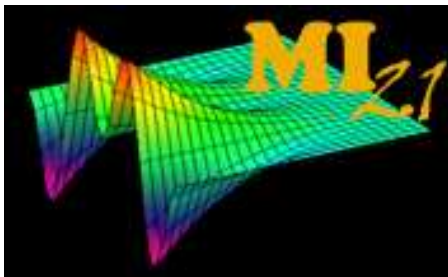


Pokročilé programovací techniky v Matlabu

T. Kozubek



Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB-Technická univerzita Ostrava



Obsah

1. Matlab

1. Funkce, řízení toku
2. Homogenní pole (číselné matice, řetězce)
3. Struktury
4. Pole buněk
5. OOP
6. GUI
7. Import, export, MEX

2. Fourierova řada (spektrální analýza)

3. Diskrétní Fourierova transformace (spektrální analýza)

4. Konvoluce (vyhlazování signálu)

5. Laplaceova transformace (řešení DR)

6. Úlohy mechaniky – MatSol, paralelní programování

Matlab – funkce

```
function [mean_x,std_x,var_x]=statistics(x)
% STATISTICS Spocte zakladni statistiky vstupniho vektoru x
% UZITI: [mean,std,var]=statistics(x)

% Test na pocet vstupnich argumentu
if nargin~=1; error('Spatny pocet vstupnich argumentu!'); end;

% Test na korektnost vstupnich argumentu (x je ciselny vektor)
if ~isvector(x); error('Vstupni argument x neni vektor!'); end;
if ~isnumeric(x); error('Vstupni argument x neni ciselny vektor!'); end;

% Spocti zakladni statistiky
mean_x=mean(x); std_x=std(x); var_x=var(x);

end

% Podfunkce MEAN (upraveny vypocet stredni hodnoty)
function f=mean(x)
% Vypocet stredni hodnoty
f=sum(x)/(length(x)+1);
end
```

Matlab – homogenní pole

Vytvoření matice

```
A=[1 2 3  
  2 3 4  
  5 6 7];
```

```
B=[0,0,0; 2,3,4; 5,6,7];
```

```
C=[A,B; B,A]; S=['abc'; 'def']
```

Přístup k prvkům, rušení

```
A(1:2,3)=[1;3]
```

```
A(A<7)=0
```

```
A(2,:)=[]
```

Řídké matice

```
e=ones(5,1); A = spdiags([-e 2*e -e], -1:1, 5, 5)
```

```
B=sprand(5,4,0.1)
```

```
I=[1 1 2 3]; J=[1 3 2 4]; V=[1 1.5 2 3.7]; S=sparse(I,J,V,5,6)
```

```
spy(S,'o',8); imagesc(S); colorbar;
```

Matlab - struktury

- **Struktura** je heterogenní pole, jehož prvky jsou libovolné objekty – instance tříd (řetězce, číselné pole, buněčné pole, struktury, atd).
- **Přístup k položkám** je prostřednictvím názvů položek užitých v definici struktury.

```
>>student=struct('jmeno','Jan Madlo', 'predmet','LAM','znamky',[1 2 1]);
```

- **Paměťové požadavky:** není třeba souvislý blok paměti pro celou strukturu, ale pouze pro položky struktury.
- **Využití:** snížení počtu vstupních a výstupních argumentů funkce, zvýšení čitelnosti kódu.

- **Přístup k položkám**

```
>>student.jmeno
```

```
>>student.jmeno='Miro Vilik'
```

Matlab – pole buněk

- **Pole buněk** je heterogenní pole, jehož prvky jsou libovolné objekty – instance tříd (řetězce, číselné pole, buněčné pole, struktury, atd).
- **Přístup k položkám** je prostřednictvím indexů buněk.

```
>>c={'Hello',rand(3),student};
```

- **Paměťové požadavky:** není třeba souvislý blok paměti pro celé pole, ale pouze pro jeho položky.
- **Využití:** uložení řetězců různé velikosti, uložení matic různého řádu u metody rozložení oblastí, nahrazení vstupních a výstupních argumentů funkce, zvýšení čitelnosti kódu.

- **Přístup k položkám**

```
>>c{1}, c{1}(2:end), c{3}.jmeno
```

```
>>c{1}='Ahoj', c{3}.jmeno='Ivan Leden'
```

Matlab – OOP

MATLAB má zabudovánu kompletní podporu OO přístupu

- Jak systém roste, je čím dál složitější sledovat předávání dat mezi funkcemi.
- Opakování kódu copy&paste – redundance – nekonzistence.
- Deklarace třídních proměnných – předem se ví, jaké má proměnné.
- Srozumitelnost a přirozenost návrhu.

Matlab – OOP

```
classdef MyClass < MySuperClass
    properties
        Property1
        Property2 = sin(pi/12); % default value
    end
    properties (SetAccess = private, GetAccess = private)
        Stress
        Strain
    end
    methods
        function value = get.Property1(obj)
            %optionally implement get value
        end
        function obj = set.Property1(obj,value)
            %optionally implement set value
        end
    end
end
```

Dědičnost, přetěžování operátorů, konstruktory, proměnné s omezeným přístupem.

Matlab – GUI

1. Pomocí LAYOUT editoru (.fig, .m)

2. Pomocí Matlabovských příkazů

(figure, uicontrol, uibuttongroup, guidata, guidata, uihandles, uimenu, ...)

Matlab – import, export, MEX

1. Podpora práce s textovými i binárními soubory
2. Načtení obrázků, zvuků, videa, MS Excell, ...

Načtení a zobrazení obrázku

```
rgb = imread('hokej.jpg');  
image(rgb);
```

Načtení a přehrání zvukového souboru

```
[y,fs,nbits]=wavread('babycry.wav');  
sound(y,fs); plot(y);
```

MEX – přilinkování k Matlabu samostatných funkcí
i celých knihoven v C, C++, Fortranu, Javě

Lineární zobrazení

$$f(z) = az, \quad z \in \mathbb{C}, \quad a \in \mathbb{C}$$

$$f(z) = az = |a| e^{i \arg(a)} |z| e^{i \arg(z)} = |a| |z| e^{i(\arg(a) + \arg(z))}$$

Fourierova řada (spektrální analýza)

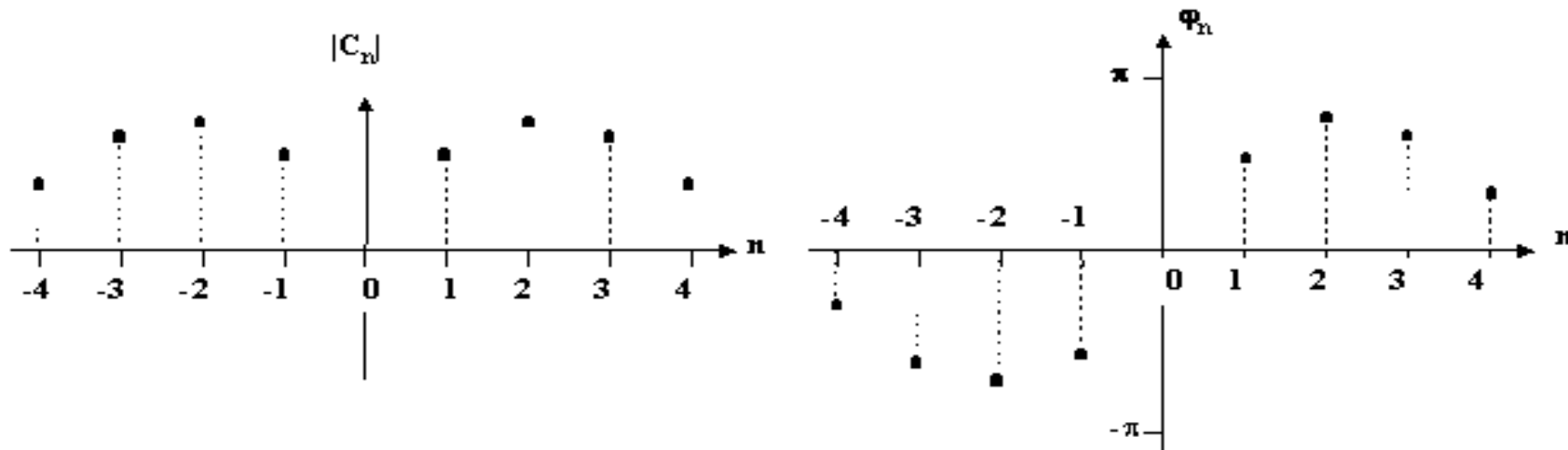
$\left\{ e^{i\omega n t} / \sqrt{T} \right\}_{n \in \mathbb{Z}} \dots$ ortonorm. systém fcí $L^2(0, T)$

$\omega = \frac{2\pi}{T} \dots$ úhlová rychlost, $T \dots$ perioda

$$f(t) \in L^2(0, T), \quad f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i\omega n t}, \quad t \in \mathbb{R}$$

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-i\omega n t} dt, \quad n \in \mathbb{Z},$$

- Dvoustranné spektrum : $\{ |c_n|, \varphi_n \}$



Diskrétní Fourierova transformace (spektr. analýza)

Fourierova řada

$$f(t) = \sum_{-\infty}^{+\infty} c_n e^{i\omega n t}, \quad t \in \mathbb{R}, \quad c_n = \frac{1}{T} \int_0^T f(t) e^{-i\omega n t} dt, \quad n \in \mathbb{Z},$$

$$c_n = \frac{1}{N\Delta t} \sum_{k=0}^{N-1} f(t_k) e^{-i\frac{2\pi}{N\Delta t}nk\Delta t} \Delta t = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-i\frac{2\pi}{N}nk}, \quad n = 0, \dots, N-1$$

$$f(t_k) = \sum_{n=0}^{N-1} c_n e^{i\frac{2\pi}{N\Delta t}nk\Delta t} = \sum_{n=0}^{N-1} c_n e^{i\frac{2\pi}{N}nk}, \quad k = 0, \dots, N-1$$

DFT

$$f_k = \sum_{n=0}^{N-1} c_n e^{i\frac{2\pi}{N}nk}, \quad k = 0, \dots, N-1, \quad c_n = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-i\frac{2\pi}{N}nk}, \quad n = 0, \dots, N-1$$

Konvoluce posloupností (vyhlazování signálu)

$$\{a_k\}_{k=0}^{\infty}, \{b_l\}_{l=0}^{\infty} \quad \{c_n\} = \{a_k\} * \{b_l\}, \quad \{c_n\}_{n=0}^{\infty}, \quad c_n = \sum_{k=0}^n a_{n-k} b_k$$

$$\{a_k\}_{k=0}^{M-1}, \{b_l\}_{l=0}^{N-1} \quad \{c_n\}_{n=0}^{N+M-1}, \quad \{c_n\} = \{a_k\} * \{b_l\}$$

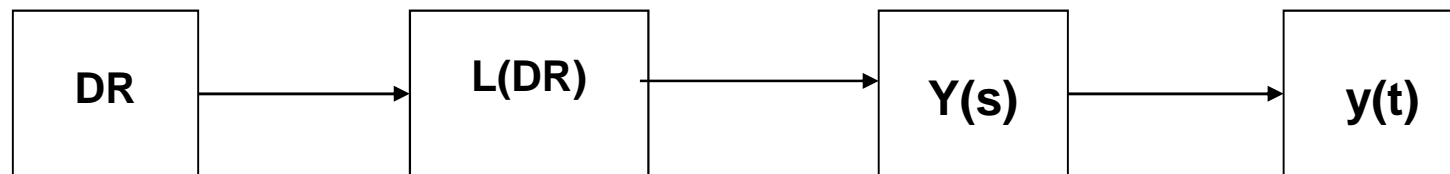
$$c = Ab = \begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ a_1 & a_0 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & a_1 & a_0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & a_{M-1} & \cdots & a_0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & a_{M-1} & a_{M-2} \\ 0 & 0 & 0 & \cdots & \cdots & 0 & a_{M-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{N-2} \\ b_{N-1} \end{bmatrix}$$

Laplaceova transformace (řešení DR)

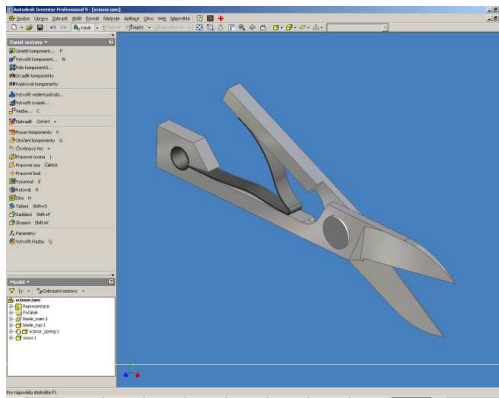
$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 y' + a_0 y = f(t),$$

$$y(t_0) = c_0, y'(t_0) = c_1, \dots; y^{(n-1)}(t_0) = c_{n-1}.$$

$$LT\{f\} = F(s) = \int_0^{\infty} f(t) \cdot e^{-st} dt,$$



modelování geometrie

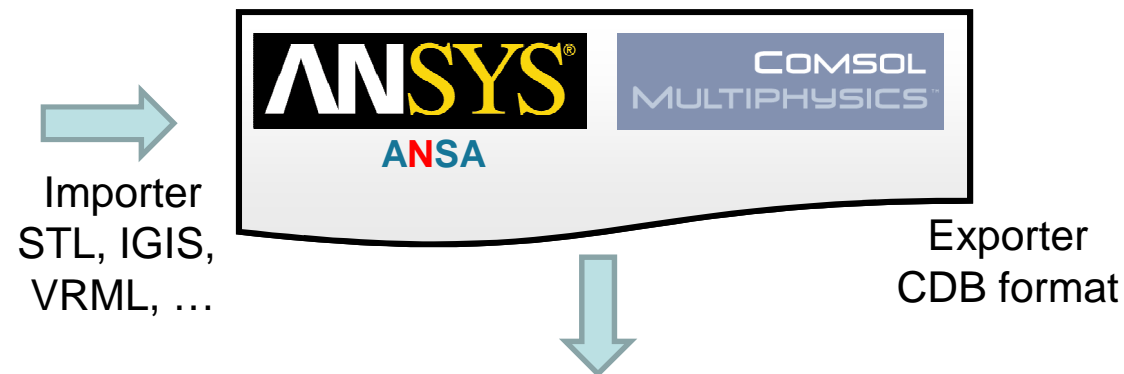


CAD System Integration

- Autodesk Inventor
- CATIA
- Pro/ENGINEER
- Solid Edge
- SolidWorks
- Unigraphics

Specifikace problému a generování sítě.

(typ úlohy, materiálové vlastnosti, počáteční a okrajové podmínky, ...)

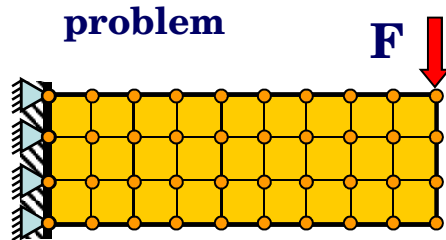


MatSol

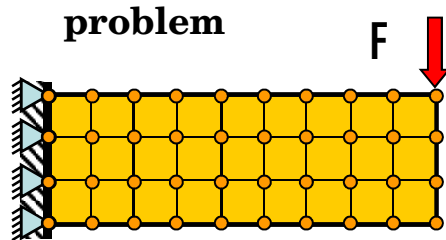
knihovna se škálovatelnými algoritmy založena na metodách rozložení oblasti **FETI** a **BETI**

**T. Kozubek, T. Brzobohatý, A. Markopoulos
Z. Dostál, V. Vondrák, R. Kučera, M. Sadowská**

Metody rozložení oblasti



<p>1. FETI</p>		<p>podoblasti jsou volné nebo uchycené (předem není znám defekt matic K)</p>
<p>2. FETI-DP</p>		<p>FETI-DP (částečné rozdělení, regulární matice tuhosti)</p>
<p>3. TFETI</p>		<p>všechny podoblasti defekt matic K je znám</p>



<p>1.</p> <p>2D</p>		<p>subdomains are fixed or free but with different subsets</p>
<p>2.</p> <p>3D</p>		<p>FEI-3D (period splitting, nonlocal)</p>

<p>3.</p> <p>TBETI</p>	<p>The diagram shows four stages of the TBETI method. Stage 1: A fixed support (blue triangle) is shown. Stage 2: A grid of nodes is shown with red arrows indicating internal forces. Stage 3: A grid of nodes is shown with red arrows indicating internal forces and a downward force F at the top-right corner. Stage 4: A grid of nodes is shown with red arrows indicating internal forces and a downward force F at the top-right corner.</p>	<p>všechny podoblasti jsou volné se shodnou dimenzí jádra</p>
--------------------------------------	--	--

Total FETI – primární proměnné

$$\min \frac{1}{2} u^T K u - u^T f$$

funkcionál energie

$$B_I u \leq c_I$$

u pole posuvů

K matice tuhosti

B matice omezení

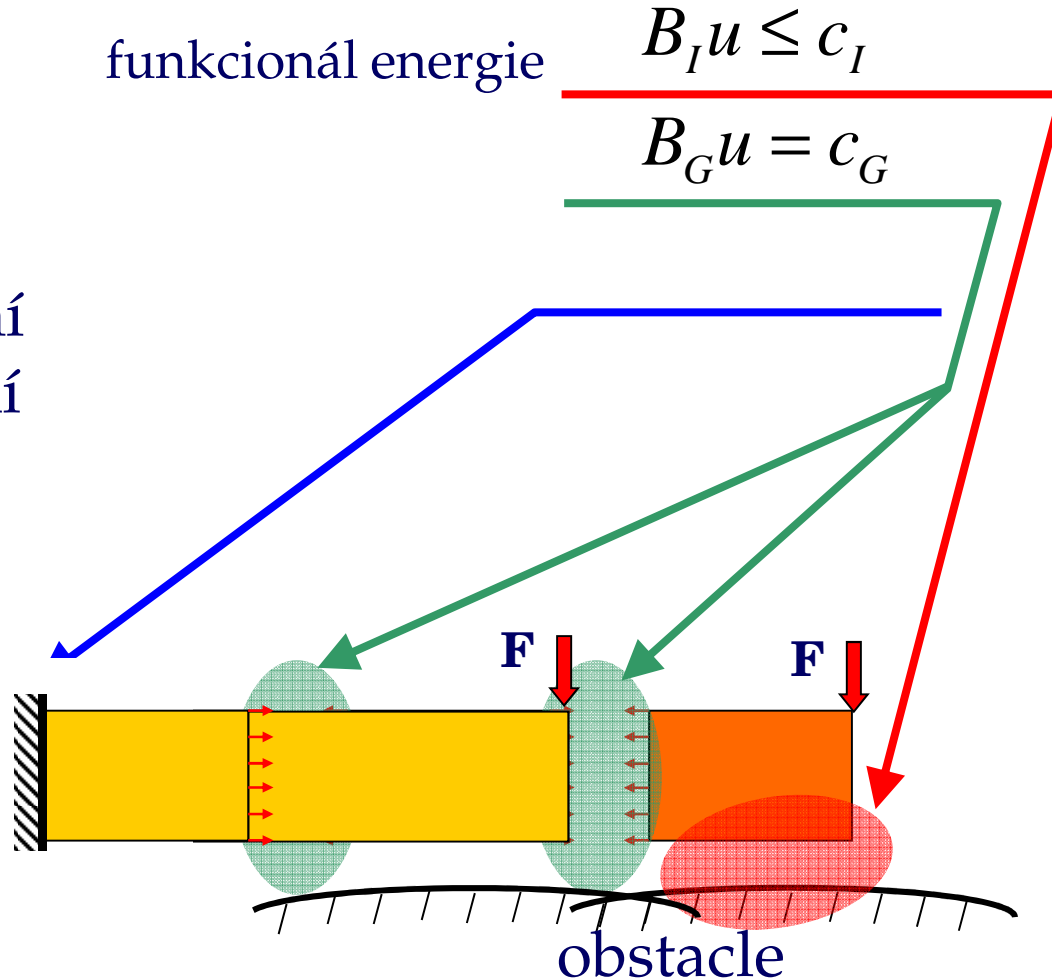
c vektor omezení

$$B_G u = c_G$$

Total FETI

Dirichletovy okrajové podmínky jsou také vynuceny pomocí LM,

$$B_B u = c_B$$



Vše a mnohem více můžete najít

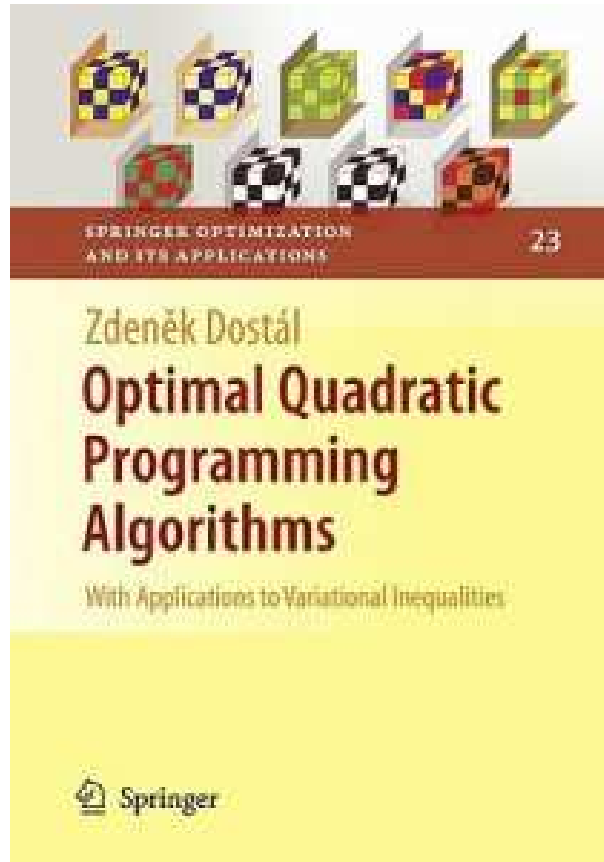


Table of Contents - Preface.

Part I. Background

1. Linear Algebra.- 2. Optimization.

Part II. Algorithms

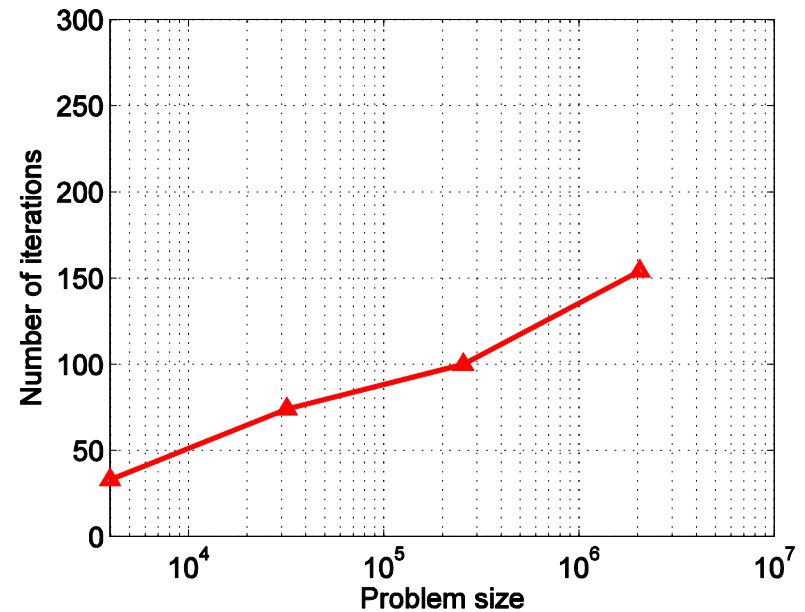
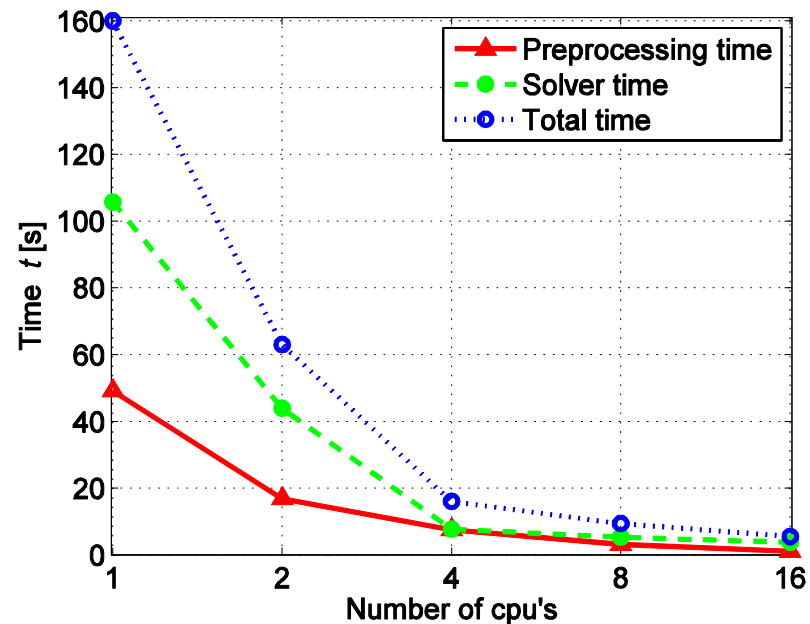
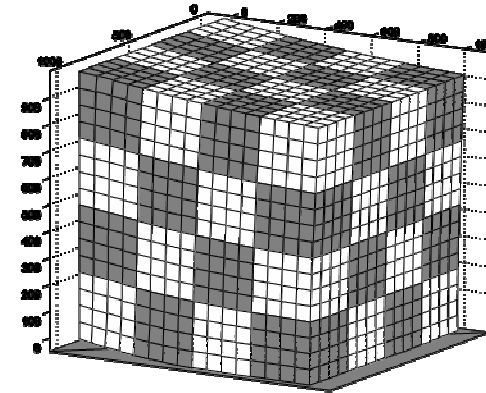
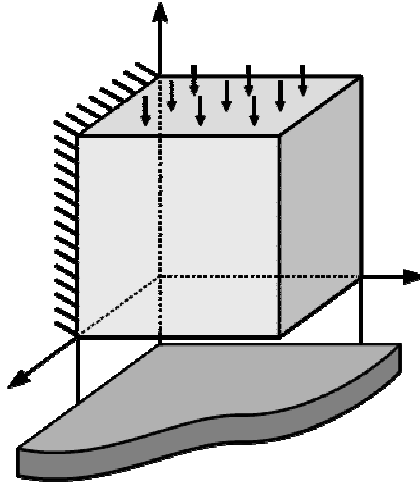
3. CG for Unconstrained Minimization
4. Equality Constrained Minimization
5. Bound Constrained Minimization
6. Bound and Equality Constrained Minimization

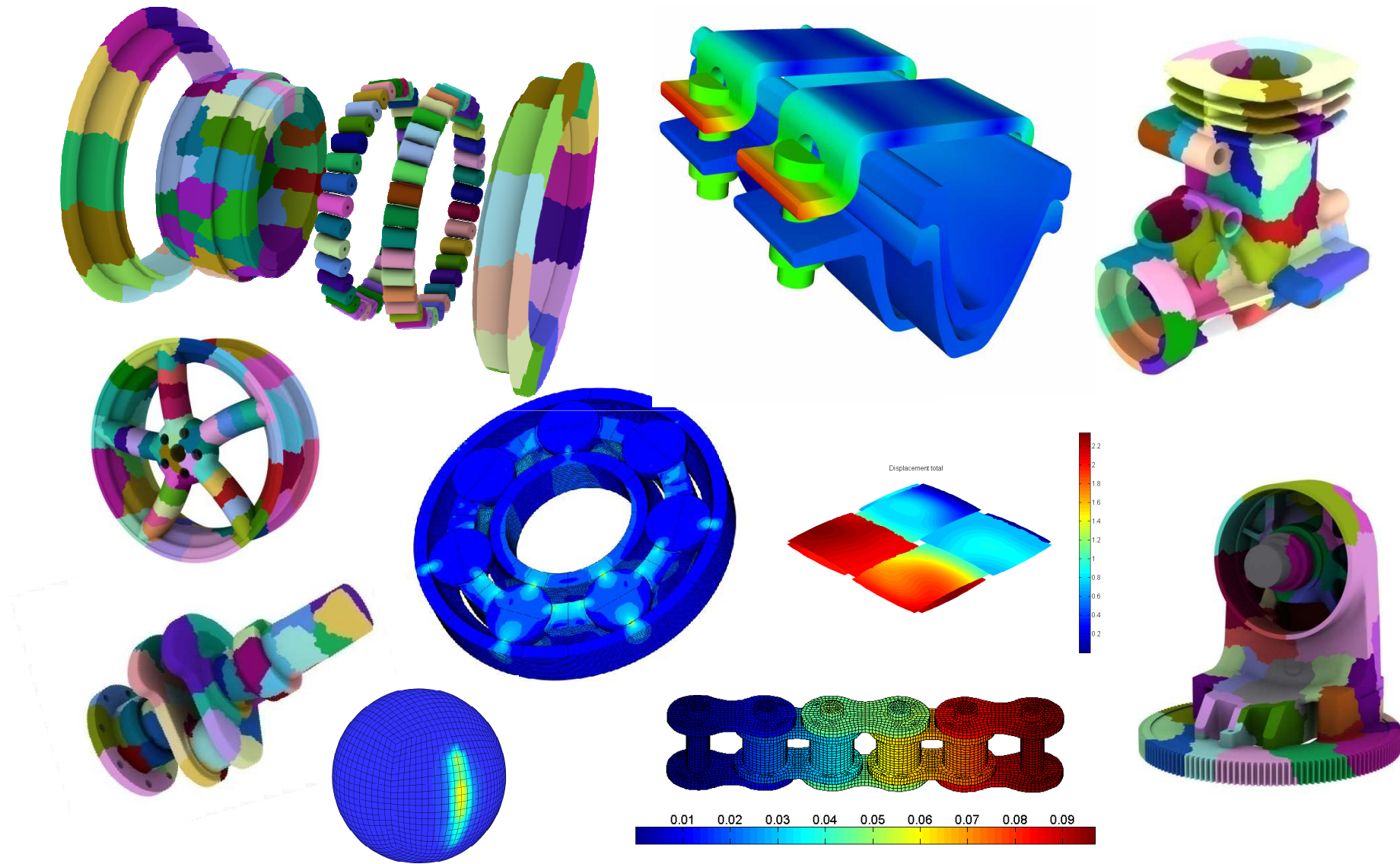
Part III. Applications to Variational Inequalities

7. Solution of a Coercive Variational Inequality by FETI-DP method
8. Solution to a Semicoercive Variational Inequality by TFETI Method.- References.- Index.

Seminář výpočetní mechaniky, 1-3. prosince 2010, VŠB-TU & Ústav geoniky AV ČR, v.v.i.

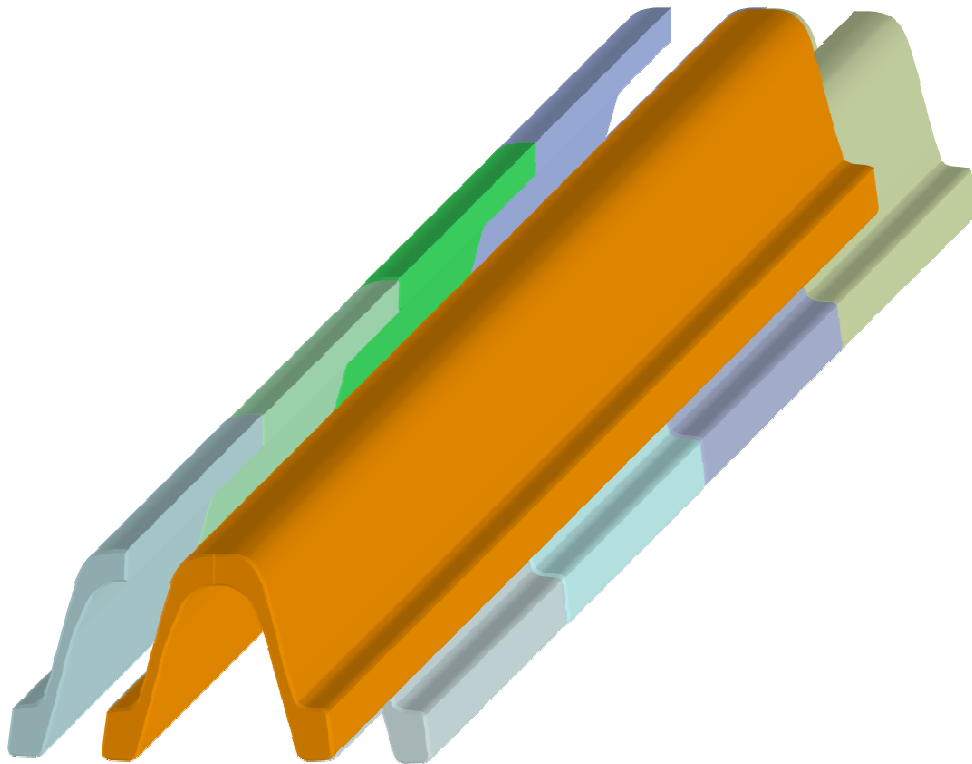
Škálovatelnost: kostka nad tuhou překážkou





1. ingredience: paralelní implementace

blokově diagonální struktura
matice tuhosti =>
vhodná pro paralelní
implementaci



$$K = \begin{pmatrix} K^1 & O & \vdots & O \\ O & K^2 & \vdots & O \\ \dots & \dots & \ddots & \dots \\ O & O & \vdots & K^n \end{pmatrix}$$

Mohou být řešeny koercivní
a semikoercivní úlohy.

Paralelní programování

Cluster HP model BLc7000 (c-class).

Typické případy

- Paralelně smyčky **for**
 - mnoho cyklů
 - dlouhé iterační cykly
- Offloading Work
- Objemné datové sady

Konfigurace výpočetních uzlů:

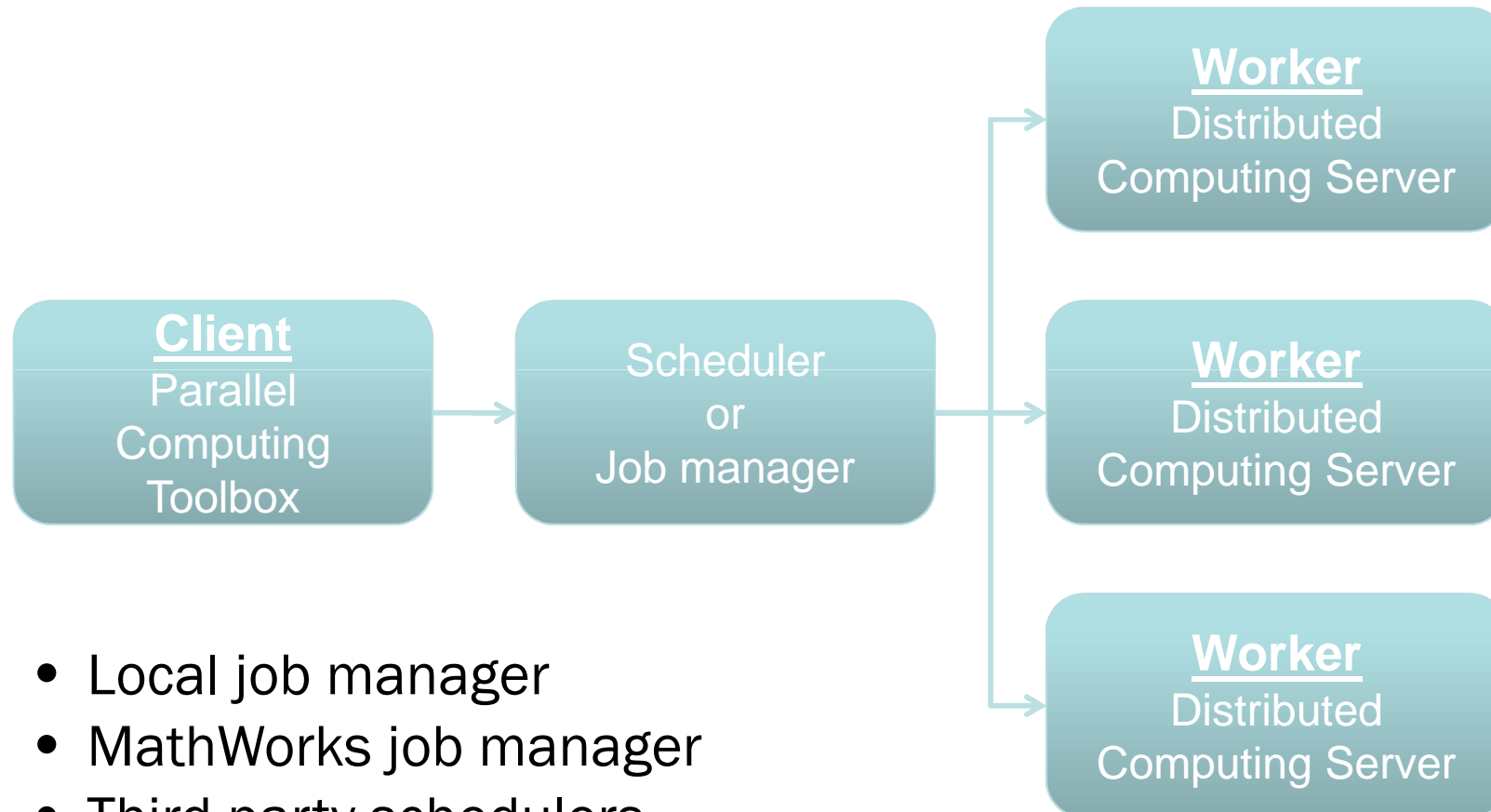
2x dual core CPU AMD Opteron 2210 HE
8GB ECC DDR2 667MHz RAM



IT4Innovations, Centrum excellence,
Ostrava, Czech Republic, cluster with
more than 30,000 processors.

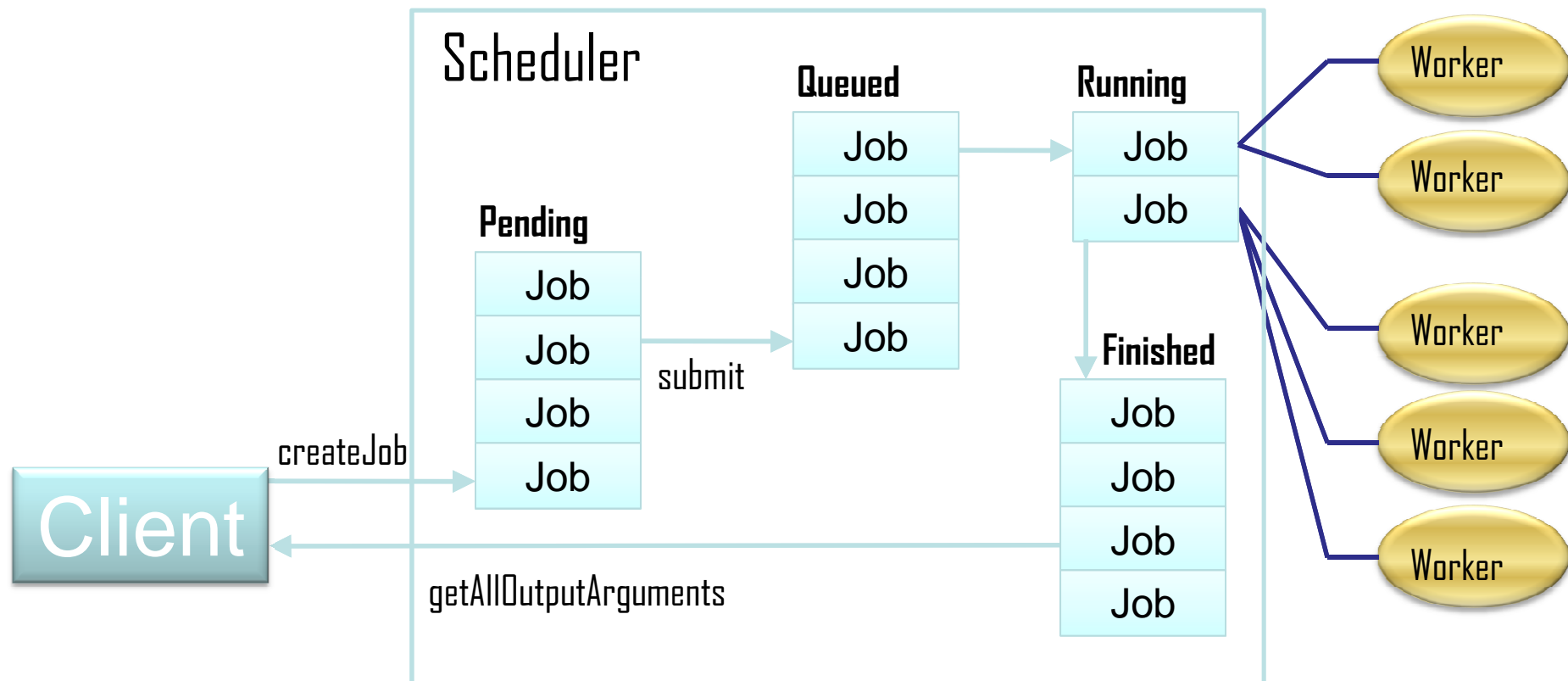
<http://www.it4i.eu>

Matlab Distributed Computing Server



- Local job manager
- MathWorks job manager
- Third-party schedulers
(Windows CCS, PBS Pro, LSF)

Life cycle of a job



Distributed computations

1. Find scheduler or job manager

```
>>sched=findResource('scheduler',  
'type','local');
```
2. Create a job

```
>>job=createJob(sched);
```
3. Create tasks and associate them with the job

```
>>createTask(job,@rand,1,{3,3});  
>>createTask(job,@eye,1,{4});  
>>createTask(job,@ones,1,{{4},{3}});
```
4. Send job to the front

```
>>submit(job);
```
5. Wait until job finishes

```
>>waitForState(job);
```
6. Gather results

```
>>results=getAllOutputArguments(job)
```
7. Destroy job

```
>>destroy(job);
```

Parallel computations

1. Find scheduler or job manager

```
>>sched=findResource('scheduler',...  
'type','local');
```
2. Create a parallel job and sets the number of tasks

```
>>pjob=createParallelJob(sched);  
set(pjob,'MaximumNumberOfWorkers',4);  
set(pjob,'MinimumNumberOfWorkers',4);
```
3. Create parallel tasks

```
>>createTask(pjob,@MyParTask,1,{});
```
4. Send job to the front

```
>>submit(pjob);
```
5. Wait until job finishes

```
>>waitForState(pjob);
```
6. Gather results

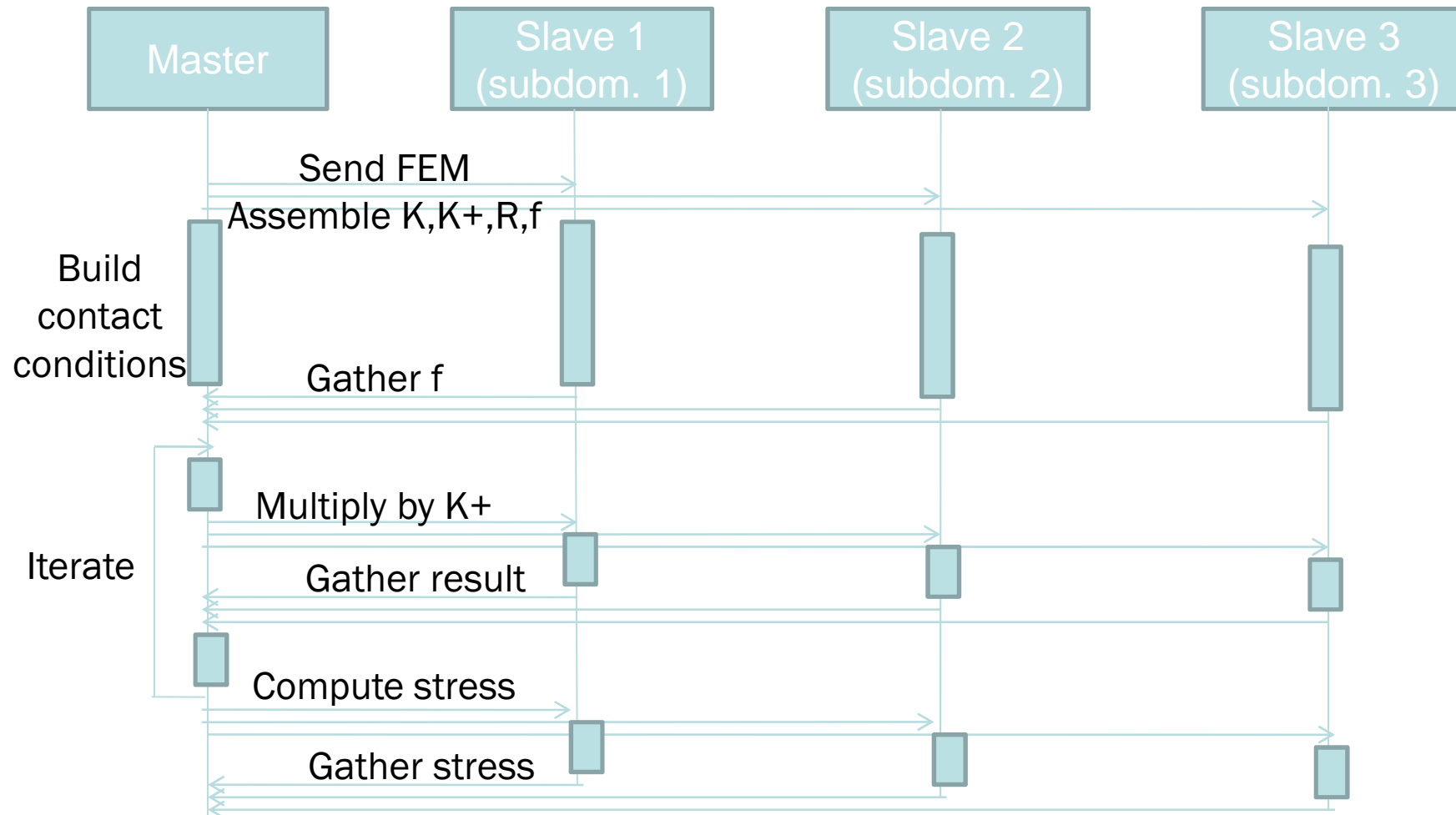
```
>>results=getAllOutputArguments(pjob)
```
7. Destroy job

```
>>destroy(pjob);
```

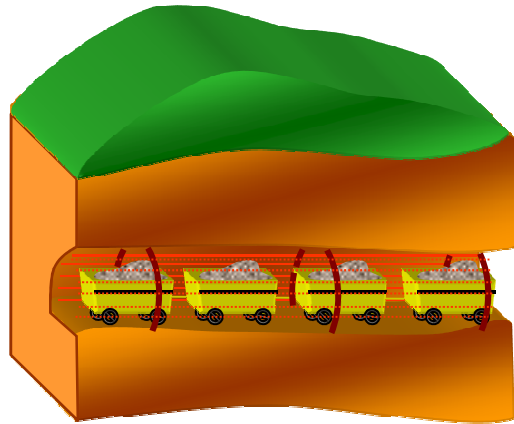
MPT functions: numlabs, labindex, labBarrier, labBroadcast, labProbe, labReceive, labSend, labSendReceive

Parallel computation scenario

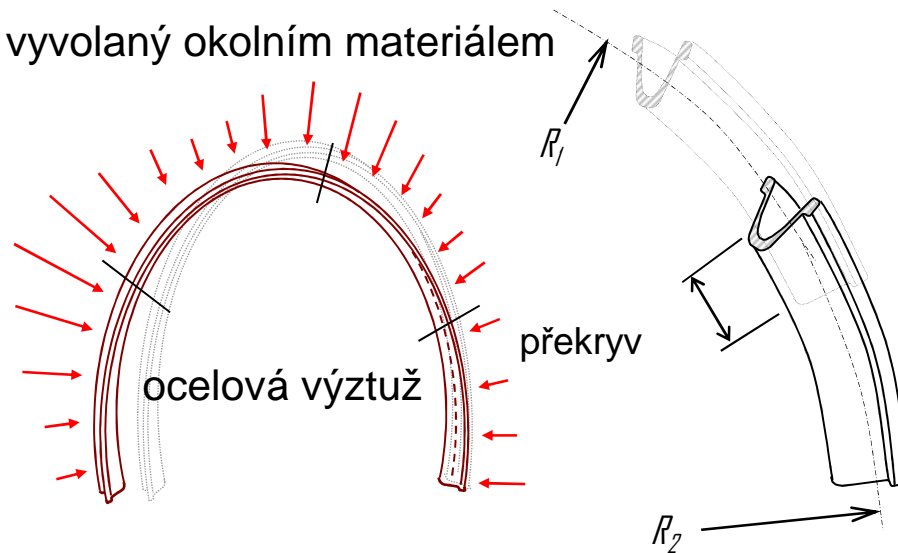
Send FEM models to workers and parallelize assembling of K , K^+ , R , f_v , stresses, searching contact pairs, multiplication procedures.



Důlní průmysl: svěrný spoj důlní výztuže



tlak vyvolaný okolním materiálem



Důlní průmysl: svěrný spoj důlní výztuže

Coulombovské tření

$$f = 0.1$$

$$E = 2.1e5 \text{ MPa,}$$

$$\mu = 0.3$$

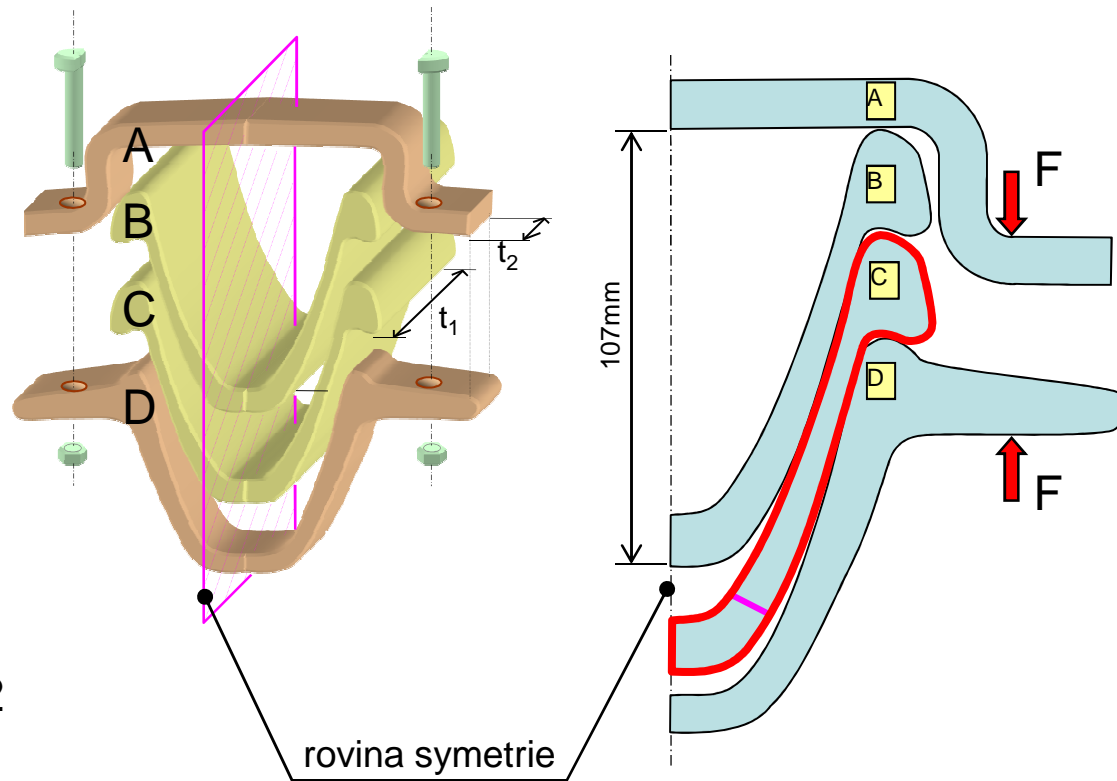
$$t_A = t_D = 10 \text{ mm}$$

$$t_B = t_C = 40 \text{ mm}$$

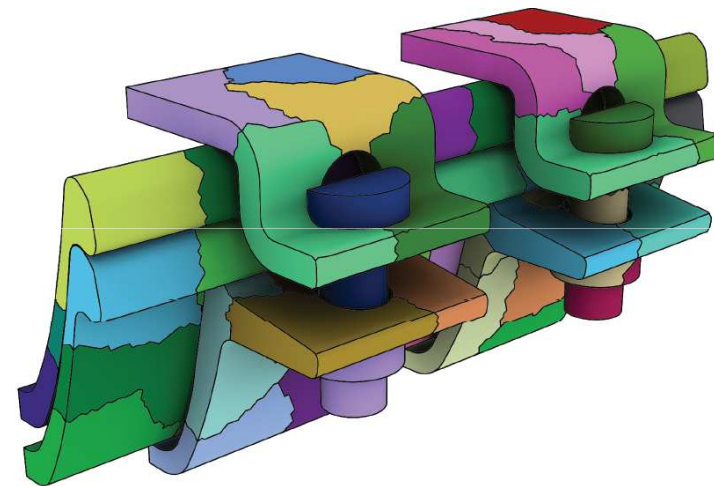
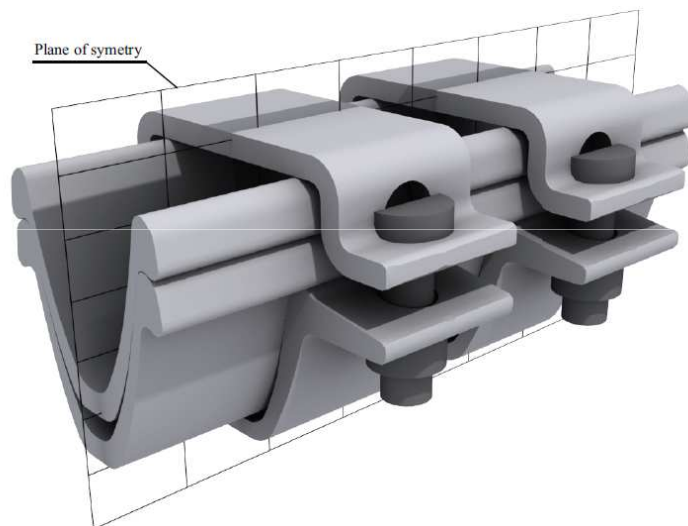
počet:

binárních proměnných 65562

duálních proměnných 3112



Důlní průmysl: svěrný spoj důlní výztuže

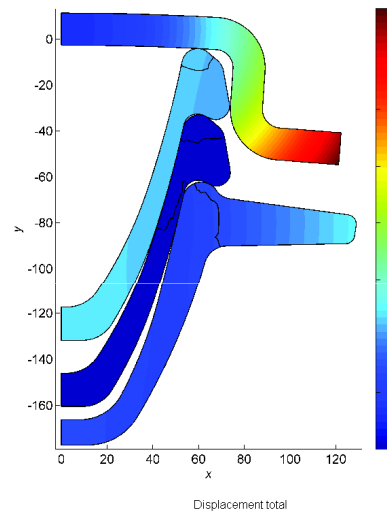


8 těles
7 volných

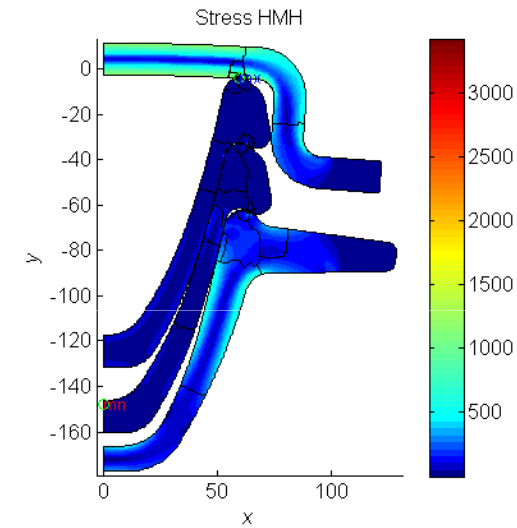
Důlní průmysl: svěrný spoj důlní výztuže

2D

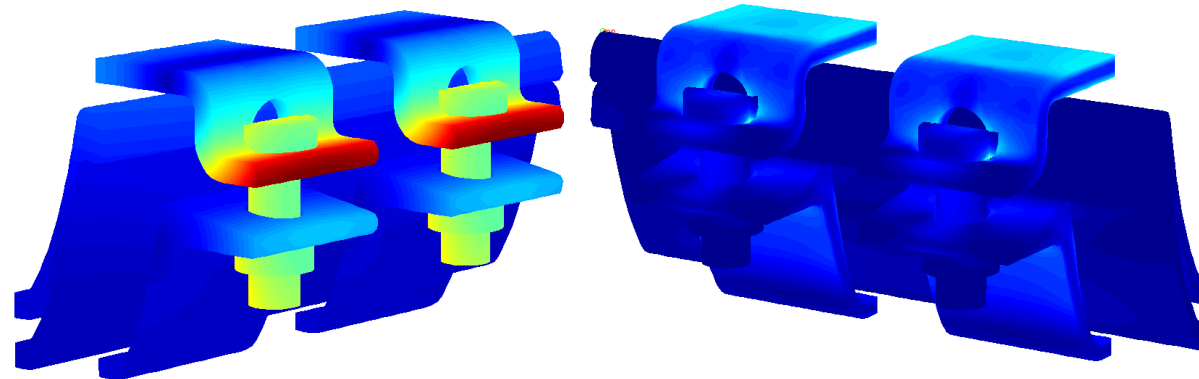
Posunutí



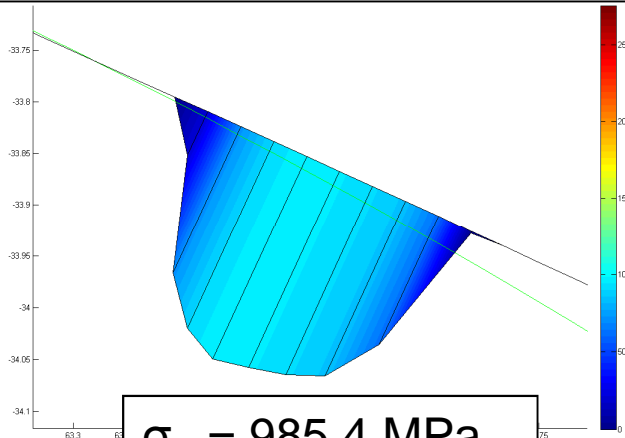
redukované napětí HMH



3D

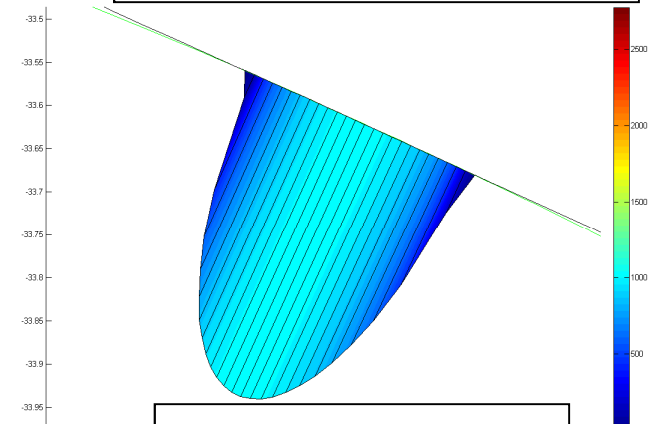


bez korekcí kontaktních směru

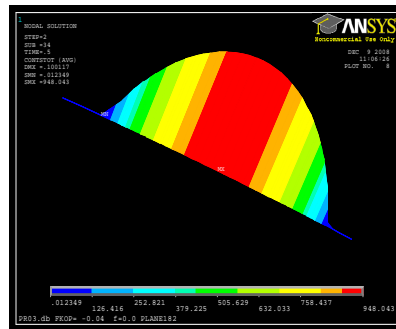
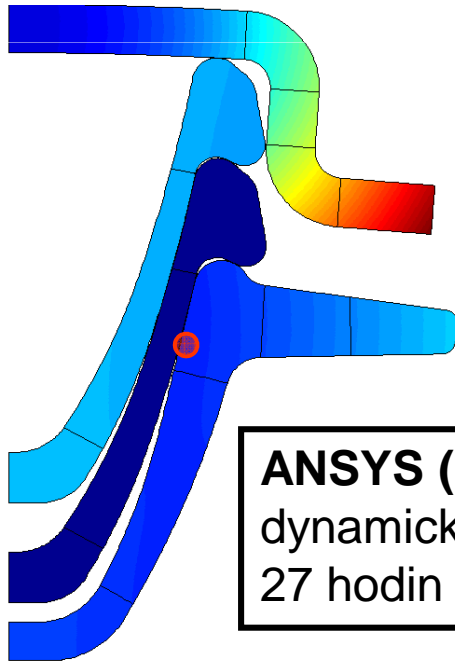


$\sigma_N = 985.4 \text{ MPa}$

3 korekce



$\sigma_N = 1020.6 \text{ MPa}$



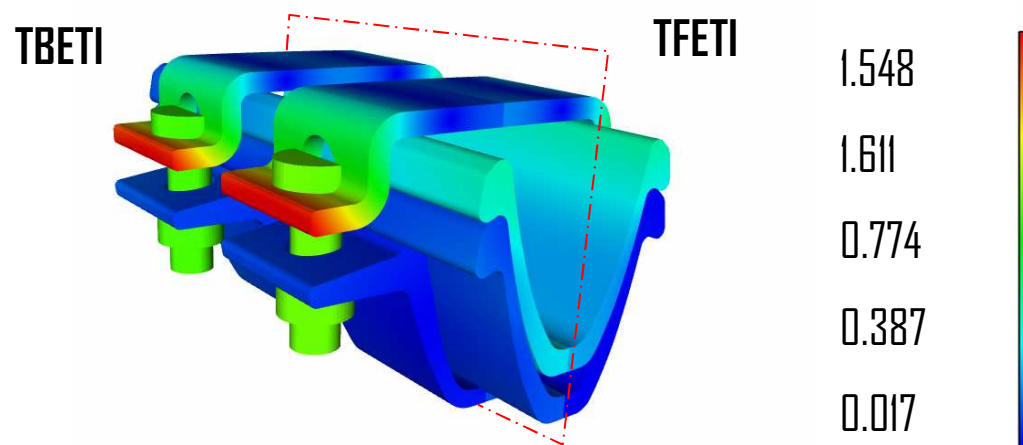
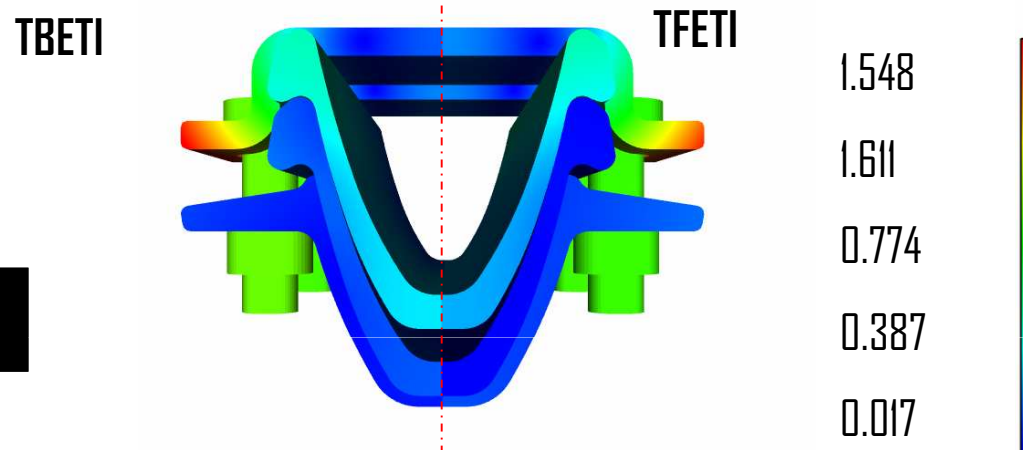
ANSYS (paralel)
dynamický výpočet:
27 hodin

ANSYS (paralel)
statický výpočet:
1 hodina

MATSOL statická
úloha, 3 korekce:
sekvenční verze 15
min.
paralelní 7 min.

Srovnání TBETI a TFETI

Celková posunutí [mm]



Srovnání TBETI a TFETI

250 podoblastí $f = 0.1$

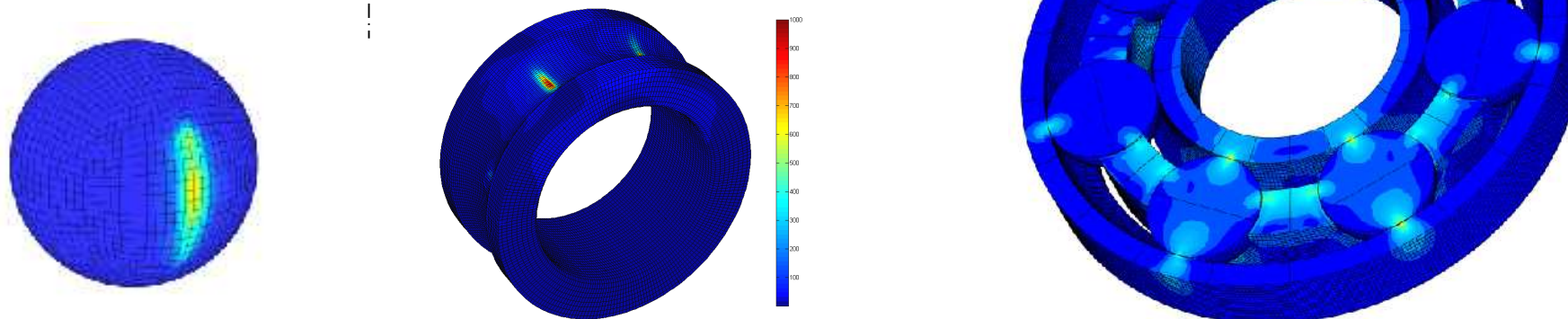
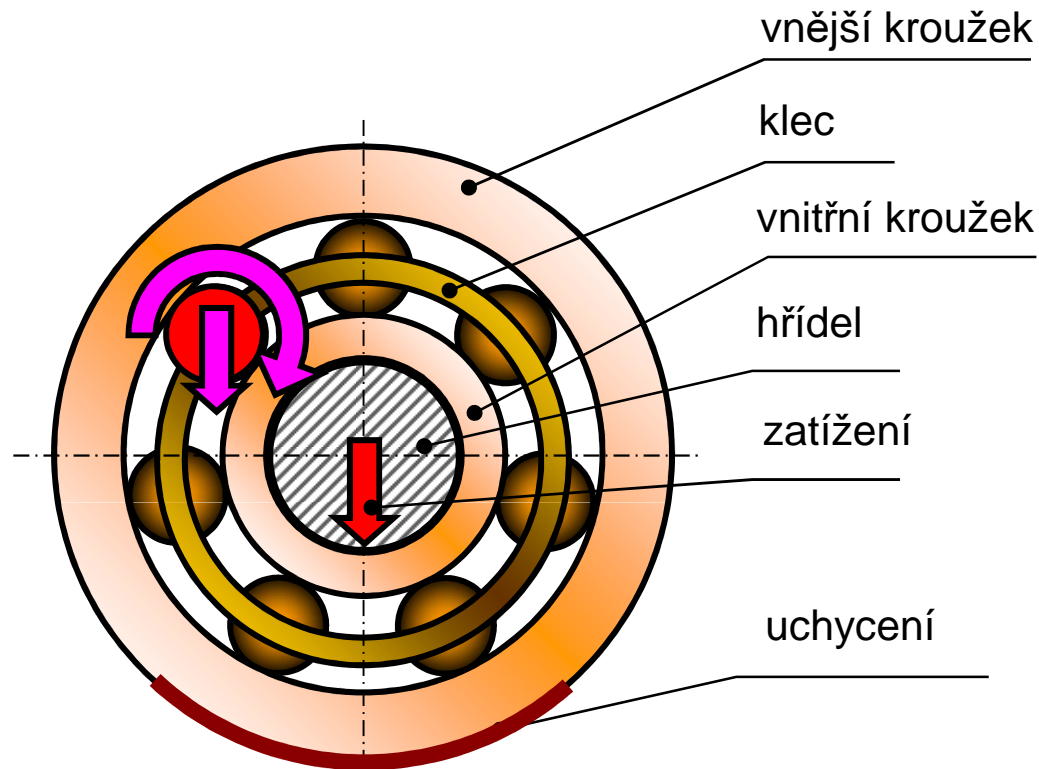
Čas výpočtu	2.41 h.
Celkový čas	2.55 h.
násobení matice vektor	2 126
primární proměnné	1 592 853
duální proměnné	261 553
max(abs(u))	1.303 mm
max(totalU)	1.548 mm

FETI

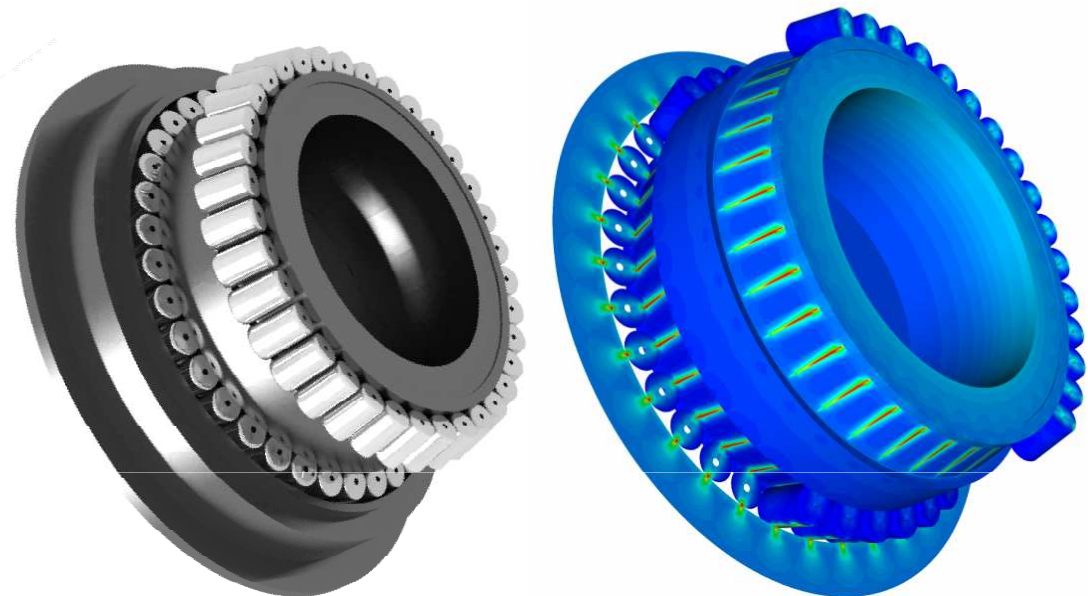
Čas výpočtu	2.54 h.
Celkový čas	3.33 h.
násobení matice vektor	1 882
primární proměnné	713 751
duální proměnné	261 553
max(abs(u))	1.378 mm
max(totalU)	1.616 mm

BETI

Automobilový průmysl: kuličkové ložisko



Válečkové ložisko větrné elektrárny



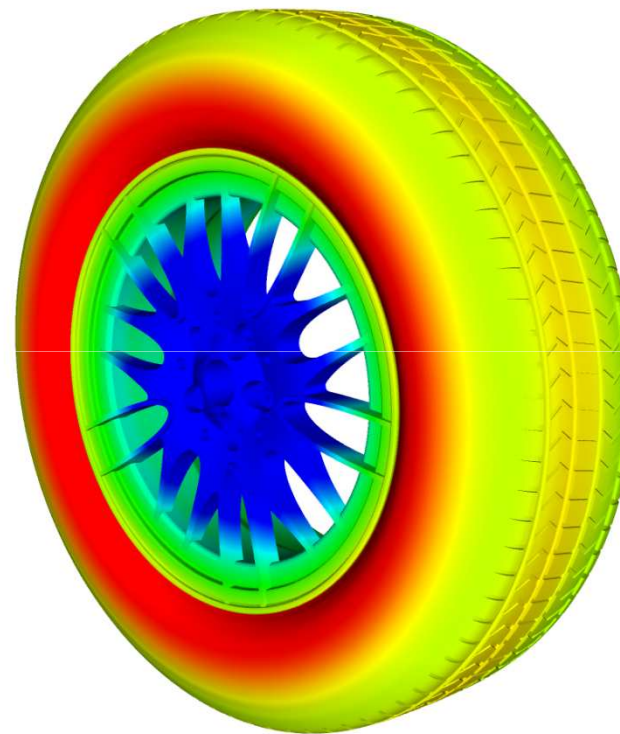
Statistics

počet těles	73
počet podoblastí	700
primární proměnné	2,73 M
duální proměnné	459,8 k
počet iterací	4270

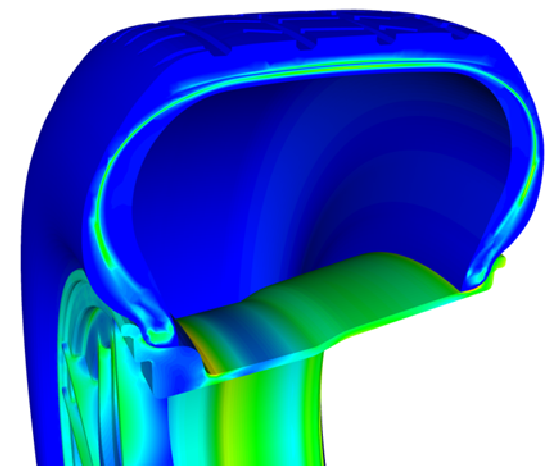


Automobilový průmysl: pneumatika

3.9 mil stupňů
volnosti,
4000 iterací



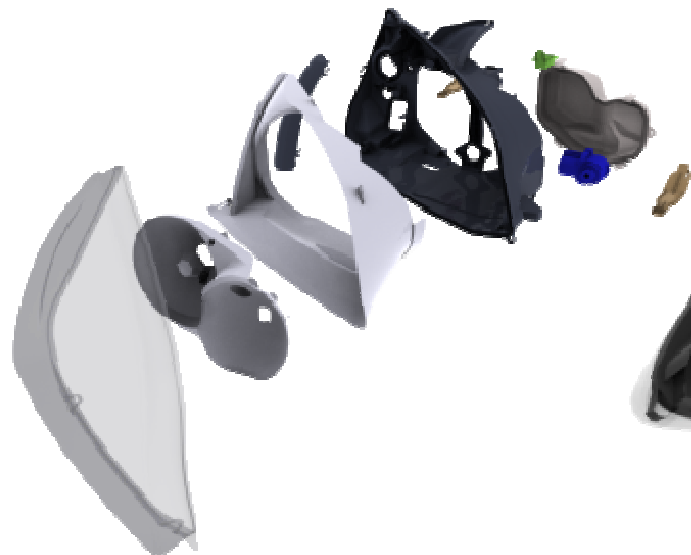
celková posunutí



redukované
napětí HMH

Světlo met – Visteon (Autopal)

12 těles

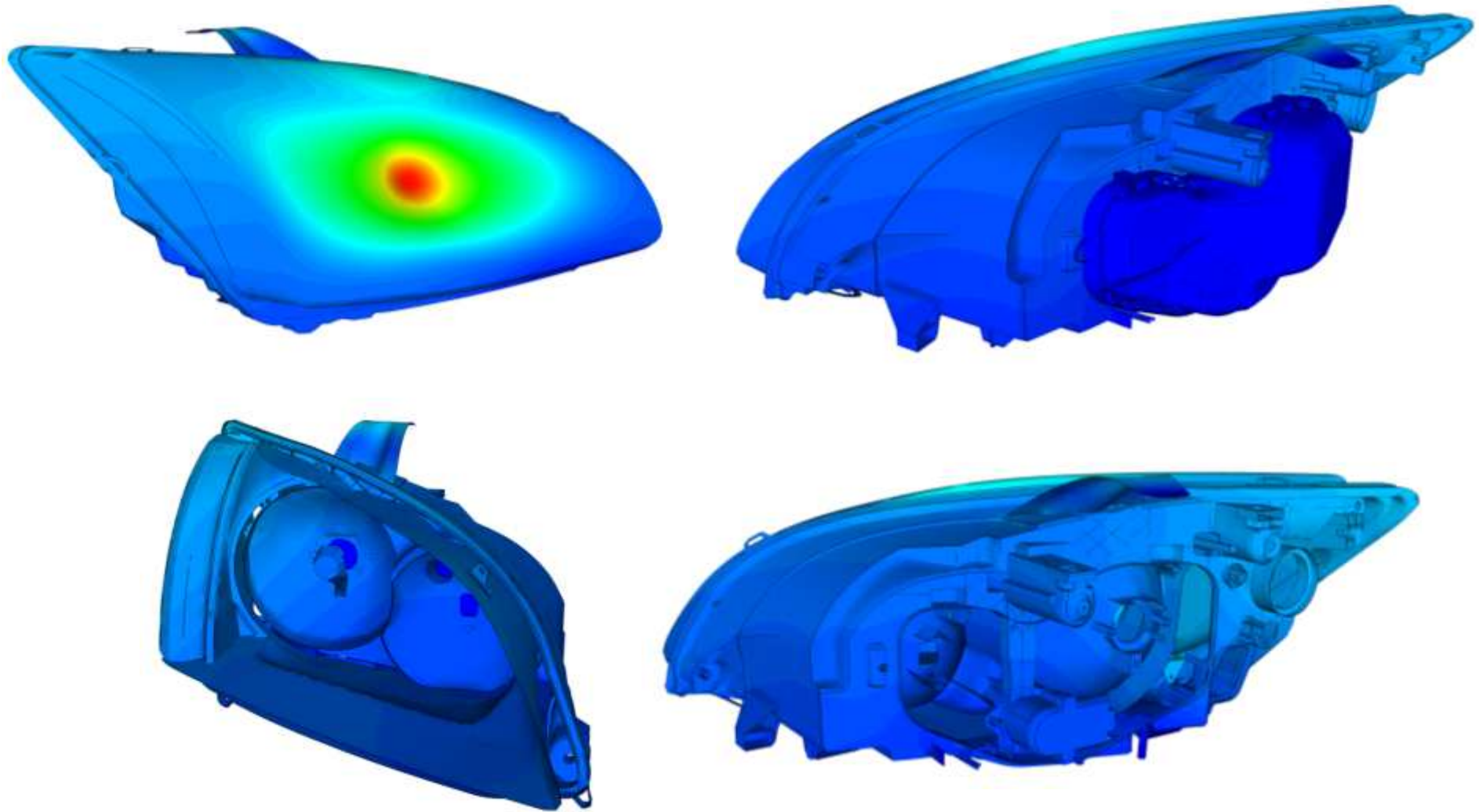


Světloomet – Visteon

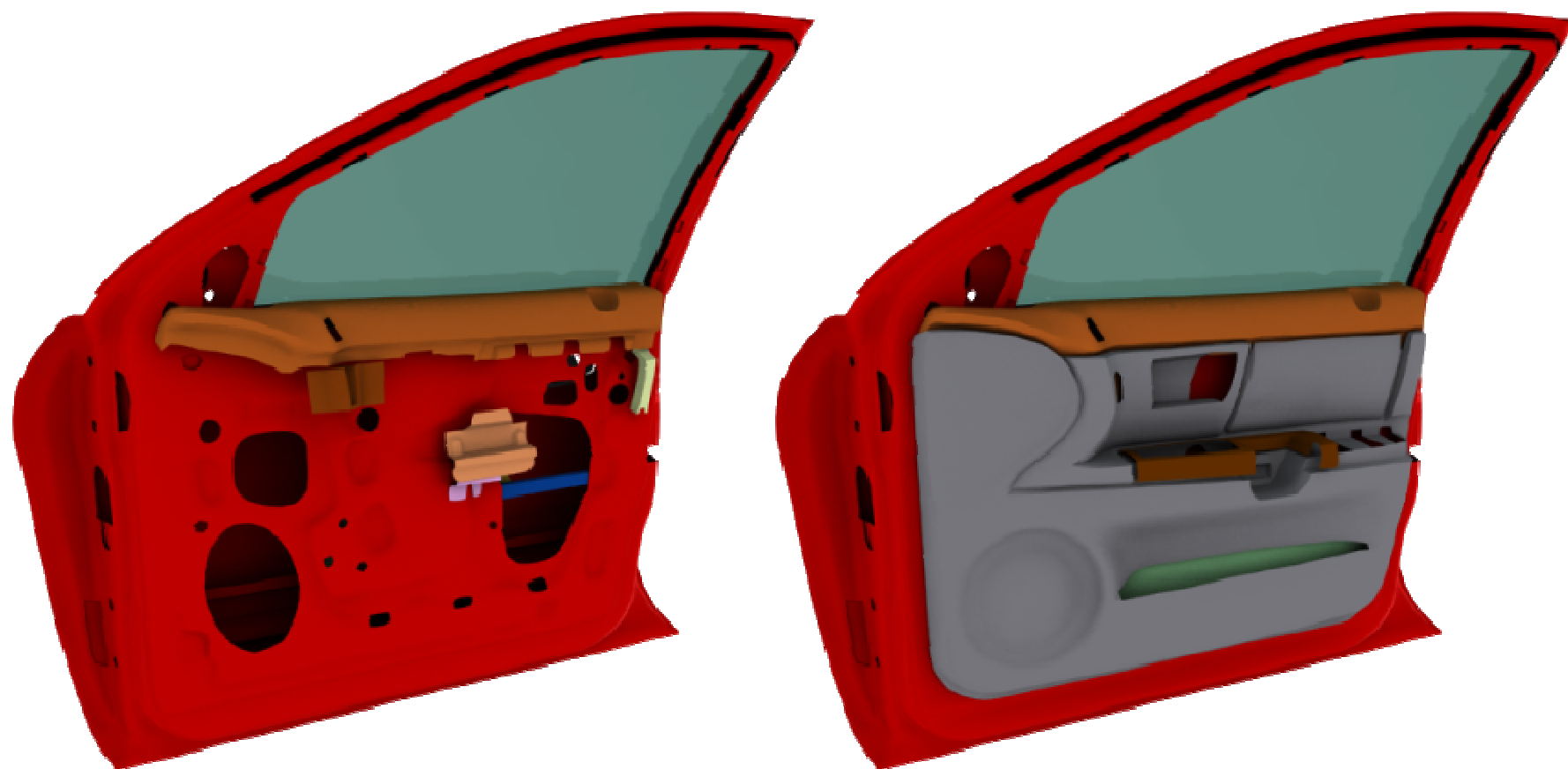
64 Oblastí



Světlomet – Visteon (Test solution-Total Displacement)

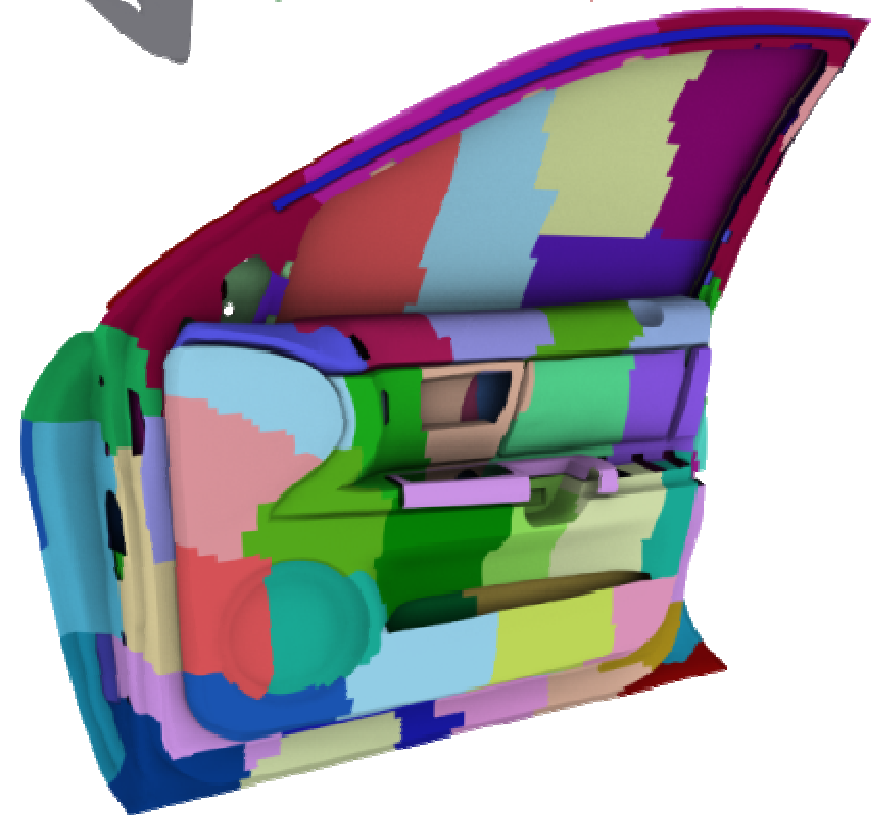
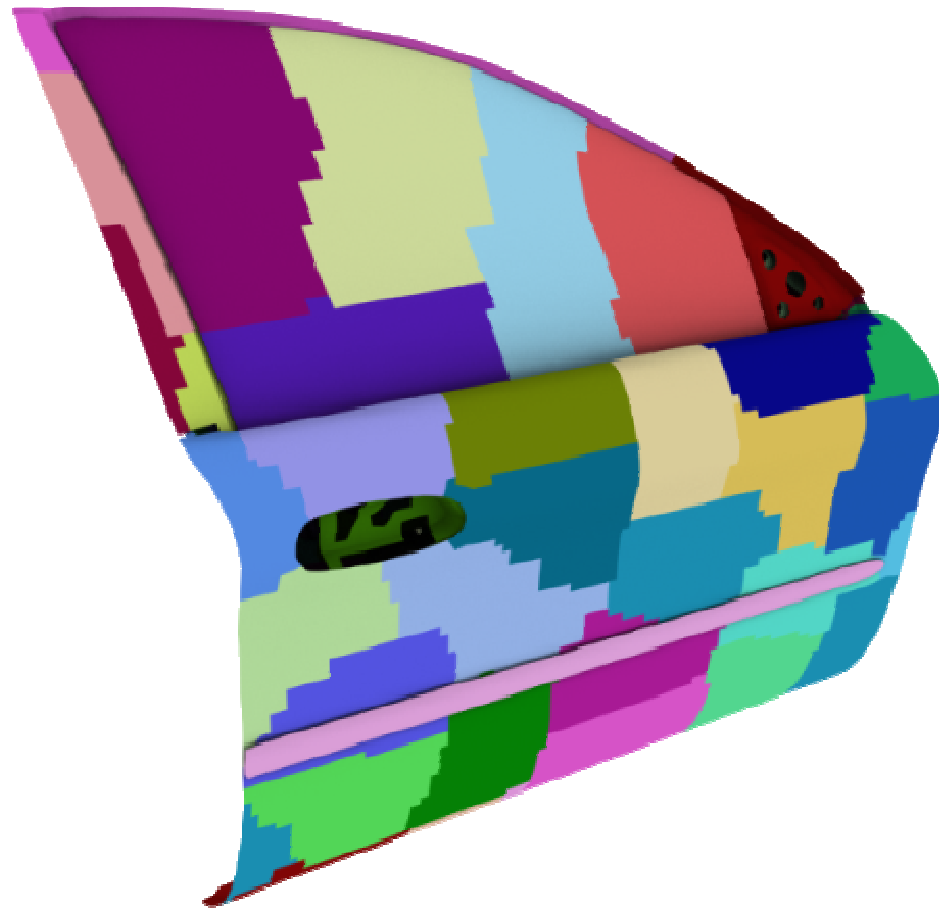


Dveře auta

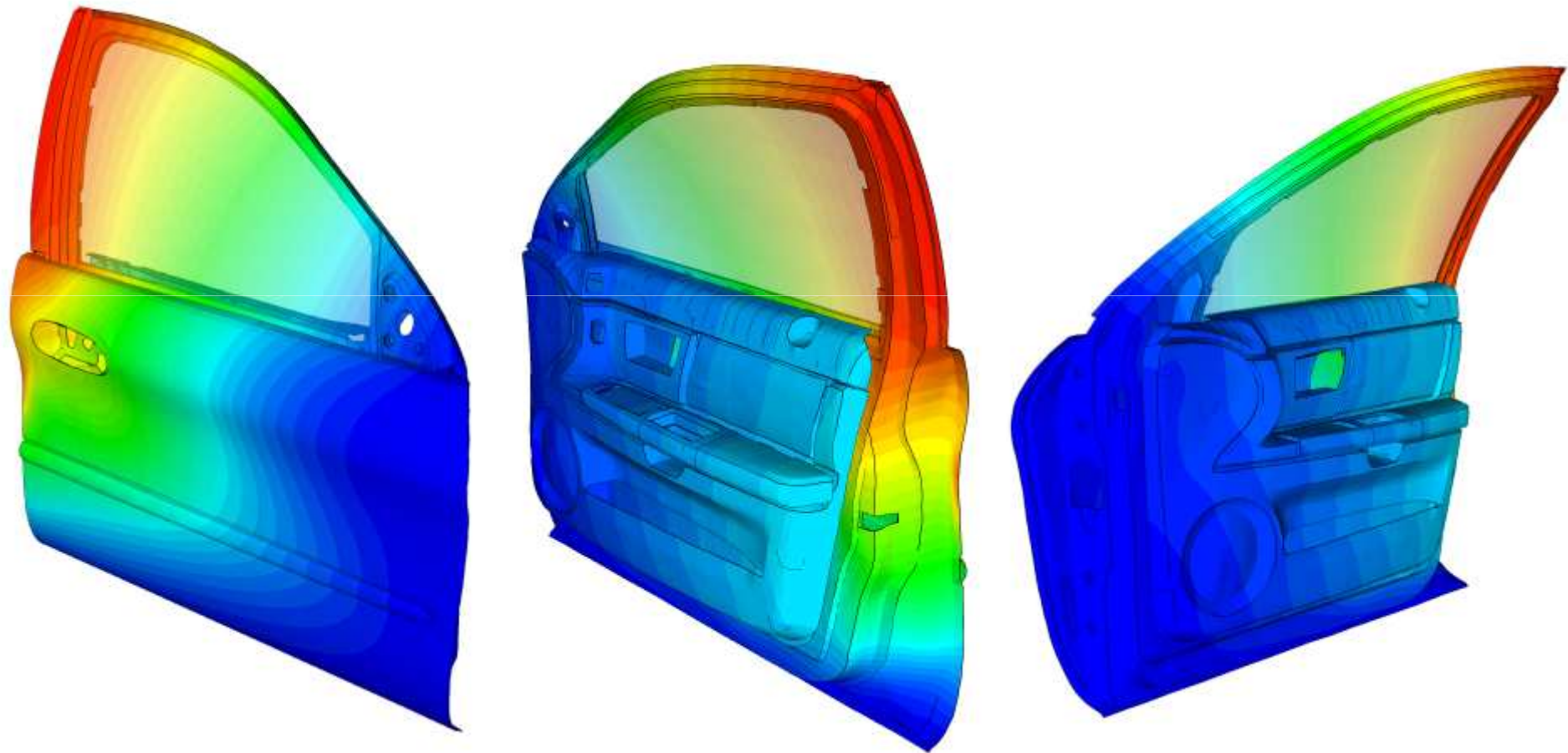


Dveře auta

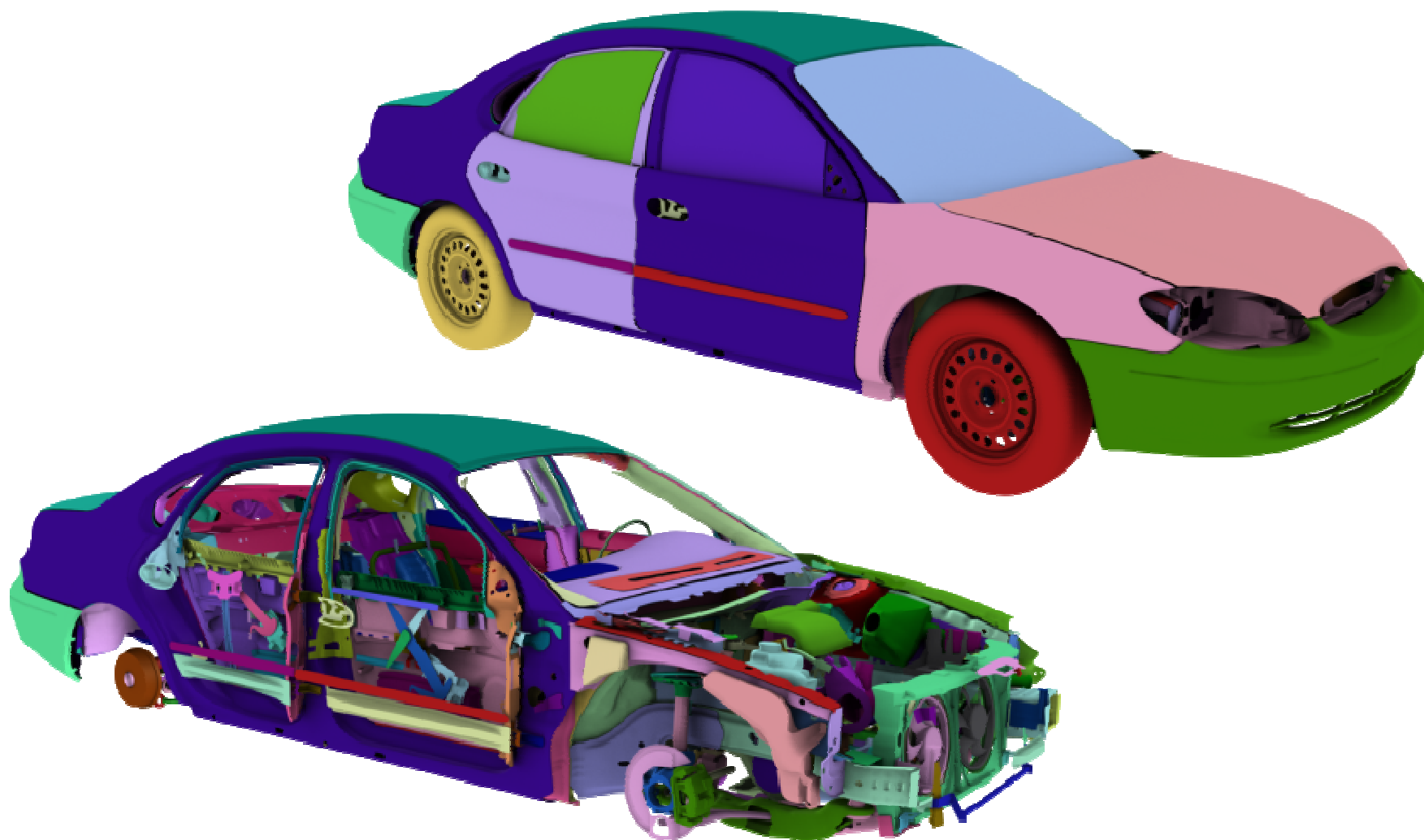
128 Oblastí



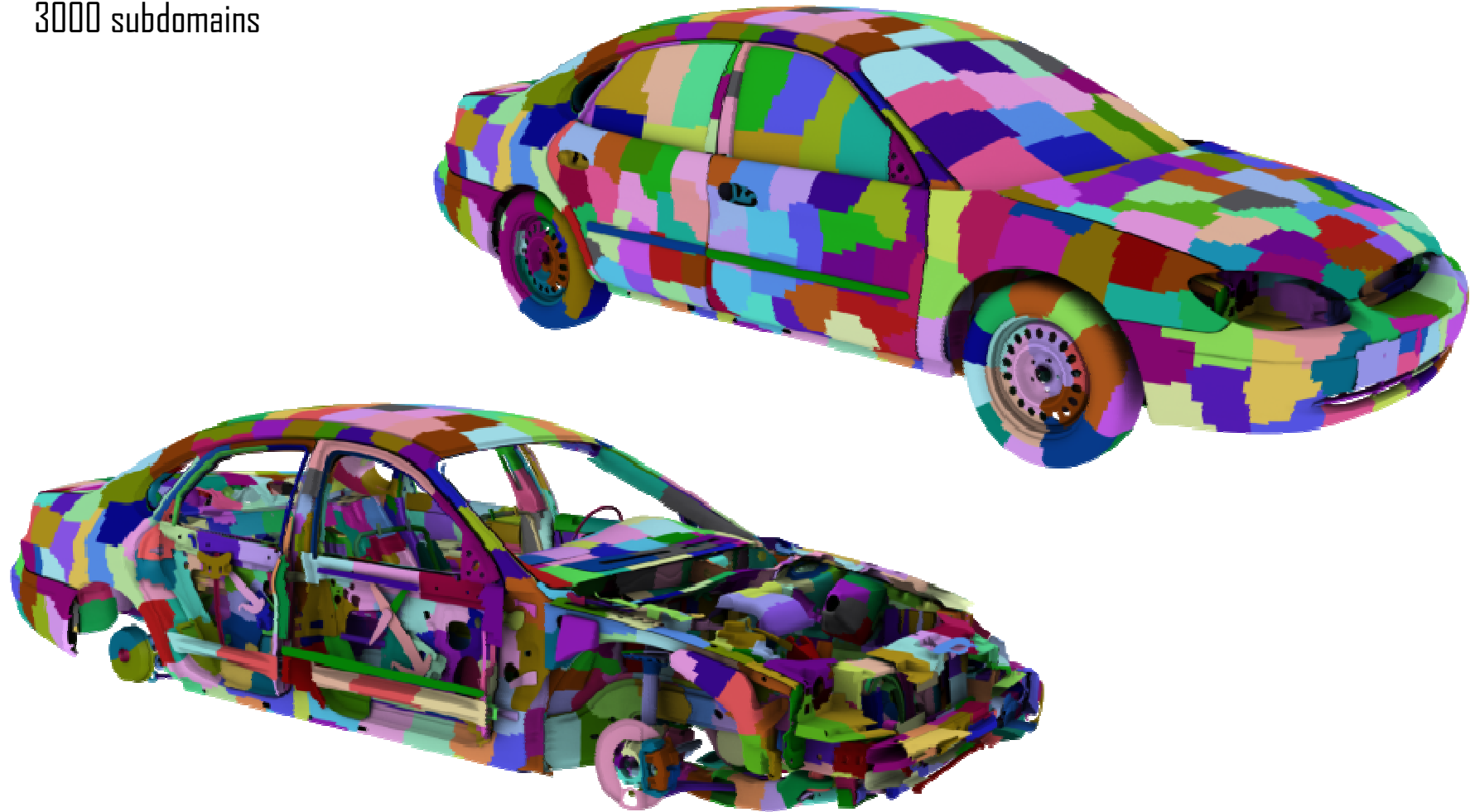
Dveře auta (Test solution-Total Displacement)



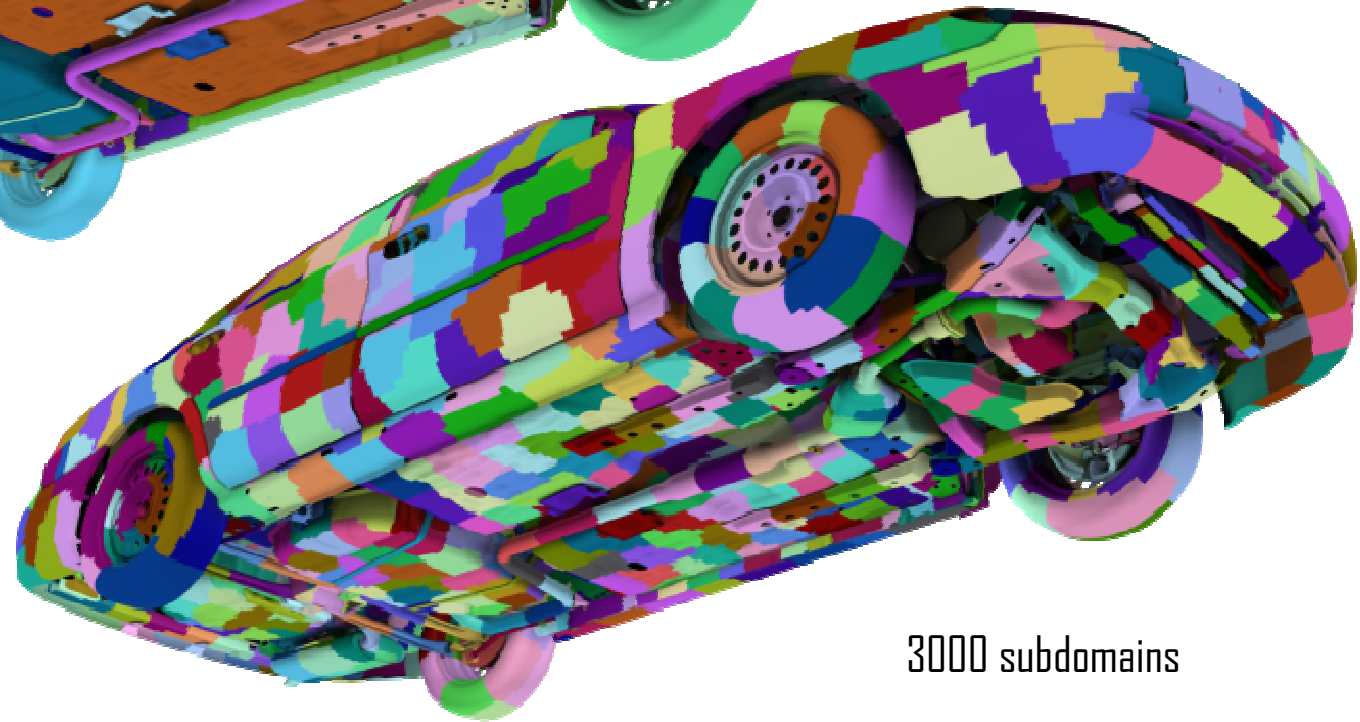
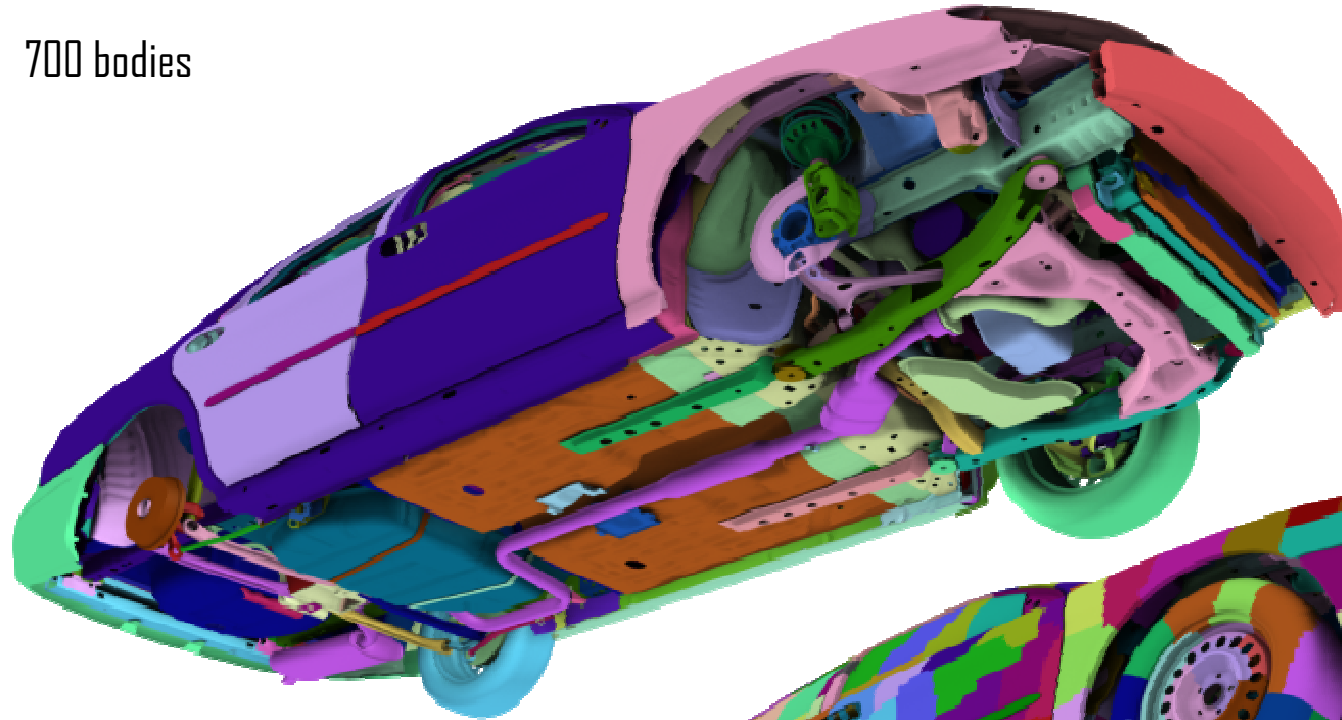
Auto – Sestava obsahuje okolo 700 těles



3000 subdomains

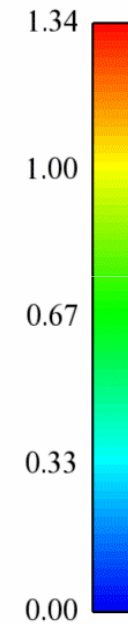
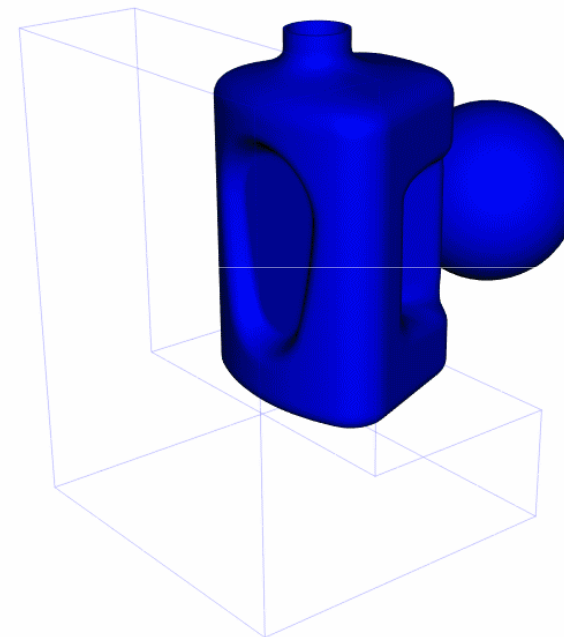
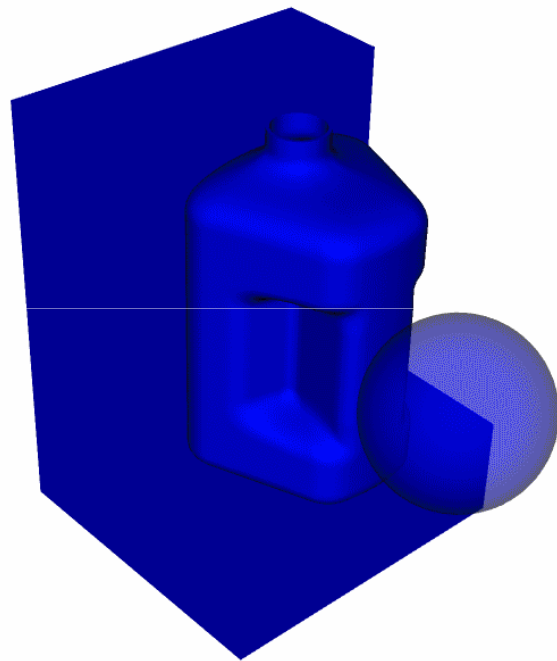


700 bodies



3000 subdomains

Rozšíření na dynamické úlohy



Děkuji za pozornost