Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science ICS

Inherently parallel solution methods for nonlinear problems in biomechanics

C. Gross, **R. Krause**

Institute of Computational Science
Università della Svizzera Italiana
Lugano

Workshop on Efficient Solvers in Biomedical Applications

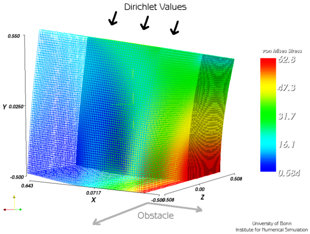
Mariatrost, July 2-5, 2012

Non–convex Model Problem and Applications

Solution of arbitrary non–linear problem

$$u \in \mathcal{B} \subset \mathbb{R}^n : J(u) = \min!$$

$\mathcal{B} = \{v \in \mathbb{R}^n : \underline{\phi} \leq v \leq \overline{\phi}\}$ a set of admissible solutions, J continuously differentiable objective function. $\underline{\phi}, \overline{\phi} \in \mathbb{R}^n$.



Contact problem with highly non–linear objective function

Applications:

- **Nonlinear Elasticity**
- Computer Vision
- Neuroinformatics
- ...

Solution is carried out employing a globalization strategy

- Trustregion Strategy
- Linesearch Strategy

Constrained Minimization

- $\mathbf{H} = (H^1(\Omega))^d$, $\mathbf{H} = (W^{1,p}(\Omega))^d$, $p > d$; $d = 2, 3$,
- $\mathcal{J} : \mathbf{H} \longrightarrow \mathbb{R}$ (non-)convex functional: stored energy function
- constraints: $\mathbf{u} \in \mathcal{K}$: equality/inequality constraints

$$\mathcal{J}(\mathbf{u}) = \min_{\mathbf{v} \in \mathcal{K}} \mathcal{J}(\mathbf{v})$$

Direct minimization $J(\mathbf{u}^0) \geq J(\mathbf{u}^1) \geq J(\mathbf{u}^2) \geq \cdots \geq J(\mathbf{u})$, $\mathbf{u}_i \in \mathcal{K}$
gradient methods, sequentiell coordinate minimization, Newton-methods,. . .

First order necessary condition (non–smooth) :
Quadratic Energy $J(\mathbf{v}) = \frac{1}{2} a(\mathbf{v}, \mathbf{v}) - f(\mathbf{v})$: variational inequality

$$\mathbf{u} \in \mathbf{H} : \quad a(\mathbf{u}, \mathbf{v} - \mathbf{u}) \geq f(\mathbf{v} - \mathbf{u}) \quad \mathbf{v} \in \mathcal{K} .$$

Active set strategies, subspace correction methods, multigrid, . . .

First order necessary conditions: solve non–linear equation

$$J'(\mathbf{u})(\mathbf{v}) = 0 , \qquad \mathbf{v} \in H .$$

Newton-methods, interior points,penalty,. . .

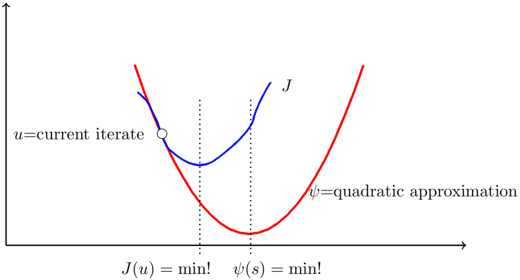
Trust-Region Methods

Iterative Method, initial iterate can be chosen almost arbitrary

1. Newton-step: Solve

$$s \in \mathbb{R}^n : \psi(s) = \frac{1}{2}\langle s, Bs\rangle + \langle \nabla J(u), s\rangle = \min!$$
$$\text{such that } \|s\| \le \Delta,\ u + s \in \mathcal{B}$$

where B is a symmetric approximation the Hessian(Quasi-Newton-Method)



Quadratic approximation to a nonlinear function

Trust-Region Methods

Iterative Method, initial iterate can be chosen almost arbitrary

1. Newton-step: Solve

$$s \in \mathbb{R}^n : \psi(s) = \frac{1}{2}\langle s, Bs\rangle + \langle \nabla J(u), s\rangle = \min!$$
$$\text{such that } \|s\| \leq \Delta,\ u + s \in \mathcal{B}$$

   where B is a symmetric approximation the Hessian(Quasi-Newton-Method)

2. Acceptance: $\rho = \frac{J(u+s)-J(u)}{\psi(s)} \geq \eta$ then: $u^{\text{new}} = u + s$, otherwise $u^{\text{new}} = u$, $\eta \in (0, 1)$.

Trust-Region Methods

Iterative Method, initial iterate can be chosen almost arbitrary

1. Newton-step: Solve

$$s \in \mathbb{R}^n : \psi(s) = \frac{1}{2}\langle s, Bs\rangle + \langle \nabla J(u), s\rangle = \min!$$
$$\text{such that } \|s\| \leq \Delta,\ u + s \in \mathcal{B}$$

where B is a symmetric approximation the Hessian(Quasi-Newton-Method)

2. Acceptance: $\rho = \frac{J(u+s)-J(u)}{\psi(s)} \geq \eta$ then: $u^{\text{new}} = u + s$, otherwise $u^{\text{new}} = u$, $\eta \in (0,1)$.
3. Update of the Trust-Region: Δ by means of ρ. Iterate!

Trust-Region Methods

Iterative Method, initial iterate can be chosen almost arbitrary

1. Newton-step: Solve

$$s \in \mathbb{R}^n : \psi(s) = \frac{1}{2}\langle s, Bs\rangle + \langle \nabla J(u), s\rangle = \min!$$
$$\text{such that } \|s\| \leq \Delta,\ u + s \in \mathcal{B}$$

   where B is a symmetric approximation the Hessian(Quasi-Newton-Method)

2. Acceptance: $\rho = \frac{J(u+s)-J(u)}{\psi(s)} \geq \eta$ then: $u^{\text{new}} = u + s$, otherwise $u^{\text{new}} = u$, $\eta \in (0,1)$.
3. Update of the Trust-Region: Δ by means of ρ. Iterate!

Theorem

If $\psi(s) = \min!$ is solved accurately enough, the gradients and B are bounded on a compact set, then the method computes a globally converging sequence of iterates

Towards Large-Scale Optimization

Trust-Region (and also Linesearch) methods

- rescale the Newton correction (a priori/a posteriori)
- $\Rightarrow$ only if a sufficient decrease of the objective function can be achieved, the (scaled) correction will be applied

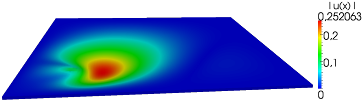
Towards Large-Scale Optimization

Trust-Region (and also Linesearch) methods

- rescale the Newton correction (a priori/a posteriori)
- $\Rightarrow$ only if a sufficient decrease of the objective function can be achieved, the (scaled) correction will be applied

Rescaling

- depends on the strongest nonlinearity of the objective function
- might tremendously slow down convergence
- does not depend on the quality of search directions s



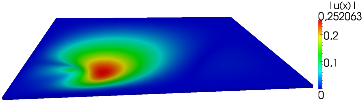
Towards Large-Scale Optimization

Trust-Region (and also Linesearch) methods

- rescale the Newton correction (a priori/a posteriori)
- ⇒ only if a sufficient decrease of the objective function can be achieved, the (scaled) correction will be applied

Rescaling

- depends on the strongest nonlinearity of the objective function
- might tremendously slow down convergence
- does not depend on the quality of search directions s



Aim

Since local nonlinearities govern the whole computation:
define strategies which improve the rates of convergence.

Towards Large-Scale Problems

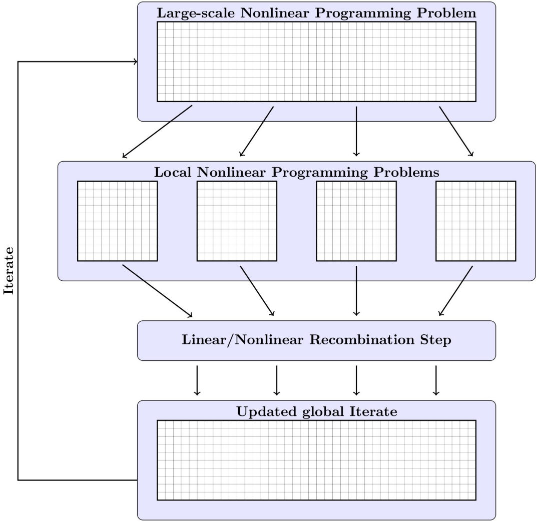
Standard Approach

- Linearize Outer nonlinear iteration
- Decompose Parallel solution of the inner linear problem
- Convergence Control Linesearch, Trust region

Alternative

- Nonlinear Decomposition Decompose into many small nonlinear problems
- Nonlinear Solve Solve small nonlinear problems in parallel
- Convergence Control Recombination step

Nonlinear Domain Decomposition Scheme



Concept: APTS

The APTS method

1. Decompose $\mathbb{R}^n$ into N subsets D_k such that $\mathbb{R}^n = \bigcup_k I_k D_k \subset \mathbb{R}^n$.

Concept: APTS

The APTS method

1. Decompose $\mathbb{R}^n$ into N subsets D_k such that $\mathbb{R}^n = \bigcup_k I_k D_k \subset \mathbb{R}^n$.
2. Employ on each D_k a Trust-Region method to solve

$$s_k \in \mathcal{B}_k : H_k(P_k u^G + s_k) < H_k(P_k u^G) \text{ such that } \|I_k s_k\| \leq \Delta^G$$

where
- $u^G \in \mathbb{R}^n$ is the current global iterate, Δ^G the current global Trust-Region radius,
- $\mathcal{B}_k$ local admissible corrections,
- $H_k : D_k \to \mathbb{R}$ a particular, local objective function,
- $I_k : D_k \to \mathbb{R}^n$ (prolongation) and $P_k : \mathbb{R}^n \to D_k$ (Projection)

Concept: APTS

The APTS method

1. Decompose $\mathbb{R}^n$ into N subsets D_k such that $\mathbb{R}^n = \bigcup_k I_k D_k \subset \mathbb{R}^n$.
2. Employ on each D_k a Trust-Region method to solve

$$s_k \in \mathcal{B}_k : H_k(P_k u^G + s_k) < H_k(P_k u^G) \text{ such that } \|I_k s_k\| \leq \Delta^G$$

   where
   - $u^G \in \mathbb{R}^n$ is the current global iterate, Δ^G the current global Trust-Region radius,
   - $\mathcal{B}_k$ local admissible corrections,
   - $H_k : D_k \to \mathbb{R}$ a particular, local objective function,
   - $I_k : D_k \to \mathbb{R}^n$ (prolongation) and $P_k : \mathbb{R}^n \to D_k$ (Projection)
3. Combine s_k as follows

$$u^{G,\text{new}} = \begin{cases} u^G + \sum_k I_k s_k & \text{if } \rho_A = \frac{J(u^G) - J(u^G + \sum_k I_k s_k)}{\sum_k (H_k(P_k u^G) - H_k(P_k u^G + s_k))} \geq \eta \\ u^G & \text{otherwise} \end{cases}$$

   where $I_k : D_k \to \mathbb{R}^n$. Update Δ^G by means of ρ_A.

Concept: APTS

The APTS method

1. Decompose $\mathbb{R}^n$ into N subsets D_k such that $\mathbb{R}^n = \bigcup_k I_k D_k \subset \mathbb{R}^n$.
2. Employ on each D_k a Trust-Region method to solve

$$s_k \in \mathcal{B}_k : H_k(P_k u^G + s_k) < H_k(P_k u^G) \text{ such that } \|I_k s_k\| \leq \Delta^G$$

   where
   - $u^G \in \mathbb{R}^n$ is the current global iterate, Δ^G the current global Trust-Region radius,
   - $\mathcal{B}_k$ local admissible corrections,
   - $H_k : D_k \to \mathbb{R}$ a particular, local objective function,
   - $I_k : D_k \to \mathbb{R}^n$ (prolongation) and $P_k : \mathbb{R}^n \to D_k$ (Projection)
3. Combine s_k as follows

$$u^{G,\text{new}} = \begin{cases} u^G + \sum_k I_k s_k & \text{if } \rho_A = \frac{J(u^G) - J(u^G + \sum_k I_k s_k)}{\sum_k (H_k(P_k u^G) - H_k(P_k u^G + s_k))} \geq \eta \\ u^G & \text{otherwise} \end{cases}$$

   where $I_k : D_k \to \mathbb{R}^n$. Update Δ^G by means of ρ_A.
4. Compute $\tilde{s}$ employing a Trust-Region method. $u^{G,\text{new}+1} = u^{G,\text{new}} + \tilde{s}$

The local Objective Function [Nash '00]

Choose the particular nonlinear, local objective function

$$H_k(u_k) = J_k(u_k) + \langle R_k \nabla J(u^G) - \nabla J_k(P_k u^G), u_k \rangle$$

- J_k is an a priori given nonlinear function (continuously differentiable)
- $R_k = (I_k)^T$

The local Objective Function [Nash '00]

Choose the particular nonlinear, local objective function

$$H_k(u_k) = J_k(u_k) + \langle R_k \nabla J(u^G) - \nabla J_k(P_k u^G), u_k \rangle$$

- J_k is an a priori given nonlinear function (continuously differentiable)
- $R_k = (I_k)^T$

Properties of the coupling term

It holds $\nabla H_k(P_k u^G) = R_k \nabla J(u^G)$. This yields

$$\frac{J(u^G + \sum_k I_k s_k) - J(u^G)}{\sum_k (H_k(P_k u^G + s_k) - H_k(P_k u^G))} \to 1 \qquad \text{for } \|s_k\| \to 0$$

Convergence to First–Order Critical Points

Convergence to first-order critical points

Theorem: If the search directions/corrections are chosen sufficiently well, the norm of the gradients and of B are either bounded on a compact set, then APTS is globally convergent.

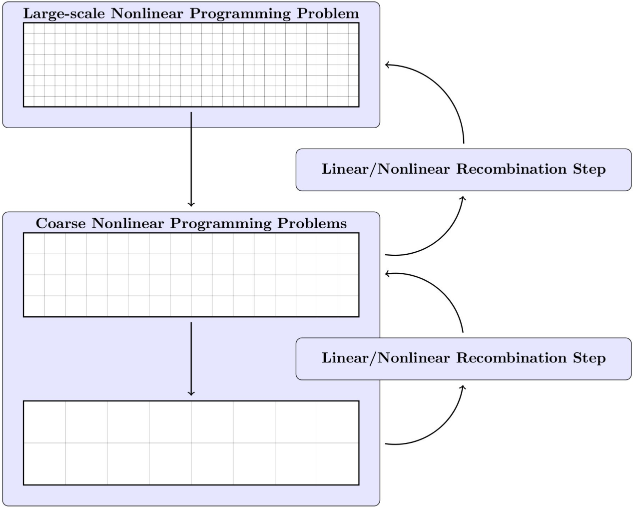
Convergence to First–Order Critical Points

Convergence to first-order critical points

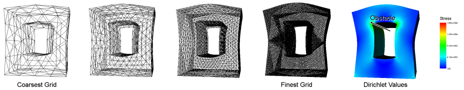
Theorem: If the search directions/corrections are chosen sufficiently well, the norm of the gradients and of B are either bounded on a compact set, then APTS is globally convergent.

Even more: global convergence can be guaranteed without global smoothing, if an (overlapping) domain decomposition is employed.

Nonlinear Domain Decomposition Scheme



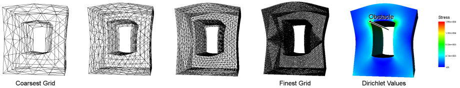
RMTR strategy [Gratton et al. 2008; Gratton et al. 2009; Groß, K' 2009]



The RMTR method

1. **compute** m_1 pre–smoothing trust–region steps to approximately solve
$H_k(u_k) < H_k(P_{k+1}u_{k+1}) \qquad \text{w.r.t } u_k \in \mathcal{B}_k, \|u_k\| \leq \Delta_k$

RMTR strategy [Gratton et al. 2008; Gratton et al. 2009; Groß, K' 2009]



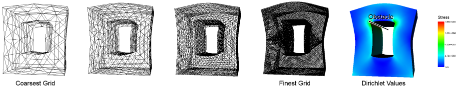
The RMTR method

1. **compute** m_1 pre–smoothing trust–region steps to approximately solve
   $H_k(u_k) < H_k(P_{k+1}u_{k+1}) \qquad$ w.r.t $u_k \in \mathcal{B}_k$, $\|u_k\| \leq \Delta_k$
2. **if** (k is not coarsest level)
   - Compute $\mathcal{B}_{k-1}$, and H_{k-1}, $u_{k-1,0} = P_k u_{k,m_1}$
   - call RMTR on level $k-1$ and receive a correction s_{k-1}

$$u_{k,m_1+1} = \begin{cases} u_{k,m_1} + I_{k-1}s_{k-1} & \text{if } \rho_M = \frac{H_k(u_{k,m_1}) - H_k(u_{k,m_1} + I_{k-1}s_{k-1})}{H_{k-1}(P_k u_{k,m_1}) - H_{k-1}(P_k u_{k,m_1} + s_{k-1})} \geq \eta \\ u_{k,m_1} & \text{otherwise} \end{cases}$$

   - Update trust-region Δ_{k,m_1+1}

RMTR strategy [Gratton et al. 2008; Gratton et al. 2009; Groß, K' 2009]



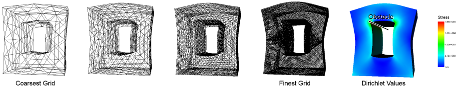
The RMTR method

1. **compute** m_1 pre–smoothing trust–region steps to approximately solve
$H_k(u_k) < H_k(P_{k+1}u_{k+1}) \qquad$ w.r.t $u_k \in \mathcal{B}_k, \|u_k\| \leq \Delta_k$
2. **if** (k is not coarsest level)
   - Compute $\mathcal{B}_{k-1}$, and H_{k-1}, $u_{k-1,0} = P_k u_{k,m_1}$
   - call RMTR on level $k-1$ and receive a correction s_{k-1}

$$u_{k,m_1+1} = \begin{cases} u_{k,m_1} + I_{k-1}s_{k-1} & \text{if } \rho_M = \frac{H_k(u_{k,m_1}) - H_k(u_{k,m_1} + I_{k-1}s_{k-1})}{H_{k-1}(P_k u_{k,m_1}) - H_{k-1}(P_k u_{k,m_1} + s_{k-1})} \geq \eta \\ u_{k,m_1} & \text{otherwise} \end{cases}$$

   - Update trust-region Δ_{k,m_1+1}
3. **compute** m_2 post–smoothing trust–region steps to approximately solve
$H_k(u_k) < H(u_{k,m_1+1}) \qquad$ w.r.t $u_k \in \mathcal{B}_k, \|u_k\| \leq \Delta_k$

RMTR strategy [Gratton et al. 2008; Gratton et al. 2009; Groß, K' 2009]



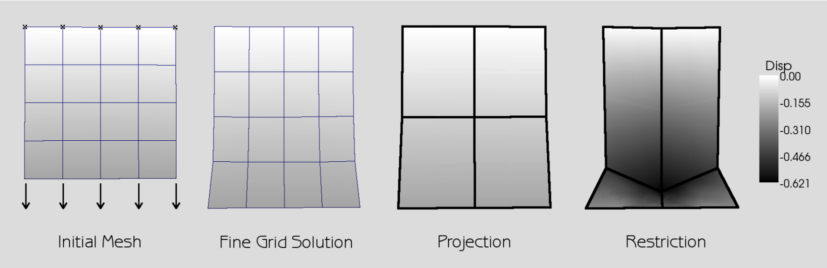
The RMTR method

1. **compute** m_1 pre–smoothing trust–region steps to approximately solve
   $H_k(u_k) < H_k(P_{k+1}u_{k+1}) \qquad$ w.r.t $u_k \in \mathcal{B}_k, \|u_k\| \leq \Delta_k$
2. **if** (k is not coarsest level)
   - Compute $\mathcal{B}_{k-1}$, and H_{k-1}, $u_{k-1,0} = P_k u_{k,m_1}$
   - call RMTR on level $k-1$ and receive a correction s_{k-1}

$$u_{k,m_1+1} = \begin{cases} u_{k,m_1} + I_{k-1}s_{k-1} & \text{if } \rho_M = \frac{H_k(u_{k,m_1}) - H_k(u_{k,m_1} + I_{k-1}s_{k-1})}{H_{k-1}(P_k u_{k,m_1}) - H_{k-1}(P_k u_{k,m_1} + s_{k-1})} \geq \eta \\ u_{k,m_1} & \text{otherwise} \end{cases}$$

   - Update trust-region Δ_{k,m_1+1}
3. **compute** m_2 post–smoothing trust–region steps to approximately solve
   $H_k(u_k) < H(u_{k,m_1+1}) \qquad$ w.r.t $u_k \in \mathcal{B}_k, \|u_k\| \leq \Delta_k$
4. **return** final iterate

Projection vs. Restriction



Comparison of initial mesh, fine level iterate, L^2-projected and restricted iterate – example in $3d$
standard restriction leads to Poor approximation of the fine level iterate

MPTS

MPTS: a generalization of RMTR

Almost arbitrary domain decomposition methods possible:

- Multigrid methods
- Alternating domain decomposition methods and nonlinear Jacobi methods

MPTS

MPTS: a generalization of RMTR

Almost arbitrary domain decomposition methods possible:

- Multigrid methods
- Alternating domain decomposition methods and nonlinear Jacobi methods

Convergence to first-order critical points

Theorem: If the search directions/corrections are chosen sufficiently well, the norm of the gradients and of B are either bounded on a compact set, then MPTS is globally convergent.

MPTS

MPTS: a generalization of RMTR

Almost arbitrary domain decomposition methods possible:

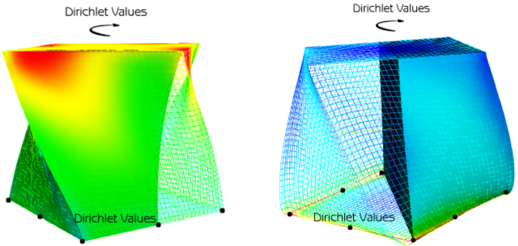
- Multigrid methods
- Alternating domain decomposition methods and nonlinear Jacobi methods

Convergence to first-order critical points

Theorem: If the search directions/corrections are chosen sufficiently well, the norm of the gradients and of B are either bounded on a compact set, then MPTS is globally convergent.

Even more: global convergence can be guaranteed without global smoothing, if an (overlapping) domain decomposition is employed.

Application: Nonlinear Mechanics of Large Deformations



Nonlinear elastic material law

Linear elastic material law

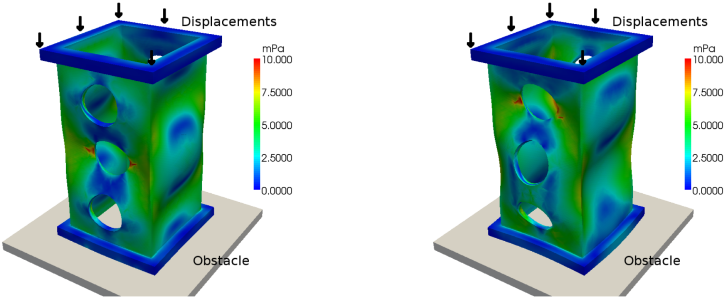
Stored energy function for Ogden materials [Ogden '72] (describes soft-tissues and rubber-like materials)

$$J(\mathbf{u}) = \int_\Omega d\,\text{tr}(E) + \frac{\lambda}{2}(\text{tr}(E))^2 + (\mu - d)\text{tr}(E^2) - d\ln(\det(I + \nabla\mathbf{u}))dx$$

$$E = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u}), d > 0$$

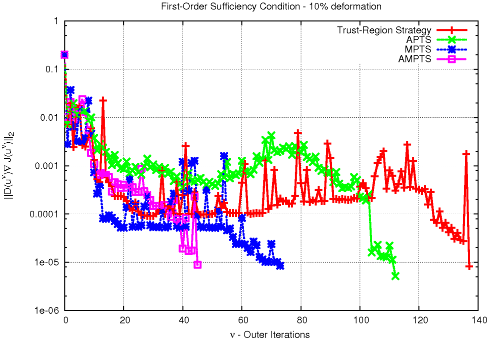
Barrier function: $\ln(\det(I + \nabla\mathbf{u}))$, penalizes element volume decrease.

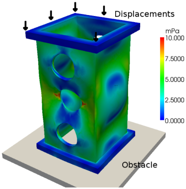
Cylinder Contact Problem

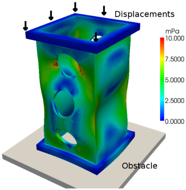


- Energy optimal displacements
- Bifurcation: energy functional is nonconvex and has at least these two solutions!
- 323,994 unknowns
- 8 processors

Cylinder Contact Problem - Performance of Trust-Region Methods



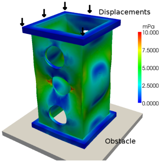


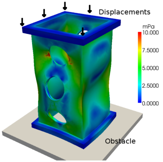


- Energy optimal displacements
- First-order sufficiency conditions $\|\nabla J(u)\|_2$ after each Trust-Region step; Comparison between seq. Trust-Region, APTS, MPTS, combined APTS/MPTS = AMPTS
  ($\mathcal{F} \widehat{=}$ 4 local Trust-Region steps on each D_k, 4 global Trust-Region steps in order to compute $\tilde{s}$)

Cylinder Contact Problem - Performance of Trust-Region Methods

	Newton it.	parallel cg it.	Time
seq. Trust-Region	137	54,800	1.0
APTS	112	44,800	1.10
MPTS	73	29,200	0.61
AMPTS	45	18,000	0.50





- Energy optimal displacements
- runtime comparison ($\mathcal{F} \mathrel{\widehat{=}} 4$ local Trust-Region steps on each D_k, 4 global Trust-Region steps in order to compute $\tilde{s}$)
- time is measured relatively to the sequential Trust-Region method
- 323,994 unknowns
- 8 processors

ASPIN Method [Cai, Keyes '00]

ASPIN

1. (Local solution phase) On each processor $k = 1, \dots, \mathcal{N}$, approximately solve

$$s_k \in \mathbb{R}^{n_k} : \nabla H_k(P_k u^i + s_k) = 0$$

ASPIN Method [Cai, Keyes '00]

ASPIN

1. (Local solution phase) On each processor $k = 1, \ldots, \mathcal{N}$, approximately solve

$$s_k \in \mathbb{R}^{n_k} : \nabla H_k(P_k u^i + s_k) = 0$$

2. (Global solution phase) Then compute the actual Newton correction s^i:

$$s^i \in \mathbb{R}^n : \qquad (C^i)^{-1} \nabla^2 J(u^i) s^i = \sum_k I_k s_k \approx -(C^i)^{-1} \nabla J(u^i)$$

Here C_i^{-1} is the additive Schwarz preconditioning matrix

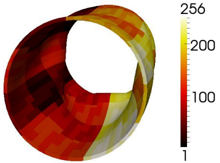
$$C_i^{-1} = \sum_k \left[I_k \left(R_k (\nabla^2 J(u^i)) I_k \right)^{-1} R_k \right]$$
$$= \begin{pmatrix} (\nabla^2 J(u^i)_{00})^{-1} & & \\ & \ddots & \\ & & (\nabla^2 J(u^i)_{NN})^{-1} \end{pmatrix}$$

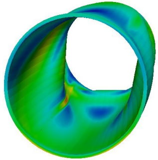
and I_k prolongation operators.

Globalized ASPIN – Overview

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$

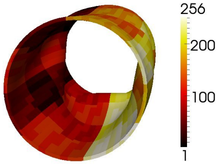


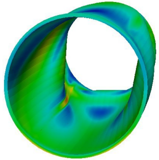


Globalized ASPIN – Overview

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$, the preconditioned gradient (based on $\sum_k I_k s_k$)

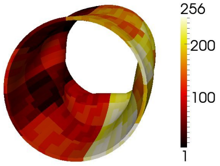


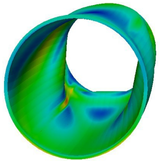


Globalized ASPIN – Overview

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$, the preconditioned gradient (based on $\sum_k I_k s_k$)
- Solve an QP problem in order to obtain the global correction s^i

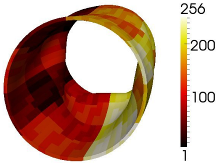


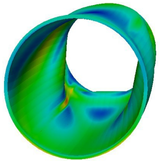


Globalized ASPIN – Overview

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$, the preconditioned gradient (based on $\sum_k I_k s_k$)
- Solve an QP problem in order to obtain the global correction s^i
- If $J(u^i) - J(u^i + s^i)$ decreases sufficiently, then $u^{i+1} = u^i + s^i$



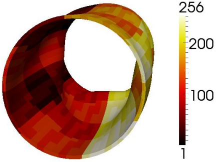


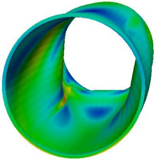
Globalization of ASPIN [GroßK' 2011]

Globalized ASPIN – Overview

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$, the preconditioned gradient (based on $\sum_k I_k s_k$)
- Solve an QP problem in order to obtain the global correction s^i
- If $J(u^i) - J(u^i + s^i)$ decreases sufficiently, then $u^{i+1} = u^i + s^i$
- Iterate!





The preconditioned Trust-Region model

We compute the global correction as the solution of $s \in \mathbb{R}^n$:

$$\widetilde{\psi}^i(s) = \frac{1}{2}\langle s, B^i s\rangle + \langle s, \tilde{g}^i\rangle = \min! \qquad \text{w.r.t. } \|s\| \leq \Delta_i^G$$

where

- $\tilde{g}^i \qquad = \qquad -C^i \sum_k I_k s_k$

The preconditioned Trust-Region model

We compute the global correction as the solution of $s \in \mathbb{R}^n$:

$$\widetilde{\psi}^i(s) = \frac{1}{2}\langle s, B^i s\rangle + \langle s, \tilde{g}^i\rangle = \min! \qquad \text{w.r.t. } \|s\| \leq \Delta_i^G$$

where

- $\tilde{g}^i \overset{\text{(just for this slide)}}{=} -C^i \sum_k I_k s_k$

The preconditioned Trust-Region model

We compute the global correction as the solution of $s \in \mathbb{R}^n$:

$$\widetilde{\psi}^i(s) = \frac{1}{2}\langle s, B^i s\rangle + \langle s, \tilde{g}^i\rangle = \min! \qquad \text{w.r.t. } \|s\| \leq \Delta_i^G$$

where

- $\tilde{g}^i \overset{\text{(just for this slide)}}{=} -C^i \sum_k I_k s_k$
- C^i is the inverse of the additive Schwarz preconditioning matrix eg

$\Rightarrow$ SQP version of ASPIN

The preconditioned Trust-Region model

We compute the global correction as the solution of $s \in \mathbb{R}^n$:

$$\widetilde{\psi}^i(s) = \frac{1}{2}\langle s, B^i s\rangle + \langle s, \tilde{g}^i\rangle = \min! \qquad \text{w.r.t. } \|s\| \le \Delta_i^G$$

where

- $\tilde{g}^i \overset{\text{(just for this slide)}}{=} -C^i \sum_k I_k s_k$
- C^i is the inverse of the additive Schwarz preconditioning matrix eg

$\Rightarrow$ SQP version of ASPIN

Preconditioned model

The preconditioned model can be considered as a perturbed Trust-Region model.

- Perturbed Trust-Region methods are well known [Toint 1988; Carter 1993; Conn et al. 1993]
- Applications for these methods: numerical differentiation and constrained optimization

The preconditioned Trust-Region model

We compute the global correction as the solution of $s \in \mathbb{R}^n$:

$$\widetilde{\psi}^i(s) = \frac{1}{2}\langle s, B^i s\rangle + \langle s, \tilde{g}^i\rangle = \min! \qquad \text{w.r.t. } \|s\| \leq \Delta_i^G$$

where

- $\tilde{g}^i \overset{\text{(just for this slide)}}{=} -C^i \sum_k I_k s_k$
- C^i is the inverse of the additive Schwarz preconditioning matrix eg

$\Rightarrow$ SQP version of ASPIN

Preconditioned model

The preconditioned model can be considered as a perturbed Trust-Region model.

- Perturbed Trust-Region methods are well known [Toint 1988; Carter 1993; Conn et al. 1993]
- Applications for these methods: numerical differentiation and constrained optimization
- Here: perturbation resulting from the nonlinear, additive solution process

Handling the Perturbation

Modified Sufficient Decrease Condition

In order to prove a sufficient decrease:

- a constraint on $\tilde{g}_i$: $\|\tilde{g}^i - g^i\| \leq \Delta_i^L \leq \Delta_i^G$ where $g_i = \nabla J(u_i)$
- Δ_i^L will be adaptively updated

Globalized ASPIN

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$ based on $C^i \cdot \sum_k I_k s_k$ and g^i such that $\|\tilde{g}^i - g^i\| \leq \Delta_i^L$

Globalized ASPIN

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$ based on $C^i \cdot \sum_k I_k s_k$ and g^i such that $\|\tilde{g}^i - g^i\| \leq \Delta_i^L$
- Solve

$$s^i \in \mathbb{R}^n : \widetilde{\psi}^i(s^i) = \min! \qquad \text{w.r.t. } \|s^i\| \leq \Delta_i^G$$

in order to obtain the global correction s^i

Globalized ASPIN

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$ based on $C^i \cdot \sum_k I_k s_k$ and g^i such that $\|\tilde{g}^i - g^i\| \leq \Delta_i^L$
- Solve

$$s^i \in \mathbb{R}^n : \widetilde{\psi}^i(s^i) = \min! \qquad \text{w.r.t. } \|s^i\| \leq \Delta_i^G$$

  in order to obtain the global correction s^i
- If the modified sufficient decrease condition holds: increase Δ_i^L otherwise decrease it

Globalized ASPIN

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \text{min!}$
  - Compute $\tilde{g}^i$ based on $C^i \cdot \sum_k I_k s_k$ and g^i such that $\|\tilde{g}^i - g^i\| \leq \Delta_i^L$
- Solve

$$s^i \in \mathbb{R}^n : \widetilde{\psi}^i(s^i) = \text{min!} \qquad \text{w.r.t. } \|s^i\| \leq \Delta_i^G$$

  in order to obtain the global correction s^i
- If the modified sufficient decrease condition holds: increase Δ_i^L otherwise decrease it
- If

$$\frac{J(u^i) - J(u^i + s^i)}{-\widetilde{\psi}^i(s^i)} \geq \eta$$

  increase Δ_i^G and $u^{i+1} = u^i + s^i$
  otherwise: decrease Δ_G^{i+1} and $u^{i+1} = u^i$

Globalized ASPIN

The Algorithm

- In parallel:
  - Compute $s_k \in \mathbb{R}^{n_k} : H_k(P_k u^i + s_k) = \min!$
  - Compute $\tilde{g}^i$ based on $C^i \cdot \sum_k I_k s_k$ and g^i such that $\|\tilde{g}^i - g^i\| \leq \Delta_i^L$
- Solve

$$s^i \in \mathbb{R}^n : \widetilde{\psi}^i(s^i) = \min! \qquad \text{w.r.t. } \|s^i\| \leq \Delta_i^G$$

  in order to obtain the global correction s^i
- If the modified sufficient decrease condition holds: increase Δ_i^L otherwise decrease it
- If

$$\frac{J(u^i) - J(u^i + s^i)}{-\widetilde{\psi}^i(s^i)} \geq \eta$$

  increase Δ_i^G and $u^{i+1} = u^i + s^i$
  otherwise: decrease Δ_G^{i+1} and $u^{i+1} = u^i$
- Iterate!

Convergence to a First-Order Critical Point

- For the given initial iterate $u^0 \in \mathbb{R}^n$ in the Algorithm we assume that the level set

$$\mathcal{L} = \{u \in \mathbb{R}^n \mid J(u) \leq J(u^0)\}$$

  is compact.
- We assume that J is continuously differentiable on $\mathcal{L}$. Then we have that the norms of the gradients are bounded by a constant $C_g > 0$, i.e., $\|\nabla J(u)\| \leq C_g$ for all $u \in \mathcal{L}$.
- There exists a constant $C_B > 0$ such that for all iterates $u^i \in \mathcal{L}$ and for each symmetric matrix B^i employed in each $\widetilde{\psi}^i$ the inequality $\|B^i\| \leq C_B$ is satisfied.

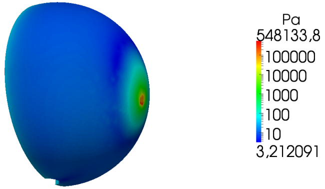
Theorem

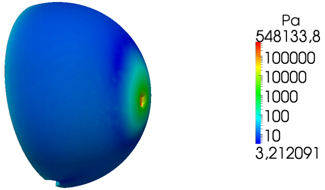
Let the assumptions on J and on B hold. In this case we obtain that the sequence of iterates generated by the globalized ASPIN algorithms has the property

$$\lim_{i \to \infty} \|\nabla J(u^i)\| = 0$$

Deformation of a Semi-Sphere

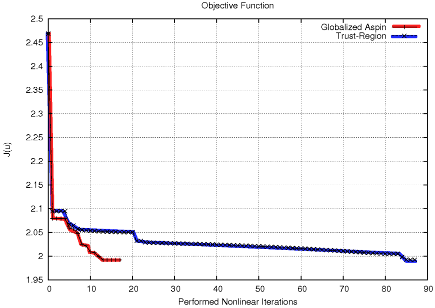
- pushing a sphere in direction of a small obstacle
- 881,280 unknowns
- No bifurcations in the simulations
  We will see
  - (highly) nonlinear behavior of the objective function
  - but: exactly the same solution
- QP solver:
  - Steihaug-Toint CG
  - Monotone Multigrid Smoother
  - Fine grid smoother: symmetric projected Gauß-Seidel
  - Coarse grid smoother: additive Schwarz
  - and Cauchy point computation + comparison
- computations carried out at CSCS, Switzerland

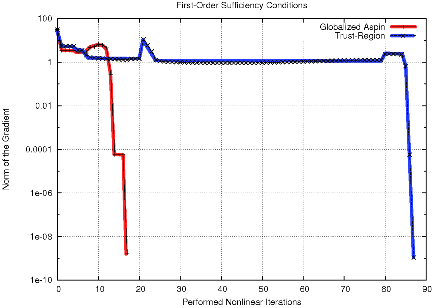




Reference geometry and deformed geometry (according to the solution)

Comparisons – 240 Cores



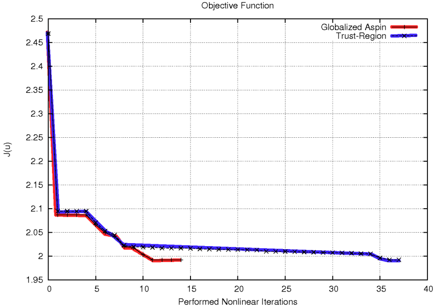


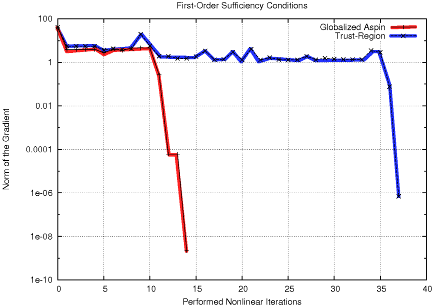
Evolution of the objective function $J(u^i)$ and the norm of the gradient $\|g^i\|$ for Trust-Region and globalized Aspin computations with 240 processors

	Trust-Region	G-ASPIN
Overall Time	460.13	196.49
Solver global QP Problem	328.15	70.72
Solver local QP Problem	—	4.43
Assembling	65.08	66.39

Computation times with 240 cores in seconds

Comparisons – 1920 Cores



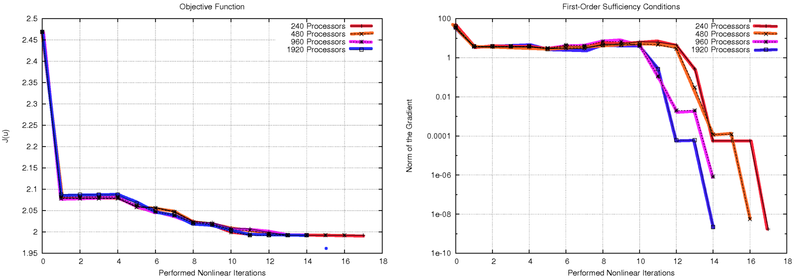


Evolution of the objective function $J(u^i)$ and the norm of the gradient $\|g^i\|$ for Trust-Region and globalized Aspin computations with 1920 processors

	Trust-Region	G-ASPIN
Overall Time	61.58	44.50
Solver global QP Problem	52.48	22.26
Solver local QP Problem	—	0.30
Assembling	6.32	13.89

Computation times with 1920 cores in seconds

Comparisons



Evolution of the objective function $J(u^i)$ and the norm of the gradient $\|g^i\|$ for globalized Aspin employing different numbers of processors

	240 cores	480 cores	960 cores	1920 cores
Overall Time	196.49	105.98	57.24	44.50
Solver global TR problem	70.72	40.43	25.25	22.26
Solver local QP Problem	4.43	1.82	0.43	0.30
Assembling	66.39	40.17	19.32	13.89
Nonlinear Iterations	17	16	14	14

Computation time in seconds.

Fault Tolerance of the APLS/APTS Strategies

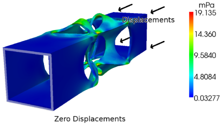
Severeness of possible fault scanarios

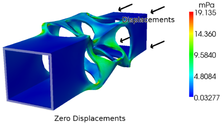
Node dies

- during local solution: having $s_k = 0$ is integral concept of APLS/APTS – almost the same convergence theory applies
- in recombination step
  - while submitting s_k: will yield $s_k = 0$ (see above)
  - while solving for α: might spoil the convergence and must be dealt with as described on the previous slides.
- in global smoothing step:
  - this step is optional (might slow down convergence)
  - if the step is computed and accepted, convergence must be ensured.

Conclusion

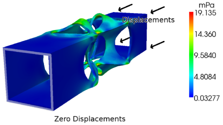
- The following multiplicative and additive Trust-Region strategies:
  - APTS
  - MPTS
- A globalization for ASPIN was presented
  - extension to ASPIN: reduces to ASPIN if "iterates are sufficiently close to local solution"
  - Convergence can be proven due to interpretation as perturbed Trust-Region approach
- Application to NLPs from nonlinear mechanics: solution is
  - efficient
  - reliable

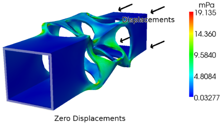




Conclusion

- The following multiplicative and additive Trust-Region strategies:
  - APTS
  - MPTS
- A globalization for ASPIN was presented
  - extension to ASPIN: reduces to ASPIN if "iterates are sufficiently close to local solution"
  - Convergence can be proven due to interpretation as perturbed Trust-Region approach
- Application to NLPs from nonlinear mechanics: solution is
  - efficient
  - reliable





Thank you for your attention.