Efficient parallelization of direct solvers for isogeometric analysis

Maciej Paszyński

Department of Computer Science
AGH University, Krakow, Poland
home.agh.edu.pl/paszynsk

Collaborators:
David Pardo (UPV / BCAM / IKERBASQUE, Spain)
Daniel Garcia(BCAM, Spain)
Victor Calo (Curtin Uniwersity, Australia)

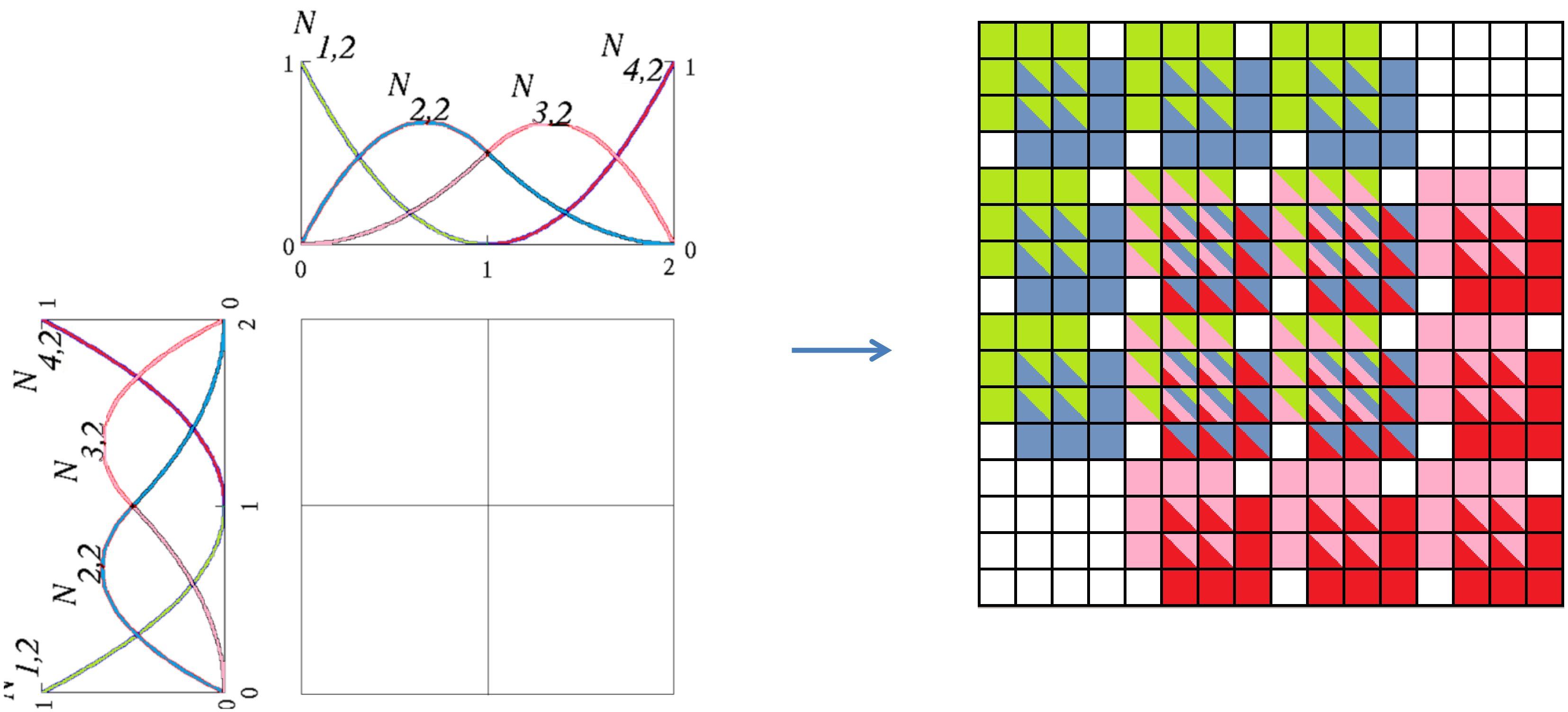
PhD Students:
Maciej Woźniak
Marcin Łoś
Konrad Jopek
Marcin Skotniczny
Grzegorz Gurgul

Motivation

In 1D/2D/3D Finite Element Method computations
it is possible to **refine basis functions** over the computational mesh
in such a way that

- the **topology of the mesh does not change**
- **accuracy** of the numerical approximation **is similar**
- computational cost of both direct and parallel solvers
  is reduced up to two orders of magnitude
- **efficiency** of parallel solver is better

Computational mesh, sparse matrix and direct solvers

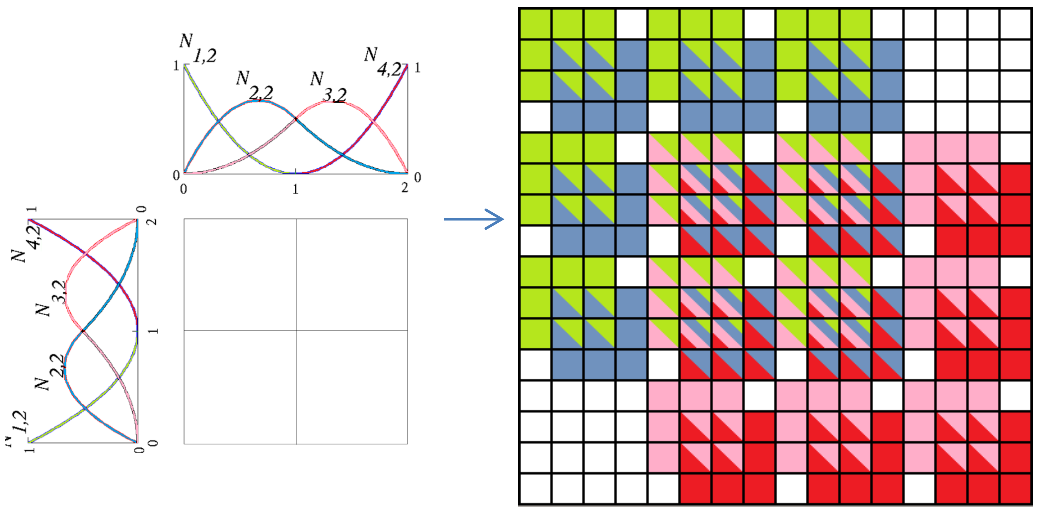


2D Isogeometric Analysis Finite Element Method (IGA-FEM)

Basis functions defined as tensor products of B-splines

Element matrices merged into the global matrix

Sparse matrix based direct solvers

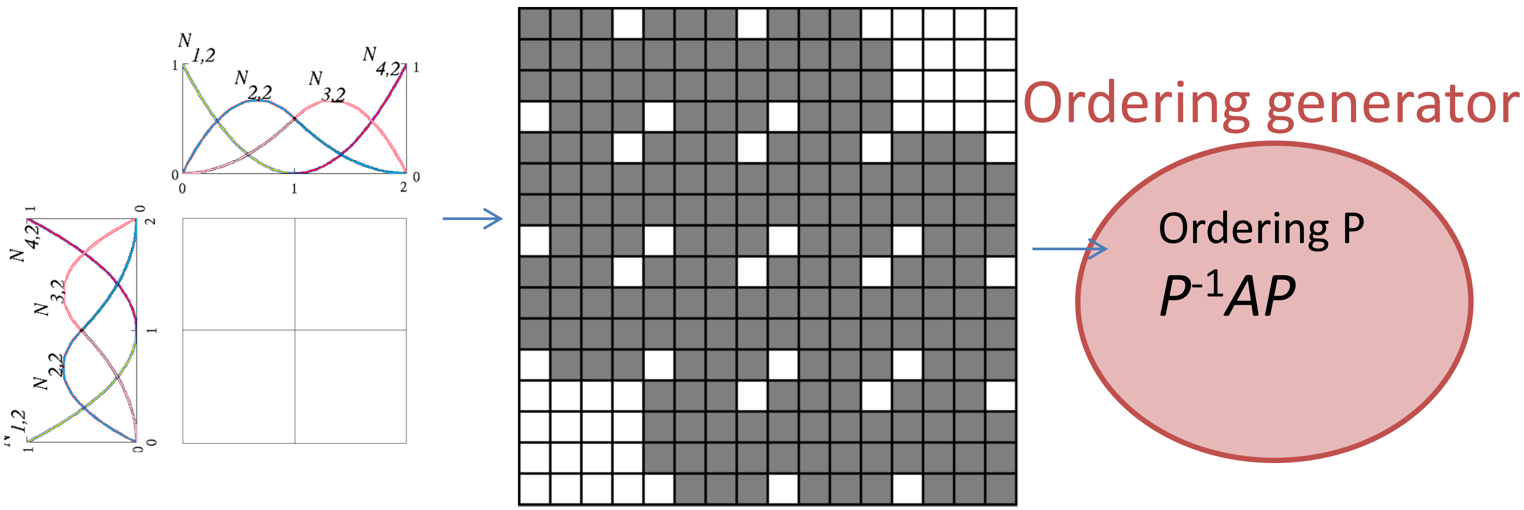


Sparse global matrix, stored in some compressed manner, e.g.

- coordinate format,
- CSC format
- CSR format

(see e.g. Sparse Matrix Computations lectures by Jean Yves L'Excellent et al. http://graal.ens-lyon.fr/~bucar/CR07/introSparse.pdf for more details)

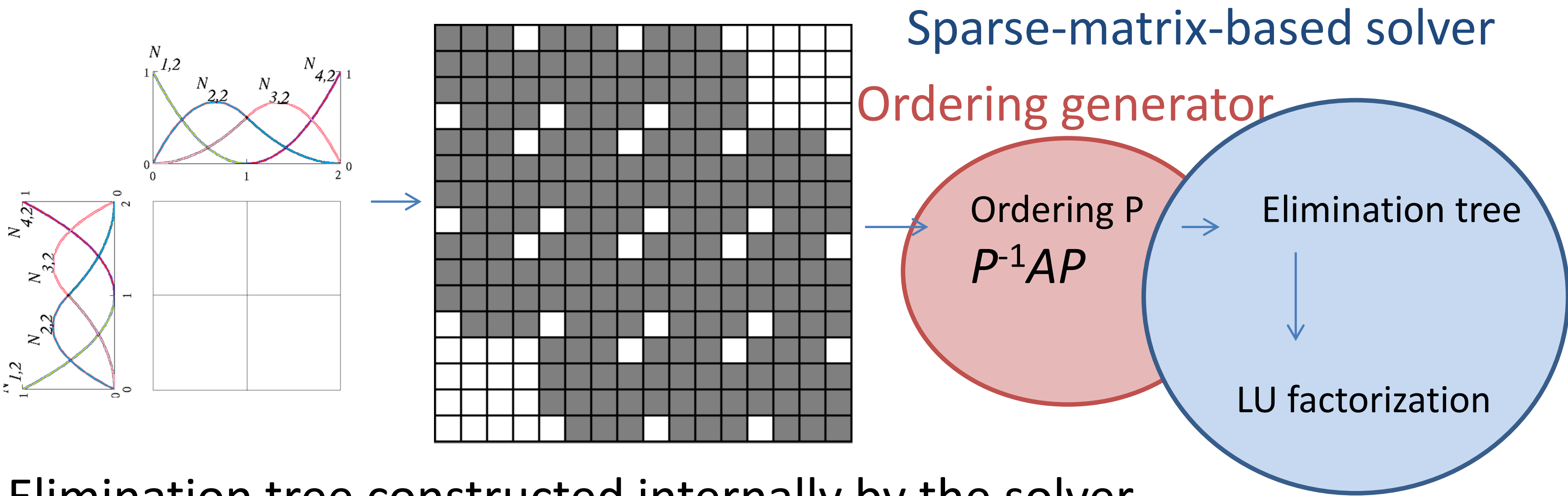
Sparse matrix based direct solvers



Several algorithms constructing ordering looking at the structure of the sparse matrix, e.g. available through MUMPS solver interface:

- nested-dissections (METIS)
- aproximate minimum degree (AMD)
- PORD

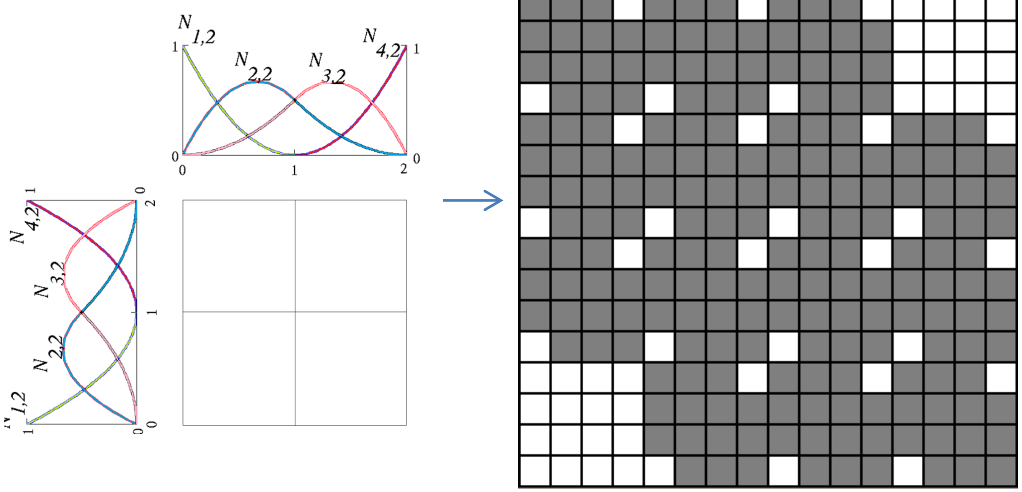
Sparse matrix based direct solvers

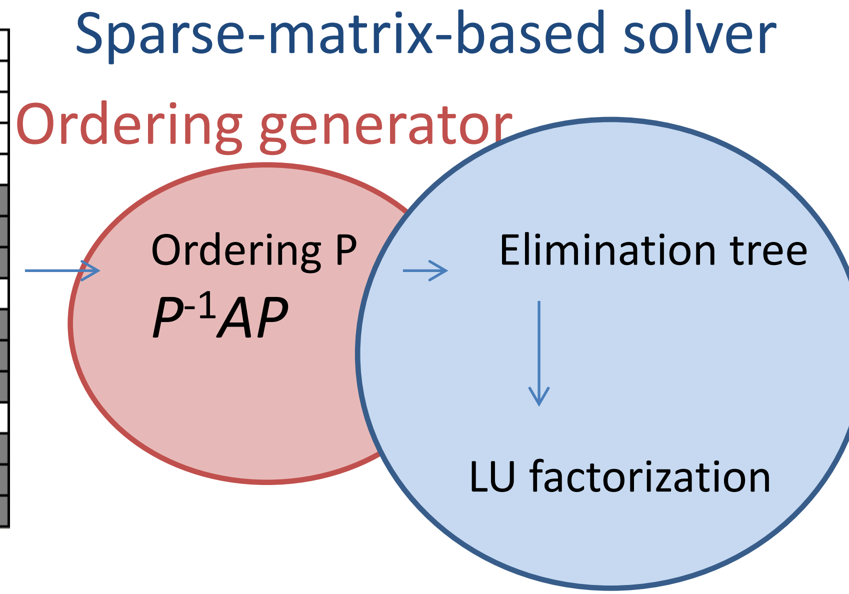


Elimination tree constructed internally by the solver followed by LU factorization

(for more details on the elimination trees
see e.g. Sparse Matrix Computations lectures by Jean Yves L'Excellent et al.:
http://graal.ens-lyon.fr/~bucar/CR07/lecture-etree.pdf
http://graal.ens-lyon.fr/~bucar/CR07/factorization.pdf
)

Sparse matrix based direct solvers

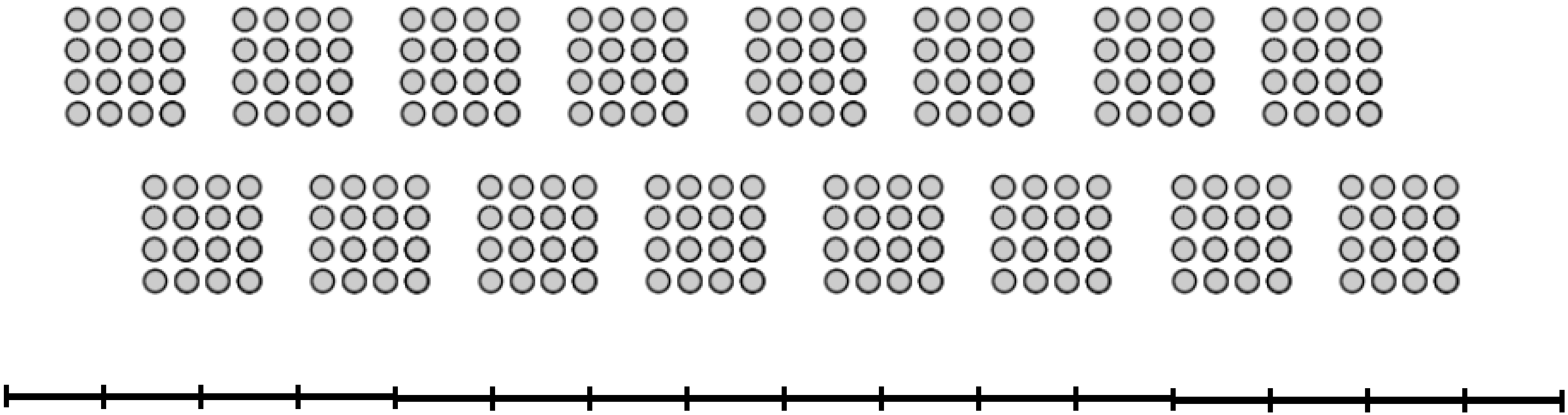




Sparse-matrix based direct solvers
lost information about basis functions spread over the mesh

Additional knowledge about the basis functions
allows to speed up both sequential and parallel solvers
up to two orders of magnitude

Isogeometric analysis



16 finite elements, 16 element matrices

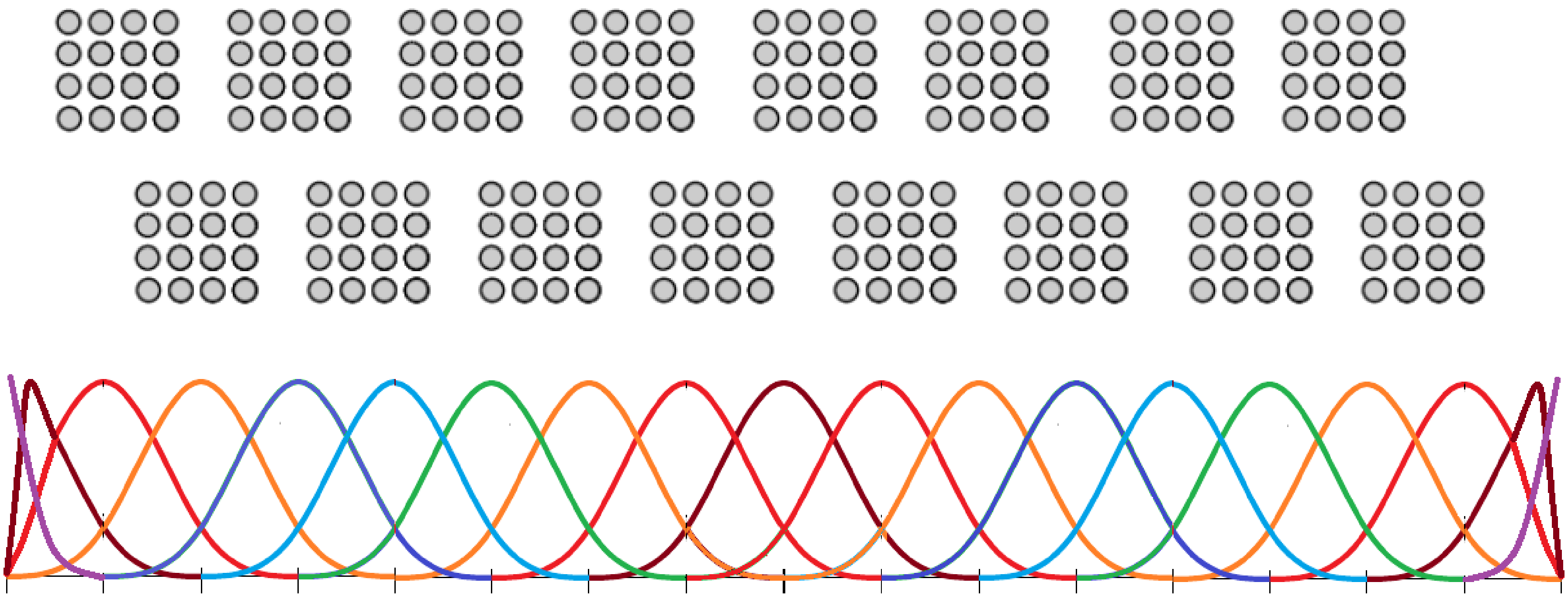
↓ merged (assembled) into

1 Global matrix

↓ submitted to

Direct solver

Isogeometric analysis



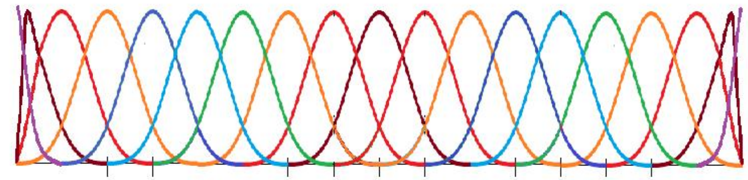
16 elements with cubic B-splines

4 basis functions per element → 4x4 element matrices

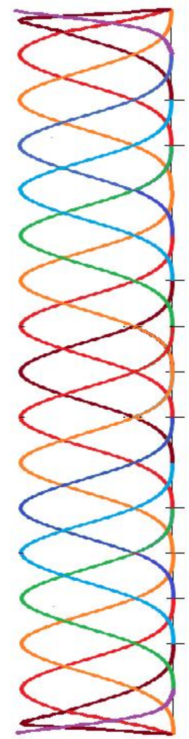
Isogeometric analysis

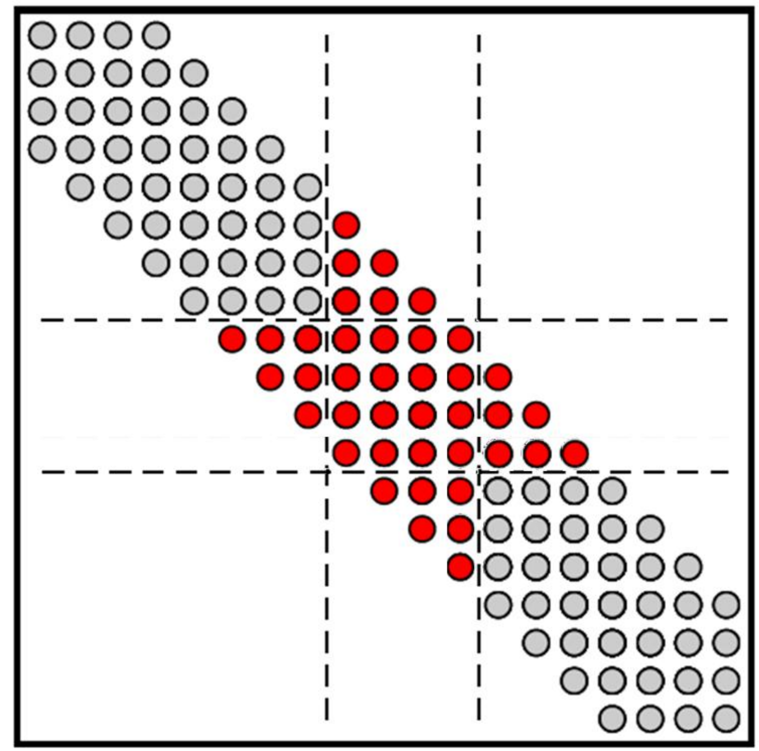
16 element frontal matrices

Size of each element matrix 4x4



assembled into





Global matrix:

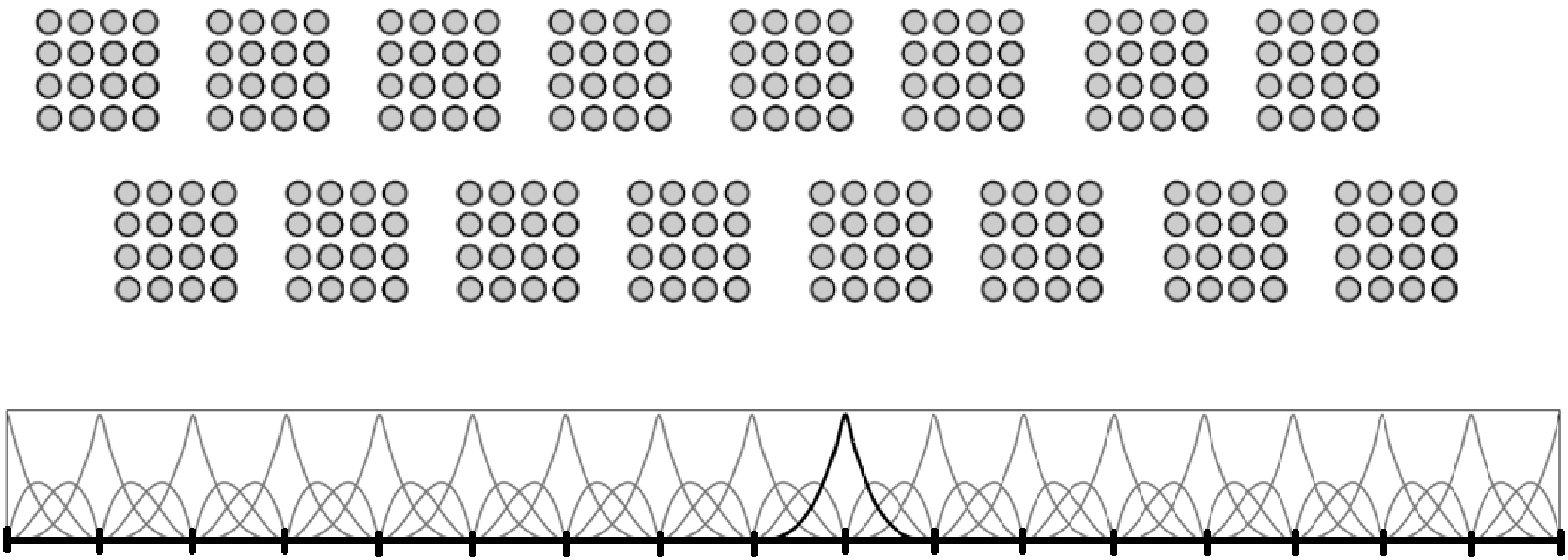
Small size N=19

(=16+3)

Dense diagonals

Element matrices overlap to the greatest extend

Traditional Finite Element Method analysis



When we introduce additional basis functions „C^0 separators”
in between finite elements we obtain tradition Finite Element Method
with third order polynomials

We enrich the space of basis functions, so the accuracy is similar

Traditional Finite Element Method analysis

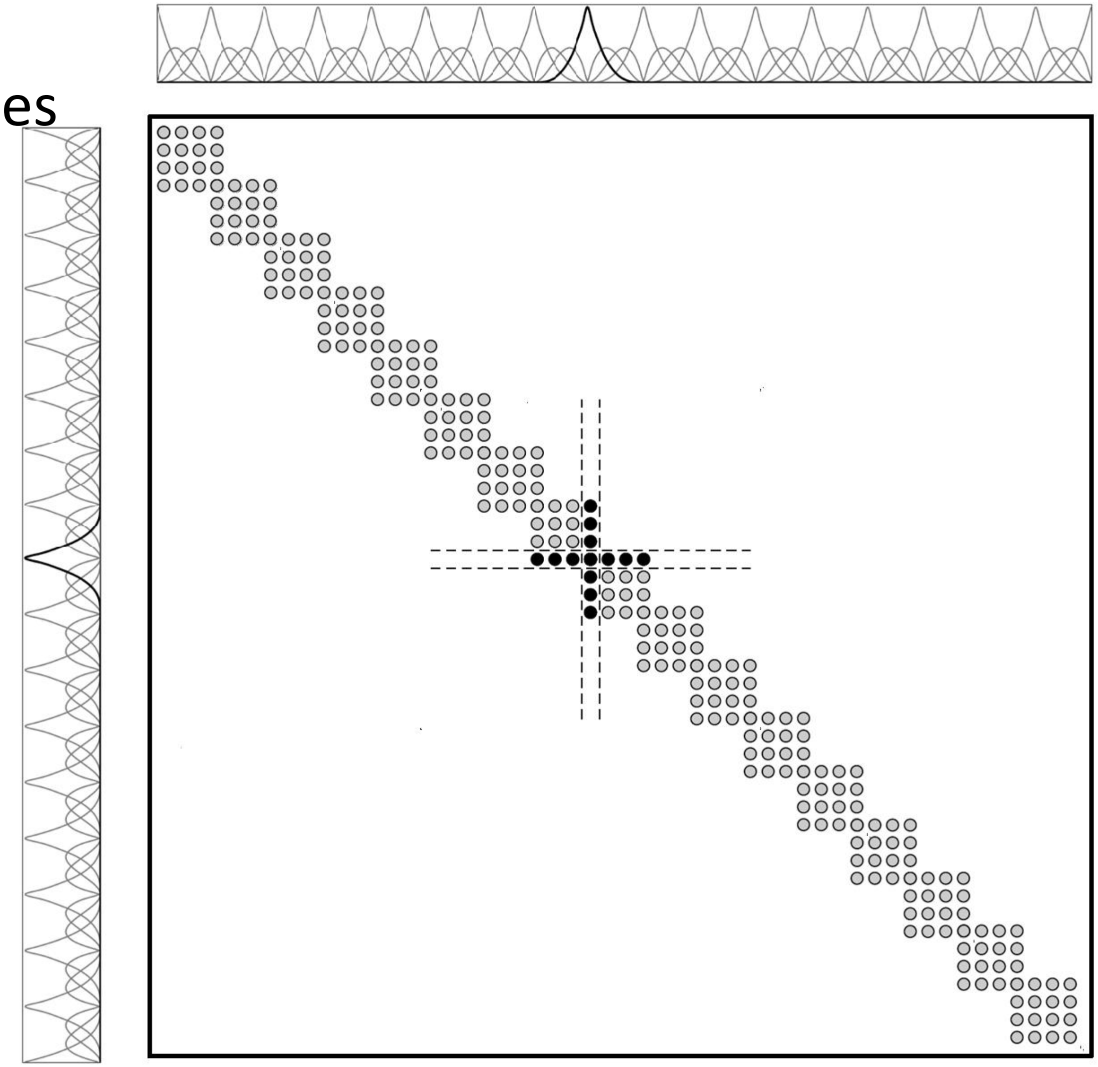
16 element frontal matrices
each element matrix 4x4

↓ assembled into

Global matrix:

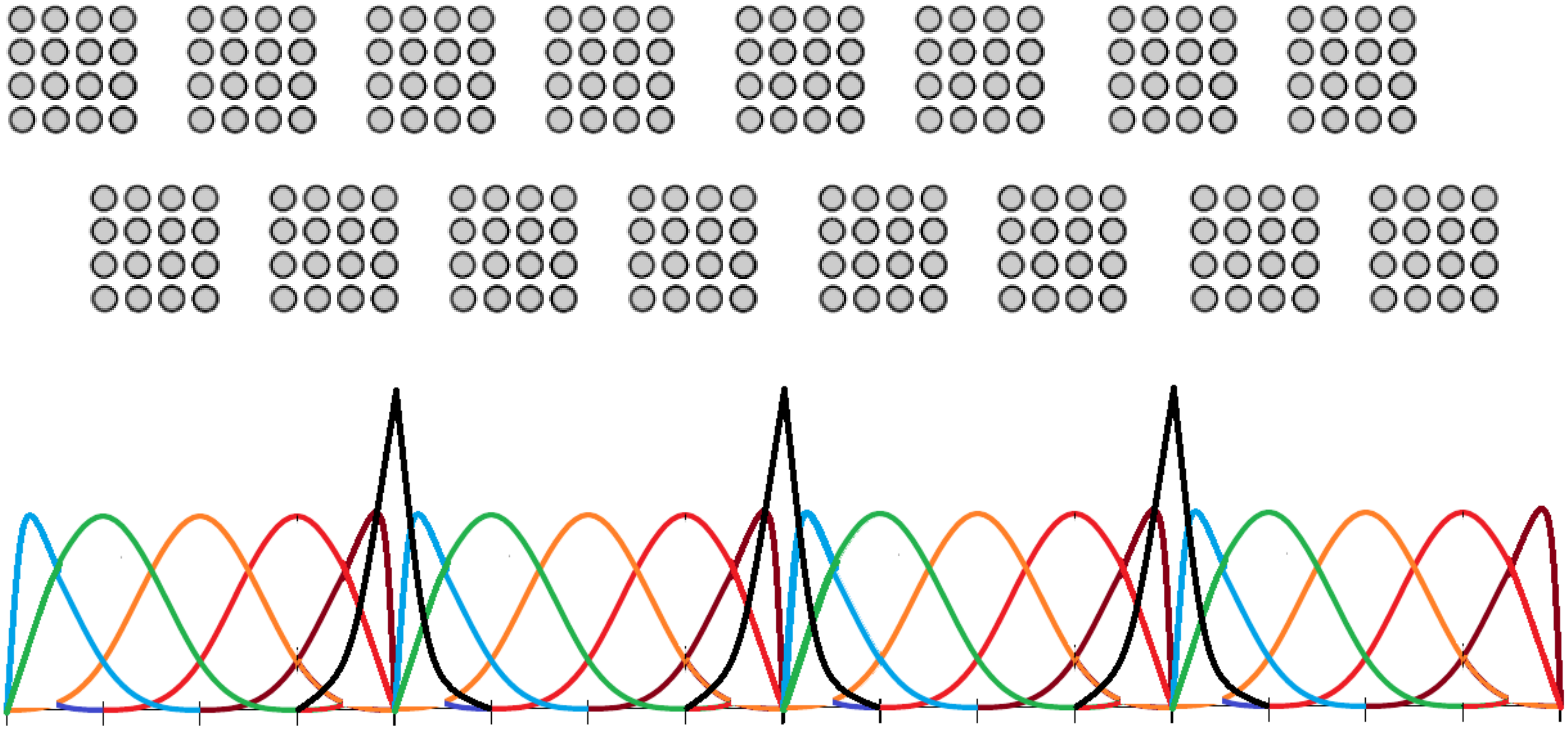
Large size N=49
(=3*16+1)

Sparse diagonals



Element matrices overlap in minimal way

refined Isogeometric Analysis (**rIGA**)



Compromise between both methods
16 elements with cubic B-splines
additional C^0 separators included every four elements

refined Isogeometric Analysis (**rIGA**)

16 element frontal matrices

Size of each element matrix 4x4

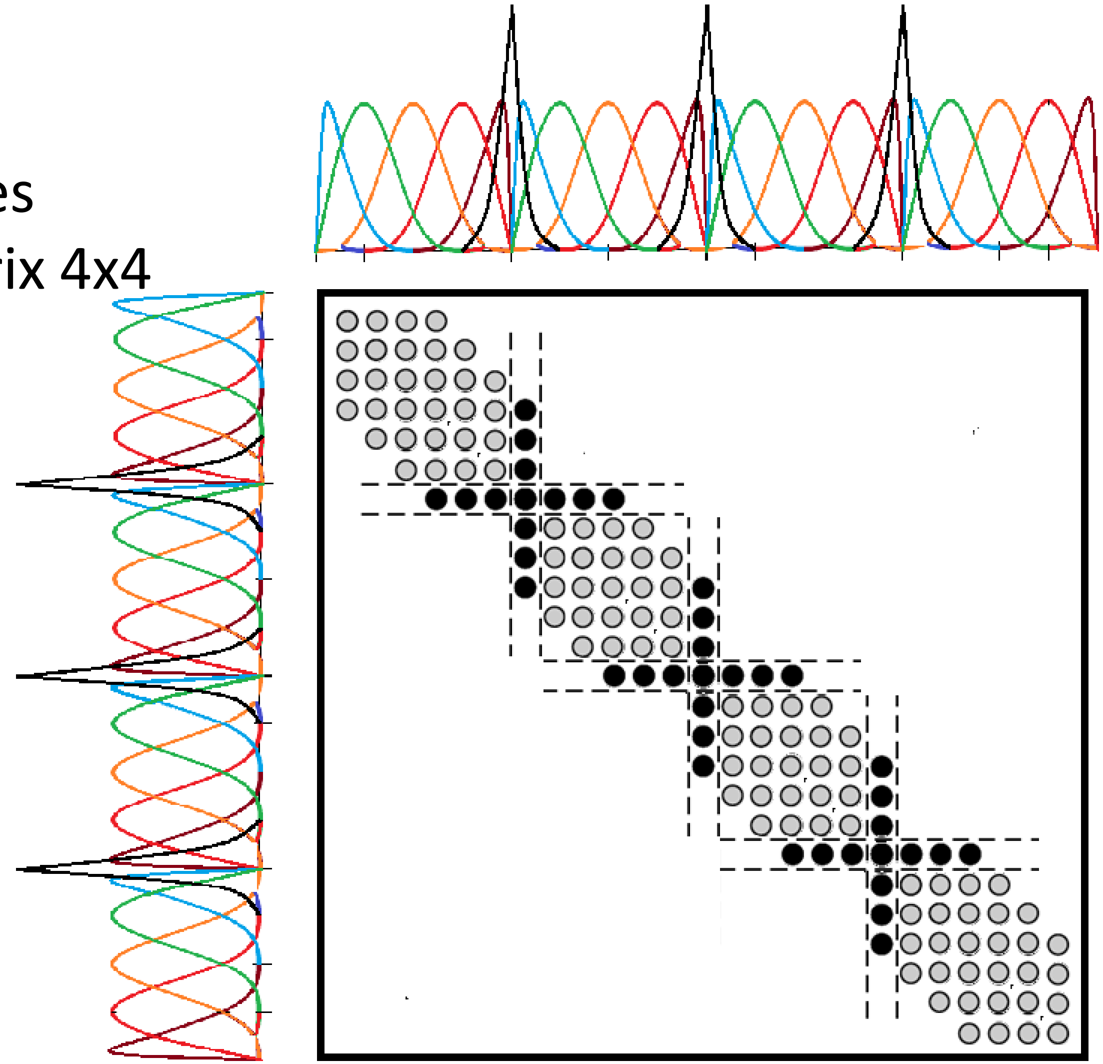
assembled into

Global matrix:

Medium size N=25

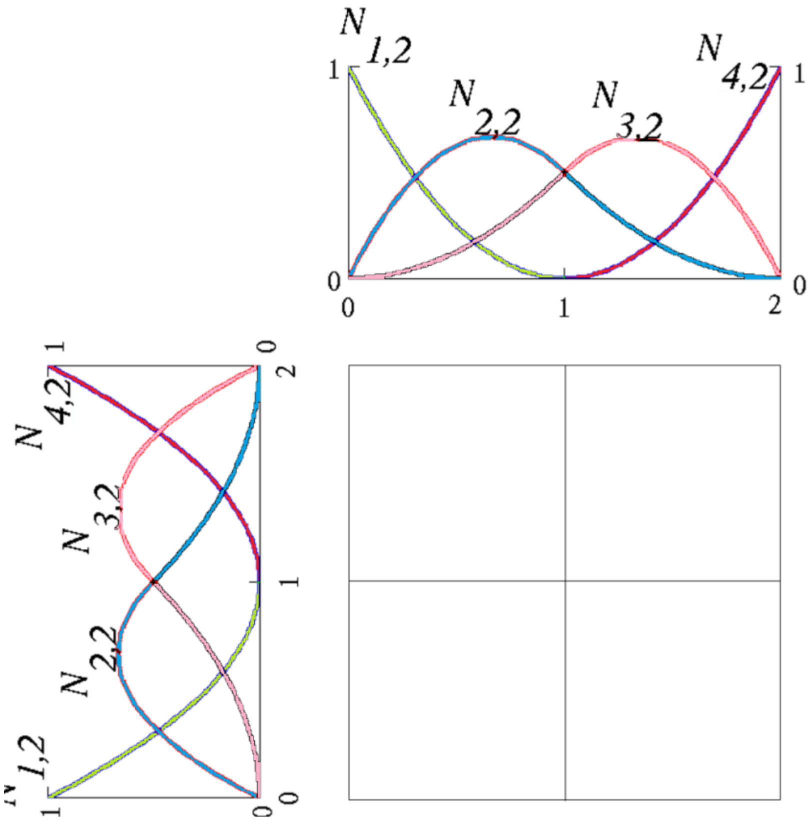
(=4*(4+2)+1)

Medium sparse diagonals

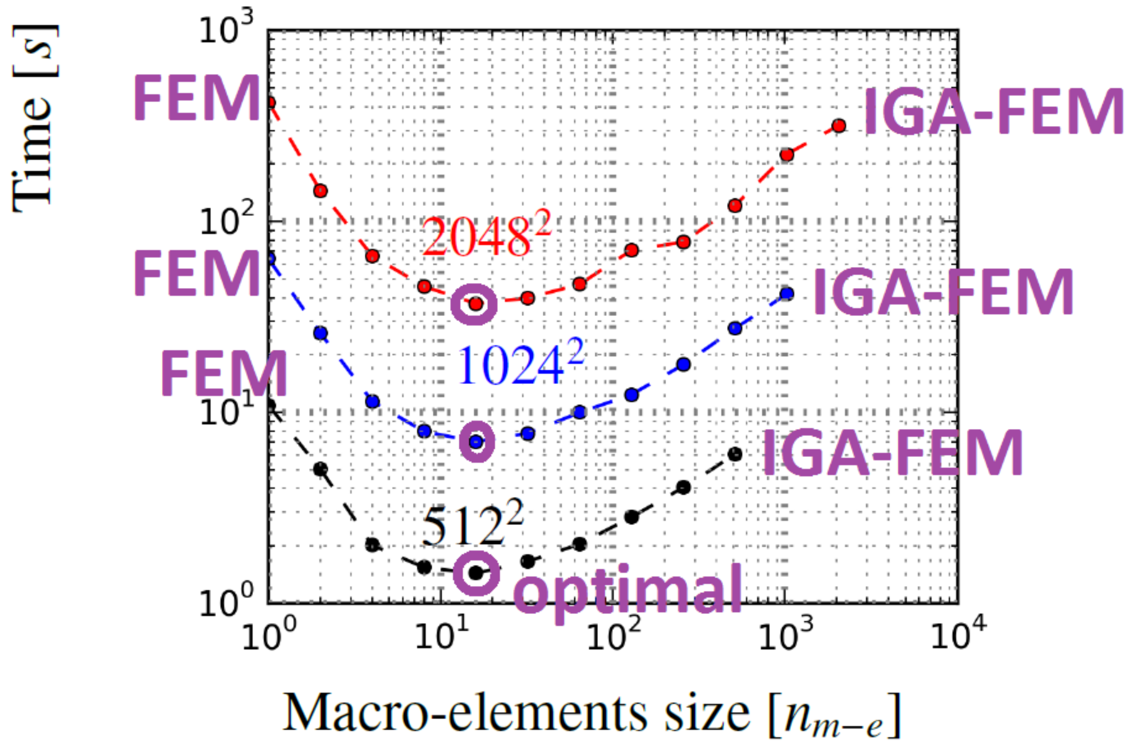


2D IGA-FEM

2D uniform mesh with basis functions = tensor products of B-splines



rIGA sequential 2D

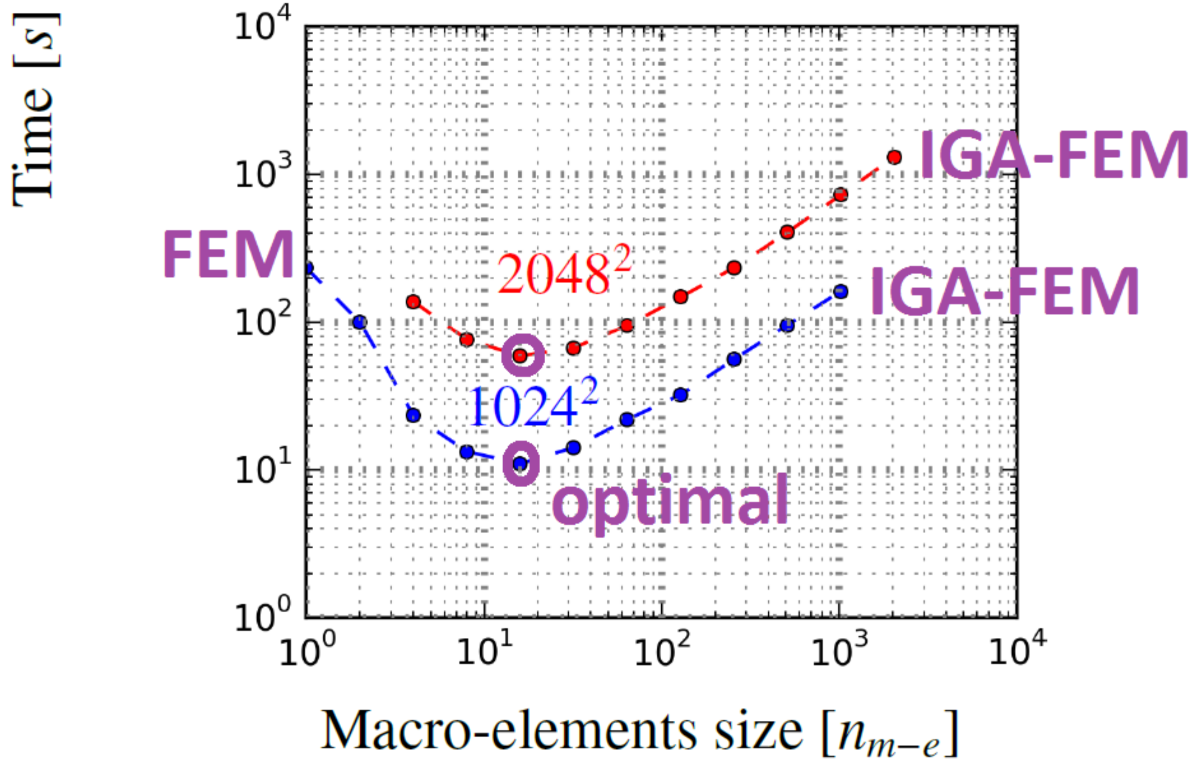


(b) Polynomial order $p = 3$

rIGA with optimal size of macro elements (16 in this case) cubic B-splines is one order of magnitude faster than FEM and IGA-FEM

Daniel Garcia, David Pardo, Lisandro Dalcin, Maciej Paszynski, Victor M. Calo, **Refined Isogeometric Analysis (rIGA): Fast Direct Solvers by Controlling Continuity**, submitted to *Computer Methods in Applied Mechanics and Engineering,* 2016

rIGA sequential 2D



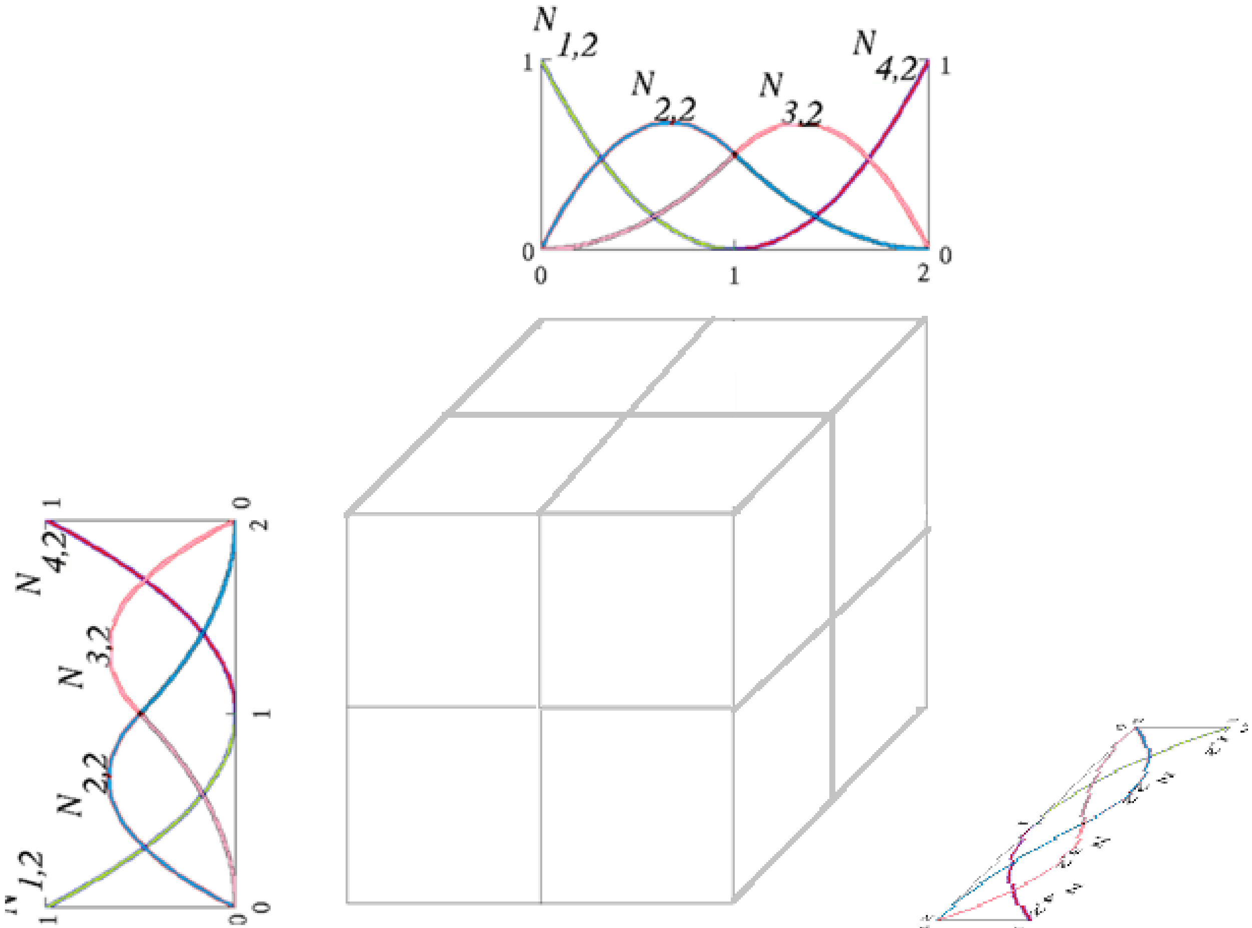
(d) Polynomial order $p = 5$

rIGA with optimal size of macro elements (16 in this case) cubic B-splines is one order of magnitude faster than FEM and IGA-FEM

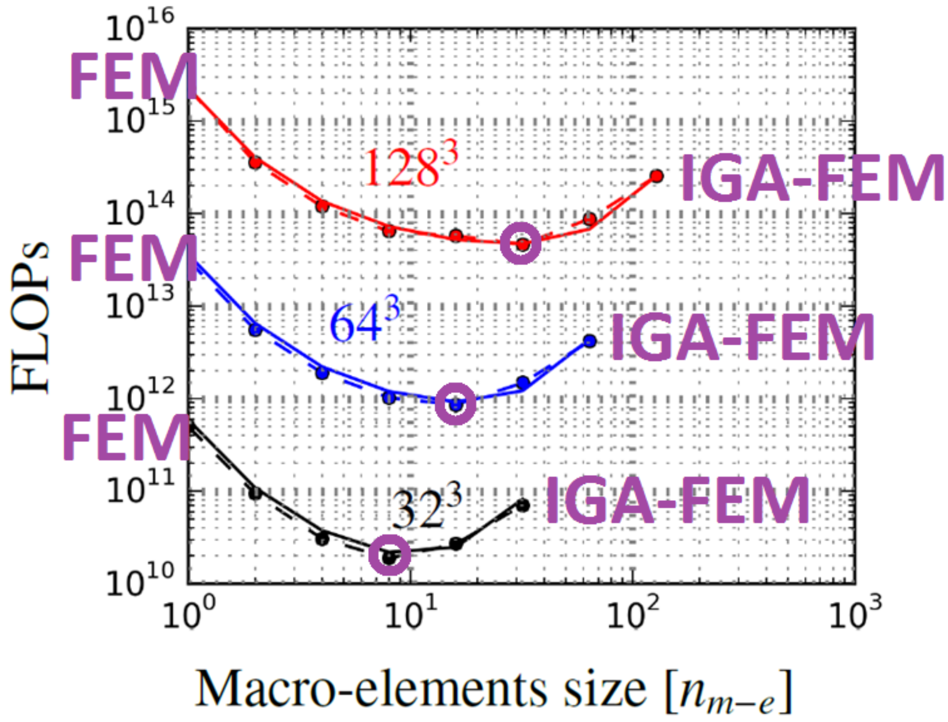
Daniel Garcia, David Pardo, Lisandro Dalcin, Maciej Paszynski, Victor M. Calo, **Refined Isogeometric Analysis (rIGA): Fast Direct Solvers by Controlling Continuity**, submitted to *Computer Methods in Applied Mechanics and Engineering,* 2016 (IF:3,456)

3D IGA-FEM

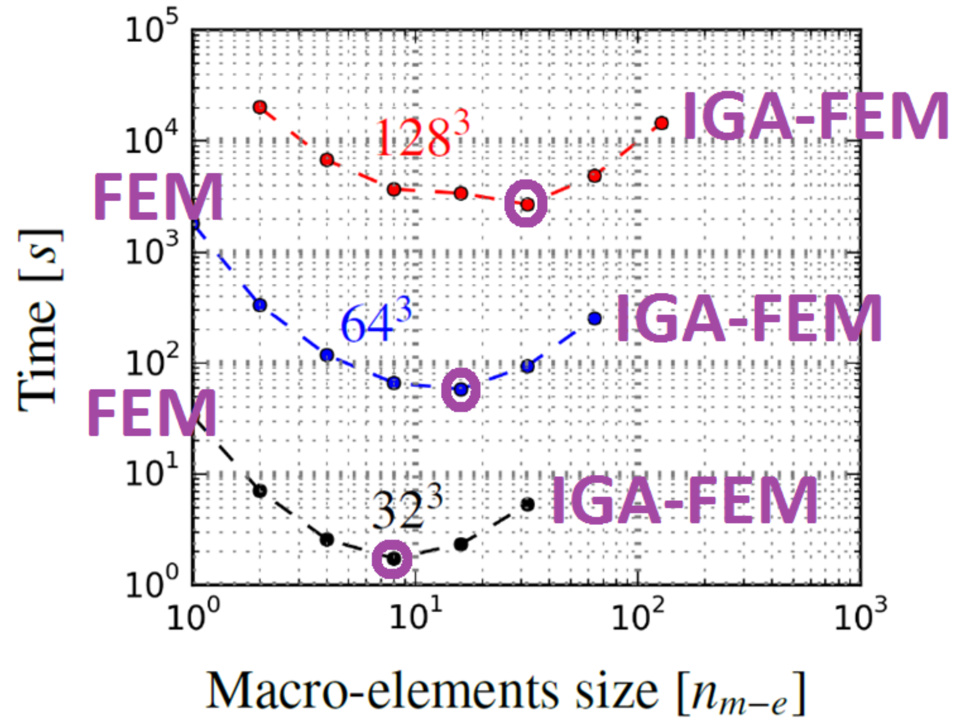
3D uniform mesh with basis functions = tensor products of B-splines



3D sequential **rIGA** with quadratic B-splines



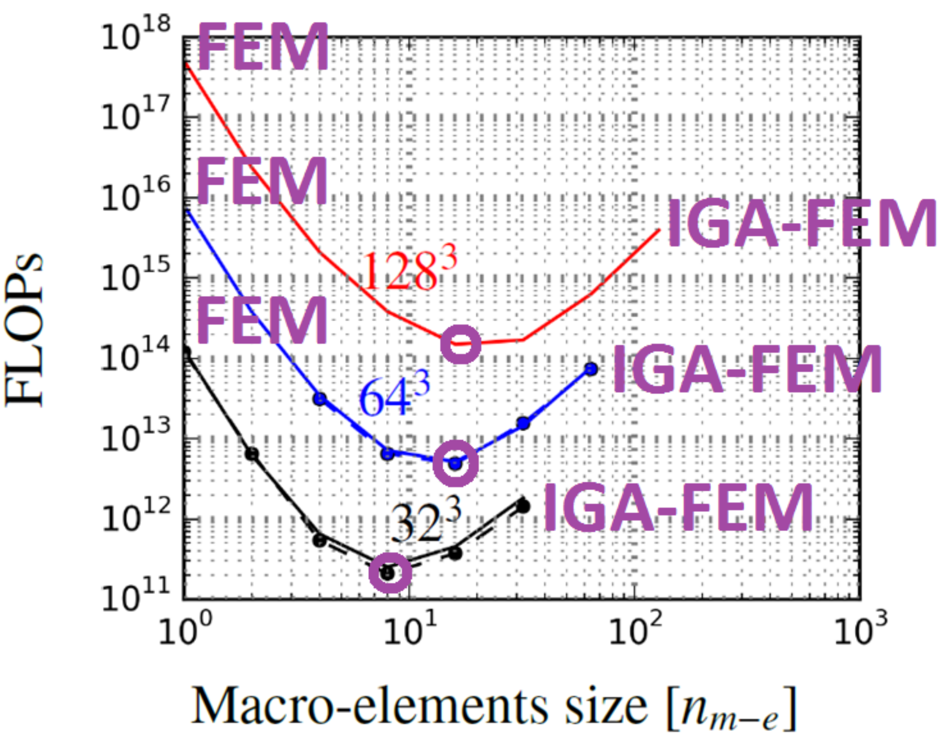
(a) Polynomial order $p = 2$



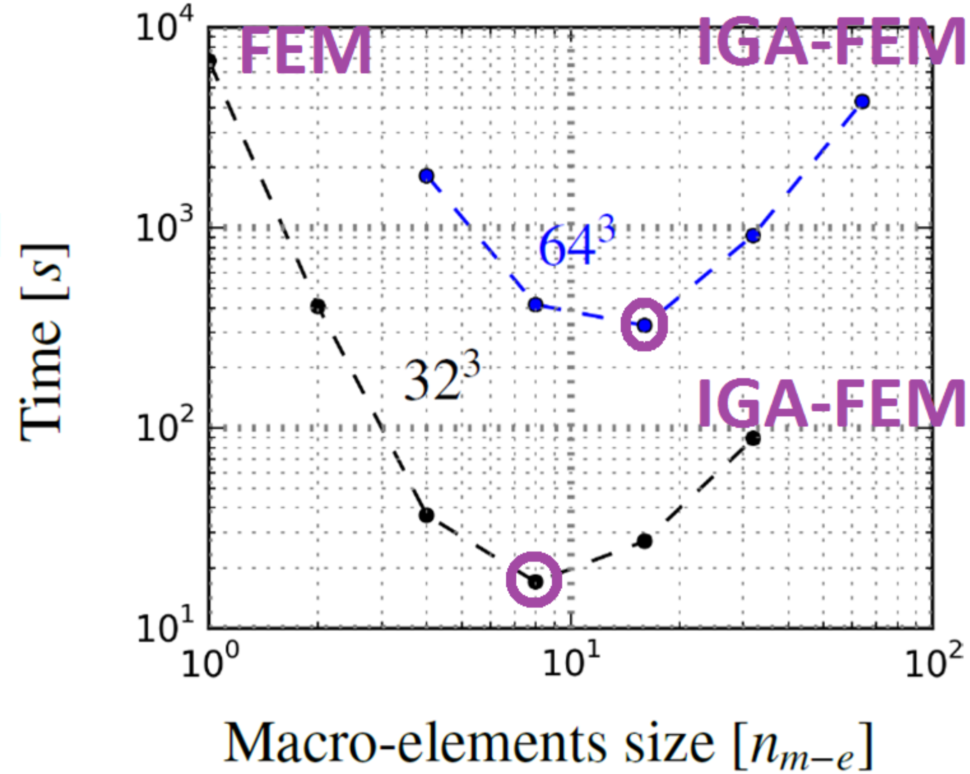
(a) Polynomial order $p = 2$

Around 15 times faster than FEM
and 4 times faster than IGA-FEM
optimal number of separators varies with mesh size (8, 16 or 32)

3D sequential **rIGA** with quintic B-splines



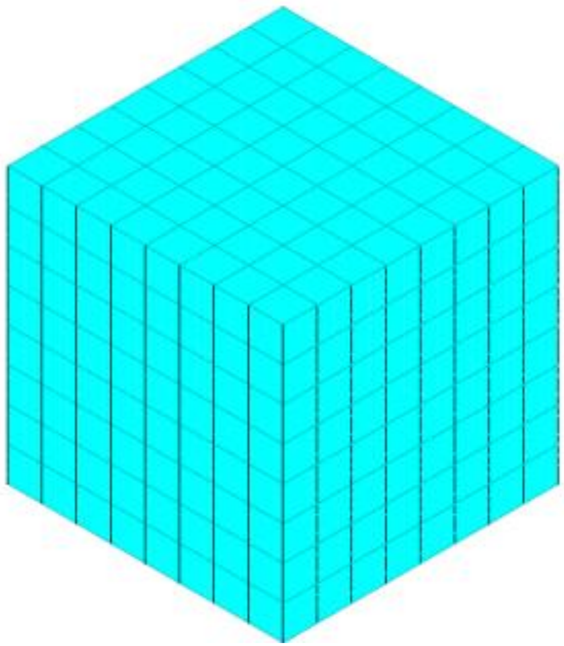
(d) Polynomial order $p = 5$



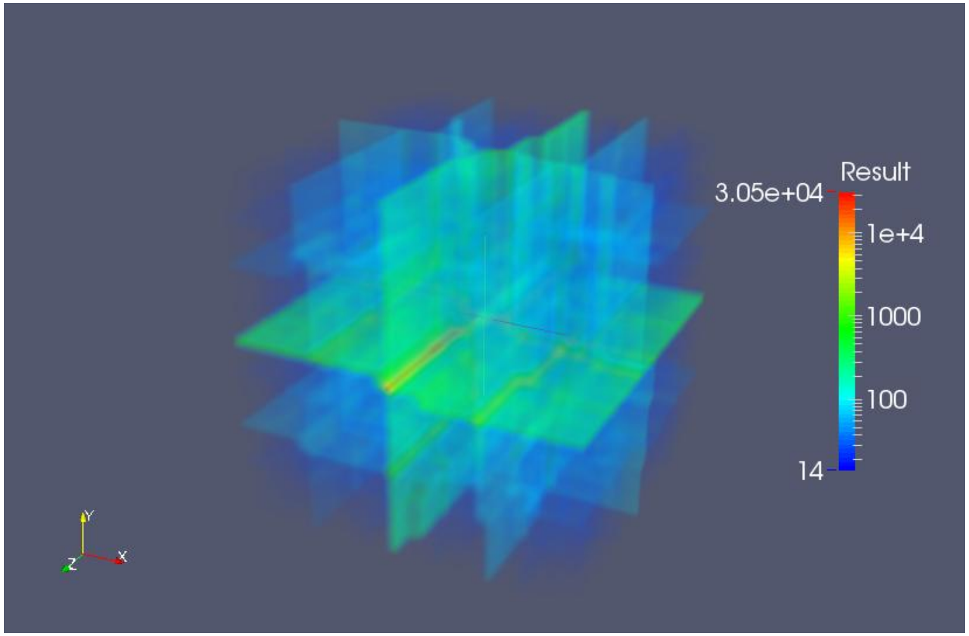
(d) Polynomial order $p = 5$

Over two orders of magnitude times faster than FEM
One order of magnitude faster than IGA-FEM
optimal number of separators varies with mesh size (8 or 16)

Automatic selection of macro-elements size



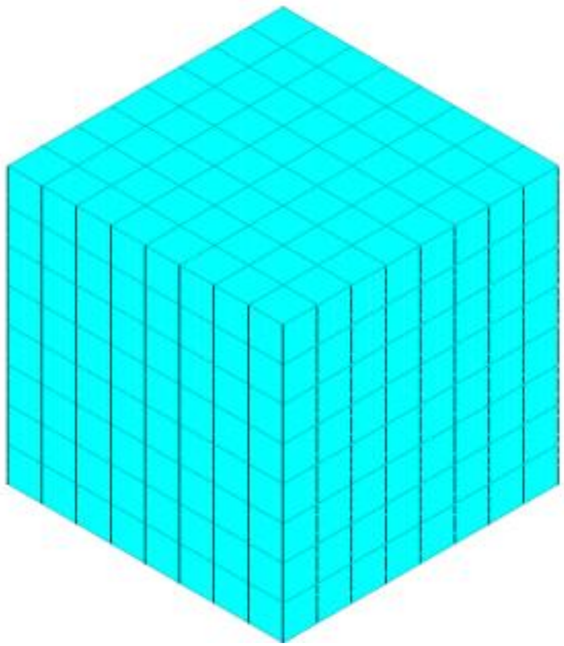
p=1



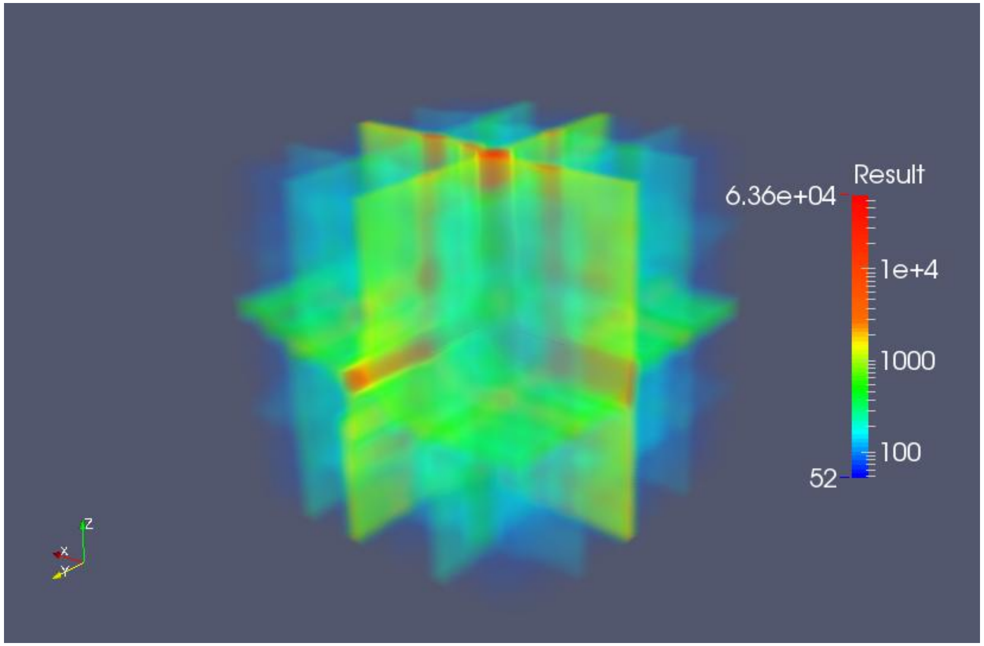
It is possible to estimate the cost (FLOPS per node) without formulation of the global matrix (we do not have the matrix assembled yet!)

Maciej Paszyński, *Fast solvers for mesh-based computations*, Taylor & Francis, CRC Press 2016

Automatic selection of macro-elements size



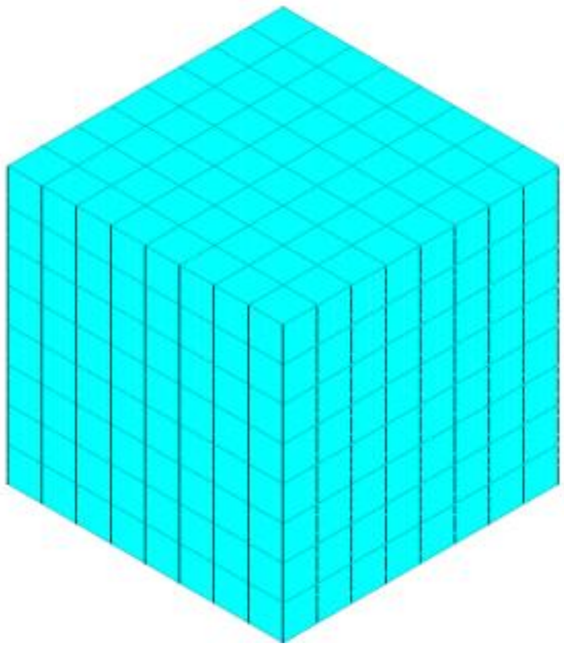
p=2



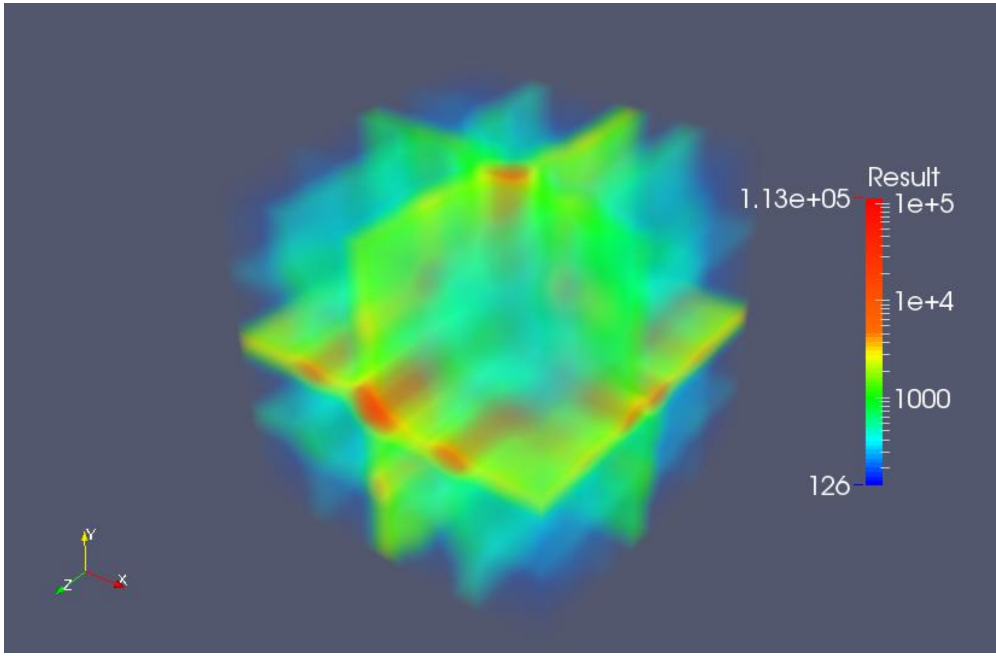
It is possible to estimate the cost (FLOPS per node) without formulation of the global matrix (we do not have the matrix assembled yet!)

Maciej Paszyński, *Fast solvers for mesh-based computations*,
Taylor & Francis, CRC Press 2016

Automatic selection of macro-elements size



p=3



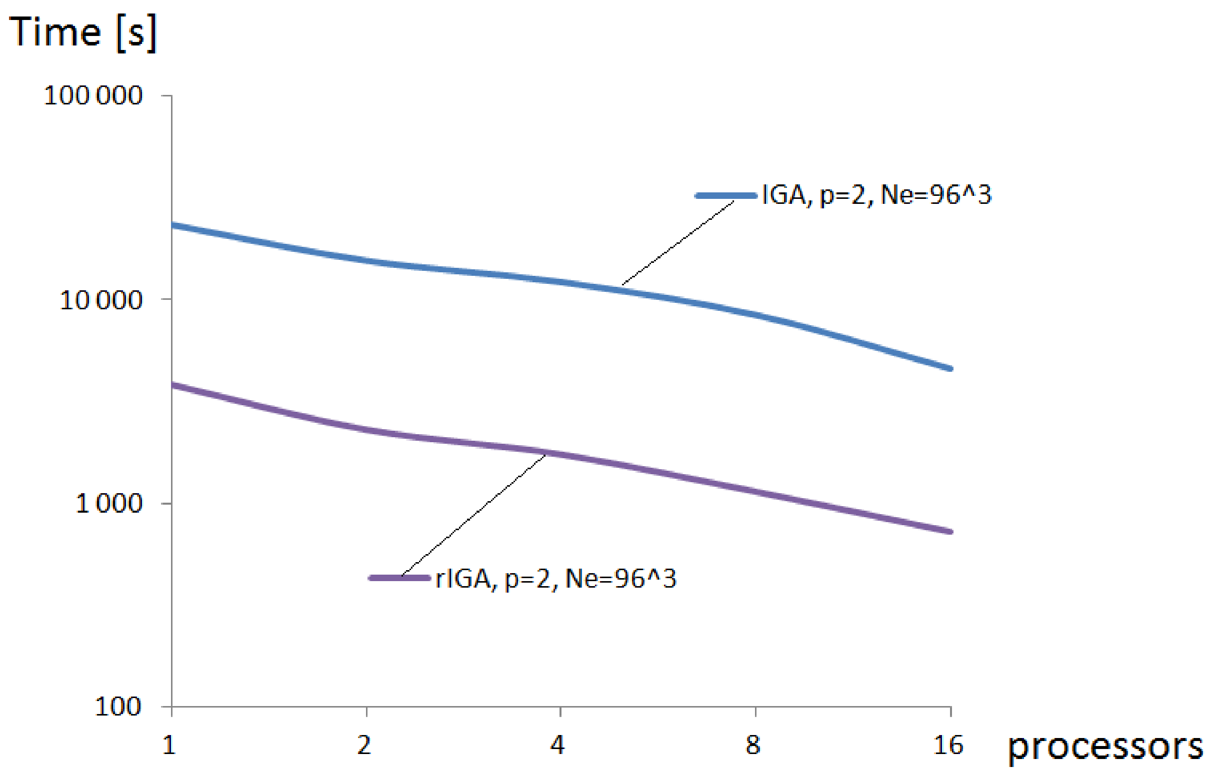
It is possible to estimate the cost (FLOPS per node) without formulation of the global matrix (we do not have the matrix assembled yet!)

Maciej Paszyński, *Fast solvers for mesh-based computations*, Taylor & Francis, CRC Press 2016

Parallel computations

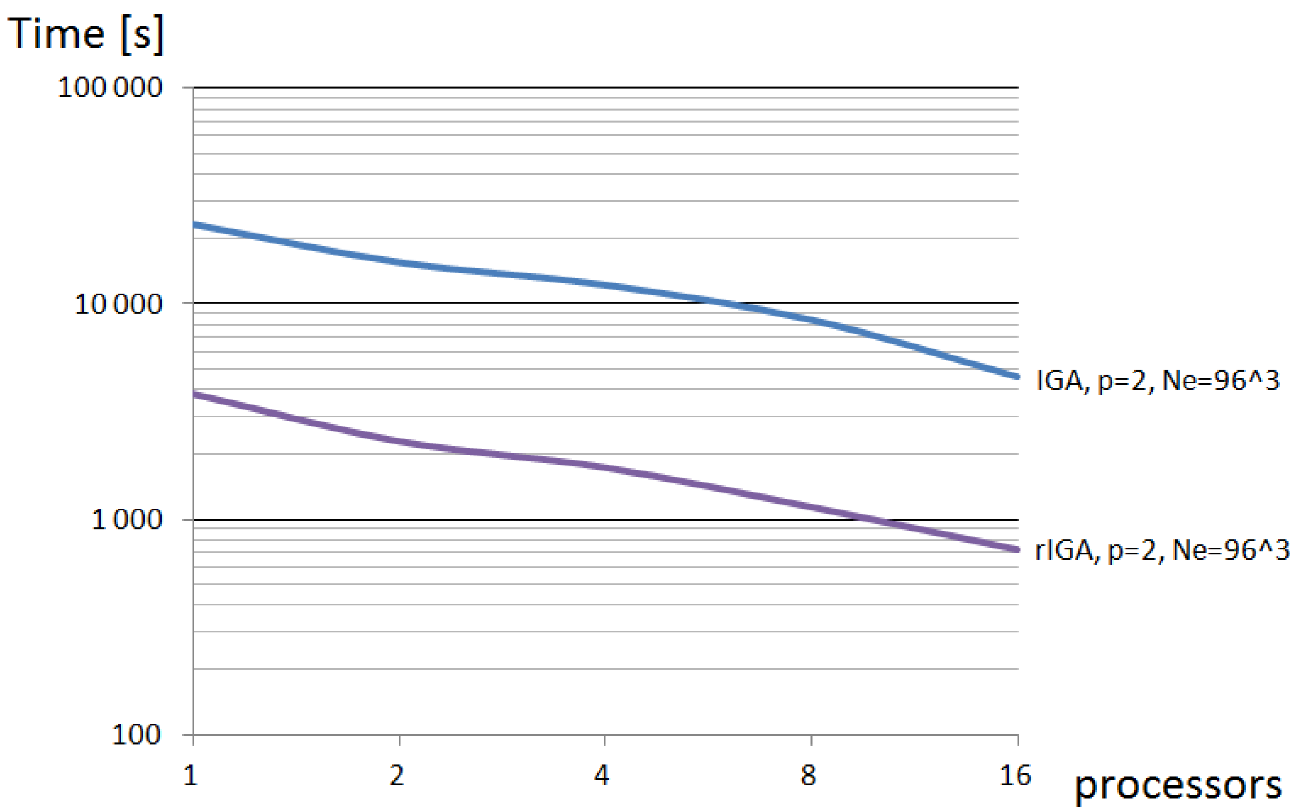
We select optimal separator and go for parallel solver
3D IGA-FEM, quadratic B-splines, 96^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

MUMPS_5.0.1
lapack-3.5.0
scalapack-2.0.2
compilers/intel/16.0.2



Parallel computations

We select optimal separator and go for parallel solver
3D IGA-FEM, quadratic B-splines, 96^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

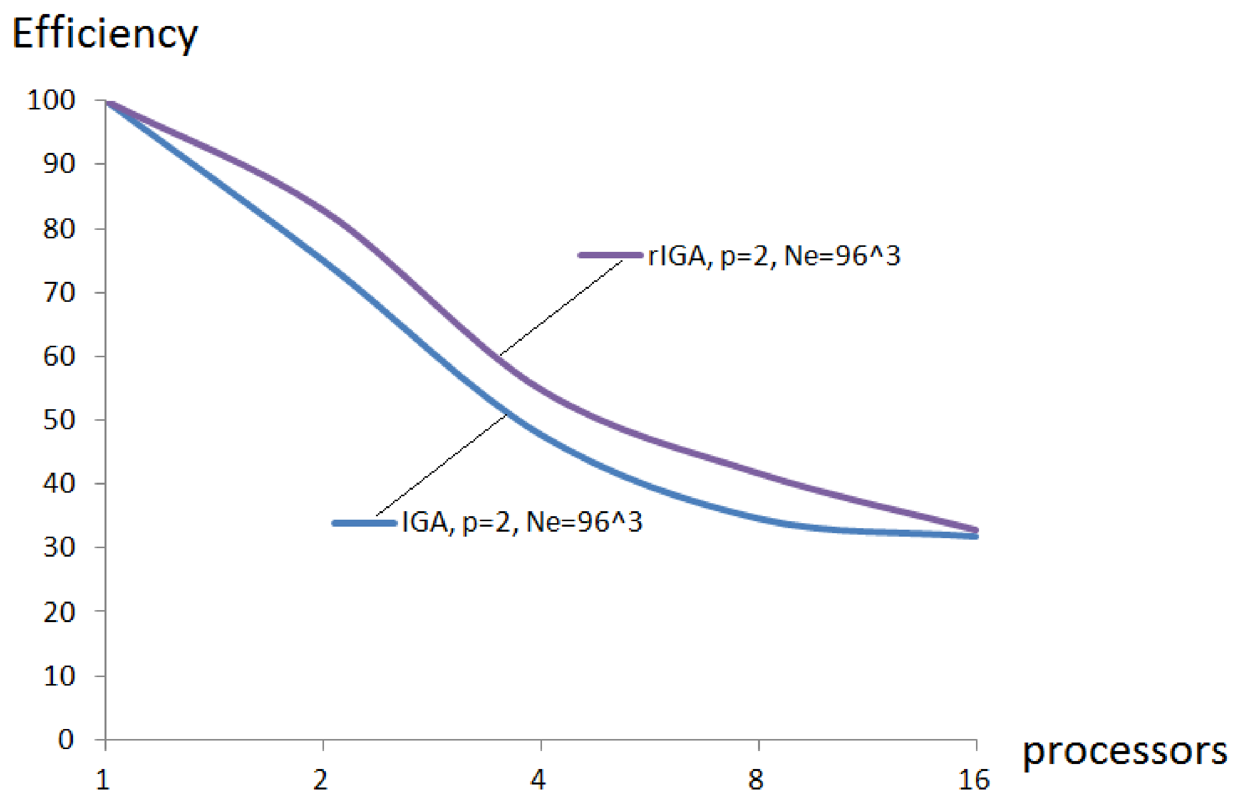


rIGA 7,5 times faster than IGA

Parallel computations

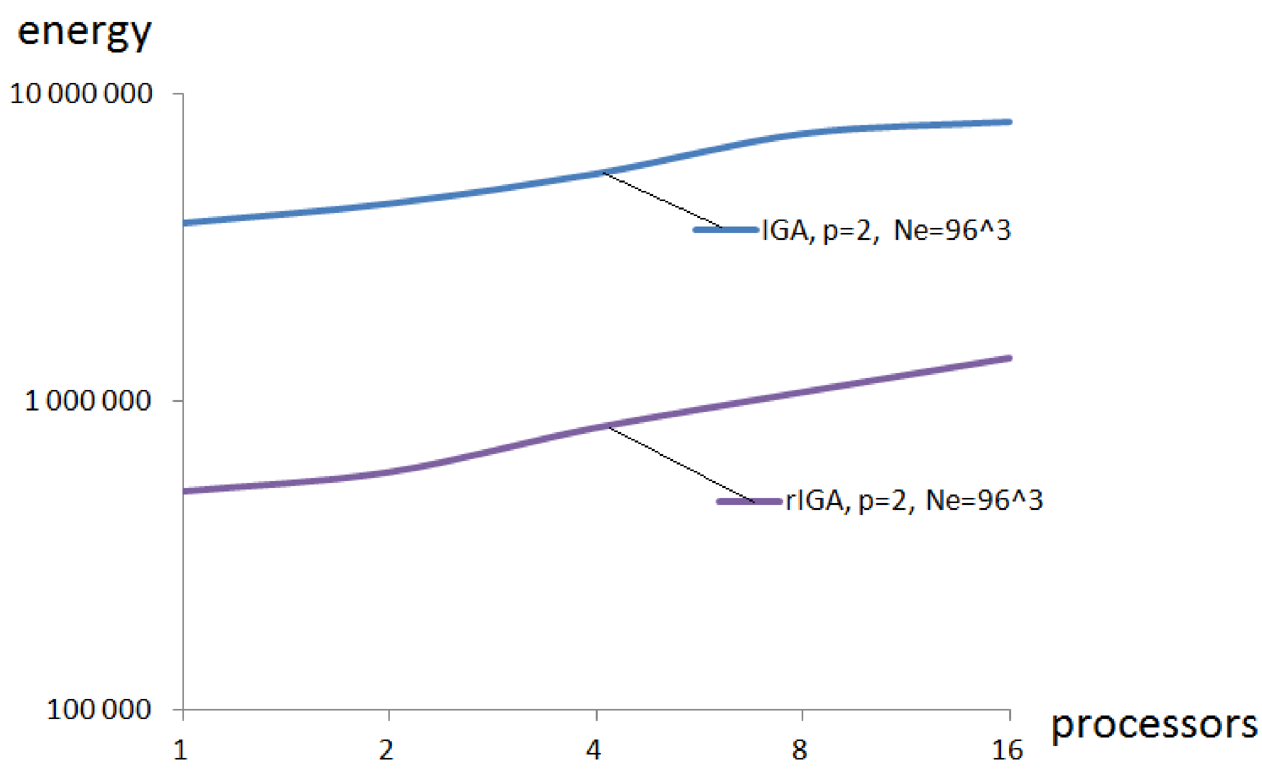
We select optimal separator and go for parallel solver
3D IGA-FEM, quadratic B-splines, 96^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

E=T1/(p*Tp)*100 %



Parallel computations

We select optimal separator and go for parallel solver
3D IGA-FEM, quadratic B-splines, 96^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

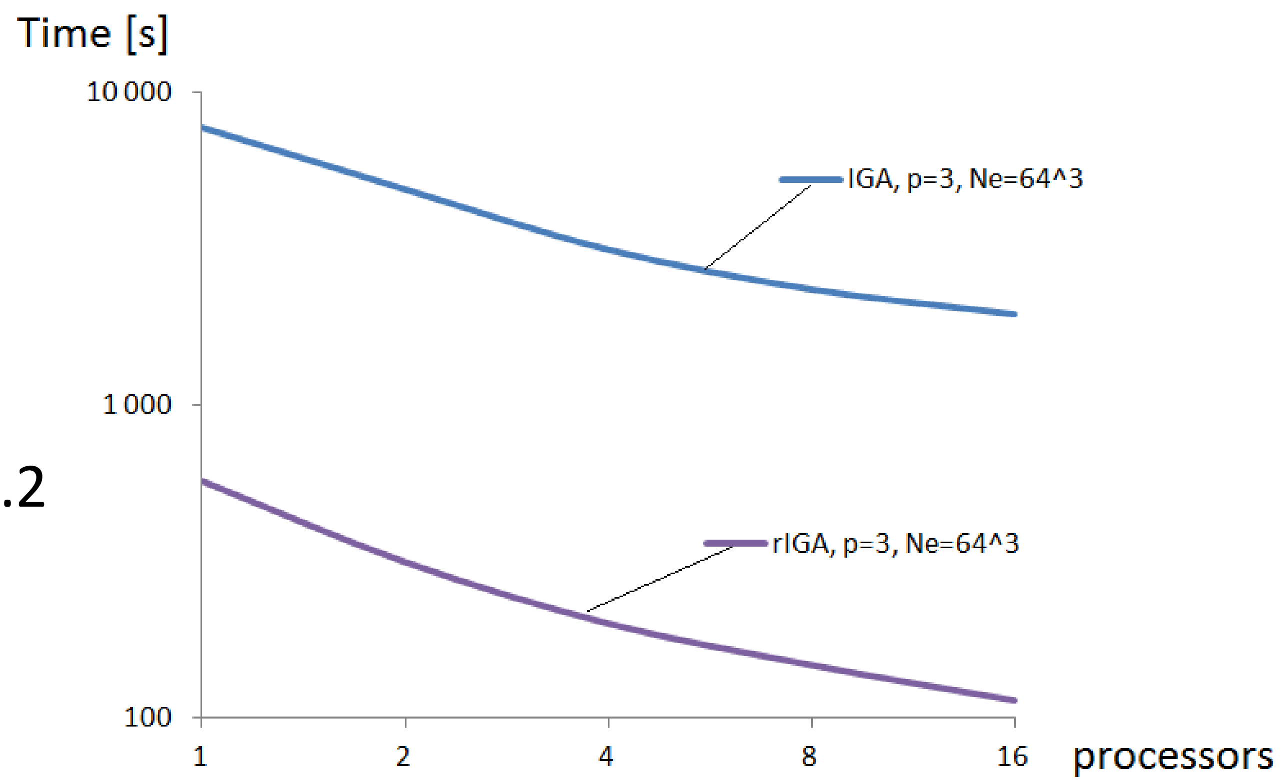


One order of magnitude lower total energy consumption

Parallel computations

We select optimal separator and go for parallel solver
3D IGA-FEM, cubic B-splines, 64^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

MUMPS_5.0.1
lapack-3.5.0
scalapack-2.0.2
compilers/intel/16.0.2

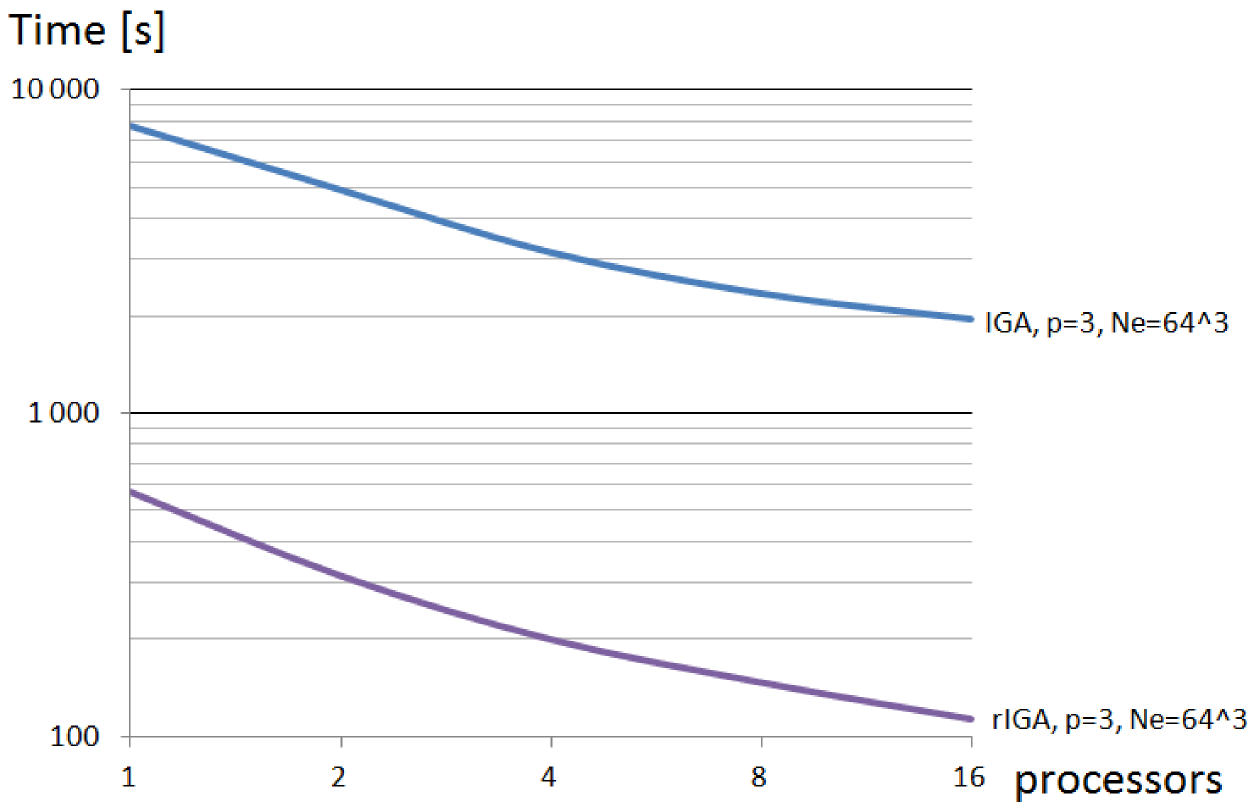


Parallel computations

We select optimal separator and go for parallel solver

3D IGA-FEM, cubic B-splines, 64^3 elements

PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

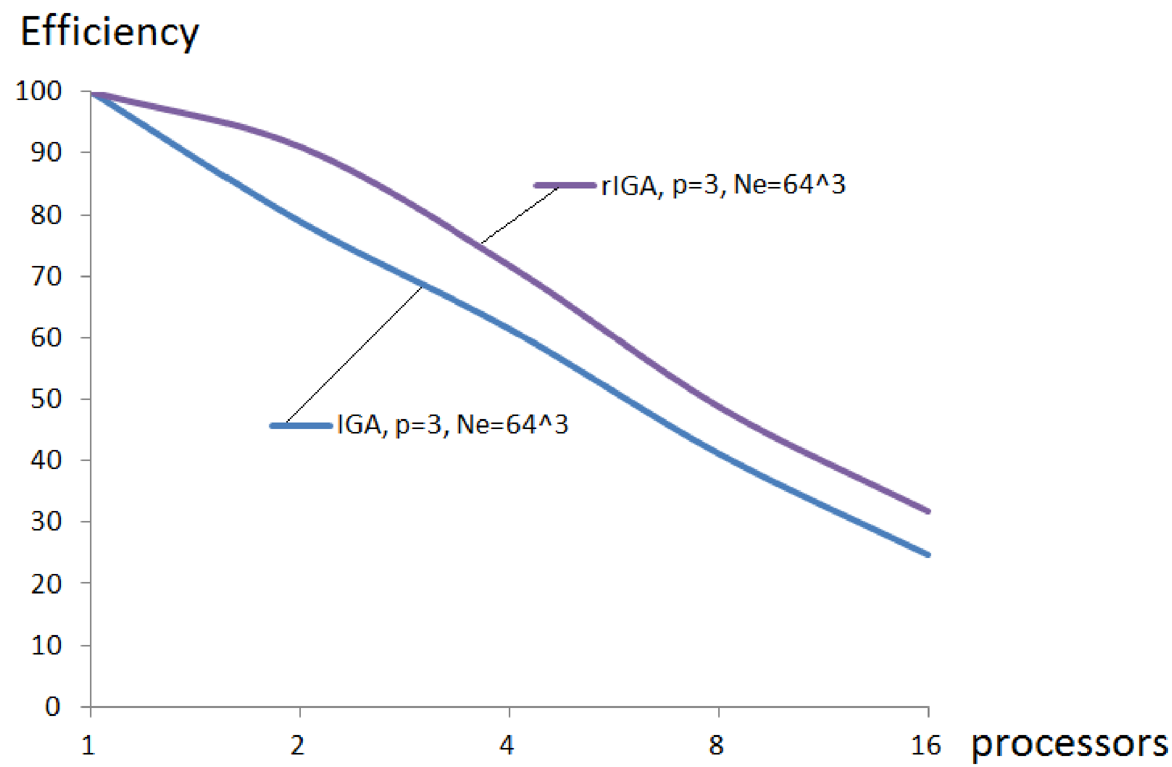


rIGA 11 times faster than IGA

Parallel computations

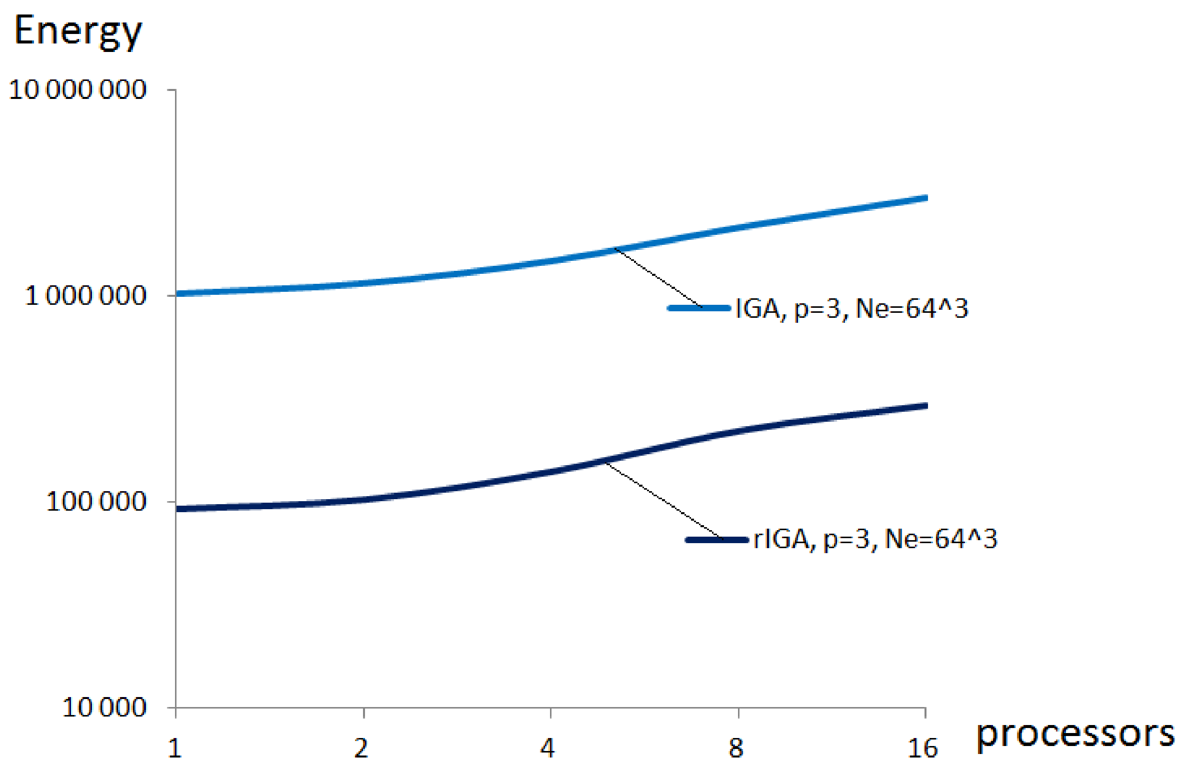
We select optimal separator and go for parallel solver
3D IGA-FEM, cubic B-splines, 64^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

E=T1/(p*Tp)*100 %



Parallel computations

We select optimal separator and go for parallel solver
3D IGA-FEM, cubic B-splines, 64^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

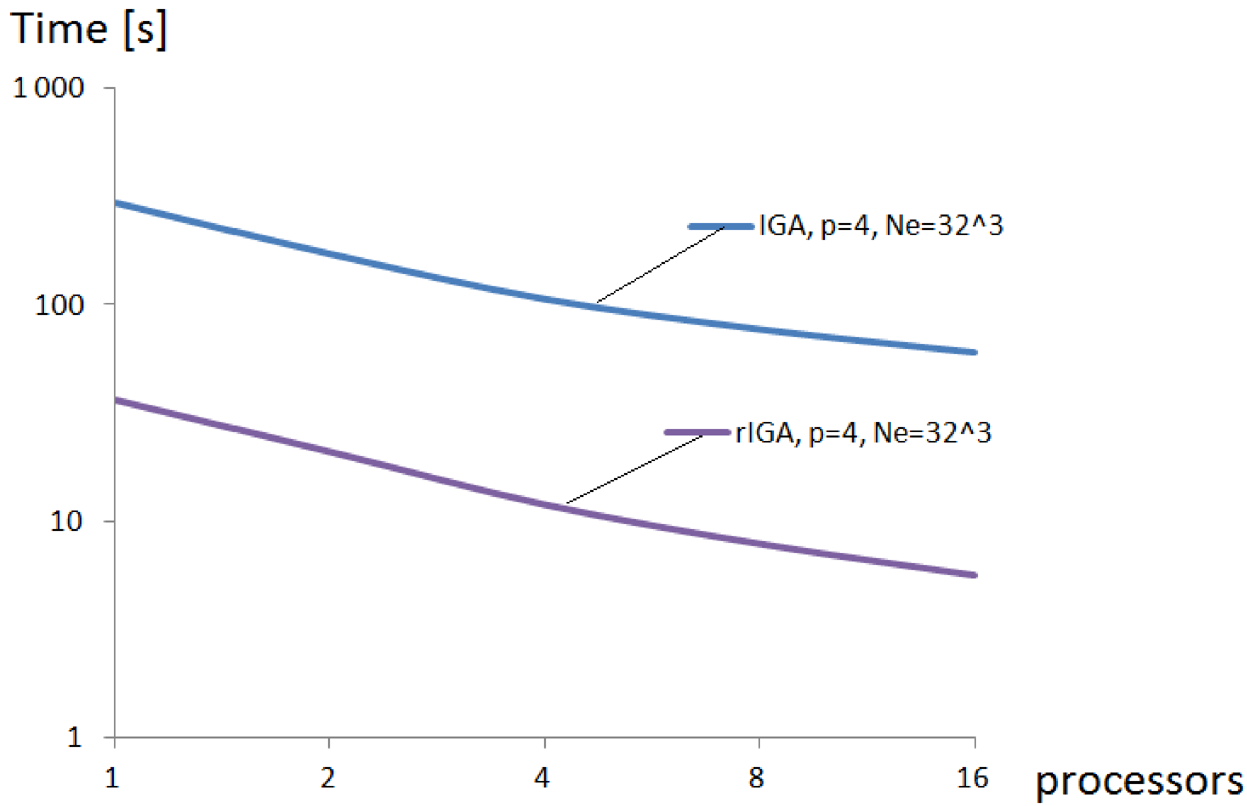


One order of magnitude lower total energy consumption

Parallel computations

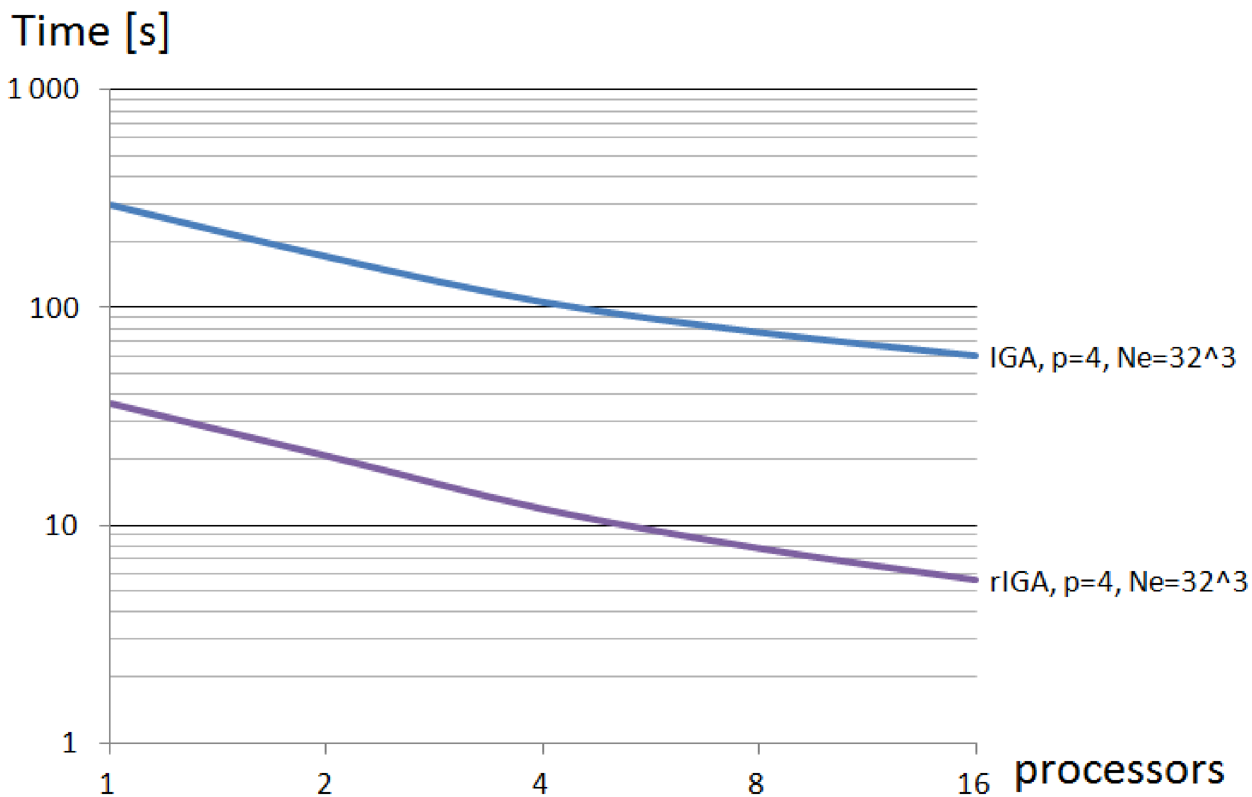
We select optimal separator and go for parallel solver
3D IGA-FEM, quartic B-splines, 32^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

MUMPS_5.0.1
lapack-3.5.0
scalapack-2.0.2
compilers/intel/16.0.2



Parallel computations

We select optimal separator and go for parallel solver
3D IGA-FEM, quartic B-splines, 32^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

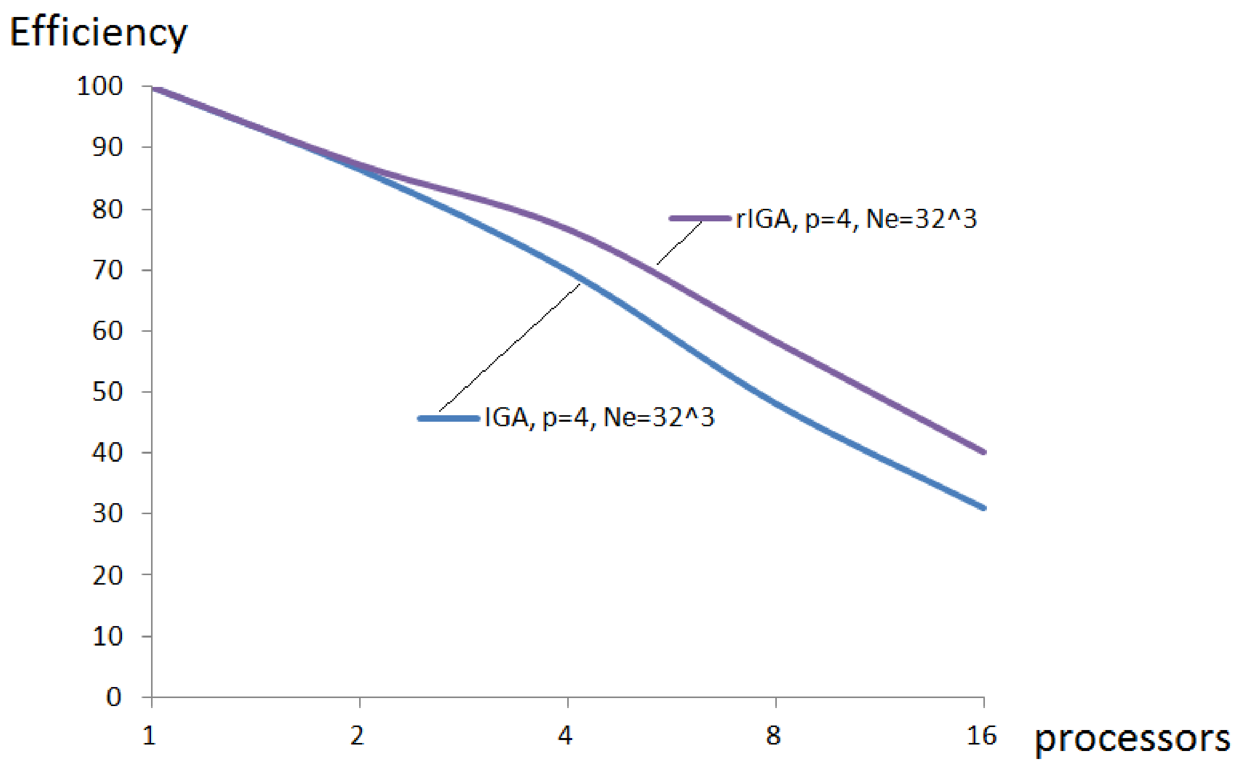


rIGA is 8 times faster than IGA

Parallel computations

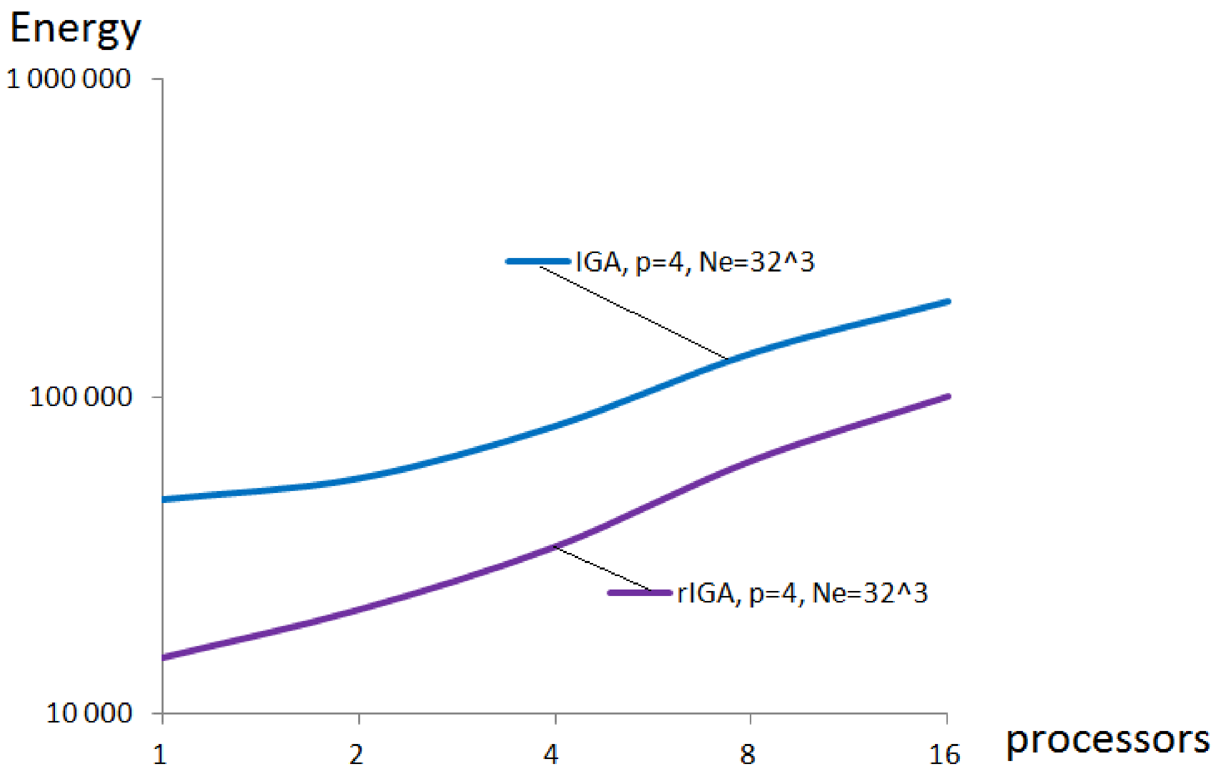
We select optimal separator and go for parallel solver
3D IGA-FEM, quartic B-splines, 32^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM

E=T1/(p*Tp)*100 %



Parallel computations

We select optimal separator and go for parallel solver
3D IGA-FEM, quartic B-splines, 32^3 elements
PROMETHEUS 16 nodes @ 2,50 GHz, 128 GB RAM



3 times lower total energy consumption

Conclusions

In 1D/2D/3D Finite Element Method computations
it is possible to **refine basis functions** over the computational mesh
in such a way that

- the **topology of the mesh does not change**
- **accuracy** of the numerical approximation **is similar**
- computational cost of both direct and parallel solvers
  is reduced up to two orders of magnitude
- **effciency** of parallel solver is better

We believe these features are **solver independent**
since we have changed the properties of the matrix