Frontal and multi-frontal solver for large sparse linear systems for high-performance finite element analysis

dr hab. Maciej Paszynski
Department of Computer Science
AGH University of Science and Technology, Krakow, Poland
e-mail:maciej.paszynski@agh.edu.pl
http://home.agh.edu.pl/paszynsk
http://www.ki.agh.edu.pl/en/staff/paszynski-maciej
http://www.ki.agh.edu.pl/en/research-groups/a2s

Lecture 1: Frontal and multi-frontal solvers: orderings, elimination trees, refinement trees

The lecture introduces the frontal and multi-frontal solver algorithms on the example of 1D finite difference and 2D finite element method. The relation between the ordering and elimination trees are discussed. The comparison of total cost of the frontal and multi-frontal solver for simple 2D finite element mesh is presented. We also relate the construction of elimination trees with the construction of the refinement trees.

Lecture 2: Frontal and multi-frontal solvers: hypermatrix generalization to higher orders and multi-physics

This lecture introduces the conception of hypernode, and presents a data structure for storing hypermatrices. In the hypermatrix the entries are replaced by blocks corresponding the finite element nodes with higher polynomial orders of approximation and / or with multi-physics problems. The exemplary run of the algorithm is discussed. We also address the merging and ordering algorithms designed on the level of hypernodes.

Lecture 3: Frontal and multi-frontal solvers: LAPACK interface on the level of hypermatrix

In this presentation we focus on implementation of the partial forward elimination algorithm, resulting in local Schur complement computations with the LAPACK interface. The matrices from nodes of the elimination tree implemented within the hypermatrix concept are interfaced with LAPACK calls in order to compute Schur complement to be merged later on the level of parent nodes.

Lecture 4: Frontal and multi-frontal solves: Dealing with singularities

In this lecture we considered six basic refinement patters for a 3D hexahedral element, namely, uniform refinements, refinements towards a point, isotropic and anisotropic refinements towards an edge, and isotropic and anisotropic refinements towards a face. We showed theoretically that in all cases except the ones corresponding to uniform refinements and face anisotropic refinements, the FLOPS required to solve the corresponding system of linear equations grows only linearly with respect to the mesh size. Theoretical results have been confirmed by numerical experiments performed with MUMPS solver and the ordering provided by the METIS library.

Lecture 5: Frontal and multi-frontal solves: Reutilization

In this lecture we present how to construct a linear computational cost solver for a sequence of grids refined towards point singularity in two or three dimensions. The linear computational cost for the entire sequence of grids is obtained by using the reutilization technique.

Lecture 6: Frontal and multi-frontal solves: Graph grammar based model of concurrency

This lecture describes how to express the frontal and multi-frontal solver algorithms by graph grammar productions. The solver algorithm is decomposed into basic undividable tasks that must be executed in sequential. The graph grammar approach allows to construct efficient parallel implementations of the solver algorithms

Lecture 7: Frontal and multi-frontal solves: Generalization to isogeometric finite element method

This lecture describes how to generalized the idea presented in previous lectures into isogeometric finite element method. In particular we will focus on the elimination pattern for the B-spline basis functions, as well as extension of the graph grammar model in order to build efficient isogeometric solver