Large Non-Hermitian Matrix Generation with Given Spectra

Xinzhe WU and Serge G. Petiton

Maison de la Simulation, CNRS, F-91191, Gif-sur-Yvette, France

CRIStAL, UMR CNRS 9189, Université de Lille, France

xinzhe.wu@ed.univ-lille1.fr, serge.petiton@univ-lille1.fr

Introduction

Eigenvalues and Eigenvalue problem:

For a square matrix A, if there is a vector $u \in \mathbb{C}^n$ such that

$$Au = \lambda u$$

for some scalar λ, then λ is called the eigenvalue of A with corresponding (right) eigenvector u.

1. numerical simulation
   - ⊖ the Schrödinger equation, molecular simulation, geology, and preconditioners for solving linear systems, e.g. UCGLE, etc.
2. machine learning and pattern recognition
   - ⊖ principal component analysis (PCA), Fisher discriminant analysis (FDA), clustering, etc.

Requirement of large-scale matrix generator

The backgroud:

- ⊖ the increase of the eigenvalue problems, and the demand to adjust the numerical algorithms to the coming exascale platforms.

The test matrix suite should has the properties:

- ⊖ the spectra should be known and can be controlled, sparse, non-Hermitian and non-trivial and have very high dimension, both the matrix size and the nnz number.

In this poster, we present a Scalable Matrix Generator with Given Spetra (SMG2S).

SMG2S Algorithm

Mathematical notations:

For all matrices $A \in \mathbb{N}^{n\times n}$, $M \in \mathbb{C}^{n\times n}$, $n \in \mathbb{N}$, a linear operator $\widetilde{A_A}$ of matrix M determined by matrix A can be set up as Formule (1):

$$\begin{cases} \widetilde{A_A} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}, \\ M \to AM - MA. \end{cases} \tag{1}$$

$$(\widetilde{A_A})^k(M_0) = \sum_{m=0}^{k} (-1)^m C_k^m A^{k-m} M_0 A^m. \tag{2}$$

$$M_{i+1} = M_i + \frac{1}{i!}(\widetilde{A_A})^i(M_0), i \in (0, +\infty). \tag{3}$$

In order to make $(\widetilde{A_A})^i$ tends to $\mathbf{0}$ in limited steps, it is necessary that $A = B^{-1}PB$, then we set the matrix P to be nilpotent, and the matrix B to be the identity matrix $I \in \mathbb{N}^{n\times n}$ for simplification.

Algorithm (Hervé Gachlier et al.):

Algorithm 1 Matrix Generation Method

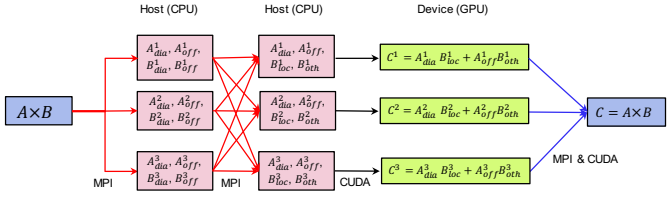
Require: $Spec_{in} \in \mathbb{C}^n$, h, d

Ensure: $M_t \in \mathbb{C}^{n\times n}$

1: Insert random elements in h lower diagonals of $M_o \in \mathbb{N}^{n\times n}$
2: Insert $Spec_{in}$ on the diagonal of M_0 and $M_0 = (2d-2)!M_0$
3: Randomly insert 1 and 0 on sub-diagonal of $A \in \mathbb{N}^{n\times n}$ with the maximum continuous length of 1 to be d
4: **for** $i = 0, \cdots, 2(d-2)-1$ **do**
5: $\quad M_{i+1} = M_i + (\prod_{k=i+1}^{2d-2} k)(\widetilde{A_A})^i(M_0)$
6: **end for**
7: $M_t = \frac{1}{(2d-2)!} M_{2d-2}$

CPU and GPU implementation

We implement SMG2S on homogenous and heterogeneous machines. The former is implemented based on MPI and PETSc, the latter is based on MPI, CUDA, and PETSc. The kernel of implementation is the SpGEMM.

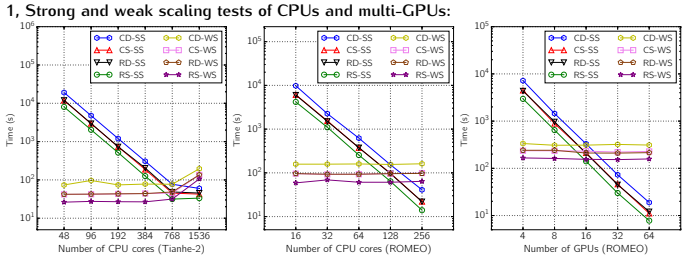


Experimental Hardwares

We implement SMG2S on the supercomputers *Tianhe-2* of National Supercomputing Center in Guangzhou and *Romeo* in University of Reims. *Tianhe-2* is a heterogeneous system made of Intel Xeon CPUs and Intel Knights Corner (KNC), with 16000 compute nodes in total. Each node composes 2 Intel Ivy Bridge 12 cores @ 2.2 GHz. *Romeo* is also a heterogeneous system made of Xeon CPUs and Nvidia GPUs, with 130 BullX R421 nodes. Each node composes 2 Intel Ivy Bridge 8 cores @ 2.6 GHz and 2 NVIDIA Tesla K20x GPUs.

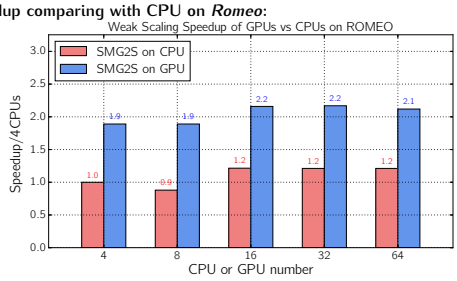
Strong and Weak Scalability

Fig 1, Strong and weak scaling tests of CPUs and multi-GPUs:



GPU Speedup

Fig 3, GPU speedup comparing with CPU on *Romeo*:



Accuracy Verification and Results

A method based on the Shifted Inverse Power Method to check the ability of keeping given spetra:

Algorithm 2 Shifted Inverse Power Method

Require: Matrix A, initial guess for desired eigenvalue σ, initial vector v_0

Ensure: Approximate eigenpair (θ, v)

1: $y = v_0$
2: **for** $i = 1, 2, 3 \cdots$ **do**
3: $\quad \theta = ||y||_\infty$, $v = y/\theta$
4: $\quad$ Solve $(A - \sigma I)y = v$
5: **end for**

$$error = \frac{||Av' - \lambda v'||}{||Av'||}$$

Table 1: Accuracy verification results with four different given spectra:

Matrix N°	Size	Spectra	precision	Accuracy	Acceptance (%)	max error
1	100	spec1	double	1×10^{-13}	100	6×10^{-14}
2	100	spec1	single	1×10^{-6}	100	3×10^{-7}
3	100	spec2	double	1×10^{-13}	100	8×10^{-14}
4	100	spec2	single	1×10^{-6}	97	3×10^{-3}
5	100	spec3	double	1×10^{-15}	100	4×10^{-16}
6	100	spec3	single	1×10^{-6}	100	6×10^{-7}
7	100	spec4	double	1×10^{-15}	94	4×10^{-4}
8	100	spec4	single	1×10^{-6}	100	9×10^{-7}

Conclusion and Perspectives

- SMG2S is a method to generate large-scale non-Hermitian matrices with good scalabilities;
- SMG2S has capacility to keep the accuracy of given spectra;
- An open source software should be implemented based on the basic C or C++, CUDA and MPI without PETSc and other large libraries;
- The matrix-matrix multiplication kernel should be optimized for both CPUs and multi-GPUs.

References

- X. Wu and S. G. Petiton. *A distributed and parallel asynchronous unite and conquer method to solve large scale non-hermitian linear systems.* In HPC Asia 2018: International Conference on High Performance Computing in Asia-Pacific Region, Tokyo, Japan, Jan. 2018.
- H. Galicher, F. Boillod-Cerneux, S. Petiton, and C. Calvin. *Generate very large sparse matrices starting from a given spectrum.* in lecture notes in computer science, 8969, springer (2014).

Maison de la Simulation · Université de Lille · CNRS · CRIStAL