Fast scalable implicit solver with convergence of physics-based simulation & data-driven learning: toward high-fidelity simulation with digital twin city

Tsuyoshi Ichimura[1,2], Kohei Fujita[1,3], Kentaro Koyama[4], Ryota Kusakabe[1], Kazuo Minami[3], Hikaru Inoue[4], Seiya Nishizawa[3], Miwako Tsuji[3], Tatsuo Nishiki[4], Muneo Hori[5,1], Lalith Maddegedara[1,5], Naonori Ueda[2]

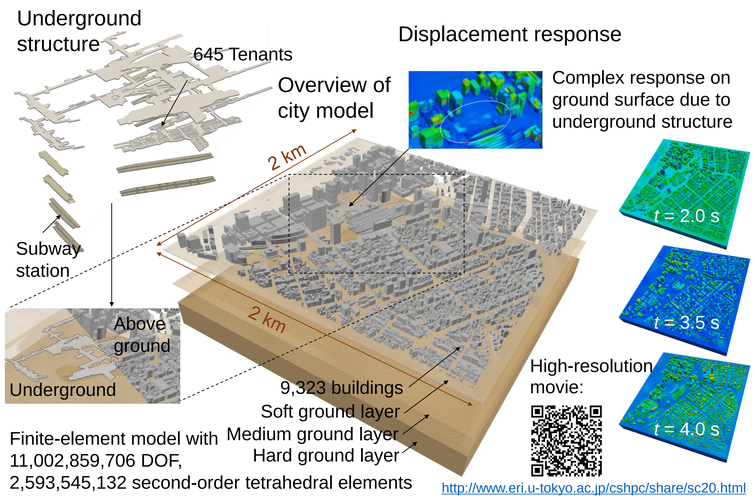
1. Earthquake Research Institute & Department of Civil Engineering, The University of Tokyo, 2. Center for Advanced Intelligence Project, RIKEN, 3. Center for Computational Science, RIKEN, 4. Fujitsu Limited, 5. Research Institute for Value-Added-Information Generation, Japan Agency for Marine-Earth Science and Technology

1. Overview of problem

- Detailed digital twin of a city, updated in real time by assimilation with measurement data, is expected to contribute towards safety and robustness of cities against threats; however, entails huge analysis cost
- We attempt to reduce the analysis cost by focusing on earthquake disasters
- Requires solving nonlinear wave propagation in coupled ground-city models of large areas ($10^3 \times 10^3 \times 10^{1\sim2}$) with complex structures in high resolutions ($10^{-2\sim-1}$) with low-order unstructured implicit finite element method:
  - $A^n \delta u^n = f^n$ …(1)
    where $A^n = \left(\frac{4}{dt^2}M + \frac{2}{dt}C^n + K^n\right)$, $f^n = F^n - Q^{n-1} + C^n v^{n-1} + M\left(a^{n-1} + \frac{4}{dt}v^{n-1}\right)$
    with $Q^n = Q^{n-1} + K^n \delta u^n, u^n = u^{n-1} + \delta u^n, v^n = -v^{n-1} + \frac{2}{dt}\delta u^n$,
    $a^n = -a^{n-1} - \frac{4}{dt}v^{n-1} + \frac{4}{dt^2}\delta u^n$.
  - Same equation solved in manufacturing industry; however, its size is limited to $10^{6\sim7}$ degrees-of-freedom (DOF) while that of the target urban earthquake problem becomes 10^{10}; thus, new development required
- **Develop a new approach by extending the merging of HPC-enhanced physics-based simulations with data-driven learning**
  - Propose an implicit solver that uses data generated during physics-based simulation for data-driven learning to accelerate the solver
  - Combine with Arm scalable vector extension (SVE) aware SIMD & multi-core tuning of core sparse matrix-vector multiplication kernel on Fugaku

2. Comparison with state of the art

- Matrix of Eq. (1) is huge and sparse; thus, iterative solvers with low memory footprint are used to solve the system of equations
- Comparison with standard solver
  - PCGE: conjugate gradient (CG) solver with block Jacobi preconditioning & matrix-free matrix-vector multiplication
  - Proposed solver IRIS attained **15.2-fold speedup over PCGE** on measurement problem on Fugaku (details shown in Perf. measurement)
- Comparison with state of the art
  - Proposed solver IRIS attained **10.3-fold speedup over GAMERA** (SC14 Gordon Bell Prize Finalist solver based on multi-grid & mixed precision preconditioning) in a fully coupled super-high resolution ground-city earthquake simulation on 98,304 cores of Fugaku
- Significant speedup attained by running developed method on large-scale supercomputers expected to advance digital twin of cities and improve resilience to earthquake disasters



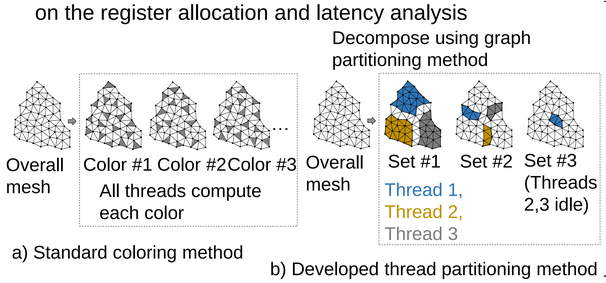
3. Innovations realized

1. Data-driven algorithm for improvement of solver convergence

- Construct a surrogate model of the physics simulation by applying data driven learning to the analysis results obtained in the previous time steps and use this surrogate model as a preconditioner in an iterative solver
- Use a three step multi-grid in the adaptive preconditioner of CG method (**Algorithm 1**)
  - Global nonlinear behavior can be solved by the coarse grids (A^n_{c1}: matrix derived from geometric coarsening of A^n, A^n_{c2}: matrix derived by algebraic-coarsening of A^n_{c1}); however, not effective in solving high-frequency modes, leading to large computational cost in fine grids
  - On the other hand, as a physical time history problem is solved, the projection $D_c^{n,i}$ from $r^{n,i} - A^n R_c z_c^{n,i}$ to $z^{n,i} - R_c z_c^{n,i}$ is expected to be similar in short period of time
- Learn the relationship between $r^{n,i} - A^n R_c z_c^{n,i}$ and $z^{n,i} - R_c z_c^{n,i}$ using past solver solutions, and use this projection for correcting the coarse solution (**Algorithm 2**)
  - Domain is partitioned using graph partitioning methods, and the projection is defined in each subdomain to increase number of effective modes with limited data
  - Leads to improvement of high-frequency modes in the coarse grid solution, leading to reduction in the number of fine grid iterations and total computational cost
  - Scalable on parallel systems as generation and application of projection is conducted independently in each subdomain without communication

2. Arm SVE aware SIMD & multi-core tuning of core kernel

- Although cost is transferred to the surrogate model, kernel of multiplying a vector to A^n remains
- Matrix-free matrix-vector multiplication involves much random access with data recurrence; not straightforward to attain performance on wide-SIMD and many-core CPUs
- Develop novel thread partitioning and SIMD buffering methods to utilize all cores and full SIMD width in matrix-free matrix-vector multiplication
- Combine with Fugaku's A64FX CPU specific tuning based on the register allocation and latency analysis



Algorithm 1 The algorithm is based on the adaptive conjugate gradient method to perform nonlinear dynamic response analysis at an n-th time step by solving Eq. (1). $\mathbf{F}^n - \mathbf{Q}^{n-1} + \mathbf{C}^n\mathbf{v}^{n-1} + \mathbf{M}(\mathbf{a}^{n-1} + \frac{4}{dt}\mathbf{v}^{n-1})$ is the right hand vector of Eq. (1). $\frac{dt}{24}(-9\mathbf{v}^{n-4} + 37\mathbf{v}^{n-3} - 59\mathbf{v}^{n-2} + 55\mathbf{v}^{n-1})$ is the initial solution estimated using the Adams-Bashforth method. The algorithm to solve $\mathbf{A}^n\delta\mathbf{u}^n = \mathbf{f}^n$ is shown in lines 4-22. ϵ is the tolerance for convergence judgment. Double-precision computation is used, except for the computation of line 9.

1: read boundary condition of n-th time step
2: $\mathbf{f}^n \Leftarrow \mathbf{F}^n - \mathbf{Q}^{n-1} + \mathbf{C}^n\mathbf{v}^{n-1} + \mathbf{M}(\mathbf{a}^{n-1} + \frac{4}{dt}\mathbf{v}^{n-1})$
3: $\delta\mathbf{u}^n \Leftarrow \frac{dt}{24}(-9\mathbf{v}^{n-4} + 37\mathbf{v}^{n-3} - 59\mathbf{v}^{n-2} + 55\mathbf{v}^{n-1})$
4: $\mathbf{r} \Leftarrow \mathbf{f}^n - \mathbf{A}^n\delta\mathbf{u}^n$
5: $\beta \Leftarrow 0$
6: $i \Leftarrow 1$
7: **while** $\|\mathbf{r}\|_2/\|\mathbf{f}\|_2 \geq \epsilon$ **do**
8: $\mathbf{r}^{n,i} \Leftarrow \mathbf{r}$
9: solve $\mathbf{A}^n\mathbf{z}^{n,i} = \mathbf{r}^{n,i}$ by CG method with ϵ_{pre}
10: $\mathbf{z} \Leftarrow \mathbf{z}^{n,i}$
11: **if** $i > 1$ **then**
12: $\beta \Leftarrow (\mathbf{z}, \mathbf{q})/\rho$
13: **end if**
14: $\mathbf{p} \Leftarrow \mathbf{z} + \beta\mathbf{p}$
15: $\mathbf{q} \Leftarrow \mathbf{A}^n\mathbf{p}$
16: $\rho \Leftarrow (\mathbf{z}, \mathbf{r})$
17: $\alpha \Leftarrow \rho/(\mathbf{p}, \mathbf{q})$
18: $\mathbf{q} \Leftarrow -\alpha\mathbf{q}$
19: $\mathbf{r} \Leftarrow \mathbf{r} + \mathbf{q}$
20: $\delta\mathbf{u}^n \Leftarrow \delta\mathbf{u}^n + \alpha\mathbf{p}$
21: $i \Leftarrow i + 1$
22: **end while**
23: $\mathbf{Q}^n \Leftarrow \mathbf{Q}^{n-1} + \mathbf{K}^n\delta\mathbf{u}^n$
24: $\mathbf{u}^n \Leftarrow \mathbf{u}^{n-1} + \delta\mathbf{u}^n$
25: $\mathbf{v}^n \Leftarrow -\mathbf{v}^{n-1} + \frac{2}{dt}\delta\mathbf{u}^n$
26: $\mathbf{a}^n \Leftarrow -\mathbf{a}^{n-1} - \frac{4}{dt}\mathbf{v}^{n-1} + \frac{4}{dt^2}\delta\mathbf{u}^n$
27: output results of n-th time step
28: set $\mathbf{A}^{n+1}$ considering nonlinearity

Algorithm 2 A solver algorithm combining equation-based modeling and data-driven learning is used to solve $\mathbf{A}^n\mathbf{z}^{n,i} = \mathbf{r}^{n,i}$ in line 9 of Algorithm 1. $diag[\mathbf{A}^n_{c2}]$ is the 3×3 block diagonal matrix of $\mathbf{A}^n_{c2}$. The 3×3 block diagonal matrix is used as the preconditioning matrix in each CG method, whose tolerance of convergence judgment is ϵ_{c2}, ϵ_{c1} and ϵ_{pre}, respectively. $\mathbf{D}^{n,i}_{c2}$ and $\mathbf{D}^{n,i}_{c1}$ are the projections defined in each small domain for the estimation of high-frequency modes obtained from analyzing data of previous steps. By using these projections for the estimation of the high-frequency modes, the number of iterations in the fine equation is transferred to the iterations on less costly coarser CG methods, even for the high-frequency mode-dominated problems. All computation is conducted in single-precision.

1: $\mathbf{r}^{n,i}_{c1} \Leftarrow \mathbf{P}_{c1}\mathbf{r}^{n,i}$
2: $\mathbf{r}^{n,i}_{c2} \Leftarrow \mathbf{P}_{c2}\mathbf{r}^{n,i}_{c1}$
3: $\mathbf{z}^{n,i}_{c2} \Leftarrow (diag[\mathbf{A}^n_{c2}])^{-1}\mathbf{r}^{n,i}_{c2}$
4: solve $\mathbf{A}^n_{c2}\mathbf{z}^{n,i}_{c2} = \mathbf{r}^{n,i}_{c2}$ by CG method with ϵ_{c2}
5: $\mathbf{z}^{n,i}_{c1} \Leftarrow \mathbf{R}_{c2}\mathbf{z}^{n,i}_{c2} + \mathbf{D}^{n,i}_{c2}(\mathbf{r}^{n,i}_{c1} - \mathbf{A}^n_{c1}\mathbf{R}_{c2}\mathbf{z}^{n,i}_{c2})$
6: solve $\mathbf{A}^n_{c1}\mathbf{z}^{n,i}_{c1} = \mathbf{r}^{n,i}_{c1}$ by CG method with ϵ_{c1}
7: construct/update $\mathbf{D}^{n+1,i}_{c2}$ using $(\mathbf{z}^{n,i}_{c1} - \mathbf{R}_{c2}\mathbf{z}^{n,i}_{c2})$ and $\mathbf{A}^n_{c1}(\mathbf{z}^{n,i}_{c1} - \mathbf{R}_{c2}\mathbf{z}^{n,i}_{c2})$
8: $\mathbf{z}^{n,i} \Leftarrow \mathbf{R}_{c1}\mathbf{z}^{n,i}_{c1} + \mathbf{D}^{n,i}_{c1}(\mathbf{r}^{n,i} - \mathbf{A}^n\mathbf{R}_{c1}\mathbf{z}^{n,i}_{c1})$
9: solve $\mathbf{A}^n\mathbf{z}^{n,i} = \mathbf{r}^{n,i}$ by CG method with ϵ_{pre}
10: construct/update $\mathbf{D}^{n+1,i}_{c1}$ using $(\mathbf{z}^{n,i} - \mathbf{R}_{c1}\mathbf{z}^{n,i}_{c1})$ and $\mathbf{A}^n(\mathbf{z}^{n,i} - \mathbf{R}_{c1}\mathbf{z}^{n,i}_{c1})$

4. Performance measurement

Measurement settings

- Measure performance on Fugaku (48-core Arm SVE CPU-based massively parallel supercomputer; 4 MPI procs.×12 OpenMP threads per node) and Oakbridge-CX (dual 28-core Cascade Lake Xeon-based system; 4 MPI procs.×14 OpenMP threads per node)
- Measure elapsed time for solving 101~200-th time steps of a nonlinear wave propagation problem in a finite-element model of an actual city
- Use previous 32 steps for constructing and updating projection $D_c^{n,i}$ with about 300 DOF per subdomain

Data-driven algorithm performance

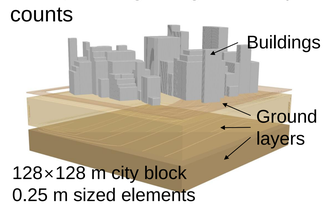
- Number of iterations of fine solver reduced by 2.18-fold from PCGE, which shifts load to coarse solvers with low computational cost

Core matrix-free matrix-vector multiplication kernel performance

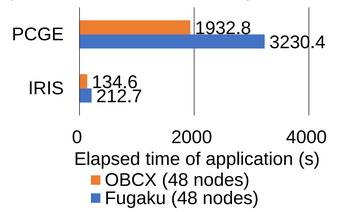
- By development of data-driven algorithm, core kernel changed from FP64 SpMV of PCGE to FP32 SpMV
- Combined with the SIMD & multi-core aware tuning led to 12.8- and 7.73-fold improvement on Fugaku and Oakbridge-CX, respectively

Total application performance

- On 48 nodes of Fugaku: Attained **15.2-fold speedup from PCGE** with 12.7% FP64 peak performance
- On 48 nodes of Oakbridge-CX: Attained **14.3-fold speedup from PCGE** with 12.8% FP64 peak performance
- Excellent **weak scalability of 96.4%** attained from 2,304 cores to 1,179,648 cores, leading to 11.8% of FP64 peak on 1,179,648 cores (8.87 PFLOPS)
- **Peak performance of 11.8%** is very high for a low-order unstructured finite-element solver (e.g., performance of HPCG benchmark, which has similar characteristics as the target problem, is 2.6% on Fugaku)
- High speedup efficiency was attained from 2304 cores to 18,432 cores, leading to significant speedup from PCGE even on high core counts

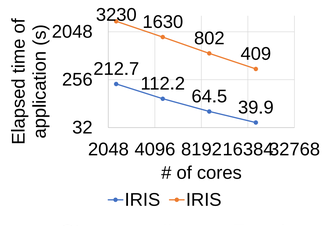


Measurement problem setting



Time-to-solution of solvers

Weak scaling up to 1,179,648 Fugaku cores



Strong scaling on Fugaku

5. Summary and implications

- Although the difference in accuracy and characteristics of the physics-based simulation methods and data-driven learning methods made it seem difficult to integrate these methods, we managed to facilitate the connection between these methods by designing a suitable interface
  - Not only improves solver convergence but also enables more efficient computation, as it replaces random access-based SpMV with sequential access-based data driven learning
- Combination with SIMD and multi-core tuning of matrix-free matrix-vector product kernel led to significant speedup from standard solver PCGE and state of the art solver GAMERA
- Similar approach expected to be useful for utilizing the highly efficient cores of recent architectures for physics-based problems with large heterogeneity and random data access
- Expect constructing more accurate data-rich data-driven methods by utilizing wide levels of storage with varying capacity and bandwidth by the developed approach

Acknowledgement: Results were obtained using Fugaku at Center for Computational Science, RIKEN, and Oakbridge-CX at Information Technology Center, The University of Tokyo. This work was supported by MEXT as "Program for Promoting Researches on the Supercomputer Fugaku" (Large-scale numerical simulation of earthquake generation, wave propagation and soil amplification: hp200126). This work was supported by Post K computer project (Priority Issue 3 – Development of integrated simulation systems for hazards and disasters induced by earthquakes and tsunamis: hp190177) and JSPS KAKENHI Grant Numbers JP18H05239, JP18H03795, JP17K14719.