EuroHPC
Joint Undertaking

Data driven Computational Mechanics at EXascale

DCoMEX

Data driven Computational Mechanics at EXascale

Work program topic: EuroHPC-01-2019
Type of action: Research and Innovation Action (RIA)

AI-SOLVE LIBRARY PROTOTYPE (software module)

DELIVERABLE D3.1

Version No 1

http://www.dcomex.eu/
This project has received funding from the European High-Performance Computing Joint Undertaking Joint Undertaking ('the JU'), under Grant Agreement No 956201

DOCUMENT SUMMARY INFORMATION

Project Title	Data driven Computational Mechanics at EXascale
Project Acronym	DCoMEX
Project No:	956201
Call Identifier:	EuroHPC-01-2019
Project Start Date	01/04/2021
Related work package	WP 3
Related task(s)	Task 3.1, 3.2, 3.3, 3.7
Lead Organisation	NTUA
Submission date	18/02/2024
Re-submission date	
Dissemination Level	PU

Quality Control:

	Who	Affiliation	Date
Checked by internal reviewer	George Stavroulakis	NTUA	17/02/2024
Checked by WP Leader	Vissarion Papadopoulos	NTUA	17/02/2024
Checked by Project Coordinator	Vissarion Papadopoulos	NTUA	17/02/2024

Document Change History:

Version	Date	Author (s)	Affiliation	Comment
1.0	15.02.2024	Ioannis Kalogeris	NTUA	

Contents

1. Description

Deliverable 3.1 is associated to WP3 of the DCoMEX project, and it describes the content of to the AI-SOLVE software module. The algorithms that constitute AI-Solve are distributed among 5 modules according to their scope: (i) linear algebra, (ii) solvers, (iii) machine learning, (iv) AISolve.core and (v) AISolve.MSolve. The content of each of these modules is illustrated below, while a detailed explanation of the theoretical background behind these algorithms along with numerical applications are provided in deliverable D3.2.

2. Contents of the AI-Solve software library

This section details the content and functionalities of the AI-Solve library that was developed within the project. All project related software implementations can be found in https://github.com/mgroupntua, and are publicly available.

2.1 Linear algebra repository

The linear algebra public repository (https://github.com/mgroupntua/LinearAlgebra) includes implementations and wrappers for linear operations in C# and in the context of the DCoMEX project the following algorithms have been developed:

- The Gauss-Seidel iterative algorithm: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/GaussSeidel/GaussSeidelAlgorithm.cs
- The conjugate gradient method: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/ConjugateGradient/CGAlgorithm.cs
- The preconditioned conjugate gradient method: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/PreconditionedConjugateGradient/PcgAlgorithm.cs
- The block preconditioned conjugate gradient method: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/PreconditionedConjugateGradient/BlockPcgAlgorithm.cs
- The generalized minimal residual method (GMRES): https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/GeneralizedMinimalResidual/GmresAlgorithm.cs
- The algebraic multigrid method: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/AlgebraicMultiGrid/AlgebraicMultiGrid.cs
- The Proper Orthogonal Decomposition algorithm: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/AlgebraicMultiGrid/PodAmg/ProperOrthogonalDecomposition.cs
- The POD-2G method described in section 3.1 to be used as a standalone iterative solver: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/AlgebraicMultiGrid/PodAmg/PodAmgAlgorithm.cs
- The POD-2G method described in section 3.1 to be used as a preconditioner in the context of CG: https://github.com/mgroupntua/LinearAlgebra/blob/develop/src/MGroup.LinearAlgebra/Iterative/AlgebraicMultiGrid/PodAmg/PodAmgPreconditioner.cs

The verify the correct implementation of these algorithms several numerical tests are included in this repository that can be found in https://github.com/mgroupntua/LinearAlgebra/tree/develop/tests/MGroup.LinearAlgebra.Tests

2.2 Solvers repository

The Solvers public repository (https://github.com/mgroupntua/Solvers) utilizes the linear algebra algorithms to solve linear systems of equations arising from engineering problems. The Solvers repository acts as a mediator between the LinearAlgebra repo and MSolve modules, making the proper connections and associations between the discretization processes of the PDEs to be solved and the actual linear algebra objects that are used for solving the emerging linear systems. Moreover, the Solvers repo has all necessary information, in order to compose problem-specific preconditioners and solution strategies.

- PCG solver: https://github.com/mgroupntua/Solvers/blob/develop/src/MGroup.Solvers/Iterative/PcgSolver.cs
- Block PCG solver: https://github.com/mgroupntua/Solvers/blob/develop/src/MGroup.Solvers/Iterative/BlockPcgSolver.cs
- GMRES solver: https://github.com/mgroupntua/Solvers/blob/develop/src/MGroup.Solvers/Iterative/GmresSolver.cs

- POD-2G solver: https://github.com/mgroupntua/Solvers/blob/develop/src/MGroup.Solvers.MachineLearning/PodAmg/AmgAISolver.cs

The family of DDM algorithms is currently at a seperate repository (https://github.com/SerafeimBakalakos/MSolveOne/tree/paper/xfem_pfetidp_mpi) and will be integrated in the main code in the following months. All DDM algorithms described in sections 2.5.5-2.5.7 can be found in:

- P-DDM (or PSM): https://github.com/SerafeimBakalakos/MSolveOne/blob/paper/xfem_pfetidp_mpi/src/MGroup.Solvers.DDM/PSM/PsmSolver.cs
- Feti-DP : https://github.com/SerafeimBakalakos/MSolveOne/blob/paper/xfem_pfetidp_mpi/src/MGroup.Solvers.DDM/FetiDP/FetiDPSolver.cs
- P-Feti-DP: https://github.com/SerafeimBakalakos/MSolveOne/blob/paper/xfem_pfetidp_mpi/src/MGroup.Solvers.DDM/PFetiDP/PFetiDPSolver.cs

2.3 Machine learning repository

The MachineLearning public repository (https://github.com/mgroupntua/MachineLearning) contains the different types of neural networks that were implemented and used for the project's purposes. These include:

- Feedforward Neural Networks: https://github.com/mgroupntua/MachineLearning/blob/develop/src/MGroup.MachineLearning.TensorFlow/NeuralNetworks/FeedForwardNeuralNetwork.cs
- Convolutional Neural Networks: https://github.com/mgroupntua/MachineLearning/blob/develop/src/MGroup.MachineLearning.TensorFlow/NeuralNetworks/ConvolutionalNeuralNetwork.cs
- Autoencoders: https://github.com/mgroupntua/MachineLearning/blob/develop/src/MGroup.MachineLearning.TensorFlow/NeuralNetworks/Autoencoder.cs
- Convolutional Autoencoders: https://github.com/mgroupntua/MachineLearning/blob/develop/src/MGroup.MachineLearning.TensorFlow/NeuralNetworks/ConvolutionalAutoencoder.cs

Based on these building blocks, the customized surrogate modeling technique mentioned in D2.3 that combines FFNN and CAEs to provide predictions for complex systems (step 1 of the data-driven solution framework in Section 3.1) can be found in https://github.com/mgroupntua/MachineLearning/blob/develop/src/MGroup.Constitutive.Structural.MachineLearning/Surrogates/CaeFffnSurrogate.cs

In addition, dedicated surrogate modeling techniques were developed to accelerate the DCoMEX-Mat applications in WP 7. These include neural networks that emulate the constitutive behavior of composite materials and can be found in: https://github.com/mgroupntua/MachineLearning/tree/develop/src/MGroup.Constitutive.Structural.MachineLearning/Continuum

2.4 AISolve.core repository

AISOLVE.core (https://github.com/mgroupntua/AISolve.Core/tree/develop/src) is a collection of interfaces and classes for the workflow definition of AISolve. AISolve has been designed in a fashion allowing for the interconnection of heterogeneous software modules that will tackle the solution of the various phases of the AISolve workflow.

2.5 AISolve.MSolve repository

AISolve.MSolve repository (https://github.com/mgroupntua/AISolve.MSolve) is the MSolve implementation of AISolve.Core interfaces and workflow. Specifically, this repository implements all the interfaces defined in the AISolve.Core repository and connects it with the appropriate MSolve modules for the solution of engineering problems with the aid of AISolve.