivalent to the real equation

$$\boldsymbol{S}\begin{bmatrix}\boldsymbol{x} & \boldsymbol{y}\end{bmatrix} = \boldsymbol{T}\begin{bmatrix}\boldsymbol{x} & \boldsymbol{y}\end{bmatrix}\begin{bmatrix}a & b \\ -b & a\end{bmatrix}$$

Real arithmetic

The complex equation

$$\boldsymbol{S}(\boldsymbol{x} + i\boldsymbol{y}) = (a + ib)\boldsymbol{T}(\boldsymbol{x} + i\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m, \quad a, b \in \mathbb{R}$$

is equivalent to the real equation

$$\boldsymbol{S}\begin{bmatrix}\boldsymbol{x} & \boldsymbol{y}\end{bmatrix} = \boldsymbol{T}\begin{bmatrix}\boldsymbol{x} & \boldsymbol{y}\end{bmatrix}\begin{bmatrix}a & b \\ -b & a\end{bmatrix}$$

Small twist to allow infinite ($a/0$) and indefinite problems ($0/0$)

$$\lambda = \frac{\text{complex numerator}}{\text{denominator}} = \frac{a + ib}{\gamma}$$

Real arithmetic

The complex equation

$$\boldsymbol{S}(\boldsymbol{x} + i\boldsymbol{y}) = (a + ib)\boldsymbol{T}(\boldsymbol{x} + i\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m, \quad a, b \in \mathbb{R}$$

is equivalent to the real equation

$$\boldsymbol{S} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

Small twist to allow infinite ($a/0$) and indefinite problems ($0/0$)

$$\lambda = \frac{\text{complex numerator}}{\text{denominator}} = \frac{a + ib}{\gamma}$$

Small modification

$$\boldsymbol{S} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} \begin{bmatrix} \gamma & 0 \\ 0 & \gamma \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

Equivalent matrix equation

- Pencil $(\boldsymbol{S}, \boldsymbol{T})$ in real Schur form
- User's selection is closed under complex conjugation
- Diagonal matrix $\boldsymbol{D}$ represents selected denominators
- Block diagonal matrix $\boldsymbol{B}$ represents selected numerators
- Matrix $\boldsymbol{V}$ represents selected eigenvectors
- Equivalent matrix equation is

$$\boldsymbol{SVD} - \boldsymbol{TVB} = \mathbf{0}$$

Blocked algorithm

Partition all matrices conformally. Then

$$\boldsymbol{SVB} = \begin{bmatrix} \boldsymbol{S}_{11} & \boldsymbol{S}_{12} \\ & \boldsymbol{S}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{11} & \boldsymbol{V}_{12} \\ & \boldsymbol{V}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{D}_{11} & \\ & \boldsymbol{D}_{22} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{T}_{11} & \boldsymbol{T}_{12} \\ & \boldsymbol{T}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{11} & \boldsymbol{V}_{12} \\ & \boldsymbol{V}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_{11} & \\ & \boldsymbol{B}_{22} \end{bmatrix} = \boldsymbol{TVB}$$

is equivalent to

$$\boldsymbol{S}_{22}\boldsymbol{V}_{22}\boldsymbol{D}_{22} - \boldsymbol{T}_{22}\boldsymbol{V}_{22}\boldsymbol{B}_{22} = \mathbf{0}$$

$$\boldsymbol{S}_{11}\boldsymbol{V}_{12}\boldsymbol{D}_{22} - \boldsymbol{T}_{11}\boldsymbol{V}_{12}\boldsymbol{B}_{22} = -(\boldsymbol{S}_{12}\boldsymbol{V}_{22}\boldsymbol{D}_{22} - \boldsymbol{T}_{12}\boldsymbol{V}_{22}\boldsymbol{B}_{22})$$

$$\boldsymbol{S}_{11}\boldsymbol{V}_{11}\boldsymbol{D}_{11} - \boldsymbol{T}_{11}\boldsymbol{V}_{11}\boldsymbol{B}_{11} = \mathbf{0}$$

generalized eigenvector
multi-shift linear solve   linear update
generalized eigenvector

Necessary kernels

1. Kernel solving tiny linear systems

$$\gamma \boldsymbol{S}\boldsymbol{x} - \lambda \boldsymbol{T}\boldsymbol{x} = \boldsymbol{f}$$

   of dimensions 1 or 2 using real arithmetic only

2. Kernel solving matrix equations

$$\boldsymbol{SXD} - \boldsymbol{TXB} = \boldsymbol{F}$$

3. Generalized xGEMM doing updates

$$\boldsymbol{Y} \leftarrow \boldsymbol{Y} - (\boldsymbol{SXD} - \boldsymbol{TXB})$$

Overflow protection

Overflow protection

1. If $|b| \leq |t|\Omega$ and $t \neq 0$, then

$$y \leftarrow \frac{b}{t}$$

   cannot execeed Ω.

Overflow protection

1. If $|b| \leq |t|\Omega$ and $t \neq 0$, then

$$y \leftarrow \frac{b}{t}$$

   cannot execeed Ω.

2. If

$$\|\boldsymbol{y}\|_\infty + \|\boldsymbol{T}\|_\infty \|\boldsymbol{x}\|_\infty \leq \Omega,$$

   then

$$\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{T}\boldsymbol{x}$$

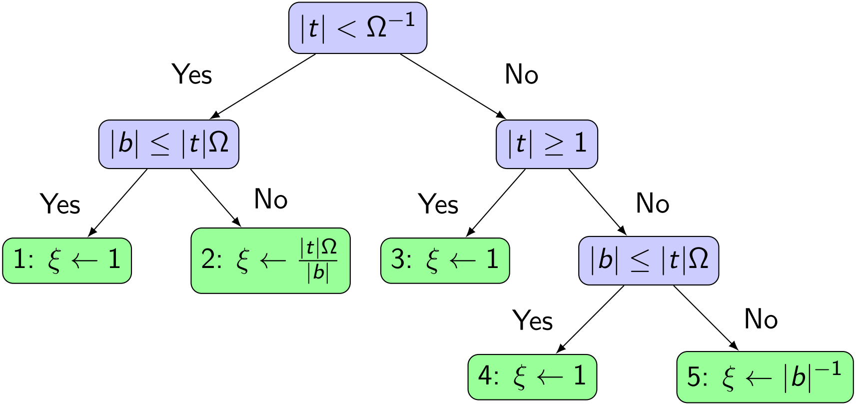
   can be computed without exceeding Ω.

Division

The division

$$y \leftarrow \frac{(\xi b)}{t}$$

cannot overflow

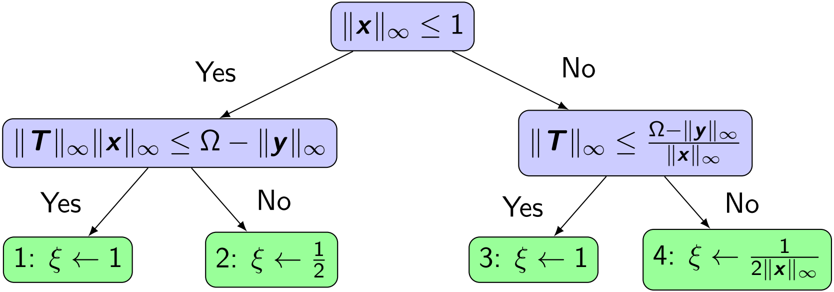


Linear update

The linear update

$$\boldsymbol{y} \leftarrow \xi \boldsymbol{y} - \boldsymbol{T}(\xi \boldsymbol{x})$$

can be computed without overflow



Parallel, blocked and robust algorithms

Start with your favorite blocked algorithm ...

1. Augment each block $\boldsymbol{X}$ with a row vector $\boldsymbol{\xi}$
2. The augmented block $\langle \boldsymbol{\xi}, \boldsymbol{X} \rangle$ represents $\boldsymbol{Y}$ given by

$$\boldsymbol{y}_j = \xi_j^{-1} \boldsymbol{x}_j$$

3. Replace all kernels with robust kernels on augmented blocks

Robust equivalent of $Z = Y - TX$

Construct

$$(\langle \boldsymbol{\alpha}, \boldsymbol{X} \rangle, \quad \langle \boldsymbol{\beta}, \boldsymbol{Y} \rangle) \quad \rightarrow \quad \langle \boldsymbol{\xi}, \boldsymbol{Z} \rangle \tag{1}$$

such that

$$\xi_j^{-1} \boldsymbol{z}_j = \alpha_j^{-1} \boldsymbol{x}_j - T(\beta_j^{-1} \boldsymbol{y}_j)$$

where ξ_j ensures $\boldsymbol{z}_j$ can be computed without overflow.

1. For each pair of columns $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ do
   1.1 Compute scaling needed for consistency
   1.2 Compute scaling needed to prevent overflow
   1.3 Apply accumulated scaling factor
2. Apply linear update (xGEMM)
3. For each column z_j of $\boldsymbol{Z}$ do
   3.1 Compute vector norm

Implementation and numerical results

1. Shared memory machine
2. Task-parallelism using StarPU
3. Integer scaling factors

- Experiments were performed on the Kebnekaise system, HPC2N, Umeå University.
- Relevant compute node types:
  - **Regular compute node:** 28 Intel Xeon E5-2690v4 Broadwell cores. 128 GB memory, 2 NUMA islands. FDR Infiniband.
  - **Large memory node:** 72 Intel Xeon E7-8860v4 Broadwell cores. 3072 GB memory, four NUMA islands.

Eigenvectors (shared memory performance)

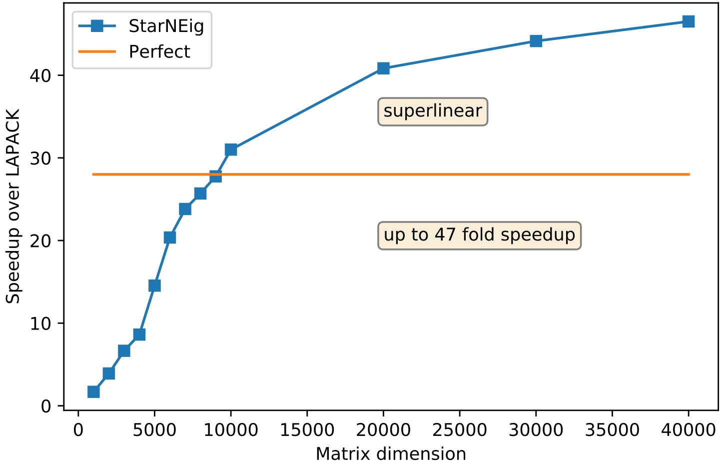


Figure: Speedup over LAPACK DTGEVC, 100% selected, 28 cores.

Eigenvectors (shared memory scalability)

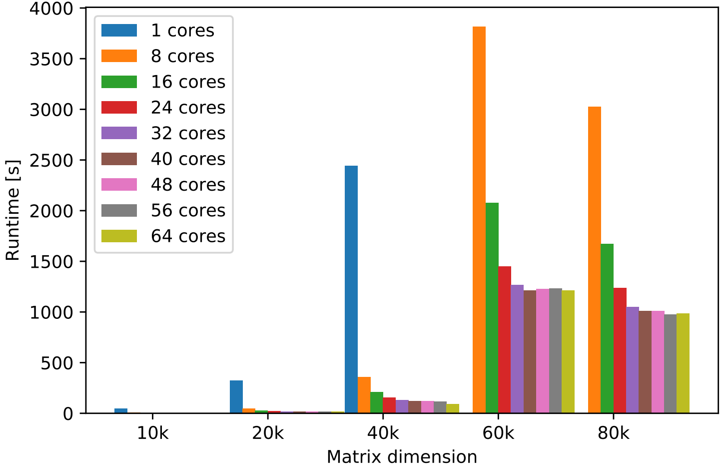


Figure: 35% of eigenvalues selected.

Eigenvectors (shared memory scalability)

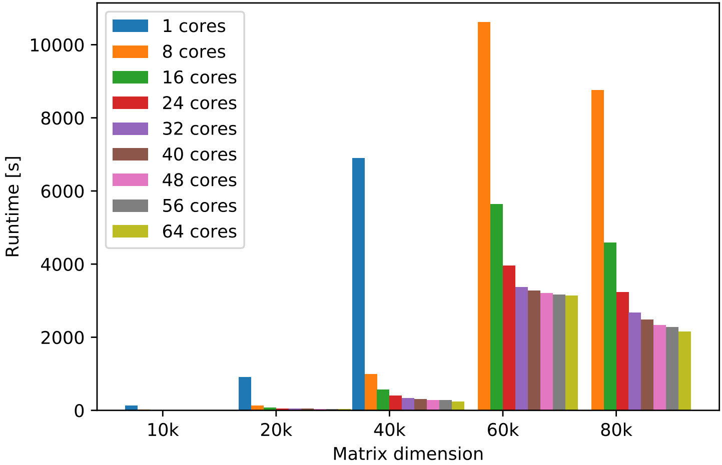


Figure: 100% of eigenvalues selected.

Questions?