Accelerating computation of eigenvectors

Mark Gates, Azzam Haidar, Khairul Kabir

March 21, 2014

ICL UT
INNOVATIVE
COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

Eigenvalue Applications

- Air flow over bridges
- Airplane design
- Chemical engineering reaction-diffusion
- Molecular resonance in electro-magnetic fields
- Integrated circuit design

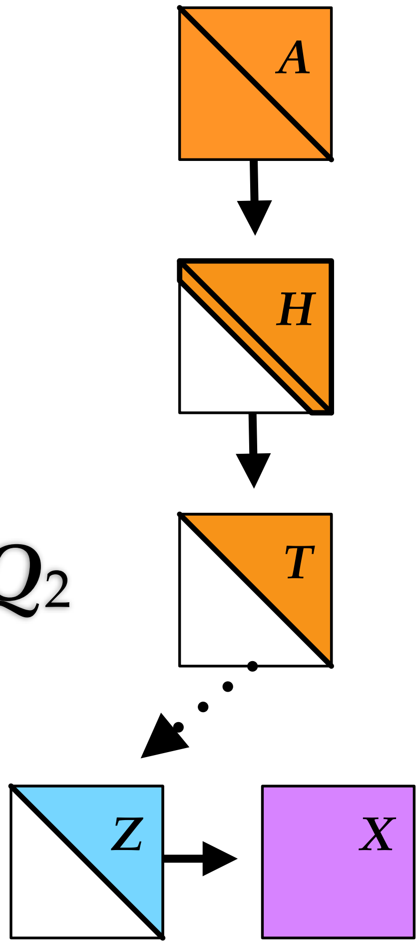
1940 Tacoma Narrows
bridge collapse
in 40 mph wind
Video: Barney Elliott

Nonsymmetric eigenvalues

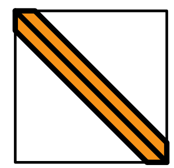
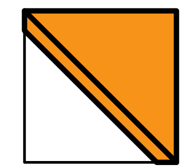
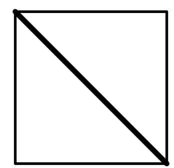
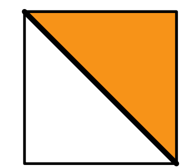
- A is n x n, nonsymmetric

$$Ax = \lambda\, x$$

- Three phases:
  - Hessenberg reduction, $H = Q_1^T A\, Q_1$
  - QR iteration to triangular form, $T = Q_2^T H\, Q_2$
  - Compute eigenvectors Z of T and back-transform to eigenvectors X of A

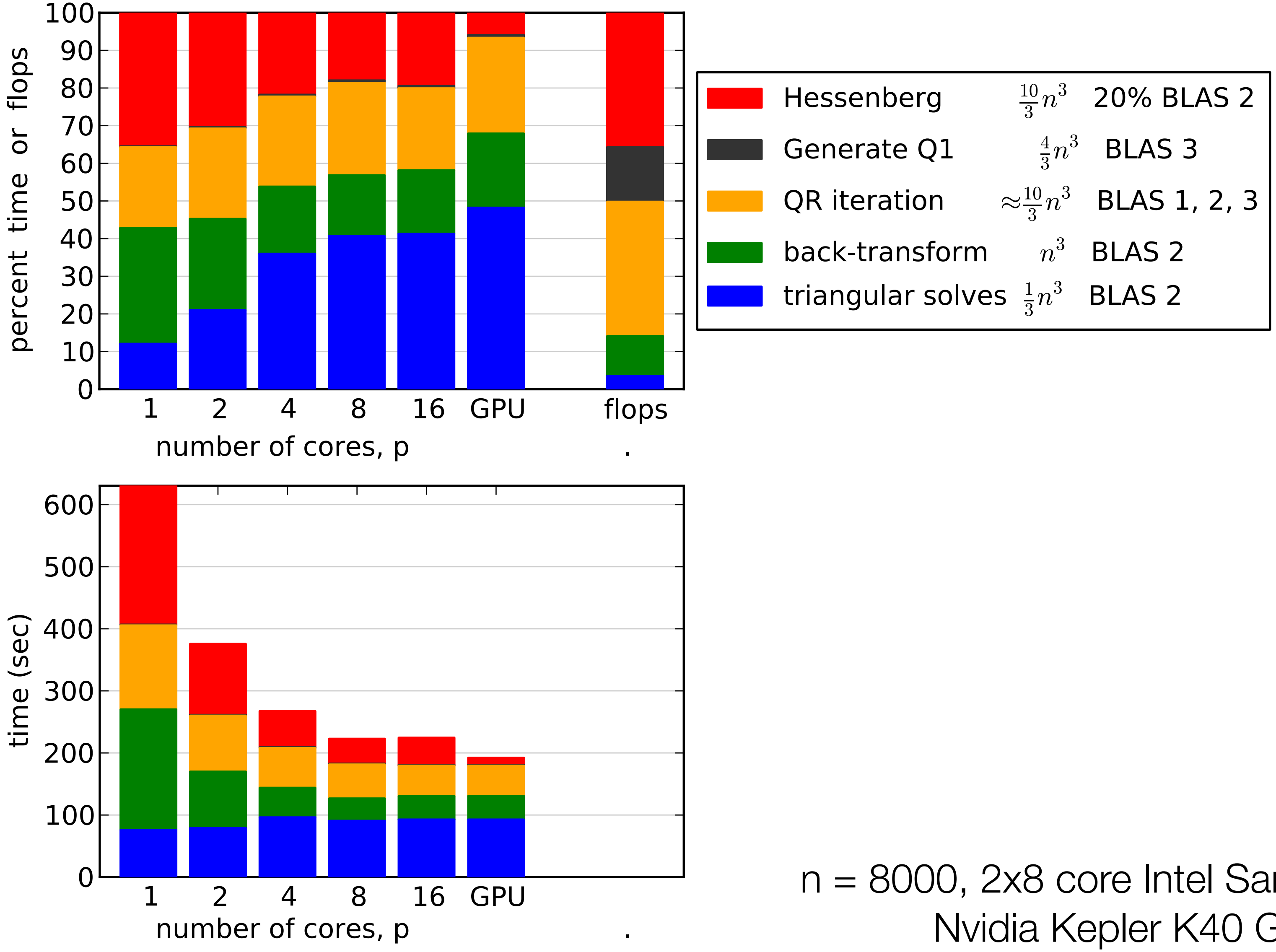


Comparison with symmetric

	Symmetric (Hermitian)		Nonsymmetric	
Condensed form	tridiagonal, $O(n)$ entries		Hessenberg, $O(n^2)$ entries	
Final form	diagonal, $\mathbf{\Lambda}$		triangular, T	
Eigenvectors of $\mathbf{\Lambda}$, T	trivial – identity		triangular solves – Z	
Back-transform	$X = Q_1 Q_2$		$X = (Q_1 Q_2)\, Z$	
Eigenvalues	real, even if A is complex		complex, even if A is real	

LAPACK Performance

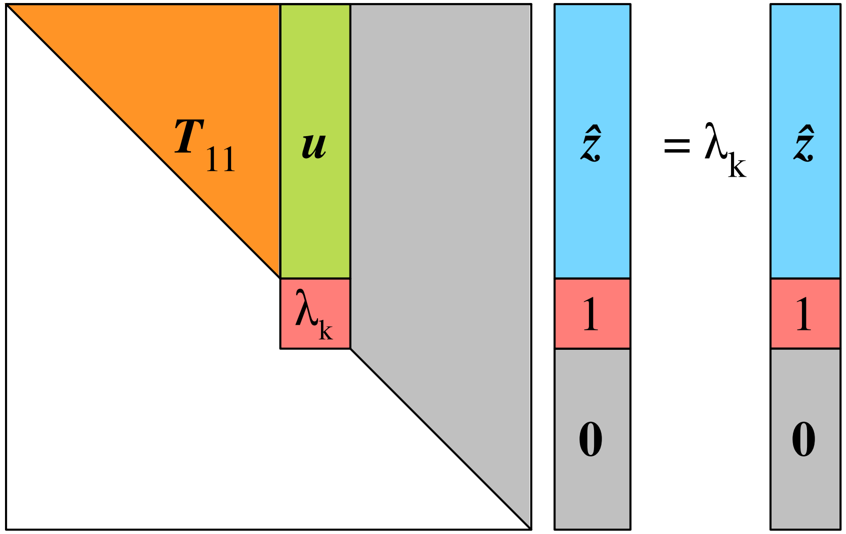
- BLAS 2 operations dominate overall cost



n = 8000, 2x8 core Intel Sandy Bridge,
Nvidia Kepler K40 GPU

Compute eigenvector

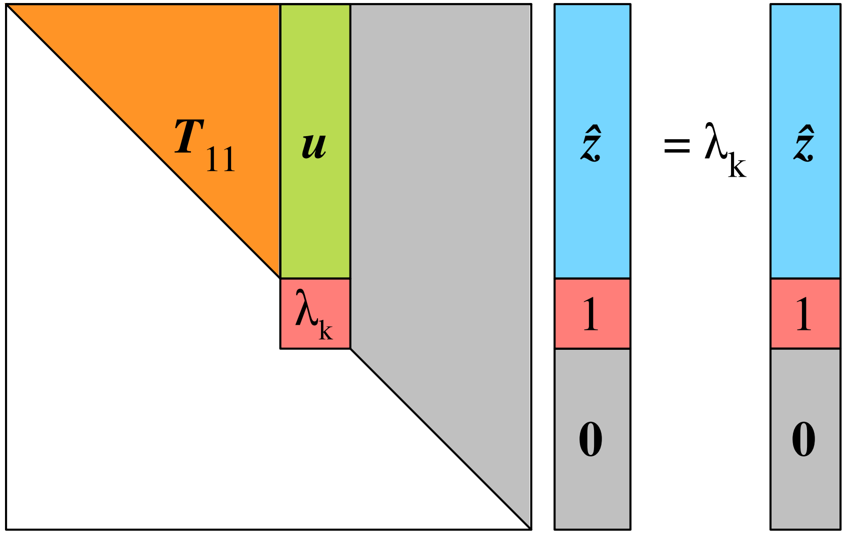
- Eigenvalues λ_k are diagonal elements of $\boldsymbol{T}$
- Solving $\boldsymbol{T}\boldsymbol{z} = \lambda_k \boldsymbol{z}$ yields $(\boldsymbol{T}_{11} - \lambda_k \boldsymbol{I})\,\hat{\boldsymbol{z}} = -\boldsymbol{u}$

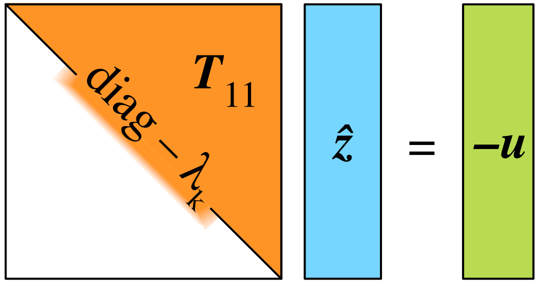


- $n - 1$ triangular solves, diagonal modified by each λ_k

Compute eigenvector

- Eigenvalues λ_k are diagonal elements of $\boldsymbol{T}$
- Solving $\boldsymbol{Tz} = \lambda_k \boldsymbol{z}$ yields $(\boldsymbol{T}_{11} - \lambda_k \boldsymbol{I})\,\hat{\boldsymbol{z}} = -\boldsymbol{u}$

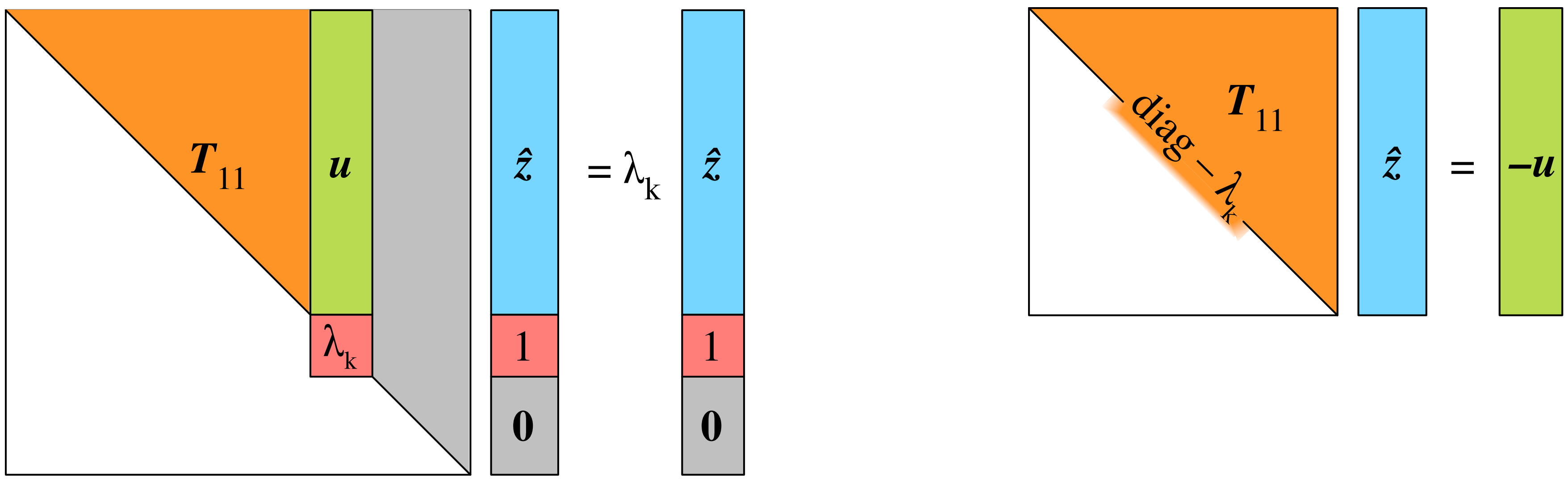




- $n - 1$ triangular solves, diagonal modified by each λ_k

Compute eigenvector

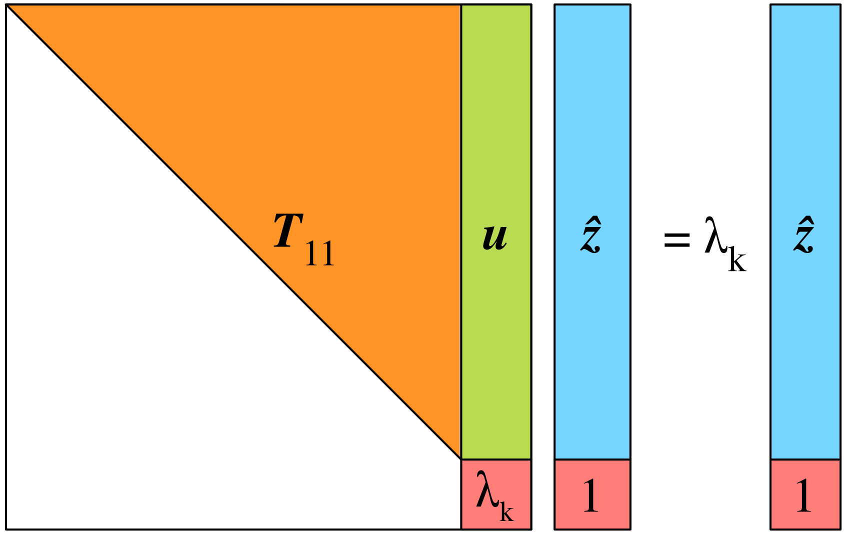
- Eigenvalues λ_k are diagonal elements of $\boldsymbol{T}$
- Solving $\boldsymbol{T}\boldsymbol{z} = \lambda_k \boldsymbol{z}$ yields $(\boldsymbol{T}_{11} - \lambda_k \boldsymbol{I})\, \hat{\boldsymbol{z}} = -\boldsymbol{u}$

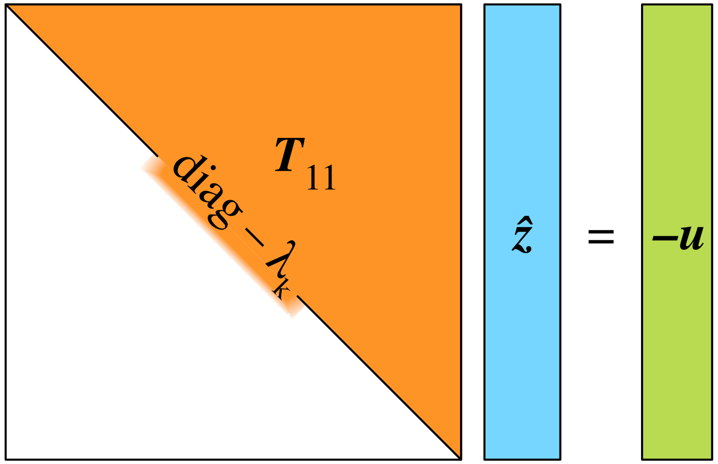


- $n - 1$ triangular solves, diagonal modified by each λ_k

Compute eigenvector

- Eigenvalues λ_k are diagonal elements of $\boldsymbol{T}$
- Solving $\boldsymbol{Tz} = \lambda_k \boldsymbol{z}$ yields $(\boldsymbol{T}_{11} - \lambda_k \boldsymbol{I})\,\hat{\boldsymbol{z}} = -\boldsymbol{u}$

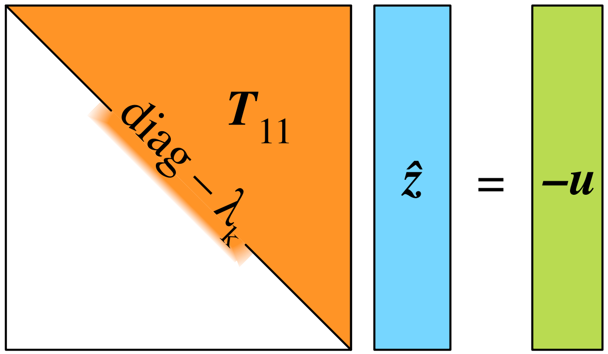




- $n - 1$ triangular solves, diagonal modified by each λ_k

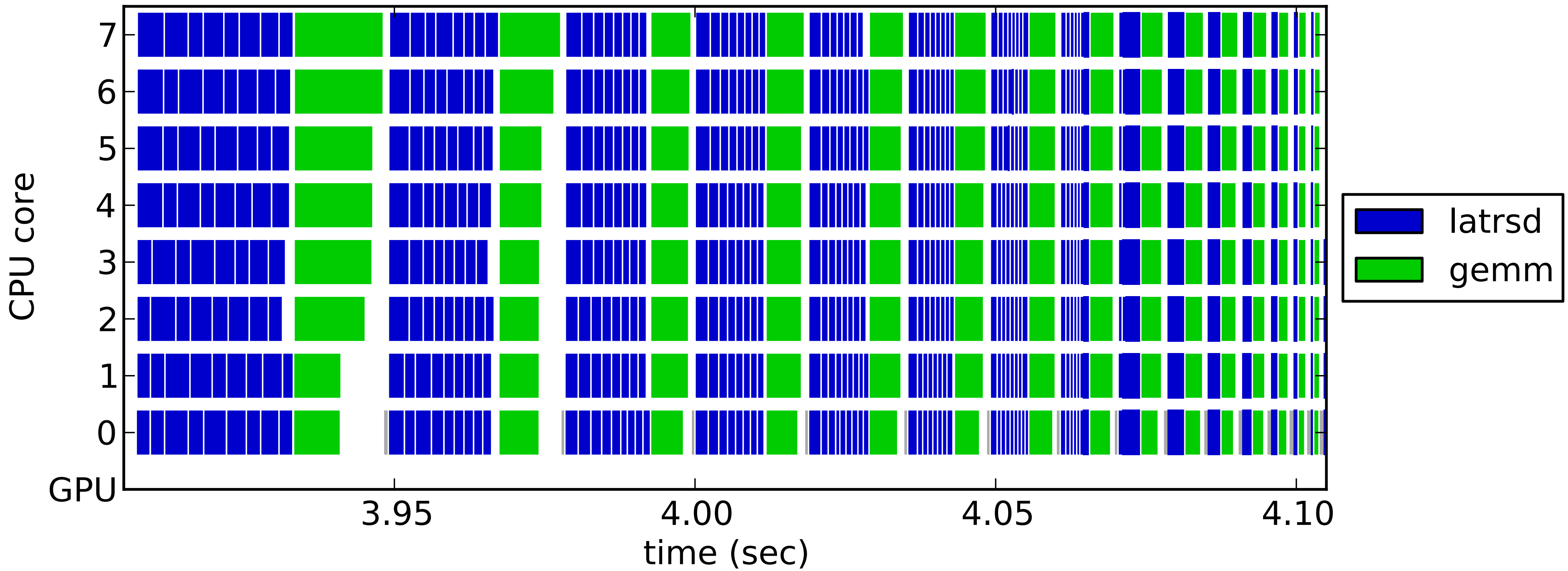
Triangular solve

- Safe solver **latrs** instead of **trsv**
  - Handles singular and ill-conditioned T matrix
  - Computes column-by-column → single threaded
- Previously, LAPACK modified T for each **latrs**
- New routine **latrsd** — subtracts λ_k from diagonal, without modifying T
- Thread pool executes multiple solves in parallel



Trace

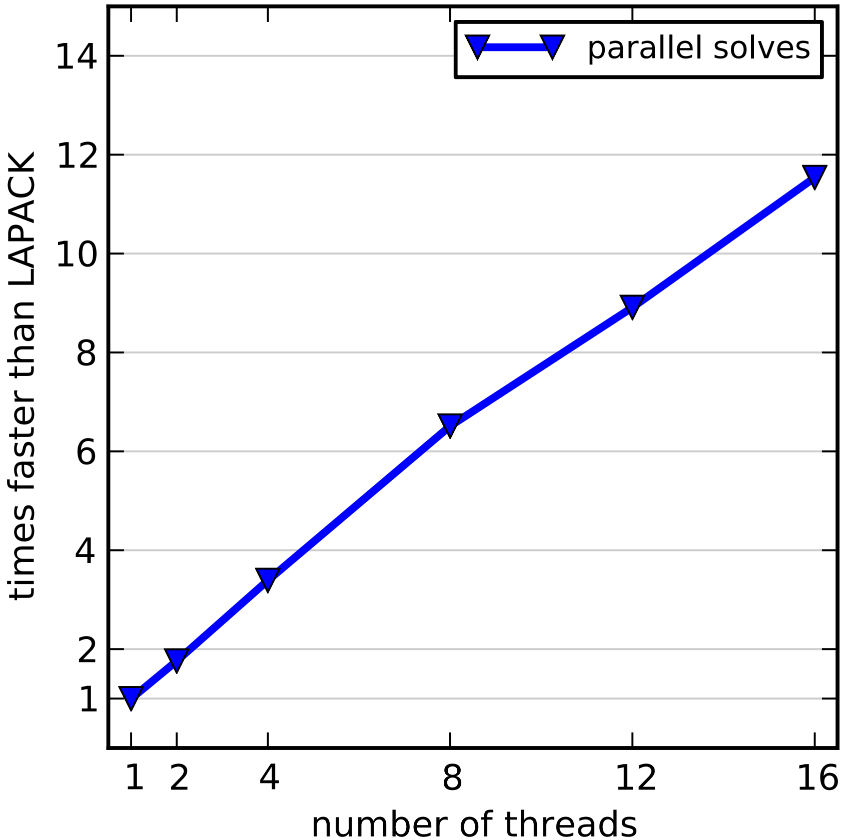
- **latrsd** decreases in time as $k \to 0$
- Thread pool maintains load balance

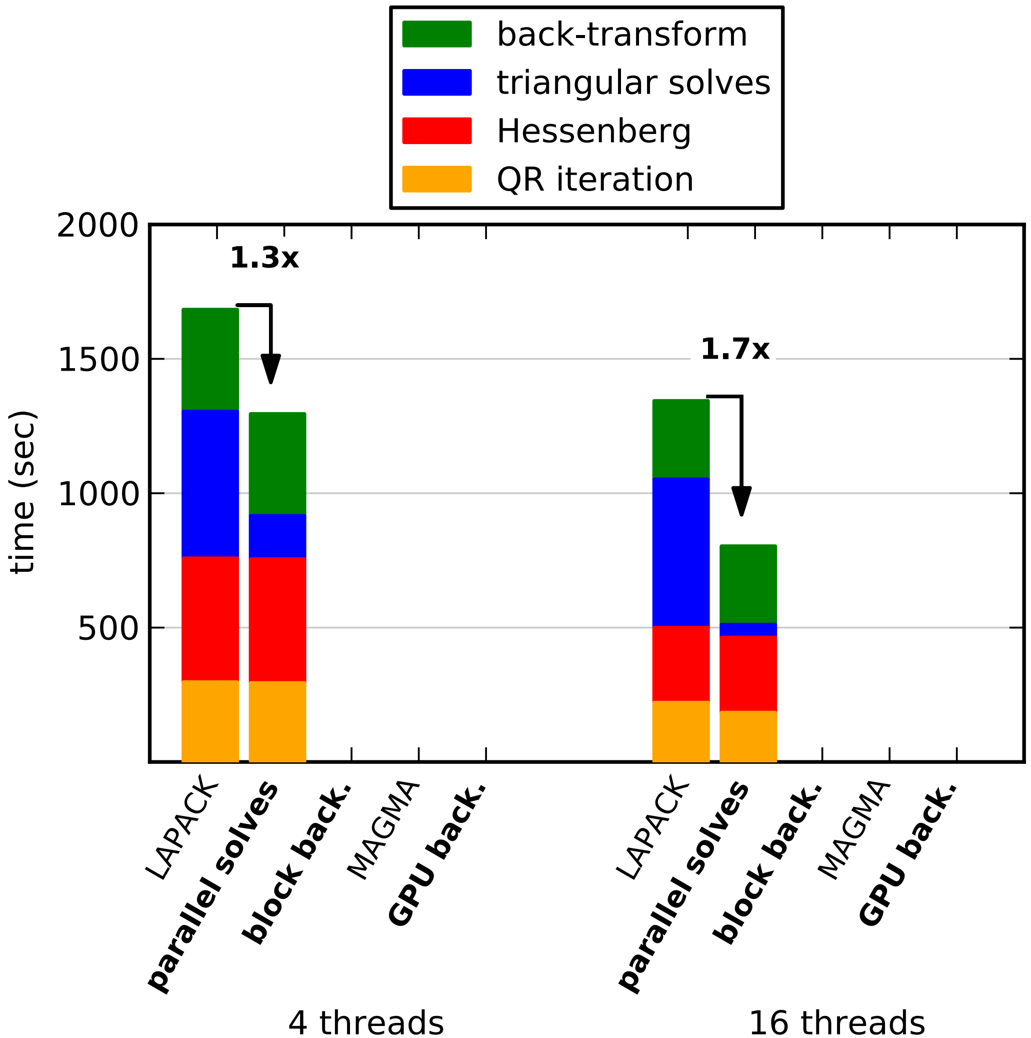


n = 2000, n_b = 64 on 8 core Intel Sandy Bridge

Performance

- Good parallel scaling



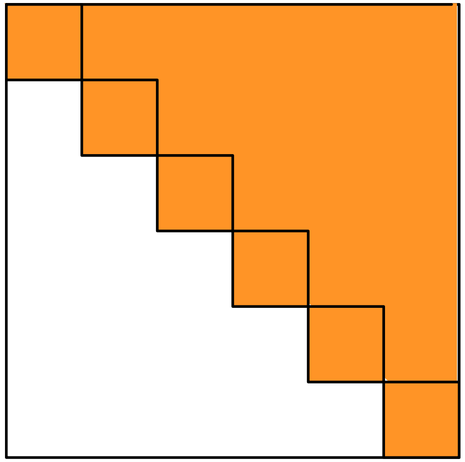


n = 16000, 2x8 core Intel Sandy Bridge

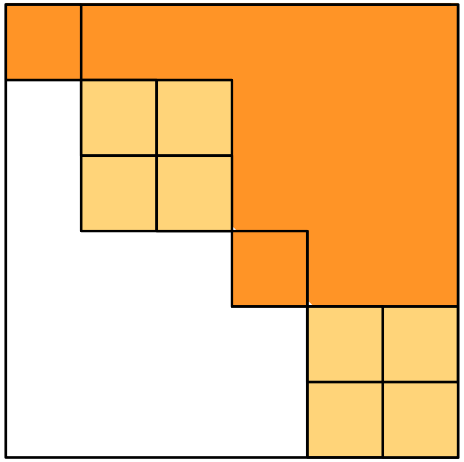
Complications for real

- Eigenvalues can be complex, even if A is real
- 2x2 diagonal blocks for conjugate eigenvalue pairs

Complex Schur form T



Real Schur form T

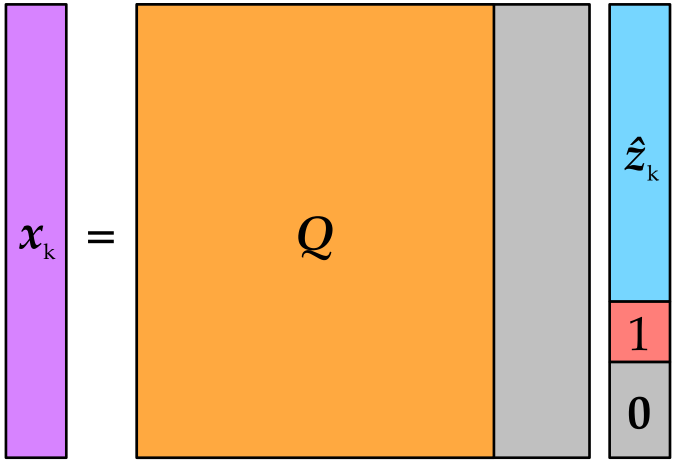


- Needs specialized quasi-triangular solver
- Factor quasi-triangular solver **dlaqtrsd** out of **dtrevc**

Back-transform

Original, unblocked

- Each vector, $\boldsymbol{x}_k = \boldsymbol{Q}\boldsymbol{z}_k$



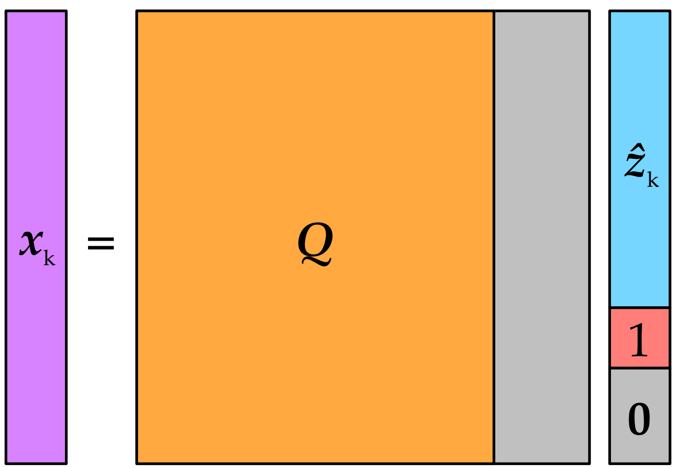
- n **gemv**, size n x k

  for $k = n, n-1, \ldots, 1$

Back-transform

Original, unblocked

- Each vector, $\boldsymbol{x}_k = \boldsymbol{Q}\boldsymbol{z}_k$

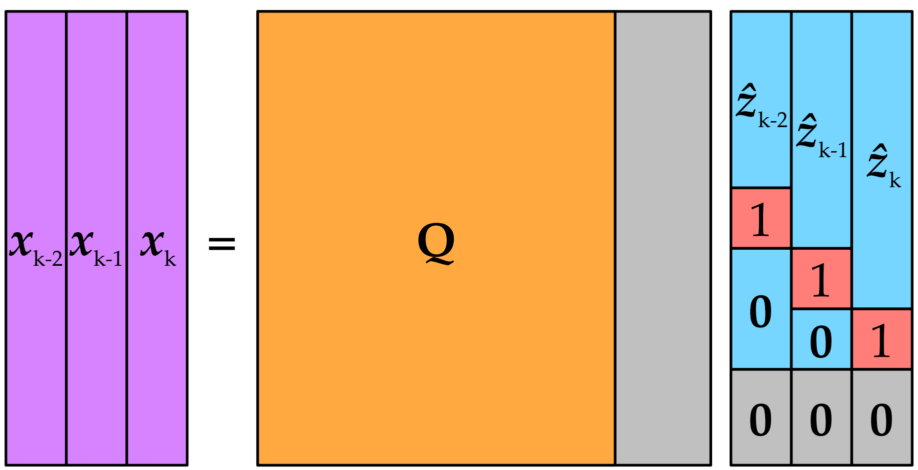


- n **gemv**, size n x k

  for $k = n, n-1, \ldots, 1$

New, blocked

- Block n_b vectors, $\boldsymbol{X}_k = \boldsymbol{Q}\boldsymbol{Z}_k$

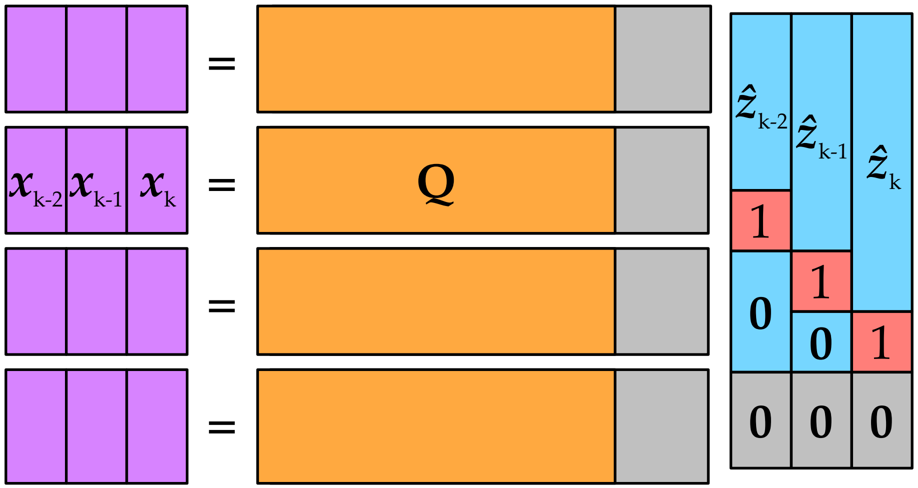


- $\frac{n}{n_b}$ **gemm**, size n x k x n_b

  for $k = n, n - n_b, \ldots$

Parallel GEMM

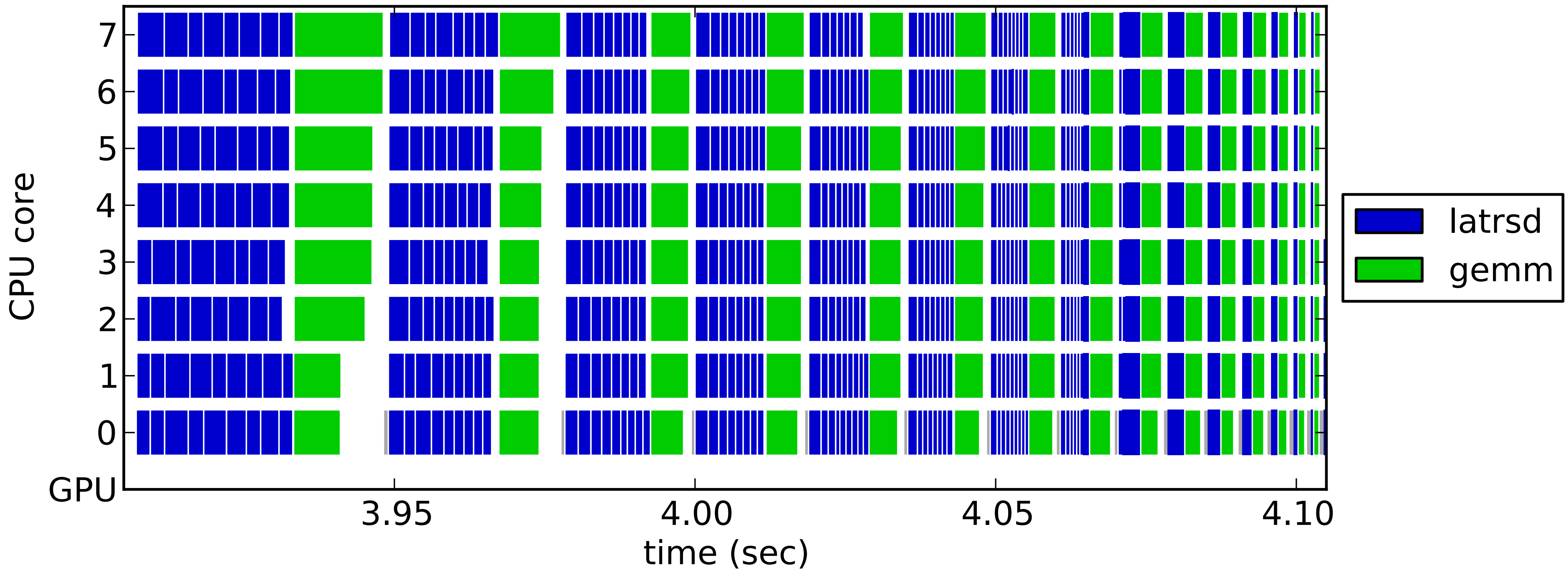
- Split Q by block-rows
- Each block-row is single-threaded **gemm** task
- Use same thread pool as triangular solves



Trace

(same as before)

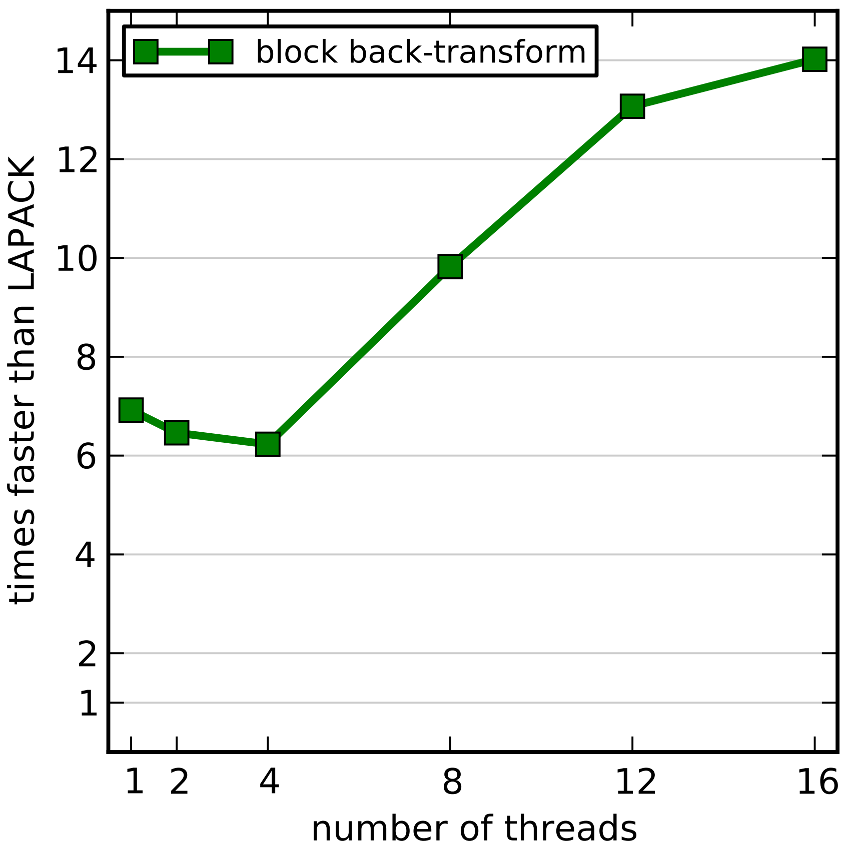
- **gemm** decreases in time as $k \to 0$

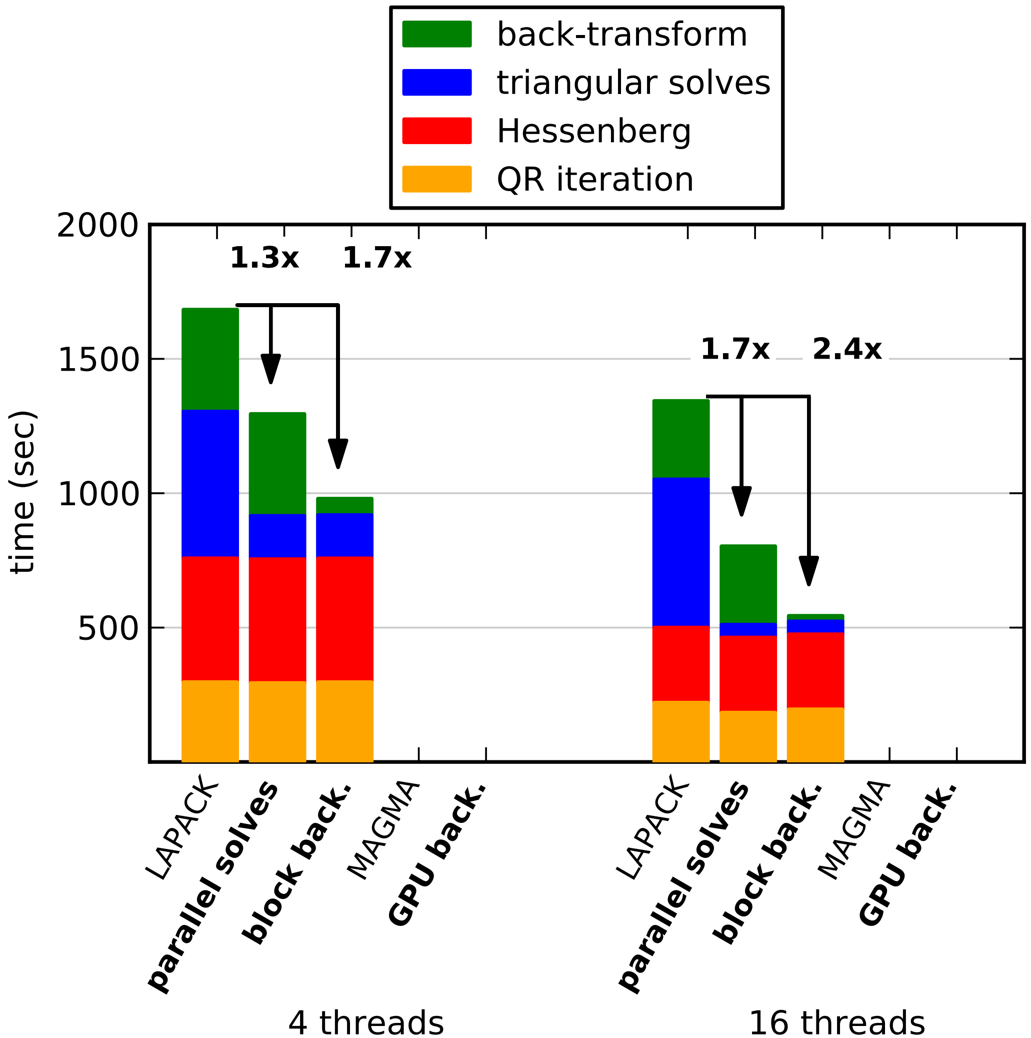


n = 2000, n_b = 64 on 8 core Intel Sandy Bridge

Performance

- Improvement for all number of threads

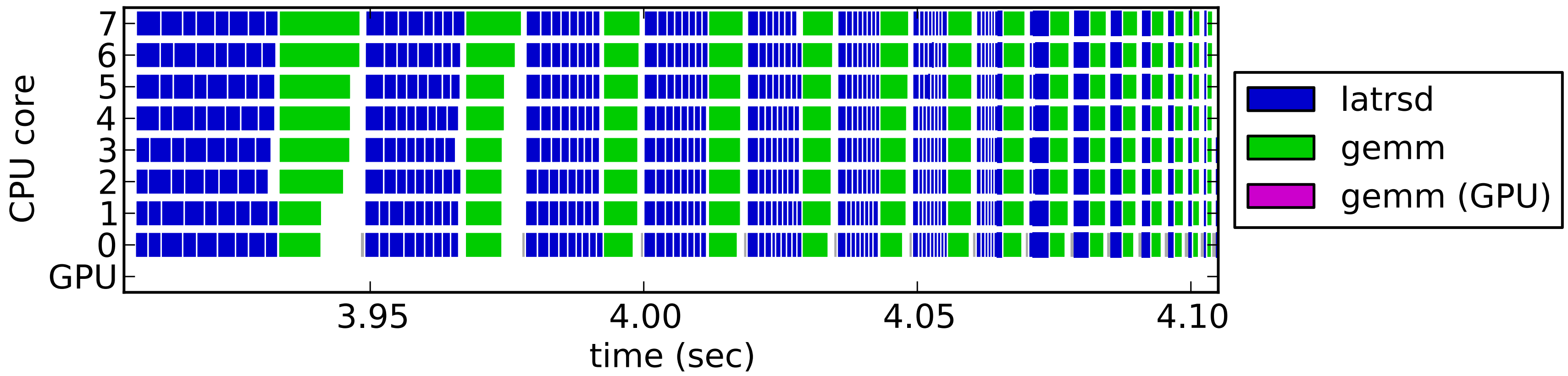


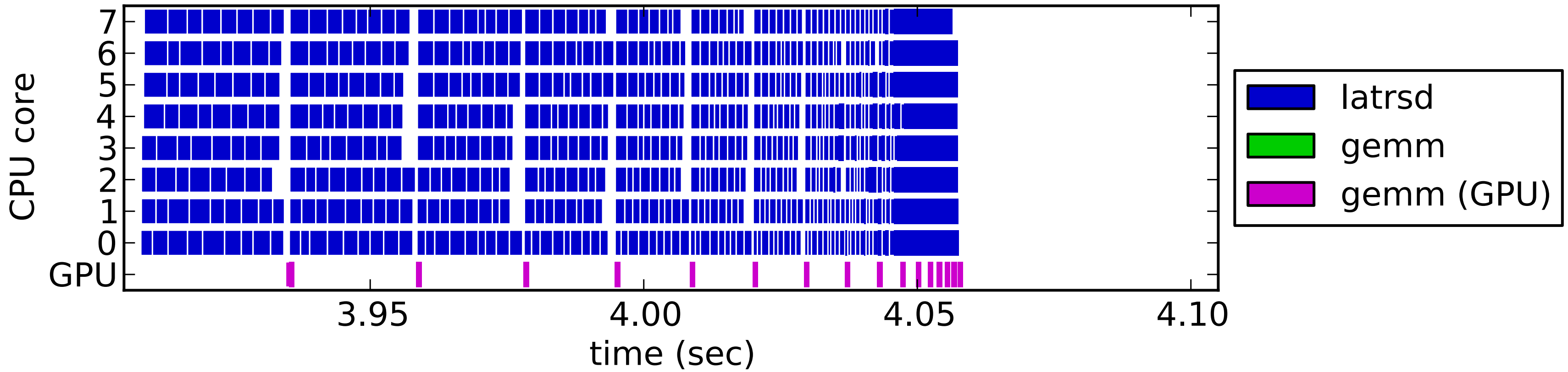


n = 16000, 2x8 core Intel Sandy Bridge

GPU acceleration

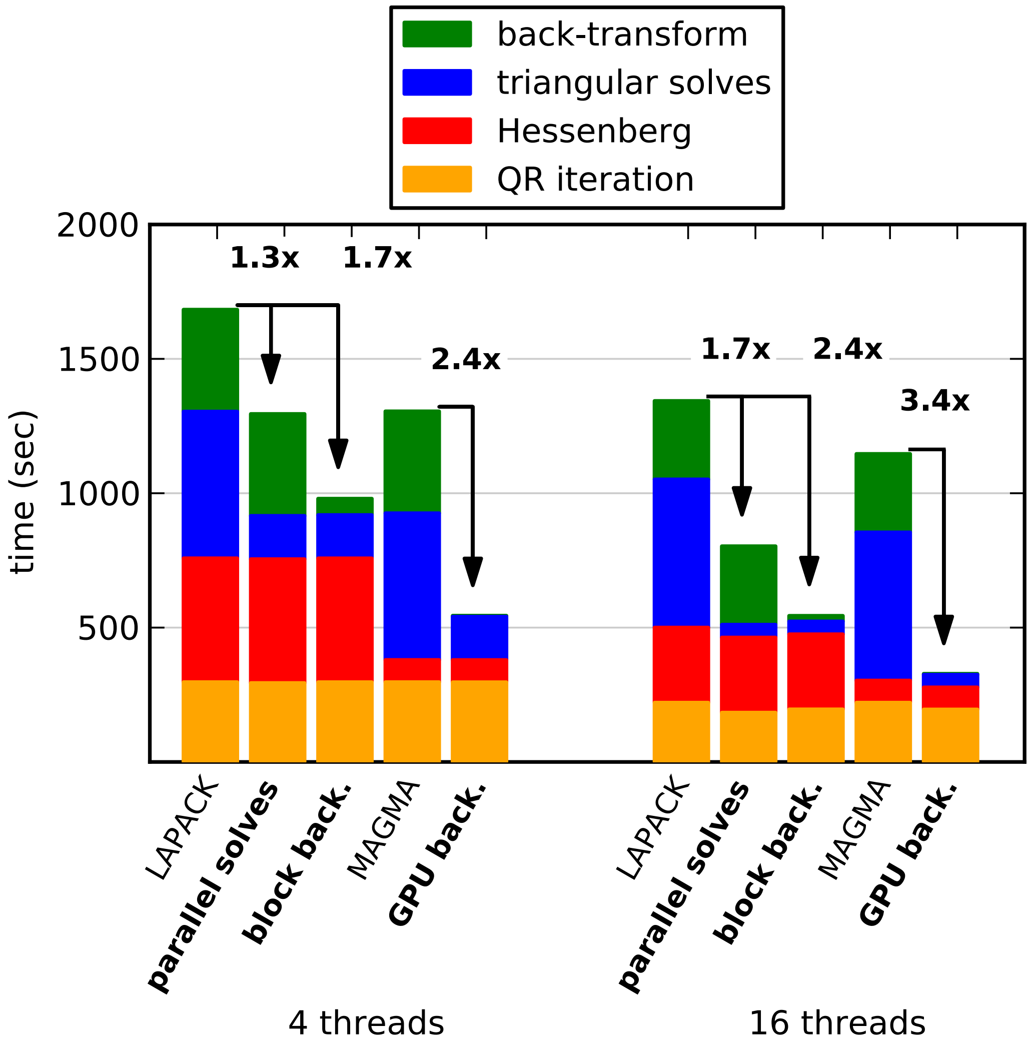
- Next **latrsd** does not depend on **gemm** results
- Perform **gemm** on GPU; overlap with **latrsd** on CPU





Performance

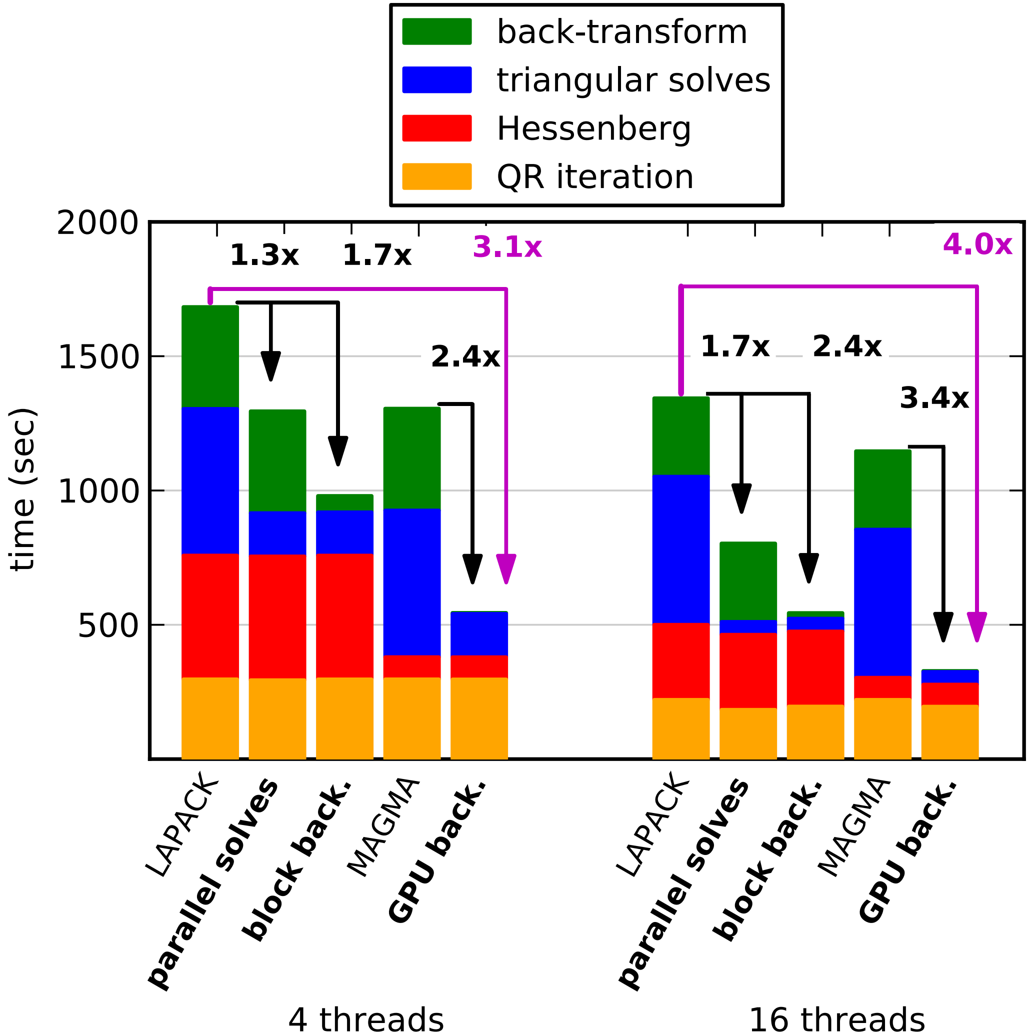
- Compare to MAGMA Hessenberg on GPU
- Back-transformation time completely hidden



n = 16000, 2x8 core Intel Sandy Bridge, NVIDIA Kepler K40 GPU

Performance

- Compare to MAGMA Hessenberg on GPU
- Back-transformation time completely hidden
- Overall improvement compared to LAPACK



n = 16000, 2x8 core Intel Sandy Bridge, NVIDIA Kepler K40 GPU

Xeon Phi acceleration

- Difficult to parallelize **latrsd** with CUDA or OpenCL
  - Especially **dlaqtrsd** with 2x2 diagonal blocks
- Intel Xeon Phi offers easier programming model
- Perform both **latrsd/laqtrsd** and **gemm** on Xeon Phi
- Ongoing work

ICL UT
INNOVATIVE
COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE