Domain Decomposition Methods for Constrained and Nonlinear Problems

R. Krause

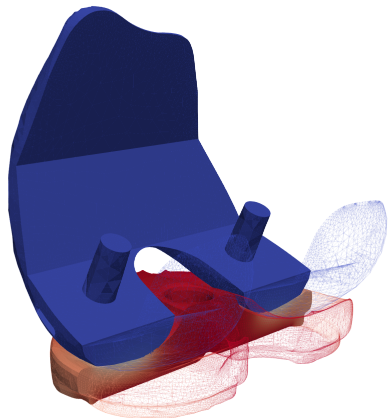
V. Braglia, L. Gaedke-Merzhäuser, A. Gharaviri,
A. Kopaničáková, H. Kothari, M. Nestola, S. Pezzuto, C. von Planta, P. Zulian

Euler Institute
USI
Università della Svizzera italiana, Lugano

CIRM 2022, Marseille

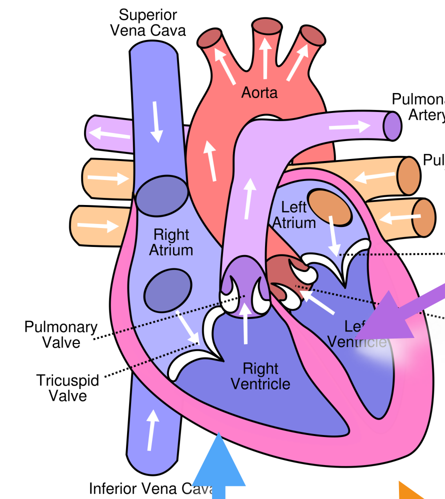
Constraints and Multiphysics

Computational Mechanics



High-performance computing

Computational Medicine



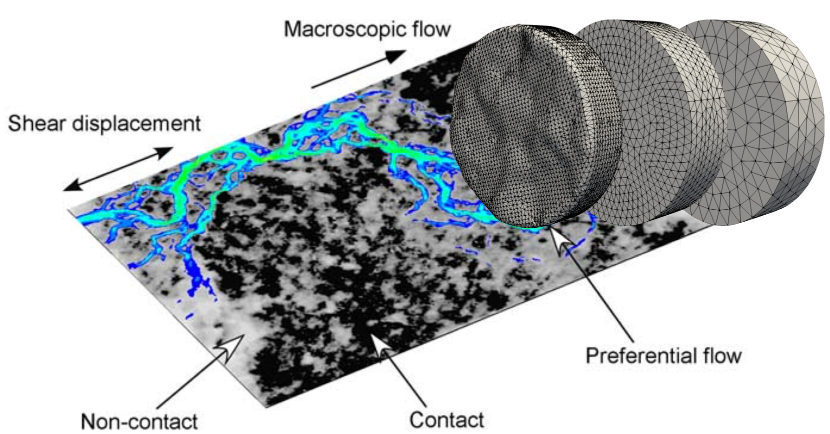
Contact

Fluid

Elasticity/

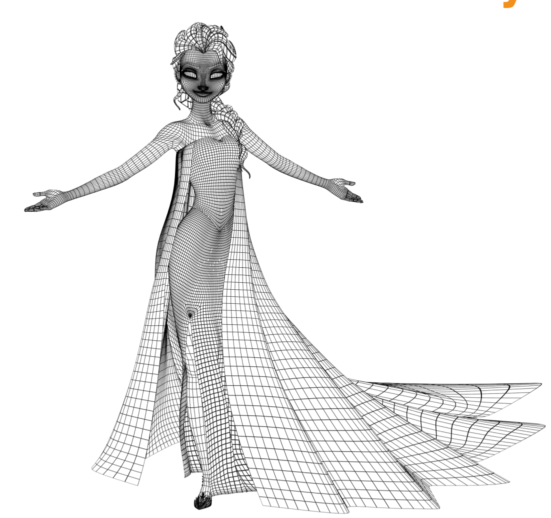
Constraints
Nonlinearities
Coupled Problems

SCCER SoE
SWISS COMPETENCE CENTER for ENERGY RESEARCH
SUPPLY of ELECTRICITY



Computational Geophysics

Machine Learning



Cloth simulations for animations

Multilevel Solution Methods

Optimisation - Minimization - Solution

- $\mathcal{J} : \mathbf{H} \longrightarrow \mathbb{R}$ (non-)convex functional: stored energy function, loss function, error, ...
- 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 conditions: solve non–linear equation

$$\mathrm{F}(\mathrm{u}) = \nabla \mathrm{J}(\mathrm{u}) = 0 \Leftrightarrow \quad J'(\mathbf{u})(\mathbf{v}) = 0\,, \qquad \mathbf{v} \in H\,.$$

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

Dual view: Either minimize J or solve F(u) = 0

Nonlinear Problem

Necessary condition of first order for a minimizer

H, W Banach spaces, $F\colon D \subset H \longrightarrow W$, D open, $F \in C^1(D)$. Find $u^* \in D$ such that

$$F(u^*) = 0$$

Construct sequence if iterates $x^k \in X$, in D, $k > 0$, via

$$u^{k+1} = u^k + \alpha^k c^k$$

with $u^k \to u^*$.

- u^k correction or step, $\alpha^k > 0$ steplength
- special case $F = \nabla J$
- solution may not be unique

Newton's method

$u^k \in H$, Newton's method replaces F by the linear model

$$F(u^k + c^k) \approx F(u^k) + F'(u^k)c^k = 0$$

leading to the the Newton correction

$$c^k = -F'(u^k)^{-1}F(u^k)$$

and the Newton update

$$c^{k+1} = c^k + \alpha^k c^k\,, \tag{1}$$

- $\alpha^k \geq 0$ damping or line-search parameter for the Newton correction.
- $F'(u^k)$ Fréchet derivative / Jacobian. Here: $H = \mathbb{R}^n$
- Invariance under affine transformations [Ortega '70, Deuflhard, Heindl '79, Deuflhard ..., '11]

First linearize, then solve

From Optimization to Multigrid

Properties of the Newton Direction [Deuflhard '11]

General level set function $T(u|A) = \frac{1}{2}\|AF(u)\|_2^2$, A regular:

$$\nabla T(u|A) = (AF'(u))^T(AF(u))$$

The "natural" choice $A = F'(u)^{-1}$ leads to

$$-\nabla T(u|F'(u)^{-1}) = -F'(u)^{-1}F(u)$$

For $J(u) = \frac{1}{2}(Au, u) - (f, u)$ we get

$$-F'(u)^{-1}F(u) = -\nabla^2 J(u)\nabla J(u) = A^{-1}(f - Au) = A^{-1}(Au^* - Au)$$

- The Newton correction is direction of steepest descent for $T(u|F'(u)^{-1})$
- Damping strategy for the exact Newton method can be derived using $T(\cdot|A)$
- J convex and quadratic: Newton step leads to minimizer $u^* = u + A^{-1}(Au^* - Au)$
- Isolines of J and $T(\cdot|I)$ will form our energy-landscape

Newton Path

$$
\begin{aligned}
F(\bar{u}(\lambda)) &= (1-\lambda)F(u^0), \\
T(\bar{u}(\lambda)|A) &= (1-\lambda)^2 T(u^0|A), \\
\frac{d\bar{u}}{d\lambda} &= -F'(\bar{u})^{-1}F(u^0), \\
\bar{x}(0) &= x^0, \quad \bar{u}(1) = u^*, \\
\left.\frac{d\bar{u}}{d\lambda}\right|_{\lambda=0} &= -F'(u^0)^{-1}F(u^0) \equiv c^0,
\end{aligned}
$$

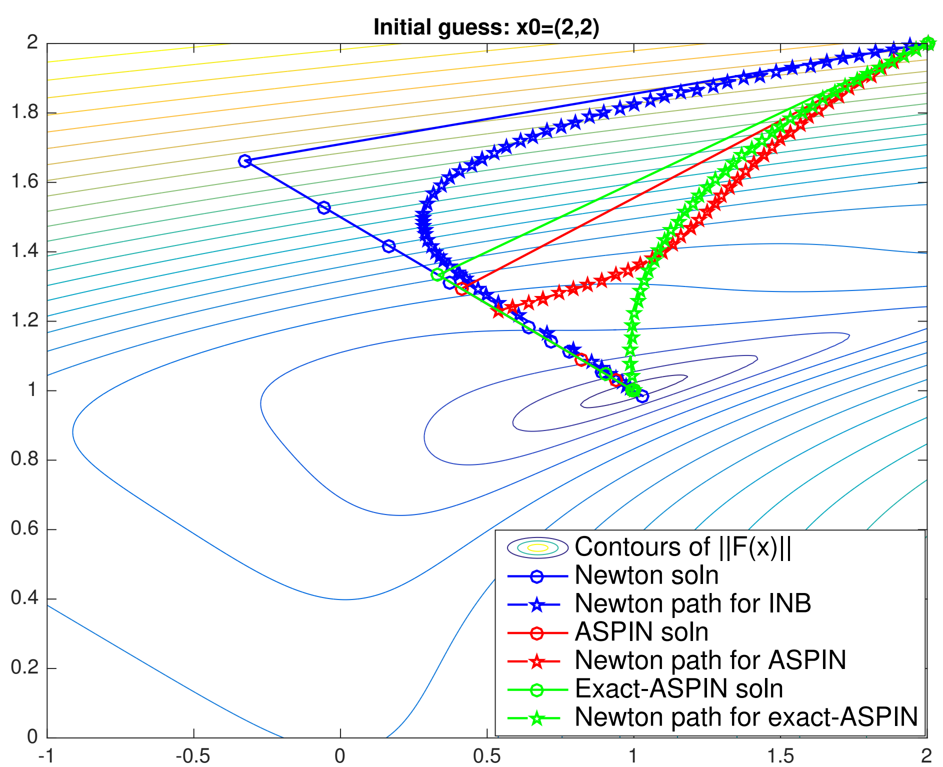
- connects start value and 'nearest' solution, or collapses (F' singular), or ends at ∂D[Davidenko '53, Deuflhard '72, '11]
- Newton correction (in the first step) ist tangent to the Newton path

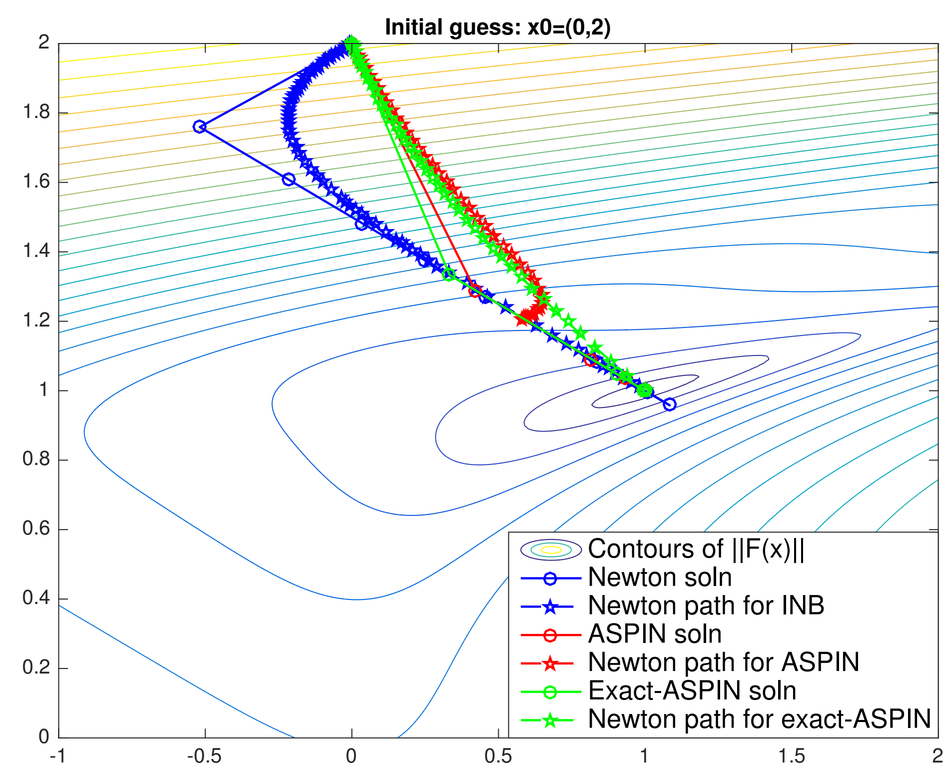
Newton Path: It's the direction

$$
\begin{aligned}
F_1(x_1, x_2) &= (x_1 - x_2^3 + 1)^3 - x_2^3, \\
F_2(x_1, x_2) &= x_1 + 2x_2 - 3.
\end{aligned}
$$

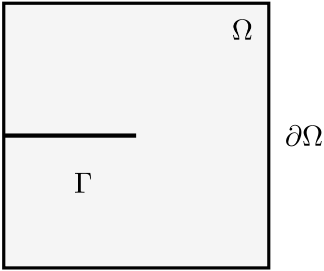
The exact solution is $u^* = [1, 1]^T$.

- INB: Inexact Newton with backtracking
- ASPIN: standard ASPIN
- Exact-ASPIN: ASPIN with analytical Jacobian for the preconditioned system





[Cai, Keyes; 2002; L. Liu, D. Keyes, K', 2018]



Outline:

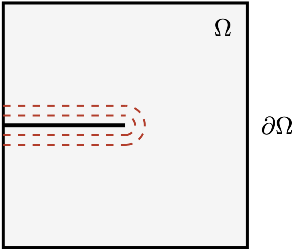
- Phase-field fracture model
- Affine similar trust-region method
  - Pseudo-transient continuation
  - Algorithmic design
  - Numerical results
- Large-scale simulation framework for pressure-induced fracture

Total potential energy: [Ambrosio, Tortorelli '90]

$$\Psi(\boldsymbol{u}) = \underbrace{\int_{\Omega\setminus\Gamma} \psi_e(\boldsymbol{u})\ d\Omega}_{\text{elastic energy}} + \underbrace{\int_{\Gamma} \mathcal{G}_c\ d\Gamma}_{\text{fracture energy}}$$

Regularized total potential energy: [Bourdin et al. '00; Miehe et al. '10]

$$\tilde{\Psi}(\boldsymbol{u}, c) = \underbrace{\int_{\Omega} g(c)\psi_e^+(\boldsymbol{u}) + \psi_e^-(\boldsymbol{u})\ d\Omega}_{\text{elastic energy}} + \underbrace{\int_{\Omega} \frac{\mathcal{G}_c}{c_w} \left(\frac{w(c)}{l_s} + l_s \mid \nabla c \mid^2 \right) d\Omega}_{\text{volumetric approximation of fracture energy}}$$



Phase field approach:

- The fracture problem is transformed into a continuous problem
- The crack is modeled in a diffused manner
- Transition represented by the phase field parameter, $c \in [0, 1]$
- l_s controls the thickness of the damaged region

Inexact Newton's method[Dembo et al. '82; Eisenstat and Walker '96]

Solve coupled problem: Find $(\boldsymbol{u}^*, c^*)$ such that

$$F(\boldsymbol{u}^*, c^*) = \begin{bmatrix} F_u(\boldsymbol{u}^*, c^*) \\ F_c(\boldsymbol{u}^*, c^*) \end{bmatrix} = \mathbf{0}$$

Inexact Newton's iteration to find coefficients $\mathbf{x} = \begin{bmatrix} \boldsymbol{u} \\ c \end{bmatrix}$

1. Find $\mathbf{p}^{(k)}$ by solving $F'(\mathbf{x}^{(k)})\mathbf{p}^{(k)} = -F(\mathbf{x}^{(k)})$, such that
$$\|F(\mathbf{x}^{(k)}) + F'(\mathbf{x}^{(k)})\mathbf{p}^{(k)}\| \leq \eta^{(k)} \|F(\mathbf{x}^{(k)})\|$$
2. Find $\alpha^{(k)}$ using a backtracking algorithm
3. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^k$

- Far from the solution, avoid solving Newton's equation exactly
- Linear system is solved using preconditioned Krylov method
- Forcing term $\eta^{(k)} \in [0, 1)$ induces tolerance with which Newton's eq. is solved

Properties of (inexact) Newton's methods

- Converges quadratically, if good initial guess is provided
- Typically exhibit very slow convergence until a local neighborhood of a solution is found
- Slow convergence due to unbalanced and highly localized nonlinearities
  - coupling between the displacement and the phase-field
  - locally varying material stiffness
  - steep gradients of the phase-field function

Nonlinear preconditioners can alleviate the aforementioned drawbacks

Nonlinear preconditioning[Cai, Keyes '02; Dolean et al. '16]

Instead of solving $F(\mathbf{x}) = 0$, we solve

$$\mathcal{H}(\mathbf{x}) = G(\underbrace{F(\mathbf{x})}_{\mathbf{y}}) = 0$$

Properties of the preconditioner G :

- $\mathcal{H}$ should have more balanced nonlinearities
- Solving $G(F(\mathbf{x})) = 0$ should be easier than solving $F(\mathbf{x}) = 0$
- If $G(\mathbf{y}) = 0$, then $\mathbf{y} = 0$
- $G \approx F^{-1}$ in some sense
- Evaluation of $G(F(\mathbf{v}))$, for some $\mathbf{v}$, should be cheap
- Multiplication of $(G(F(\mathbf{v})))'$ with $\mathbf{w}$ should be also easy

SPIN method[Cai, Keyes '02; Liu, Keyes '15 '16]

- Employ field-split approach, thus by decomposing $\mathbf{x}$ as $[\boldsymbol{u}, c]^\top$, i.e.,

$$G(F(\boldsymbol{u}, c)) := \mathcal{H}(\boldsymbol{u}, c) := \begin{bmatrix} \mathcal{H}_u(\boldsymbol{u}, c) \\ \mathcal{H}_c(\boldsymbol{u}, c) \end{bmatrix} = \mathbf{0}$$

- Explicit knowledge of preconditioner G is typically not available
- Construct $\mathcal{H}$ implicitly using knowledge about F and $\mathbf{x}$

ASPIN: Find $\mathcal{H}_u$ such that $F_u(\boldsymbol{u} + \mathcal{H}_u, c) = 0$

Find $\mathcal{H}_c$ such that $F_c(\boldsymbol{u}, c + \mathcal{H}_c) = 0$

$$\implies \mathcal{H}_A(\boldsymbol{u}, c) = \begin{bmatrix} \mathcal{H}_u(\boldsymbol{u}, c) \\ \mathcal{H}_c(\boldsymbol{u}, c) \end{bmatrix}$$

MSPIN: Find $\mathcal{H}_u$ such that $F_u(\boldsymbol{u} + \mathcal{H}_u, c) = 0$

Find $\mathcal{H}_c$ such that $F_c(\boldsymbol{u} + \mathcal{H}_u, c + \mathcal{H}_c) = 0$

$$\implies \mathcal{H}_M(\boldsymbol{u}, c) = \begin{bmatrix} \mathcal{H}_u(\boldsymbol{u}, c) \\ \mathcal{H}_c(\boldsymbol{u}, c) \end{bmatrix}$$

Jacobian of the preconditioned system (additive)

- Jacobian $\mathcal{H}'$ can be approximated as follows:

$$\mathcal{H}'_{\mathsf{A}} \approx \begin{bmatrix} F'_{uu} & \\ & F'_{cc} \end{bmatrix}^{-1} \underbrace{\begin{bmatrix} F'_{uu} & F'_{uc} \\ F'_{cu} & F'_{cc} \end{bmatrix}}_{F'}$$

- The global linear system is solved using Krylov methods, which require product $\mathcal{H}'_A \mathbf{v}$ to be performed efficiently.

$$\mathcal{H}'_A \mathbf{v} = \begin{bmatrix} F'_{uu} & \\ & F'_{cc} \end{bmatrix}^{-1} \underbrace{\begin{bmatrix} F'_{uu} & F'_{uc} \\ F'_{cu} & F'_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{v}_u \\ \mathbf{v}_c \end{bmatrix}}_{F'\mathbf{v}=\mathbf{w}} = \begin{bmatrix} F'_{uu} & \\ & F'_{cc} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{w}_u \\ \mathbf{w}_c \end{bmatrix}$$

$$\mathcal{H}'_A \mathbf{v} = \begin{bmatrix} (F'_{uu})^{-1} \mathbf{w}_u \\ (F'_{cc})^{-1} \mathbf{w}_c \end{bmatrix} = \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_c \end{bmatrix}$$

Considered solution strategies

- AM-ND: Alternate minimization with exact Newton (direct linear solver)
- AM-NK: Alternate minimization with exact Newton (Krylov linear solver)
- AM-INK: Alternate minimization with inexact Newton (Krylov linear solver)
- ASPIN: Additive Schwarz preconditioned inexact Newton
- MSPIN: Multiplicative Schwarz preconditioned inexact Newton

Implementation details

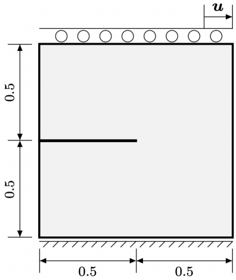
FEM discretization:

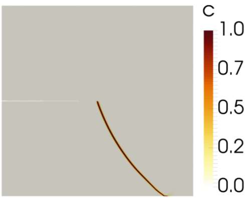
- Finite element framework MOOSE[Permann et al. '20]

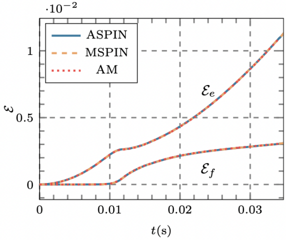
Implementation of solution strategies:

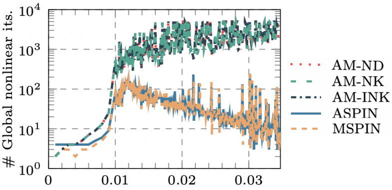
- Utopia[Zulian, Kopanicakova et al. '21] (`https://bitbucket.org/zulianp/utopia`)
- PETSc backend is used for linear algebra and linear solvers (Krylov/direct)

Shear test (gradual crack propagation)









Solver	Time (min)	Speedup with respect to			
		AM-ND	AM-NK	AM-INK	ASPIN
AM-ND	5,792.72	–	–	–	–
AM-NK	3,886.39	1.49	–	–	–
AM-INK	3,476.33	1.66	1.12	–	–
ASPIN	118.56	48.86	32.78	29.32	–
MSPIN	115.23	50.27	33.72	30.17	1.03

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,...

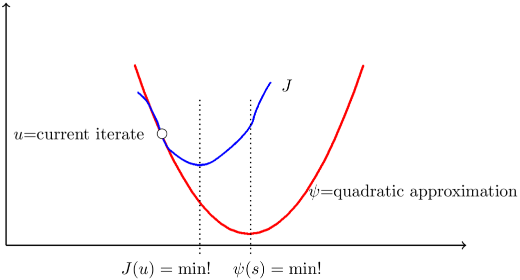
Trust-Region Methods

Iterative Method, initial iterate can be chosen almost arbitrary

1. Newton-step: Solve

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

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

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

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

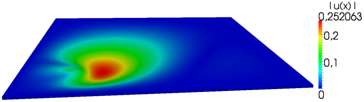
Towards Large-Scale Optimization

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

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

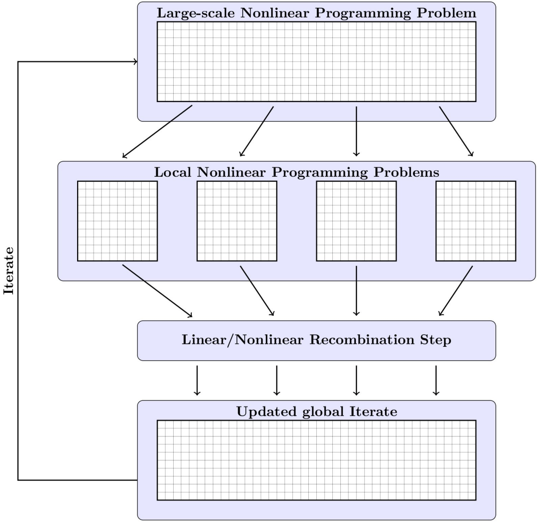
Standard Approach

- 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 $\mathbb{R}^n = \bigcup_k I_k D_k \subset \mathbb{R}^n$.
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

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

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

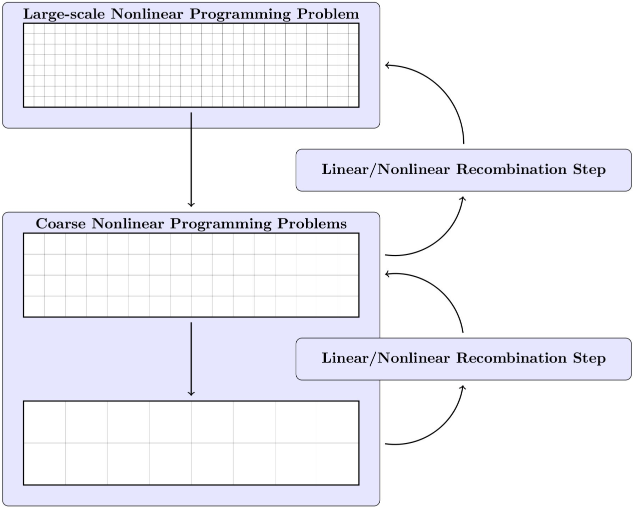
Convergence to First–Order Critical Points

Convergence to first-order critical points

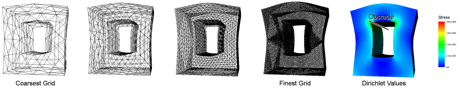
Theorem: If the search directions/corrections are chosen sufficiently well, the norm of the gradients and of B are either bounded on a compact set, then APTS is globally convergent.

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]



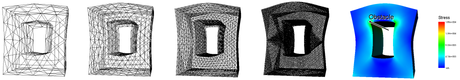
The RMTR method

1. **compute** m_1 pre–smoothing trust–region steps to approximately solve
   $H_k(u_k) < H_k(P_{k+1}u_{k+1}) \qquad$ 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]



Coarsest Grid | Finest Grid | Dirichlet Values

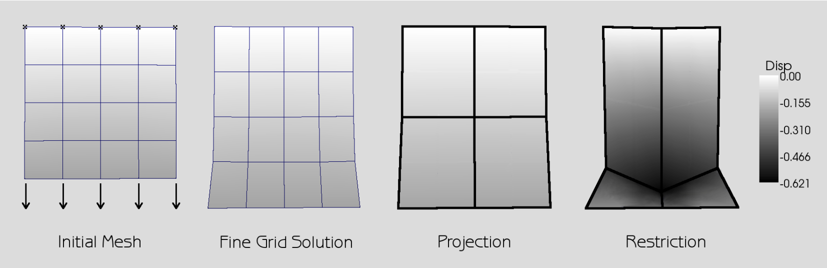
The RMTR method

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:

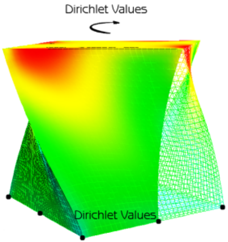
- Multigrid methods
- Alternating domain decomposition methods and nonlinear Jacobi methods

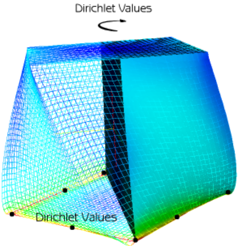
Convergence to first-order critical points

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

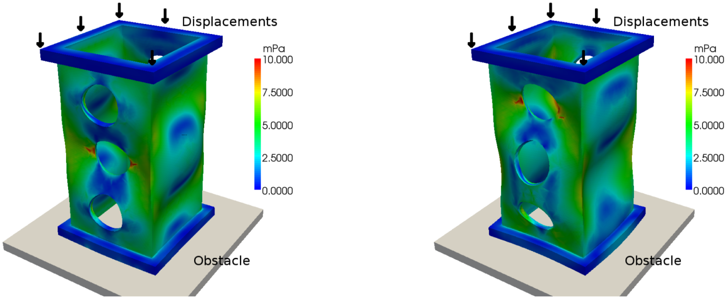
Stored energy function for Ogden materials [Ogden '72] (describes soft-tissues and rubber-like materials)

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

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

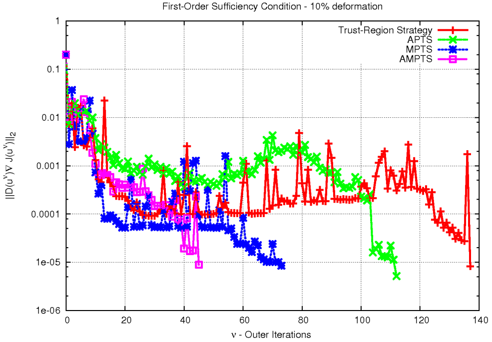
Barrier function: $\ln(\det(I + \nabla\mathbf{u}))$, penalizes element volume decrease.

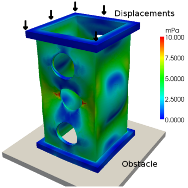
Cylinder Contact Problem

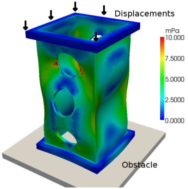


- Energy optimal displacements
- Bifurcation: energy functional is nonconvex and has at least these two solutions!
- 323,994 unknowns
- 8 processors

Cylinder Contact Problem - Performance of Trust-Region Methods



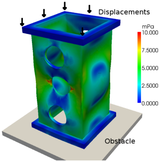


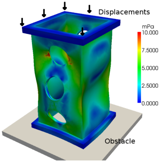


- Energy optimal displacements
- First-order sufficiency conditions $\|\nabla J(u)\|_2$ after each Trust-Region step; Comparison between seq. Trust-Region, APTS, MPTS, combined APTS/MPTS = AMPTS
  ($\mathcal{F} \widehat{=}$ 4 local Trust-Region steps on each D_k, 4 global Trust-Region steps in order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

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

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

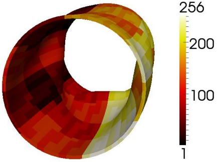
$$\begin{aligned} C_i^{-1} &= \sum_k \left[I_k \left(R_k(\nabla^2 J(u^i)) I_k \right)^{-1} R_k \right] \\ &= \begin{pmatrix} (\nabla^2 J(u^i)_{00})^{-1} & & \\ & \ddots & \\ & & (\nabla^2 J(u^i)_{NN})^{-1} \end{pmatrix} \end{aligned}$$

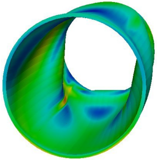
and I_k prolongation operators.

Globalized ASPIN – Overview

The Algorithm

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

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

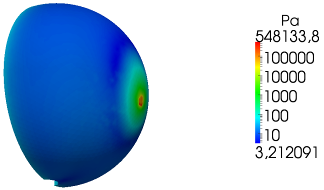
Theorem

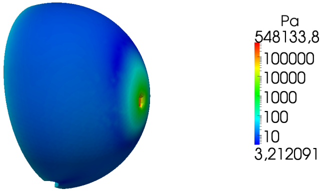
Let the assumptions on J and on B hold. In this case we obtain that the sequence of iterates generated by the globalized ASPIN algorithms has the property

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

Deformation of a Semi-Sphere

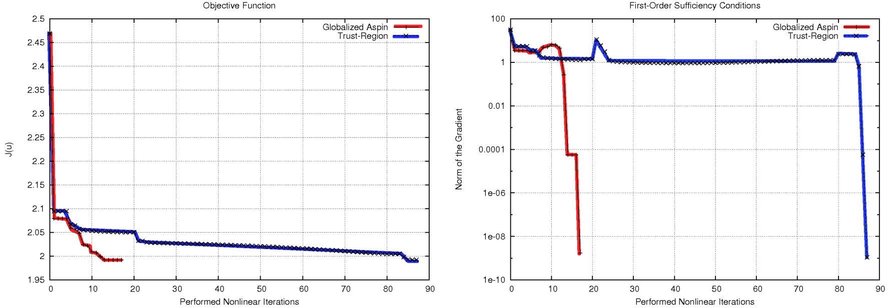
- pushing a sphere in direction of a small obstacle
- 881,280 unknowns
- No bifurcations in the simulations
  We will see
  - (highly) nonlinear behavior of the objective function
  - but: exactly the same solution
- QP solver:
  - Steihaug-Toint CG
  - Monotone Multigrid Smoother
  - Fine grid smoother: 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

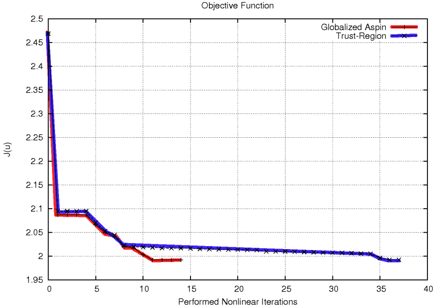


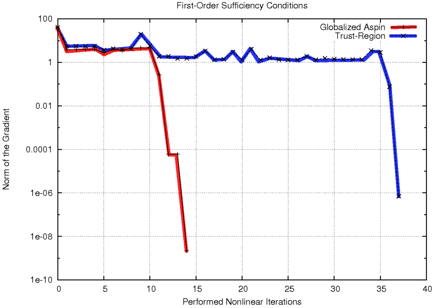
Evolution of the objective function $J(u^i)$ and the norm of the gradient $\|g^i\|$ for Trust-Region and globalized Aspin computations with 240 processors

	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



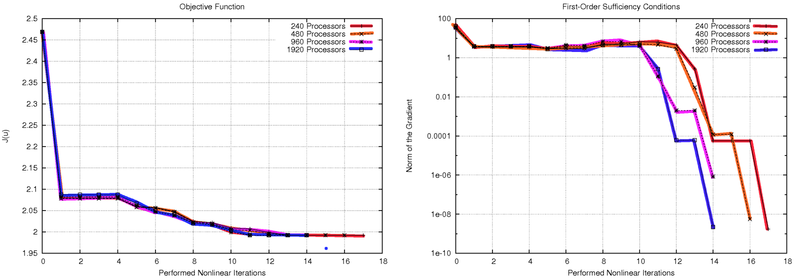


Evolution of the objective function $J(u^i)$ and the norm of the gradient $\|g^i\|$ for Trust-Region and globalized Aspin computations with 1920 processors

	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.

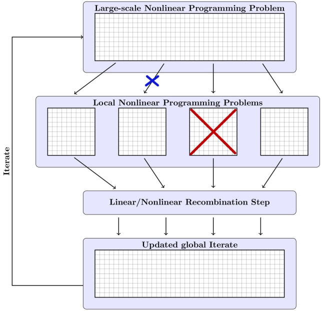
Fault Tolerance of the APLS/APTS Strategies

Possible fault scanarios

Node dies

- during submission to the processor: can be caught and submission can be tried to another node
- during local solution: will yield no local correction $\Rightarrow$ equivalent to $s_k = 0$ and thus

$$\mathcal{F}(u^G) = u^G + \alpha \sum_{p \neq k} I_p s_p$$



Nonlinearly preconditioned training via DNN decomposition

Nonlinear preconditioning framework [Cai, Keyes '02; Dolean '16,...]

- Consider the framework of nonlinear system of equations

$$\nabla \mathcal{L}(\boldsymbol{\theta}) = 0$$

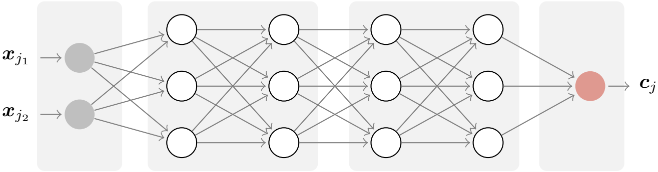
- Instead of solving $\nabla \mathcal{L}(\boldsymbol{\theta}) = 0$, our goal is to construct and solve nonlinearly preconditioned system of equations

$$\mathcal{H}(\boldsymbol{\theta}) = 0,$$

  where

  - $\mathcal{H}$ has same solution as an original system
  - $\mathcal{H}$ should be easier to solve (have more balanced nonlinearities)
  - Numerical computations with $\mathcal{H}$ should be computationally trackable

Decomposition of DNN



Example of the horizontal decomposition of network.

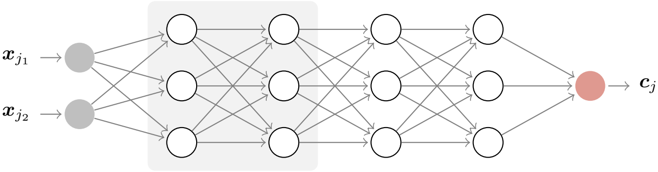
- Decompose the network into S subdomains
- Transfer operators
  - Restriction operator $\boldsymbol{R}_s : \mathbb{R}^n \to \mathbb{R}^{n_s}$ extracts the parameters associated with subdomain s, i.e.,

$$\boldsymbol{\theta}_s = \boldsymbol{R}_s \boldsymbol{\theta}, \quad \text{for } s = 1, \dots, S$$

  - Extension operator $\boldsymbol{E}_s : \mathbb{R}^{n_s} \to \mathbb{R}^n$ extends quantities related to subdomain s to the whole DNN, i.e.,

$$\boldsymbol{\theta} = \sum_{s=1}^{S} \boldsymbol{E}_s \boldsymbol{\theta}_s$$

Local solves



- Let $G_s : \mathbb{R}^n \to \mathbb{R}^{n_s}$ be a local solution operator for $1 \leq s \leq S$, such that

$$\boldsymbol{R}_s \nabla \mathcal{L}(\boldsymbol{E}_s G_s(\boldsymbol{\theta}) + (\boldsymbol{I} - \boldsymbol{E}_s \boldsymbol{R}_s)\boldsymbol{\theta}) = 0,$$

- This corresponds to minimizing $\mathcal{L}$ wrt. $\boldsymbol{\theta}_s$, thus

$$\min_{\boldsymbol{\theta}_s} \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{p} \sum_{j=1}^{p} \ell(f_m(\boldsymbol{x}_j, \boldsymbol{\theta}), \boldsymbol{c}_j),$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_s, \ldots, \boldsymbol{\theta}_S]^T$, while parameters of all other subdomains are kept fixed

Preconditioned nonlinear systems

- Using subdomain solution operators, we define nonlinear additive domain decomposition method as

$$\boldsymbol{\theta}^{k+1} = \sum_{s=1}^{S} \boldsymbol{E}_s G_s(\boldsymbol{\theta}^k),$$

which allows us to formulate the nonlinearly preconditioned system of equations as follows

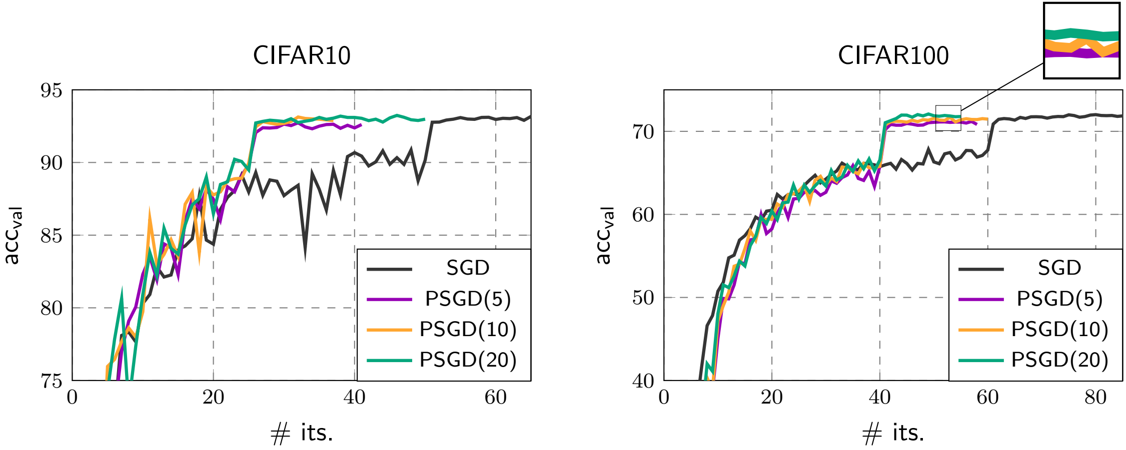
$$\mathcal{H}(\boldsymbol{\theta}) = \boldsymbol{\theta} - \sum_{s=1}^{S} \boldsymbol{E}_s G_s(\boldsymbol{\theta}) = 0,$$

- We can solve the nonlinearly preconditioned system of equations $\mathcal{H}(\boldsymbol{\theta})$ using XYZ algorithm, where XYZ can be Newton's method (ASPIN/ASPEN/RASPEN), but also (S)GD method, Adam,...

Pseudo-algorithm

1. For a given $\boldsymbol{\theta}^k$, perform local step:
   Find $\boldsymbol{\theta}_s^*$ such that $\nabla \mathcal{L}_s(\boldsymbol{\theta}_s^*) = 0, \quad$ for $s = 1, \ldots, S$
2. Evaluate preconditioned gradient as $\mathcal{H}(\boldsymbol{\theta}^k) = \sum_{s=1}^{S} \boldsymbol{E}_s(\boldsymbol{R}_s \boldsymbol{\theta}^k - \boldsymbol{\theta}_s^*)$
3. Perform global parameter update by performing one step of XYZ method

Preconditioned SGD - image classification with ResNets-101

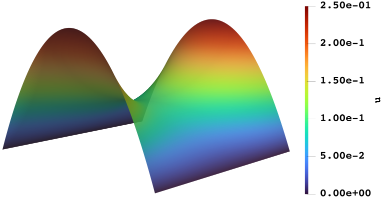


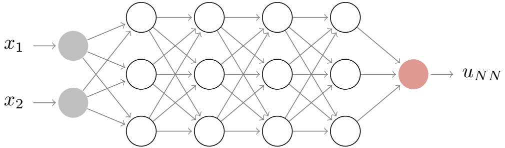
Validation accuracy as function of training steps for ResNet-101 and CIFAR10/CIFAR100 datasets. The results obtained for SGD and preconditioned SGD (PSGD) with varying number of subdomains. Local solves performed using GD method with 3 local steps.

Physics informed neural networks[5]

Minimal surface equation:

$$
\begin{aligned}
F(u) := \nabla \cdot \left(\nabla u / (1 + |\nabla u|^2)^{1/2}\right) = 0 \qquad & \text{on } (0,1)^2, \\
u = 0, \qquad & \text{on } [0, x_2), \\
u = 0, \qquad & \text{on } [1, x_2), \\
u = x_1(1 - x_1), \qquad & \text{on } (x_1, 0], \\
u = x_1(1 - x_1), \qquad & \text{on } (x_1, 1],
\end{aligned}
$$



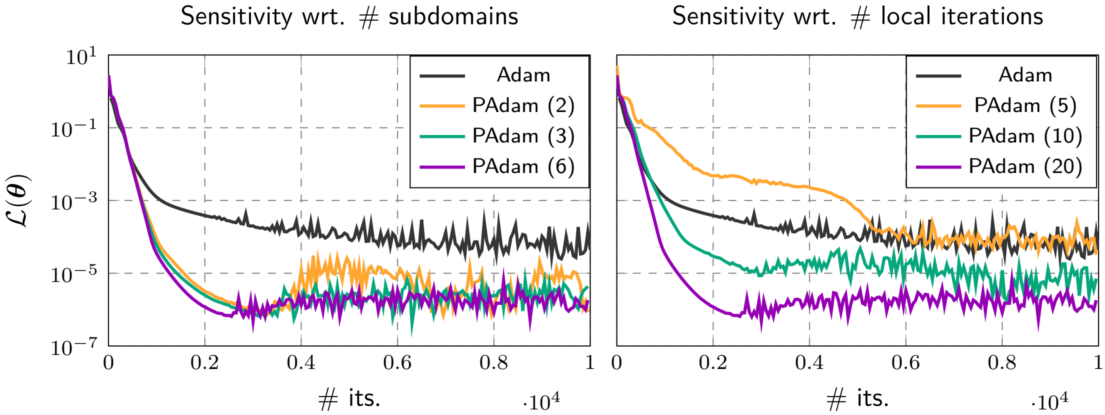


$$
\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{|\mathcal{D}_{\text{int}}|} \sum_{\boldsymbol{x}_j \in \mathcal{D}_{\text{int}}} |F(u_{NN}(\boldsymbol{x}_j, \boldsymbol{\theta}))|^2 + \frac{1}{|\mathcal{D}_{\text{bc}}|} \sum_{(\boldsymbol{x}_j, g_j) \in \mathcal{D}_{\text{bc}}} |u_{NN}(\boldsymbol{x}_j, \boldsymbol{\theta}) - g_j|^2,
$$

where g_j denotes value of u on Γ for a given $\boldsymbol{x}_j = (x_{1,j}, x_{2,j})^T$.

[5] Raissi et al., Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational physics, 2019

Preconditioned Adam and PINN (Minimal surface equation)



Convergence history of Adam (lr$=10^{-4}$) and preconditioned Adam (PADAM) for PINN-minimal surface example. PADAM method employed Adam both, locally and globally (lr$=10^{-4}$). Left: Experiment performed with varying number of subdomains, while number of local iterations is set to 20. Right: Experiment performed with varying number of local iterations, while number of subdomains is set to 6.

Conclusion

References:

- Kopaničáková A., A two-level trust-region method with adaptively sub-sampled coarse space for finite-sum minimization, To be submitted, 2022.
- Kopaničáková A., Kothari H., Krause R., Nonlinearly preconditioned training methods. In preparation, 2022.
- Kopaničáková A., Krause R., Globally Convergent Multilevel Training of Deep Residual Networks, SIAM Journal on Scientific Computing, 2022.

- Gaedke-Merzhauser* L., Kopaničáková* A., Krause R., Multilevel minimization for deep residual networks, Proceedings of French-German-Swiss Optimization Conference (FGS'2019), 2021. (*Equal contribution)
- Braglia* V., Kopaničáková* A., Krause R. A multilevel approach to training, ICML 2020 Workshop: Beyond First Order Methods in Machine Learning, 2020. (*Equal contribution)

Thank you for your attention.

Fault Tolerance of the APLS/APTS Strategies

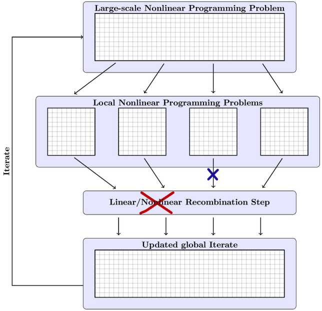
Possible fault scanarios

Node dies

- in recombination step
  - while submitting s_k: will yield $s_k = 0$ and thus

$$\mathcal{F}(u^G) = u^G + \alpha \sum_{p \neq k} I_p s_p$$

  - while solving for α: can be caught and yields a recomputation of the backtracking step on different nodes

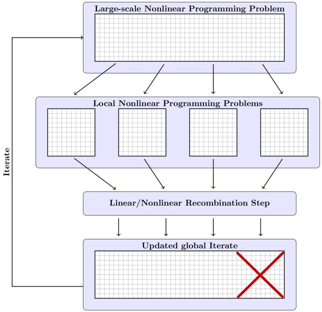


Fault Tolerance of the APLS/APTS Strategies

Possible fault scanarios

Node dies

- in (optional) global smoothing step
  - while computing a search direction: must be dealt with by the linear solver
  - while solving for α: can be caught and yields a recomputation of the backtracking step on different nodes



Updated global iterate in particular stands for the global smoothing step.

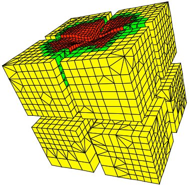
Fault Tolerance of the APLS/APTS Strategies

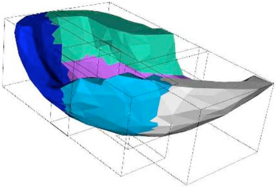
Severeness of possible fault scanarios

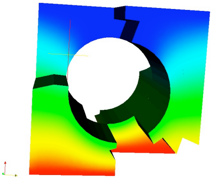
Node dies

- 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.

Parallel non-linear multiscale methods







- domain decomposition $\Omega = \bigcup_P \Omega_P$
- master-nodes m at the "processor interfaces" have several copies $C(m)$
- non-linear synchronization: solve non-linear problems along the processor boundaries

Linear parallel multigrid [Adams '06; Bastian, Birken, Lang, Wieners '96,...]

Parallel Multigrid for Frictional Contact

Exploit strong locality of constraints and friction law $\mathcal{J} = J + \phi$

For level $\ell = L, \ldots, 0$ do

1. Approximate minimization of $\mathcal{Q}^{\ell}_{\bar{\mathbf{u}}^{\ell+1}}$ on each subdomain
2. take meanvalue $\bar{r}$ of the linear residual $f(\cdot) - a(x, \cdot)$ along the processor boundaries (requires communication)
3. for all nodes at the processor boundaries solve in parallel the non-linear local problems: find $w_q \in V_q = \text{spann}\lambda_q$

$$0 \in a(w_q, v) - \bar{r}(v) + \partial\phi^{\ell}(x^k + w_q)(v) \quad v \in V_q$$

4. Local update

$$x_q^{k+1} = w_q^k$$

λ_q: basis function $\qquad C(m) = \{\text{copies of the node } m\}$

Parallel Multigrid for Frictional Contact

Exploit strong locality of constraints and friction law $\mathcal{J} = J + \phi$

For level $\ell = L, \ldots, 0$ do

1. Approximate minimization of $\mathcal{Q}^{\ell}_{\bar{\mathbf{u}}^{\ell+1}}$ on each subdomain
2. take meanvalue $\bar{r}$ of the linear residual $f(\cdot) - a(x, \cdot)$ along the processor boundaries (requires communication)
3. for all nodes at the processor boundaries solve in parallel the non-linear local problems: find $w_q \in V_q = \text{spann}\lambda_q$

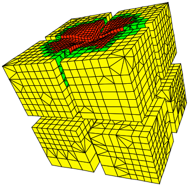
$$0 \in a(w_q, v) - \bar{r}(v) + \partial\phi^{\ell}(x^k + w_q)(v) \quad v \in V_q$$

4. Local update

$$x_q^{k+1} = w_q^k$$

λ_q: basis function $\qquad C(m) = \{\text{copies of the node } m\}$

Parallel efficiency: model problem 112 processors

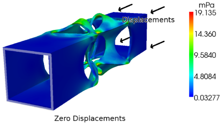


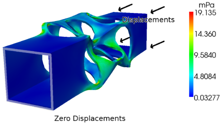
level	no. of contact	no. of dof	no of. iter
1	69	5.907	11
2	277	42.819	11
3	1.085	325.635	11
4	42.65	2.539.011	11
5	16.877	16.858752	9

Good weak scalability, strong scalability depends on surface/volume ratio of the partition

Conclusion

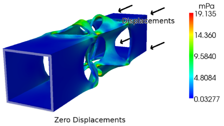
- The following multiplicative and additive Trust-Region strategies:
  - APTS
  - MPTS
- A globalization for ASPIN was presented
  - extension to ASPIN: reduces to ASPIN if “iterates are sufficiently close to local solution”
  - Convergence can be proven due to interpretation as perturbed Trust-Region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

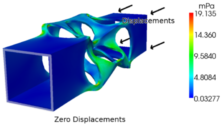




Conclusion

- 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.