Mixed-Precision Algorithm for Finding Selected Eigenvalues and Eigenvectors of Symmetric and Hermitian Matrices

Yaohung (Mike) Tsai[1,4], Piotr Luszczek[1], Jack Dongarra[1,2,3]

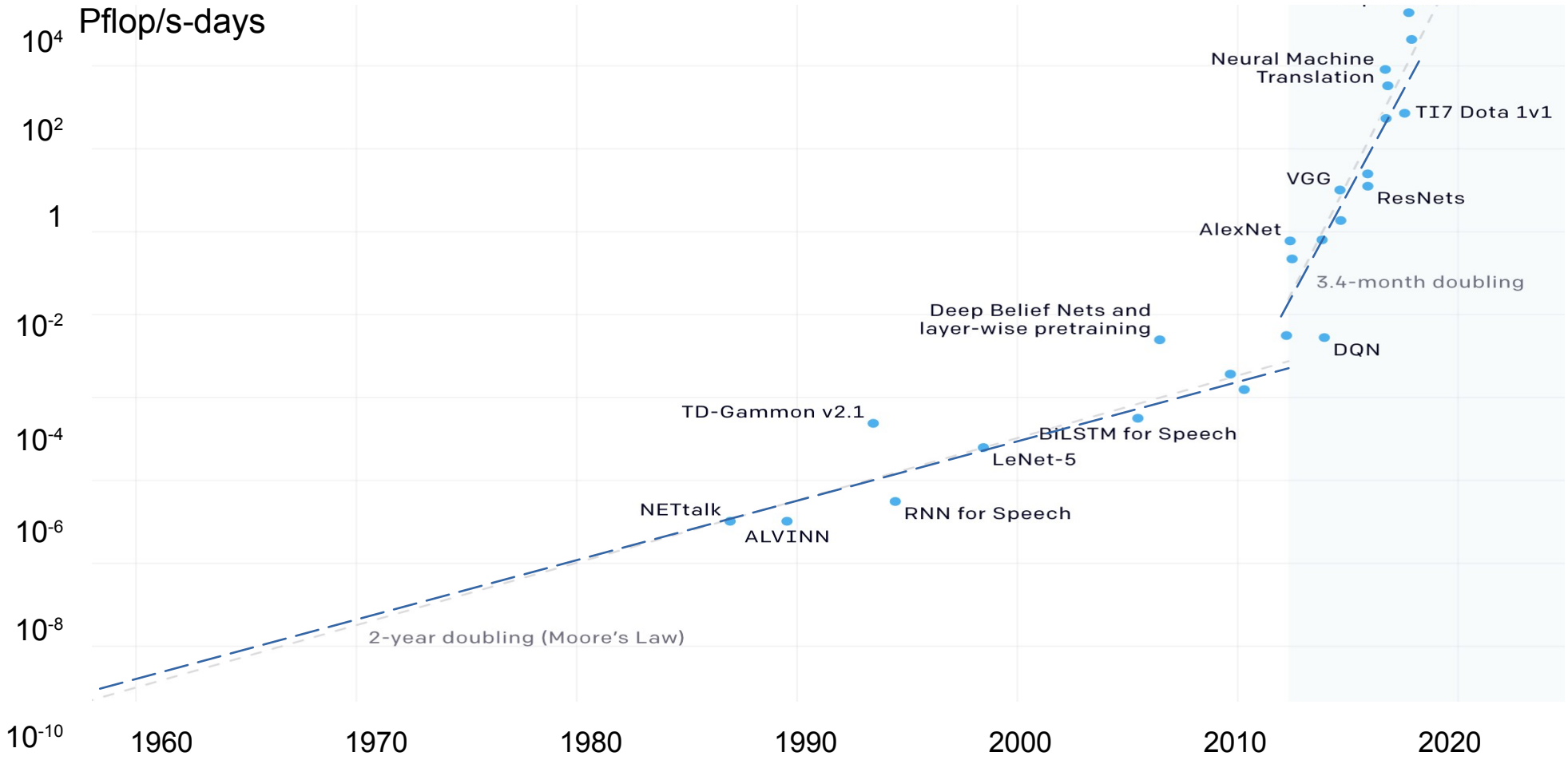
November 13, 2022

[1]University of Tennessee, [2]Oak Ridge National Laboratory, [3]University of Manchester
[4]Meta (formerly Facebook)

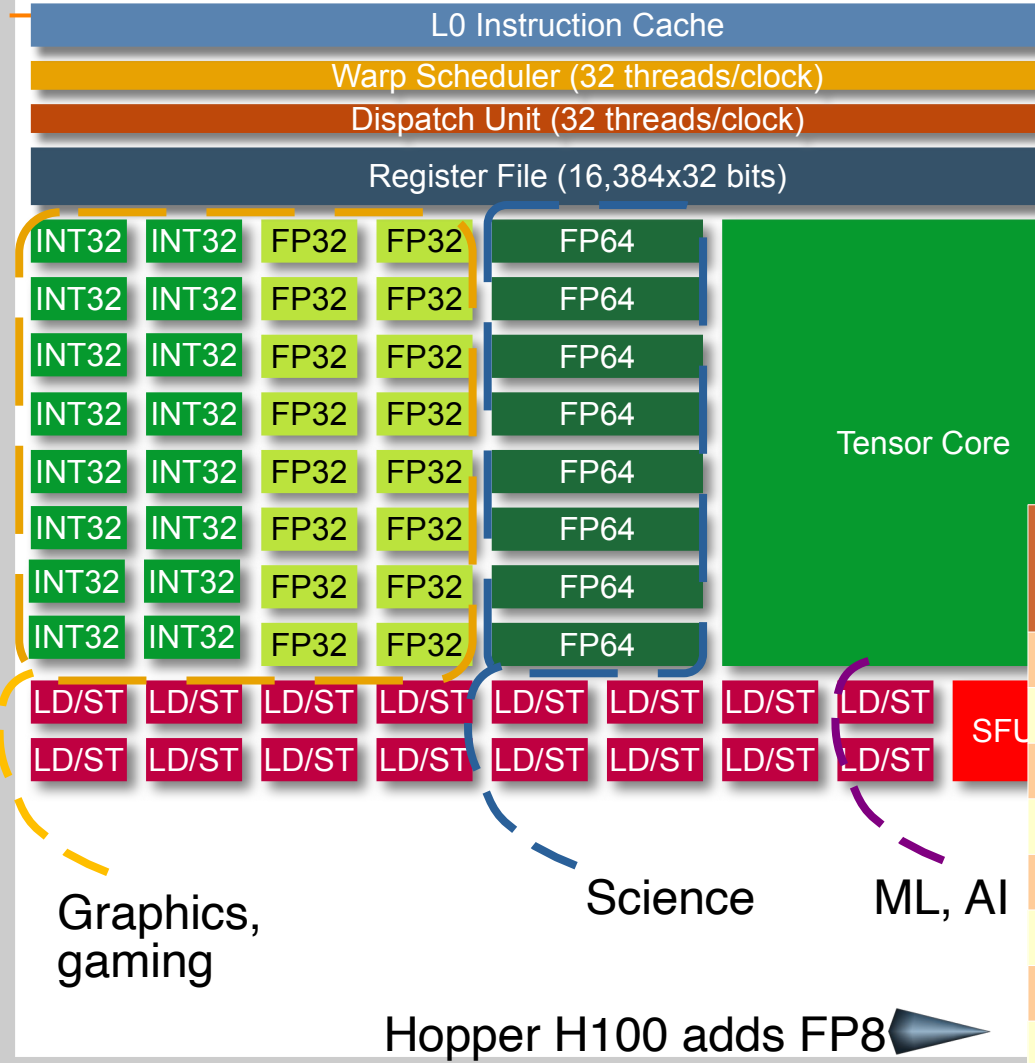
The Landscape of Mixed-Precision Hardware

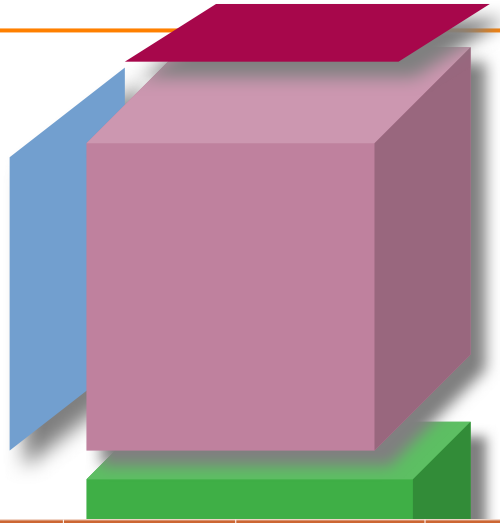
- Mixed-Precision Startups and their hardware
  - GraphCore
    - Colossus
  - Habana
    - Labs Gaudi
  - Cerebras
    - Wafer Scale Chip
  - Blaize
    - Graph Streaming Processor
  - Groq
    - Tensor Streaming Processor
  - SambaNova
    - Cardinal
  - Tenstorrent
    - Grayskull
- NVIDIA mixed-precision hardware
  - Pascal
    - FP16 units only
  - Volta
    - Tensor Cores and FP16
  - Turing
    - Tensor Cores and FP16
  - Ampere
    - Tensor Cores for FP16 and FP64
  - Hopper
    - Double-, single-, half-, and quarter-precision formats in scalar and tensor units

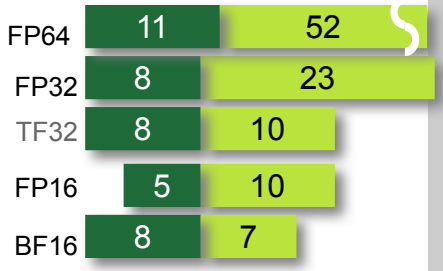
Computational Needs According to OpenAI



NVIDIA Ampere FP Hardware A100 Highlights







Precision	Width	Exponent bits	Mantissa bits	Epsilon	Max
Quadruple	128	15	112	$O(10^{-34})$	1.2×10^{4932}
Extended	80	15	64	$O(10^{-19})$	
Double	64	11	52	$O(10^{-16})$	1.8×10^{308}
Single	32	8	23	$O(10^{-7})$	3.4×10^{38}
Half*	16	5	10	$O(10^{-3})$	65504
*Only storage format is specified					
BFloat	16	8	7	$O(10^{-2})$	3.4×10^{38}
IEEE 754 2018 standard update includes 16-bit for computing					

Related Work for Eigenvalue Refinement

- Dongarra, J. J., Moler, C. B., & Wilkinson, J. H. (1983). *Improving the accuracy of computed eigenvalues and eigenvectors*. SIAM Journal on Numerical Analysis, 20(1), 23-45.
- Prikopa, K. E., & Gansterer, W. N. (2013). *On mixed precision iterative refinement for eigenvalue problems*. Procedia Computer Science, 18, 2647-2650.
- Ogita, T., & Aishima, K. (2018). *Iterative refinement for symmetric eigenvalue decomposition*. Japan Journal of Industrial and Applied Mathematics, 35(3), 1007-1035.

Algorithm 1 SICE algorithm (Dongarra, Moler, Wilkinson)

Input: Matrix $A \in \mathbb{R}^{n\times n}$. An approximate eigenvalue λ and the corresponding eigenvector x. $\mathcal{I}_{\max}$ denotes the maximum number of iterations.
Output: Refined eigenvalue λ and its eigenvector x.
function $[\lambda, x] \leftarrow$ **SICE**$(A, \lambda, x, \mathcal{I}_{\max})$
  $[Q, U] \leftarrow \text{schur}(A)$ ▹ obtain Schur decomposition $A = QUQ^\top$, $QQ^\top = I$.
  $[m, s] \leftarrow max(abs(x)); x \leftarrow x/m$ ▹ Normalizing x so that $\|x\|_\infty = s_x = 1$.
  for i in $1 : \mathcal{I}_{\max}$ **do**
    $r \leftarrow \lambda x - Ax$
    $c \leftarrow -x - a_{\lambda s}$
    $d \leftarrow Q^\top c$
    $f^\top \leftarrow Q(s,:) = e_s^\top Q$ ▹ s-th row of Q.
    $\bar{U}_\lambda \leftarrow Q_1(U - \lambda I); \bar{d} \leftarrow Q_1 d = \|d\|_2 e_1$ ▹ Givens rotations Q_1 from $Q_1 d = (P_2 P_3 \ldots P_n) d = \gamma e_1$ where $\gamma = \|d\|_2$
    $\bar{U}_\lambda \leftarrow \bar{U}_\lambda + \bar{d}(1) f^\top$
    $\bar{U}_\lambda \leftarrow Q_2 \bar{U}_\lambda$ ▹ Givens rotations Q_2 to introduce upper triangular form.
    Solve the triangular system $\bar{U}_\lambda z = Q_2 Q_1 Q^\top r$
    $y \leftarrow Qy$
    $\lambda \leftarrow \lambda + y(s)$ ▹ Update eigenvalue.
    $y(s) \leftarrow 0$ ▹ Set $y(s)$ to 0.
    $x \leftarrow x + y$ ▹ Update eigenvector.
    if desired accuracy is reached **then**
      break
    end if
  end for
end function

```
[Q, T]=schur(single(A)); Q = V; T = D;
result=zeros(1,niter+1); normA=norm(A,inf);
for i=1:niter
    r=lambda*x-A*x; %residual vector
    error(i)=norm(r,inf)/normA/norm(x,inf);
    disp(['Norm of residual vector before iteration ', num2str(i), ' : ',
num2str(norm(r))]);
    a_lambda_s = A(:,s); % eq (3.14)
    a_lambda_s(s) = a_lambda_s(s) - lambda;
    c=-x-a_lambda_s;
    d=Q'*c; % eq (3.15)
    fT=Q(s,:);
%    % Explicitly form matrix B: B=A-lambda*eye(n); B(:,s)=B(:,s)+c; y=B\r;
% .  % Explicitly form matrix T_lambda-d*fT    y=(T-lambda*eye(n)+d*fT)\(Q'*r); y=Q*y;
    rhs = Q'*r;
    T_lambda=T-lambda*eye(n);
    for k=n:-1:2    % Apply Q_1
        [P, z] = planerot(d(k-1:k)); % givens rotation
        %T_lambda(k-1:k,:) = P*T_lambda(k-1:k,:);
        T_lambda(k-1:k,k-1:n) = P*T_lambda(k-1:k,k-1:n);
        d(k-1:k) = z;
        rhs(k-1:k) = P * rhs(k-1:k);
    end
    T_lambda(1,:) = T_lambda(1,:) + d(1)*fT;
    % Apply Q_2
    for k=1:n-1
        [P, z] = planerot(T_lambda(k:k+1,k));
        %T_lambda(k:k+1,:) = P*T_lambda(k:k+1,:);
        T_lambda(k:k+1,k:n) = P*T_lambda(k:k+1,k:n);
        rhs(k:k+1) = P*rhs(k:k+1);
        T_lambda(k:k+1,k)=z;
        %T_lambda(k+1,k)=0; % forcing real zero
    end
    y=T_lambda\rhs; % T_lambda is upper triangular
    %y=(T-(lambda)*eye(n)+d*fT)\rhs;
    %y=gmres(T_lambda,rhs,5,1e-12,15,T_lambda);
    y=Q*y;
    new_x=x+y; % new eigenvector
    new_x(s)=x(s); % restore x(s) since y_s = 0
    disp(['Computed eigenvalue offset at  iteration ', num2str(i), ' : ',
num2str(y(s))]);
    lambda=lambda+y(s); % new eigenvalue
    % convergence check: break if 2*(previous correction) < current correction
    %if(i>=2 && abs(y(s))>2*abs(last_u))
    last_u = y(s);
    x=new_x;
end
```

Mixed-Precision Eigensolver Goals

$$Ax = \lambda\ x \ \text{ where } \ A = A^T \vee A = A^H$$

- Looking into iterative refinement for real symmetric and complex Hermitian eigenvalue problems.
- Apply mixed-precision to improve the performance.
- Incorporate with latest two-stages eigensolvers.
- Targeting the case that only part of the eigenpairs are required.

Algorithm 2 SICE-SM algorithm: SICE algorithm with Sherman–Morrison formula

Input: Matrix $A = A^\top \in \mathbb{R}^{n\times n}$. An approximate eigenvalue λ and the corresponding eigenvector x. $I_{\max}$ denotes the maximum number of iterations.
Output: Refined eigenvalue λ and eigenvector x.
function $[\lambda, x] \leftarrow$ **SICE_SM**$(A, \lambda, x, I_{\max})$
    $[Q, T] \leftarrow \text{tridiag}(A)$ ▹ Tridiagonalization $A = QTQ^\top$, $QQ^\top = I$.
    $[m, s] \leftarrow max(abs(x)); x \leftarrow x/m$ ▹ Normalization of x so that $\|x\|_\infty = s_x = 1$.
    for i in $1 : I_{\max}$ **do**
        $r \leftarrow \lambda x - Ax$
        $c \leftarrow -x - a_{\lambda s}$
        $d \leftarrow Q^\top c$
        $f^\top \leftarrow Q(s,:) = e_s^\top Q$ ▹ s-th row of Q.
        $rhs \leftarrow Q^\top r$
        $u \leftarrow (T - \lambda I)^{-1} d$
        $v \leftarrow (T - \lambda I)^{-1} rhs$
        $y \leftarrow v - \frac{f^\top v}{1 + f^\top u} u$ ▹ Sherman–Morrison formula
        $y \leftarrow Qy$
        $\lambda \leftarrow \lambda + y(s)$ ▹ Update eigenvalue.
        if $i \neq 1$ **then**
            $y(s) \leftarrow 0$ ▹ Set $y(s)$ to 0.
            $x \leftarrow x + y$ ▹ Update eigenvector.
        end if
        if desired accuracy reached **then**
            break
        end if
    end for
end function

Background — SICE algorithm

- General matrix A with approximated eigenvalue λ and eigenvector x.
- Solving a rank one updated system to obtain the correction to both eigenvalue y_s and eigenvector y:

$$\left(A-\lambda I+c\, e_s^T\right) y=\lambda x-Ax$$

- Solving with Schur decomposition $A=QUQ^H$ and two series of Givens rotations.

New SICE-SM Algorithm

- Looking at symmetric matrices first: $A=A^T$
- The rank one updated system under tridiagonalization $A=QTQ^T$

$$Q\left(T-\lambda I+Q^T ce_s^T Q\right)Q^T y=\lambda x-Ax$$

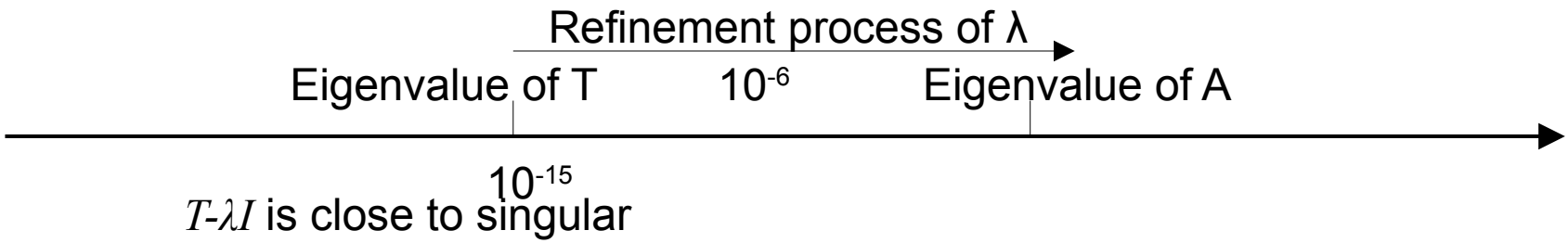
- Using Sherman-Morrison formula

$$\left(A-uv^T\right)^{-1}=A^{-1}-\frac{A^{-1}uv^T A^{-1}}{1+v^T A^{-1}u}$$

- How close is $T\text{-}\lambda I$ to being singular? In which precision?

Mixed Precision Eigensolver Process

- Consider single and double precision.
  - Tridiagonalization $A=QTQ^T$ in single precision.
- Find the eigenvalues Λ and eigenvectors *V* of the tridiagonal system *T* in double precision.
- Back transform to obtain the eigenvectors of *A: V* ← *QV*
- Iterative refinement with SICE-SM



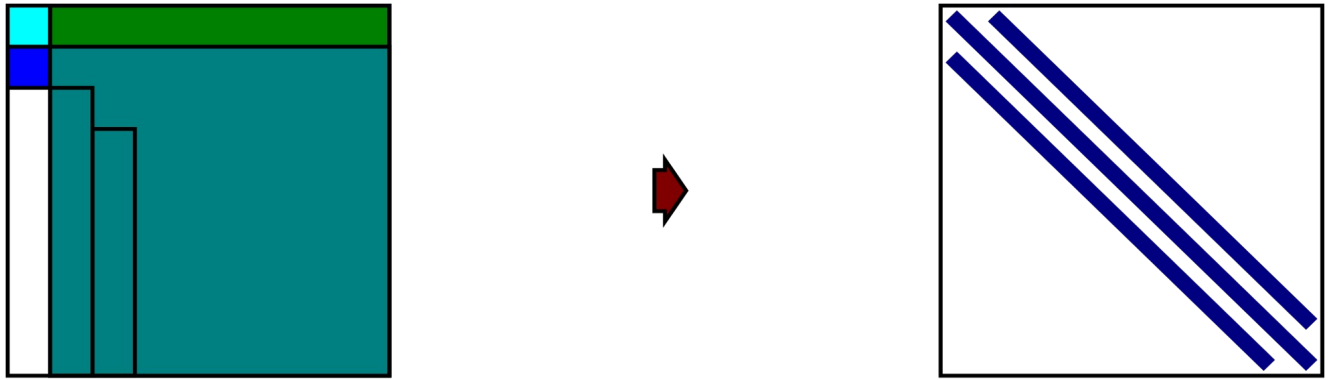
Algorithm 3 Blocked SICE-SM algorithm

Input: $A = A^T \in \mathbb{R}^{n\times n}$, initial eigenvectors $X = [x_1|x_2|...|x_\ell] \in \mathbb{R}^{n\times \ell}$ and the corresponding initial eigenvalues $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_\ell)^T \in \mathbb{R}^\ell$. $I_{\max}$ denotes the maximum number of iterations.
Output: Refined eigenvectors X and refined eigenvalues Λ.
function $[X, \Lambda] \leftarrow$ **SICE_SM_BLK**$(A, X, \Lambda, I_{\max})$
  $[Q, T] \leftarrow \text{tridiag}(A)$ ▹ Tridiagonalization $A = QTQ^\top$, $QQ^\top = I$.
  for i in $1 : I_{\max}$ **do**
    $s \leftarrow i$
    $R \leftarrow X \times \text{diag_matrix}(\Lambda) - A \times X$ ▹ Residual vectors need higher precision.
    for j in $1 : \ell$ **do**
      $c_j \leftarrow -x_j - A(:, s)$
    end for
    Compose matrix $C = [c_1|c_2|...|c_\ell]$ from column vectors c_j
    $C(s, :) \leftarrow C(s, :) + A^T$
    $D = [d_1|d_2|...|d_\ell] \leftarrow Q^T \times C$ ▹ Can be in lower precision.
    $RHS = [rhs_1|rhs_2|...|rhs_\ell] \leftarrow Q^T \times R$ ▹ Can be in lower precision.
    $f \leftarrow Q(s, :)$ ▹ s-th row of Q.
    for j in $1 : \ell$ **do**
      $u_i \leftarrow (T - \lambda I)^{-1} d_i$
      $v_i \leftarrow (T - \lambda I)^{-1} rhs_i$
      $y_i \leftarrow v_i - \frac{f^\top v_i}{1 + f^\top u_i} u_i$ ▹ Sherman–Morrison
    end for
    Compose matrix $Y = [y_1|y_2|...|y_\ell]$ from correction vectors y_j
    $Y \leftarrow Q \times Y$
    $\Lambda \leftarrow \Lambda + Y(s, :)^T$ ▹ Update eigenvalues.
    if $i \neq 1$ **then**
      $Y(s, :) \leftarrow 0$ ▹ Set $y_i(s)$ to 0.
      $X \leftarrow X + Y$ ▹ Update eigenvectors.
      Normalize eigenvectors x_i in X.
    end if
    if desired accuracy reached **then**
      break
    end if
  end for
  $X \leftarrow X + \frac{1}{2} X (I - X^\top X)$ ▹ Orthogonalization.
end function

Background: Two-Stage Tridiagonalization

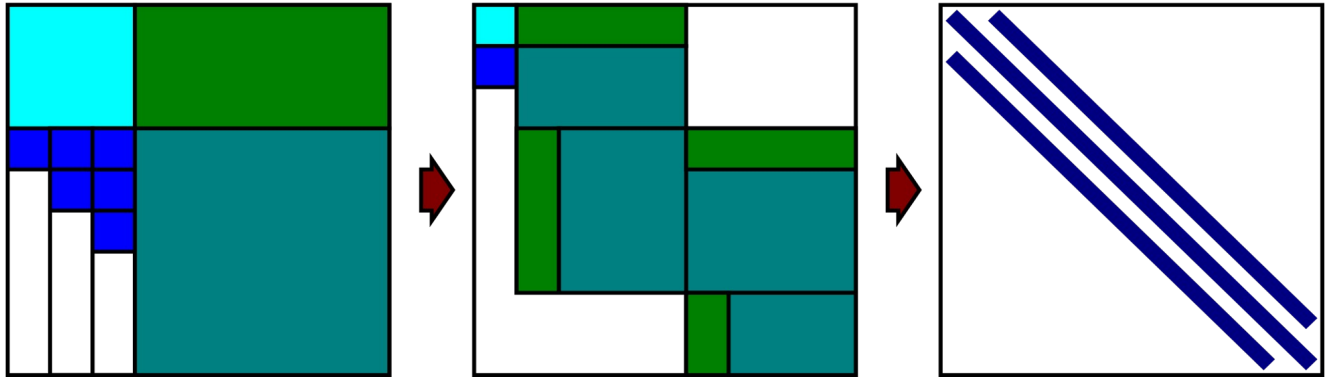
One stage (classic)

- Householder transformations
- Bound by memory speed



Two stage (post SBR)

- First stage: symmetric to band
- Second stage: band to tridiagonal



QR an LQ

Bulge chasing

Blocked SICE-SM Algorithm

- Refine multiple eigenvalues and eigenvectors simultaneously
- Performance of n × n matrix times n × m aggregated vectors on NVIDIA V100

Matrix size	Number of eigenvectors	Time (ms)	Performance (Tflop/s)
20000	1	3.76	0.212
20000	8	3.79	1.688
20000	32	6.48	3.949
20000	128	13.57	7.544

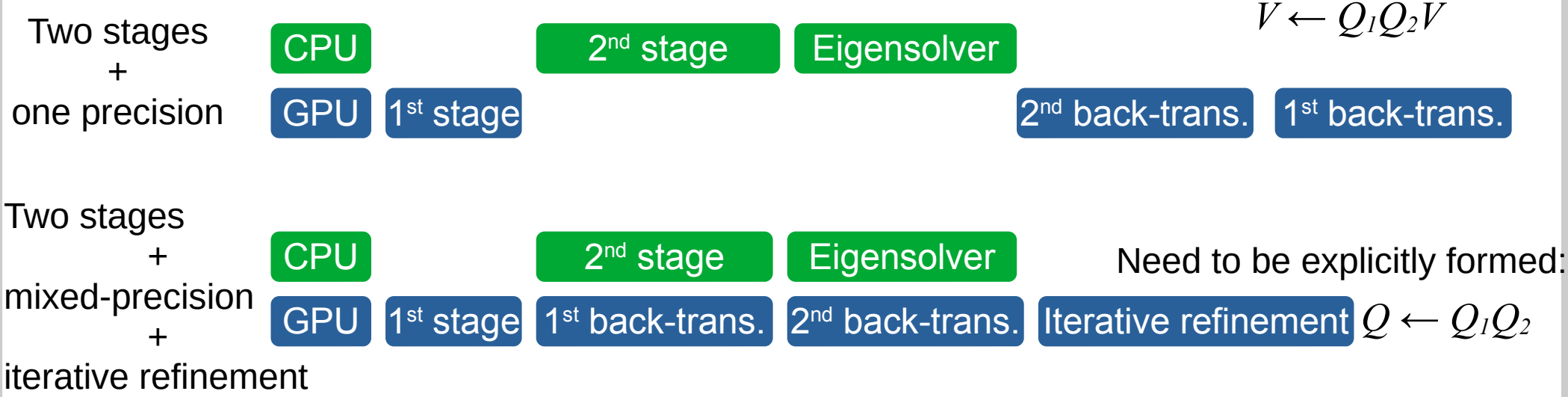
- Reorthogonalize with one Newton–Schulz iteration at the end:

$$X' = X + \tfrac{1}{2} X(I - X^T X)$$

Implementation Details

- MAGMA numerical library
  - Targeting heterogeneous systems (CPU+GPU)
- Reordering the back transformations and overlapping with CPU work

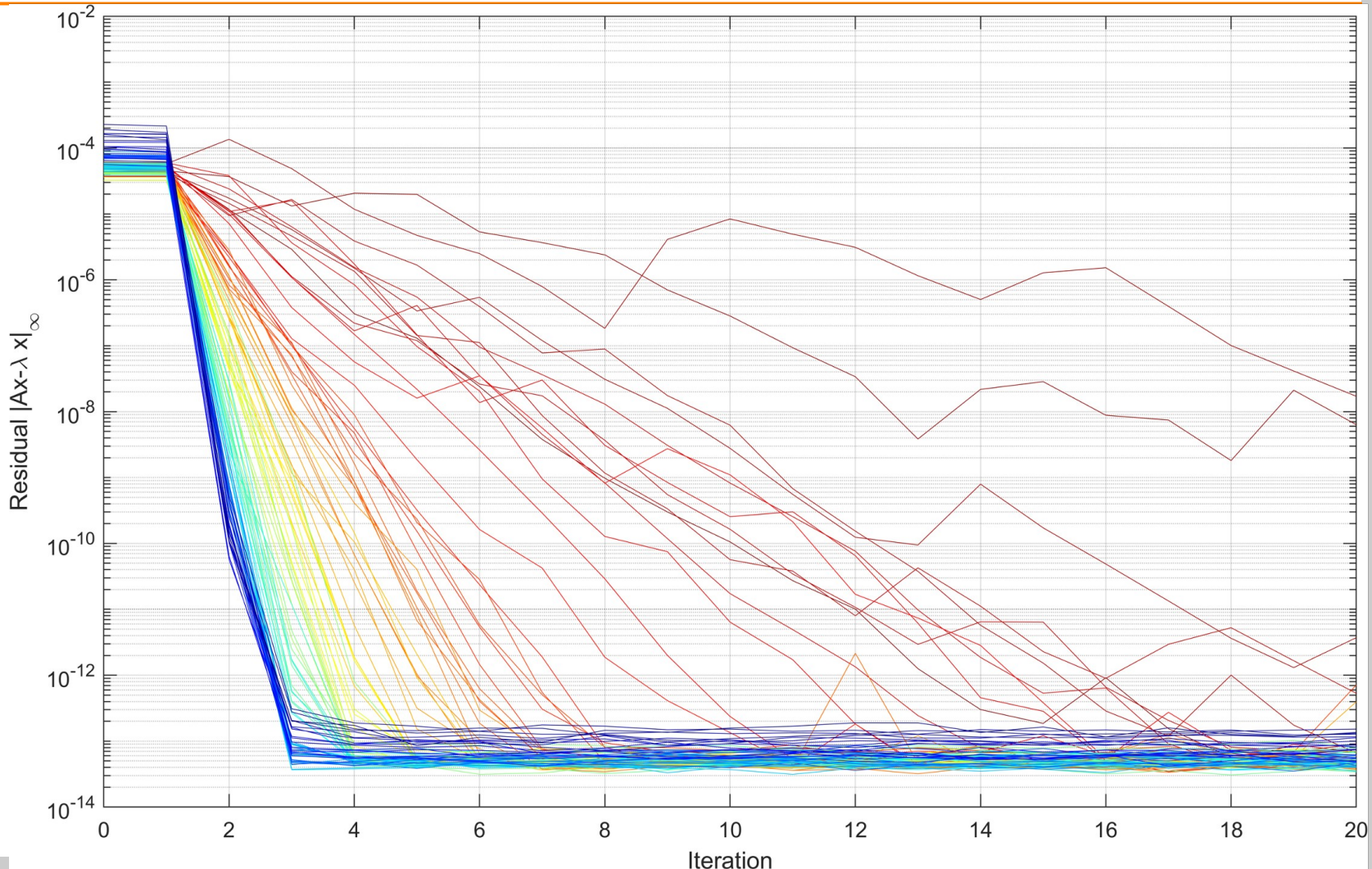
(not to scale)



- Batched tridiagonal solver on GPU

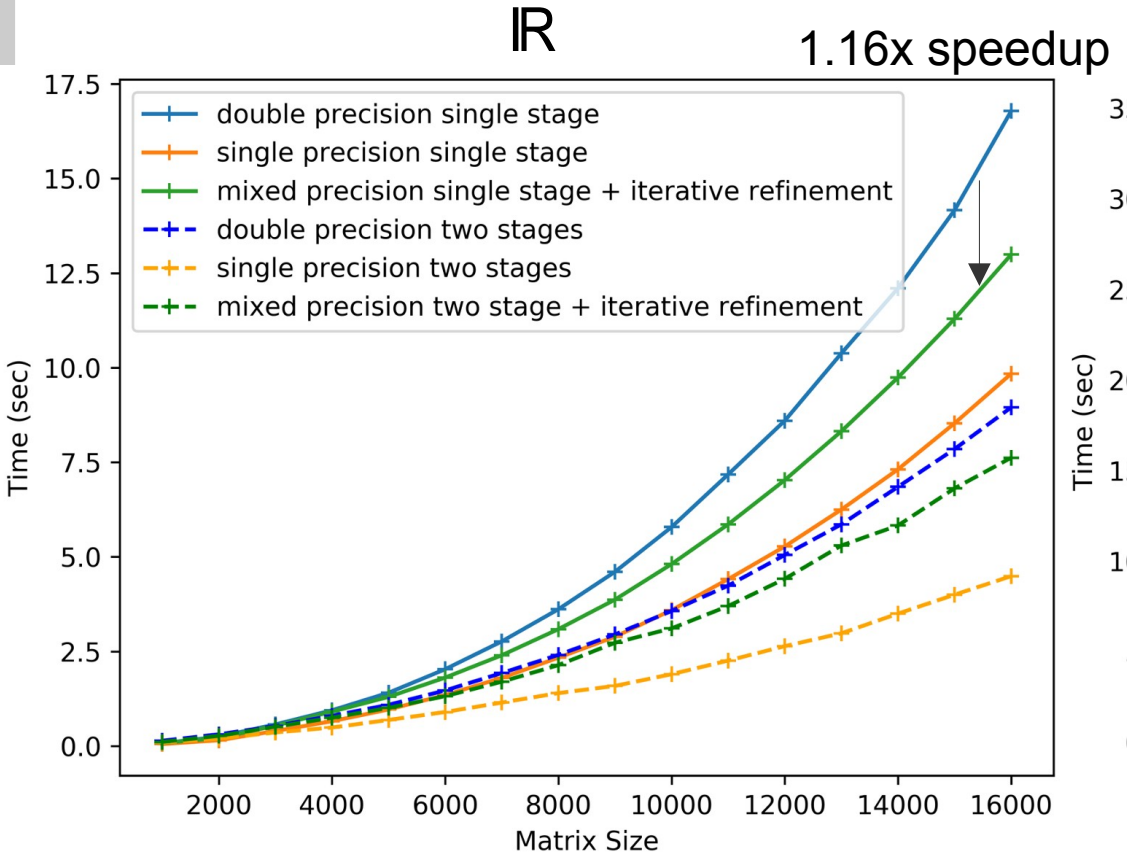
Numerical Convergence Results

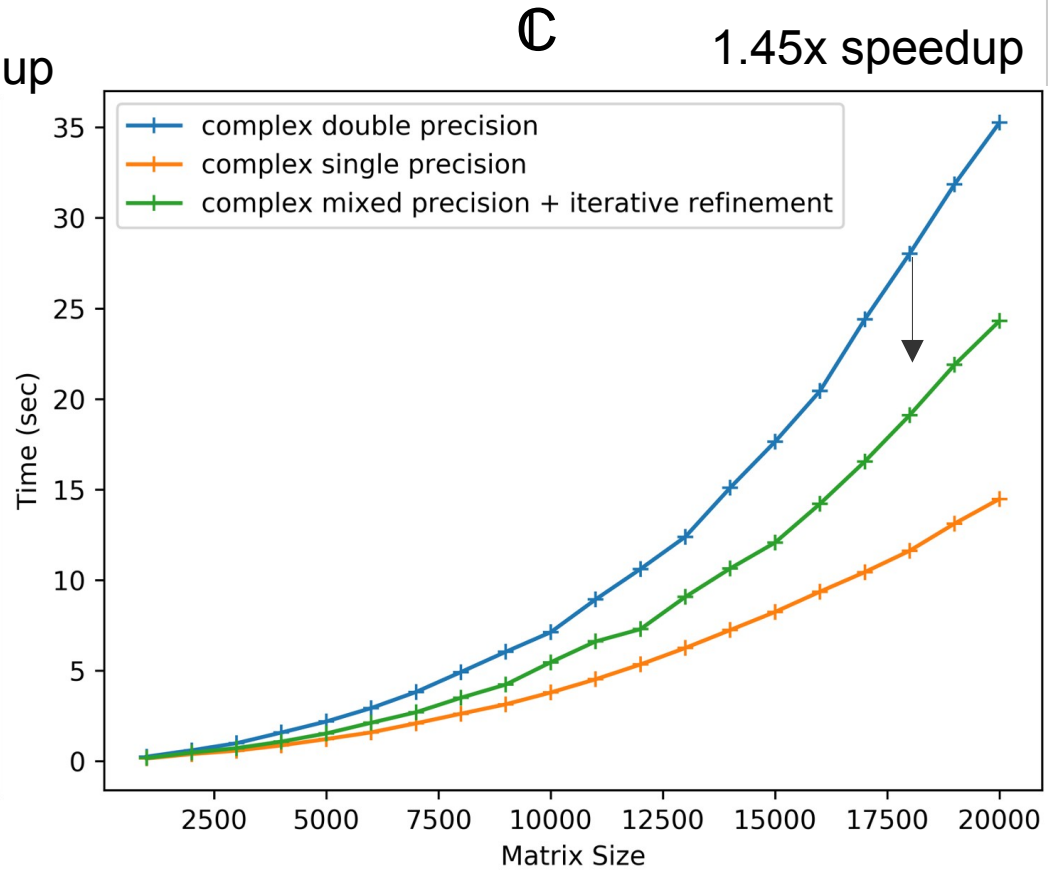
- size(A)=100
- cond(A)=10^7
- geometrically distributed eigenvalues from 1 to 10^{-7}



Performance Results on NVIDIA Volta V100

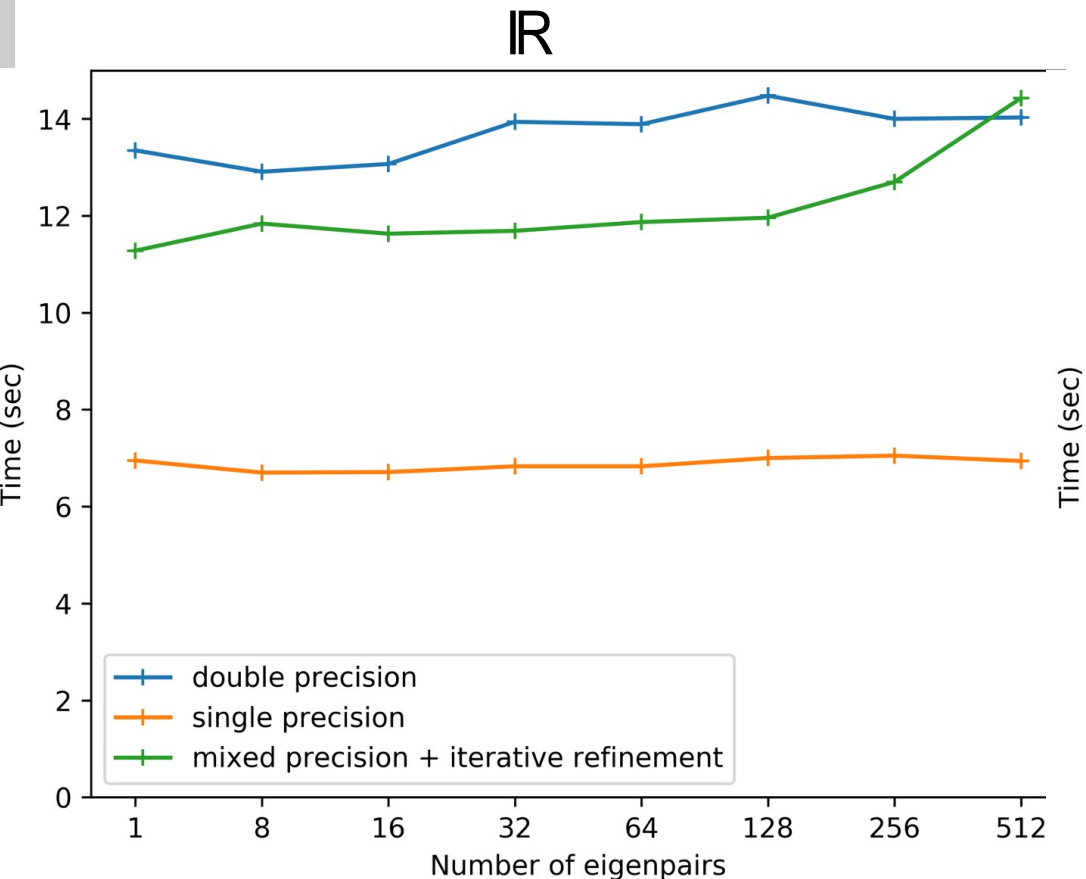
CPU: Intel Xeon E5-2650 v3 (Haswell 10C20T), GPU: NVIDIA Tesla Volta V100.
Intel compilers and MKL, CUDA 11, MAGMA 2.5.4.
Requesting largest 32 eigenpairs.

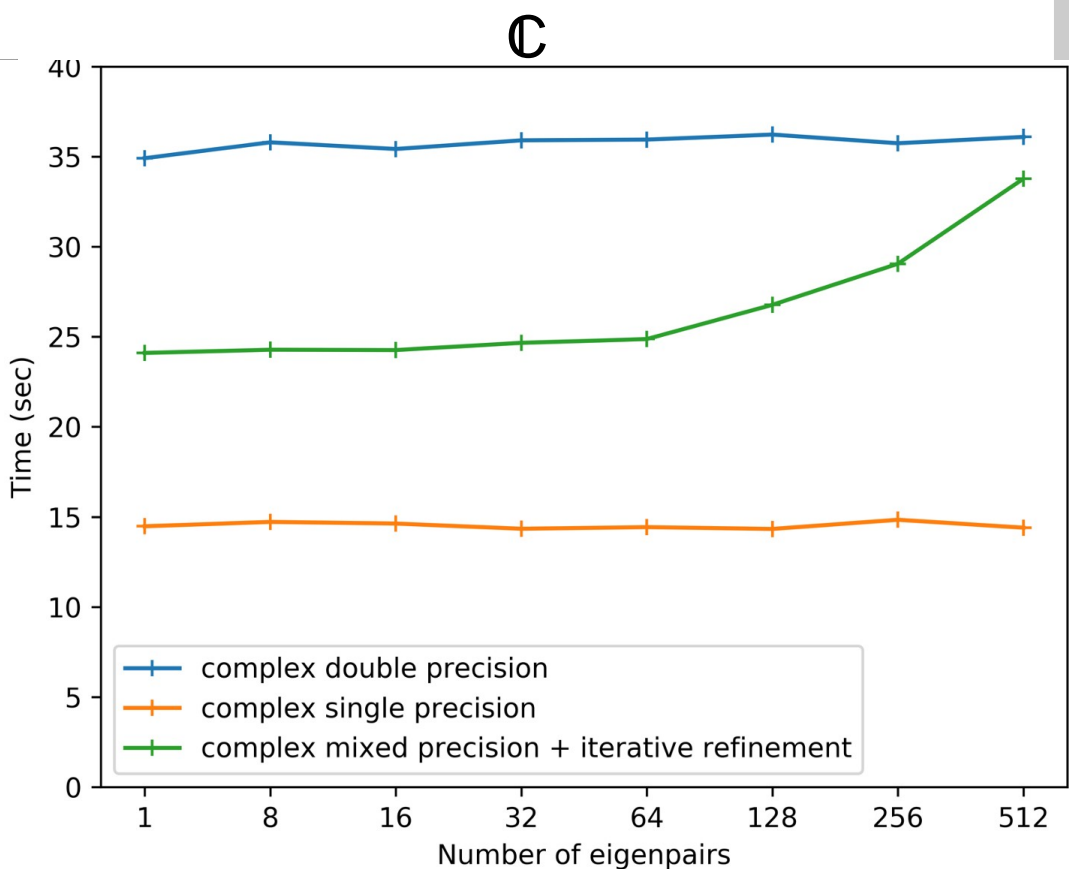




Time for Refining Different Number of Eigenpairs

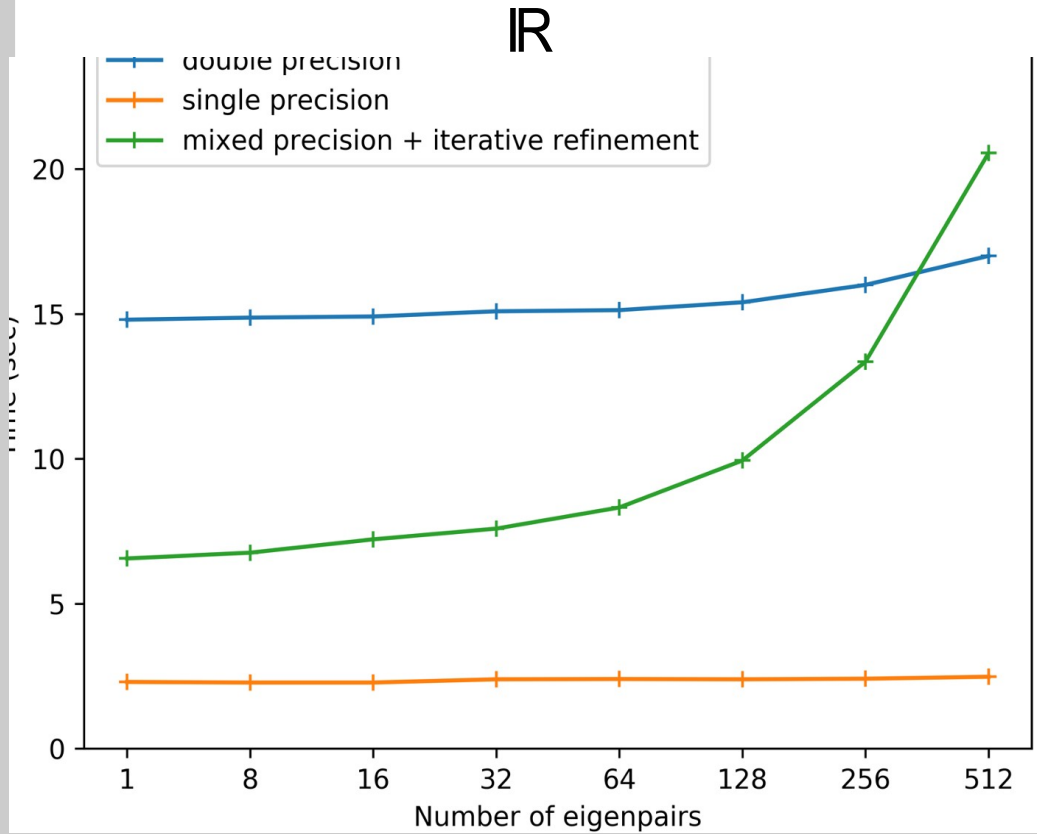
Matrix size N=20000, GPU: NVIDIA Tesla Volta V100

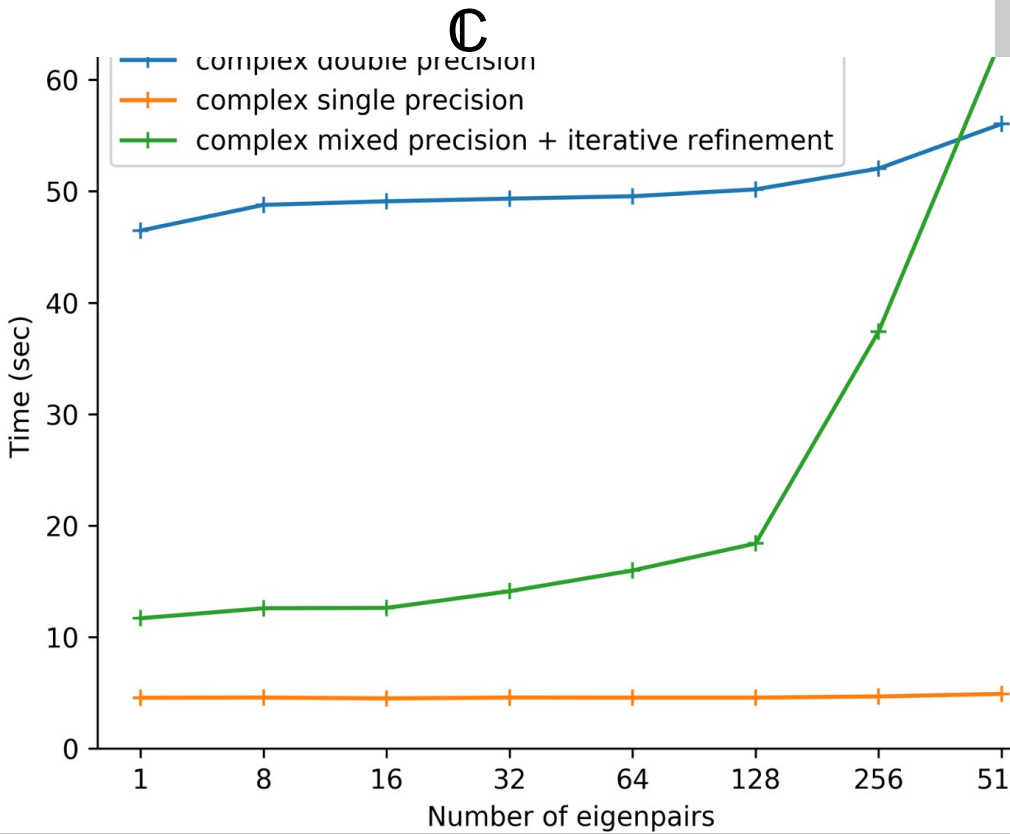




Time for Refining Different Number of Eigenpairs

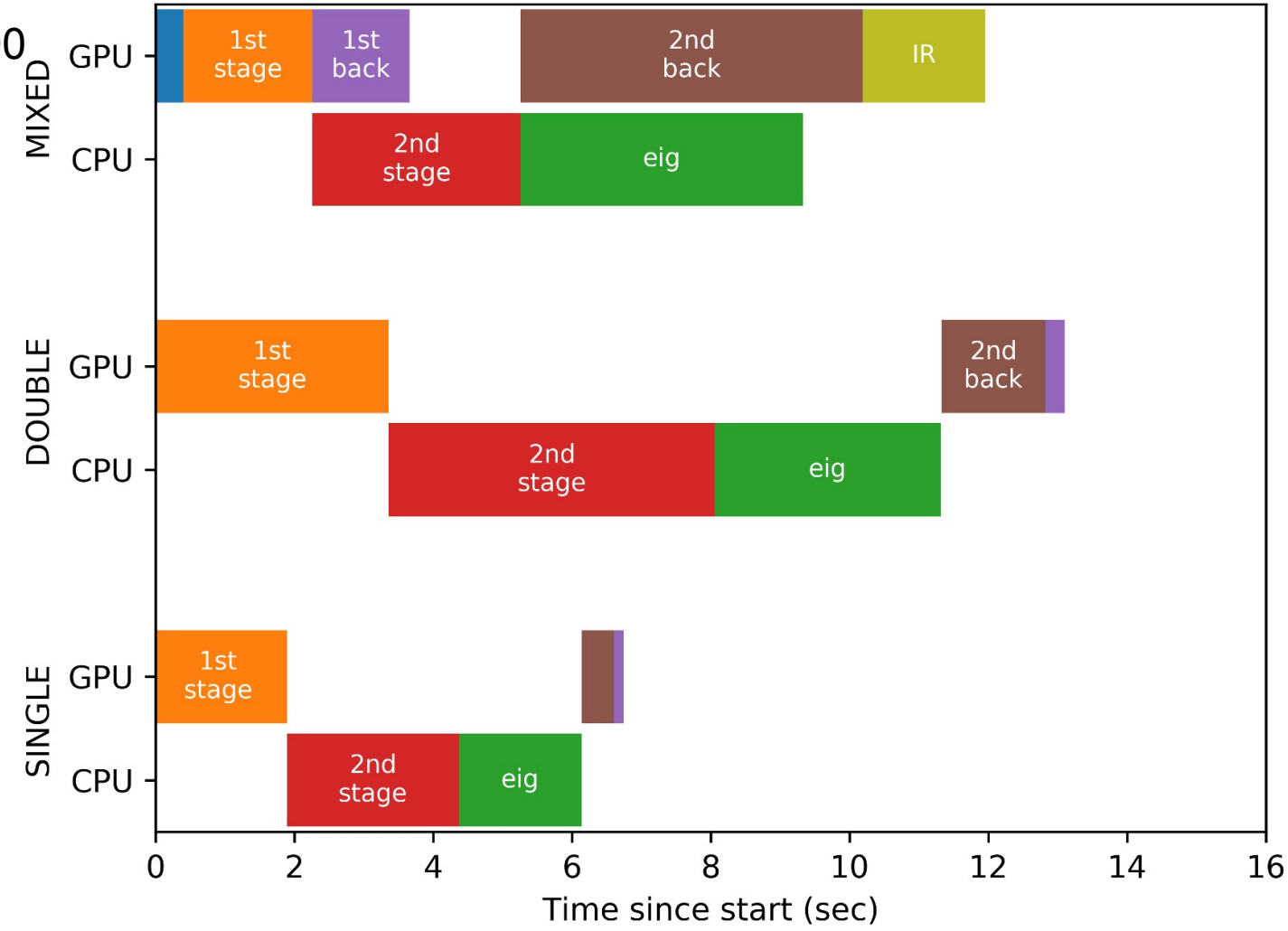
Matrix size N=20000, GPU: NVIDIA Tesla Pascal GTX1060





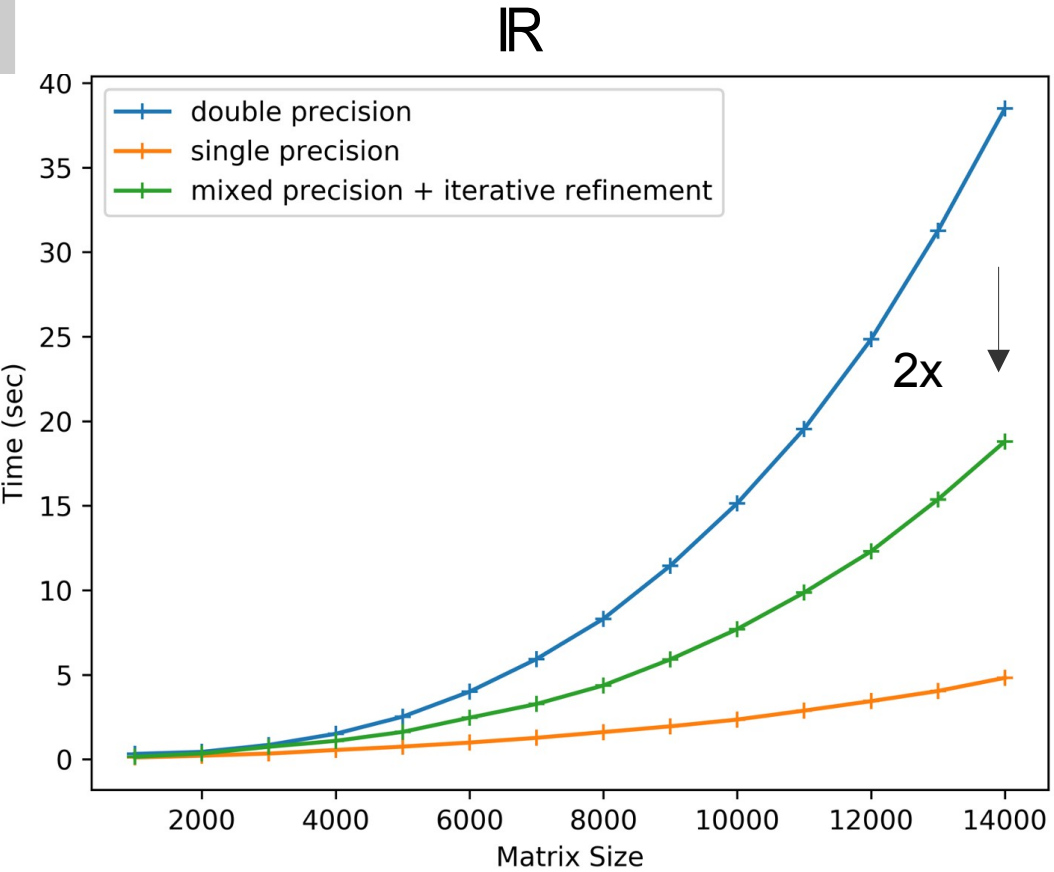
Runtime Trace

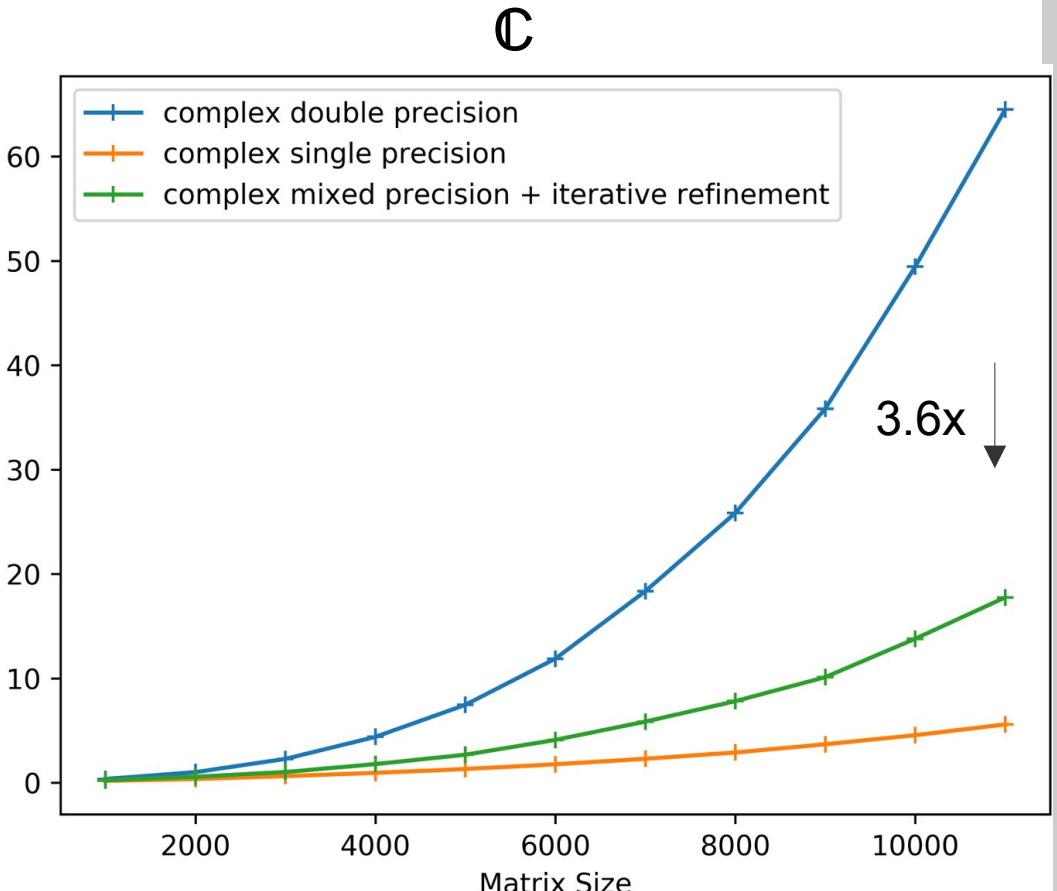
Real symmetric N=20000
32 eigenpairs



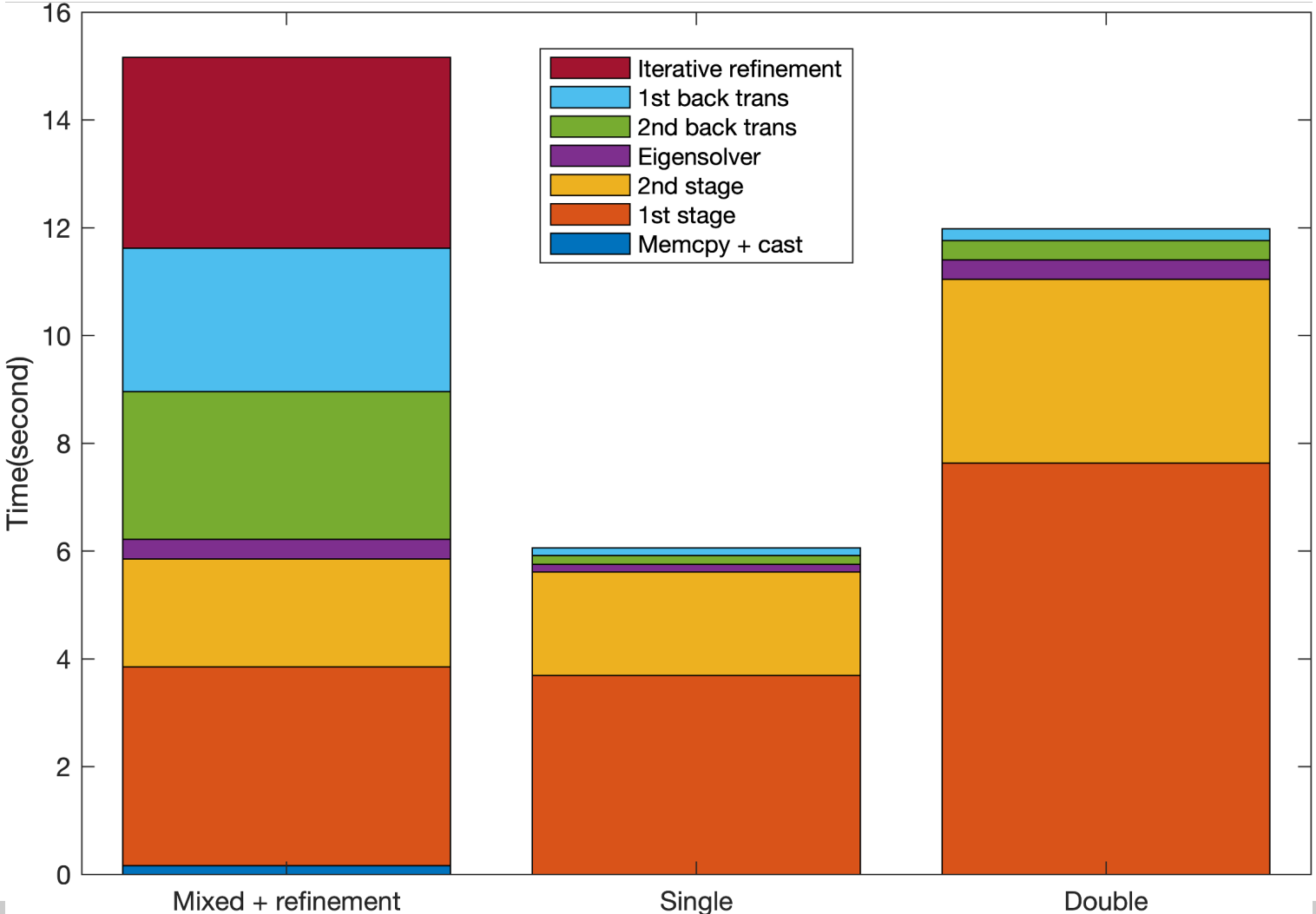
Performance Results on NVIDIA GTX1060

GPU: NVIDIA Pascal GTX1060
Peak single 4.375 TFLOP/s, double 136.7 GFLOP/s (1:32).
Requesting largest 32 eigenpairs.

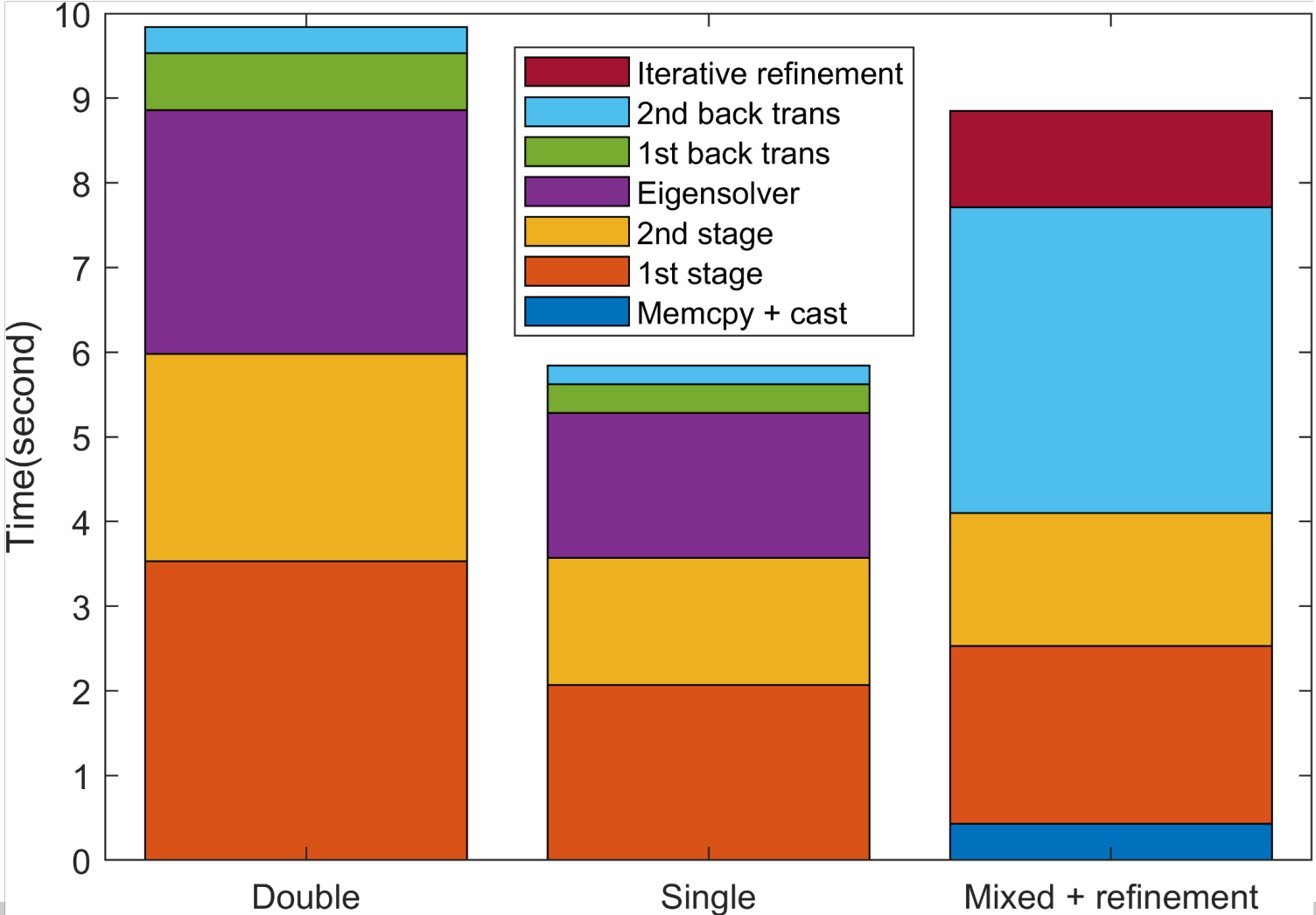




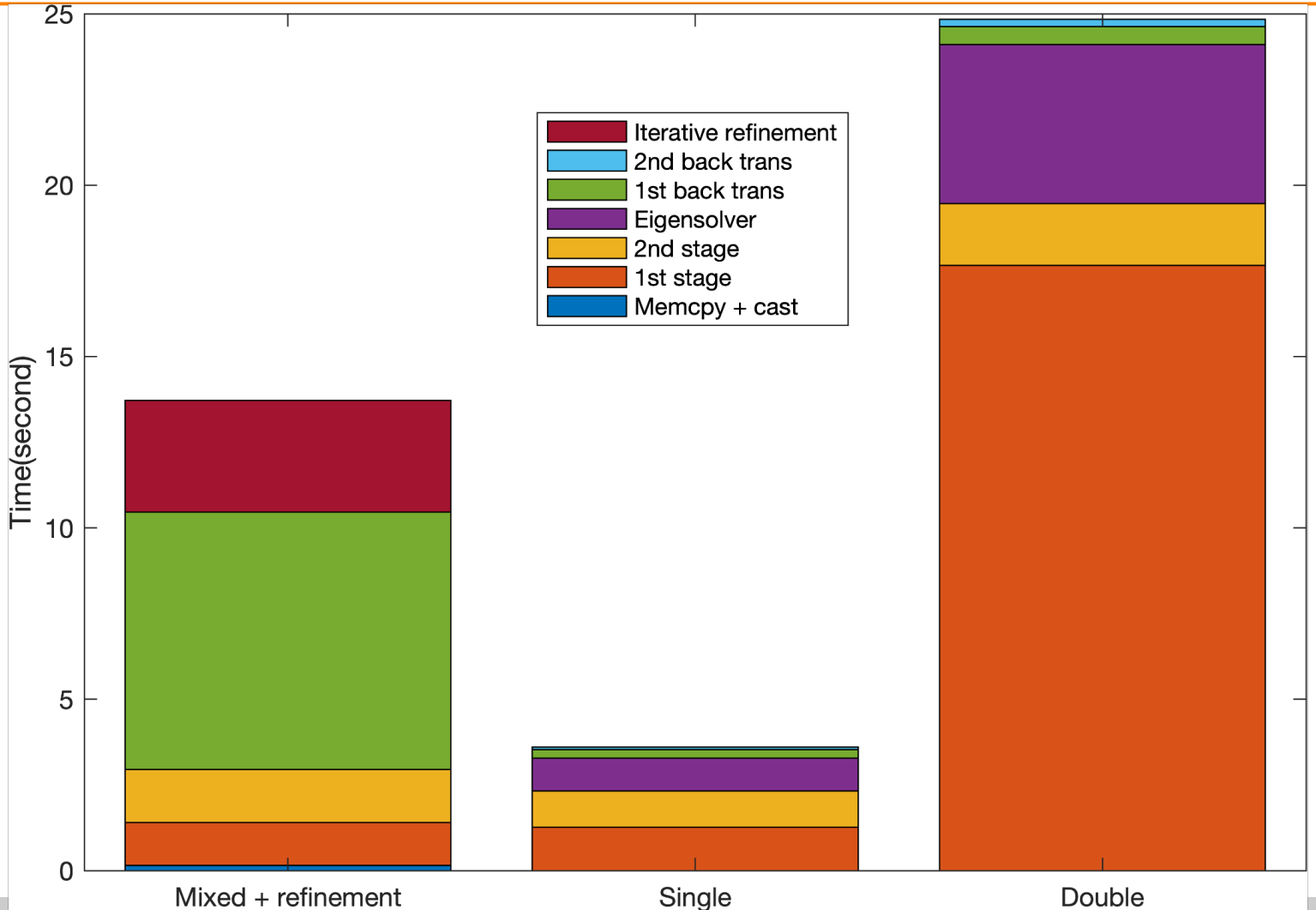
Time Breakdown on Multicore CPU



Time Breakdown on NVIDIA Volta V100



Time Breakdown on NVIDIA Pascal GTX1060



Mixed-Precision Eigensolver Summary

- Methods' summary
  - Blocked version of SICE algorithm that uses Sherman-Morrison formula
  - There are performance improvements over the state-of-the-art two-stage algorithms
- Possible future extensions
  - Convergence analysis
    - Dealing with near-singularity of $T\text{-}\lambda I$
    - $Q^T A Q$ isn't tri-diagonal while $Q^T_{32} A Q_{32}$ is
      - Also: orthogonality of Q depends on precision
  - Dealing with clustered eigenvalues
  - Implement more efficiently the second stage that forms Q
  - Use more of the custom low-precision formats
  - Implementation for distributed memory systems

ICL @ SC22

UNIVERSITY OF TENNESSEE **BOOTH #613**

@ICL_UTK ON TWITTER

FOLLOW US AT **https://icl.utk.edu/sc22**