Improving Quadratic Programming Algorithms

Jakub Kruzik
jakub.kruzik@vsb.cz

July 12, 2024
Pre-Defense, Ostrava

Outline

1. Outline of the Thesis
2. Quadratic Programming Problem
3. PERMON
4. Preconditioning MPRGP-type methods
5. Thesis Outcome and Outlook

Outline of the Thesis

1. Introduction
2. Optimization Overview
    - convexity, projections, optimality, duality, descent direction
3. QP Algorithms Implementation and Benchmarks
    - software, hardware, benchmarks
4. Unconstrained Quadratic Programming
    - steepest descent and Barzilai-Borwein, CG, deflation
5. Projection-based QP Algorithms
    - MPRGP, MPPCG + fallback, MPSPG, preconditioning
6. QP with Linear Equality Constraints
    - KKT system-based methods, penalty, SMALE
7. QP with Linear Inequality Constraints
    - dualization and FETI
8. Conclusion

Quadratic Programming (QP) Problem

$$\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad x \in \Omega,$$

where

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b},$$

$\boldsymbol{A}$ is symmetric, Ω is closed and convex.

Here

$$\operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b} \quad \text{s.t.} \quad \boldsymbol{l} \leq \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is SPD.

PERMON – Parallel, Efficient, Robust, Modular, Object-oriented, Numerical

- Collection of C libraries
- Based on/extends PETSc
- Open source (BSD-2-Clause license)
- Developed since 2011
- https://github.com/permon

PERMON

http://permon.vsb.cz

PERMON – Parallel, Efficient, Robust, Modular, Object-oriented, Numerical

- Collection of C libraries
- Based on/extends PETSc
- Open source (BSD-2-Clause license)
- Developed since 2011
- https://github.com/permon
- PermonQP
    - General, massively parallel, QP solution framework
    - QP problems, transformations, solvers
    - FETI DDM implementation

PERMON

http://permon.vsb.cz

PERMON – Parallel, Efficient, Robust, Modular, Object-oriented, Numerical

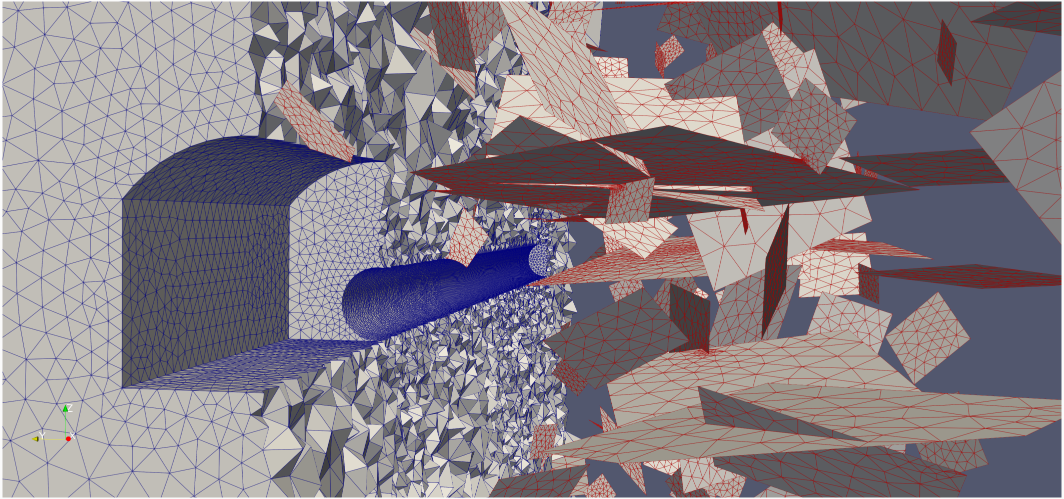
- Collection of C libraries
- Based on/extends PETSc
- Open source (BSD-2-Clause license)
- Developed since 2011
- https://github.com/permon
- **PermonQP**
  - General, massively parallel, QP solution framework
  - QP problems, transformations, solvers
  - FETI DDM implementation
- **PermonSVM**
  - Library and binaries for the solution of linear SVMs

http://permon.vsb.cz

PERMON in Applications

- DEMSI project uses PERMON QP algorithms for load balancing particles in ice sheet melting simulations.
- HyTeG library can use PERMON to solve constrained FEM problems
- Flow123d library uses PERMON to solve mechanical contact subproblems in hydro-mechanical problems
- SIFEL library can use PERMON to solve large scale mechanical contact problems in structural engineering
- Wildfires detection from satellite images using PermonSVM

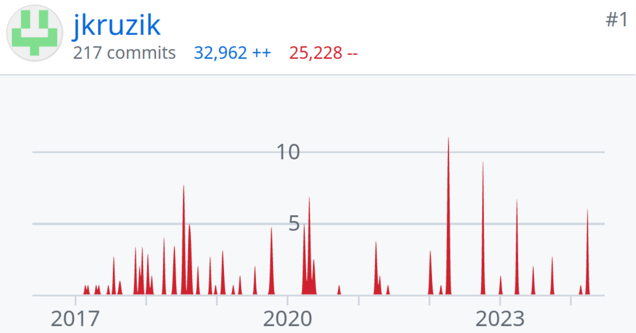
Flow123d – HM Modelling of Excavating Borehole in Fractured Porous Medium



J. Stebel, J. Kruzik et al., ”On the parallel solution of hydro-mechanical problems with fracture networks and contact conditions”, Computers & Structures (2024)

Own Contribution to PERMON

- Main contributor and maintainer of PERMON since 2018
- 15 major releases from version 3.7 and a small number of minor releases



Nearly 60% of the repository overall commits (excluding merge commits) and lines changed since the PERMON public development started in May, 2016 until June, 2024.

Active/Free Set and Gradient Splitting

Active/Free set:

$$\mathcal{A}(\boldsymbol{x}) = \{j : x_j = l_j\} \qquad \mathcal{F}(\boldsymbol{x}) = \{j : l_j < x_j\}$$

Active/Free Set and Gradient Splitting

Active/Free set:

$$\mathcal{A}(\boldsymbol{x}) = \{j : x_j = l_j\} \qquad \mathcal{F}(\boldsymbol{x}) = \{j : l_j < x_j\}$$

Gradient splitting ($\boldsymbol{g} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$):

$$g_j^f = \begin{cases} 0 & \text{if} \quad j \in \mathcal{A}, \\ g_j & \text{if} \quad j \in \mathcal{F}. \end{cases} \qquad g_j^c = \begin{cases} 0 & \text{if} \quad j \in \mathcal{F}, \\ \min(g_j, 0) & \text{if} \quad j \in \mathcal{A}, \end{cases}$$

Active/Free Set and Gradient Splitting

Active/Free set:

$$\mathcal{A}(\boldsymbol{x}) = \{j : x_j = l_j\} \qquad \mathcal{F}(\boldsymbol{x}) = \{j : l_j < x_j\}$$

Gradient splitting ($\boldsymbol{g} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$):

$$g_j^f = \begin{cases} 0 & \text{if} \quad j \in \mathcal{A}, \\ g_j & \text{if} \quad j \in \mathcal{F}. \end{cases} \qquad g_j^c = \begin{cases} 0 & \text{if} \quad j \in \mathcal{F}, \\ \min(g_j, 0) & \text{if} \quad j \in \mathcal{A}, \end{cases}$$

Projected gradient:

$$\boldsymbol{g}^P = \boldsymbol{g}^f + \boldsymbol{g}^c$$

Active/Free Set and Gradient Splitting

Active/Free set:

$$\mathcal{A}(\boldsymbol{x}) = \{j : x_j = l_j\} \qquad \mathcal{F}(\boldsymbol{x}) = \{j : l_j < x_j\}$$

Gradient splitting ($\boldsymbol{g} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$):

$$g_j^f = \begin{cases} 0 & \text{if} \quad j \in \mathcal{A}, \\ g_j & \text{if} \quad j \in \mathcal{F}. \end{cases} \qquad g_j^c = \begin{cases} 0 & \text{if} \quad j \in \mathcal{F}, \\ \min(g_j, 0) & \text{if} \quad j \in \mathcal{A}, \end{cases}$$

Projected gradient:

$$\boldsymbol{g}^P = \boldsymbol{g}^f + \boldsymbol{g}^c$$

Projection onto the feasible set Ω:

$$[P_\Omega(\boldsymbol{x})]_j = \max(l_j, x_j).$$

Modified Proportioning with Projected Conjugate Gradient (MPPCG) [JK 2020]

Input: $\boldsymbol{A}$, $\boldsymbol{x}_0 \in \Omega$, $\boldsymbol{b}$, $\Gamma > 0$

```
g_0 = A x_0 - b, p_0 = g_0^f, k = 0
while ||g_k^P|| is not small:
    if ||g_k^c|| <= Γ||g_k^f||:
        Projected CG
    else:
        Proportioning step
    k = k + 1
```

$\boldsymbol{g}_0 = \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{b}$, $\boldsymbol{p}_0 = \boldsymbol{g}_0^f$, $k = 0$

while $||\boldsymbol{g}_k^P||$ *is not small*:

  if $||\boldsymbol{g}_k^c|| \leq \Gamma ||\boldsymbol{g}_k^f||$:

    Projected CG

  else:

    Proportioning step

  $k = k + 1$

Output: $\boldsymbol{x}_k$

Proportioning step:

$\alpha_k = \boldsymbol{g}_k^T \boldsymbol{g}_k^c / (\boldsymbol{g}_k^c)^T \boldsymbol{A} \boldsymbol{g}_k^c$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k^c$

$\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \alpha_k \boldsymbol{A} \boldsymbol{g}_k^c$

$\boldsymbol{p}_{k+1} = \boldsymbol{g}_{k+1}^f$

Input: $\boldsymbol{A}$, $\boldsymbol{x}_0 \in \Omega$, $\boldsymbol{b}$, $\Gamma > 0$

```
g_0 = A x_0 − b, p_0 = g_0^f, k = 0
while ||g_k^P|| is not small:
    if ||g_k^c|| ≤ Γ||g_k^f||:
        Projected CG
    else:
        Proportioning step
    k = k + 1
```

Output: $\boldsymbol{x}_k$

Projected CG:

```
α_k^feas = max{α : x_k − α p_k ∈ Ω}
α_k^cg = g_k^T p_k / p_k^T A p_k
x_{k+1} = x_k − α_k^cg p_k
if α_k^cg ≤ α^feas:
    # CG step
    g_{k+1} = g_k − α_k^cg A p_k
    β_k = p_k^T A g_{k+1}^f / p_k^T A p_k
    p_{k+1} = g_{k+1}^f − β_k p_k
else:
    # Expansion step
    x_{k+1} = P_Ω(x_{k+1})
    g_{k+1} = A x_{k+1} − b
    p_{k+1} = g_{k+1}^f
```

Preconditioning

$$\operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b} \quad \text{s.t.} \quad \boldsymbol{l} \leq \boldsymbol{x},$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is SPD.

Apply SPD preconditioner

$$\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$$

$$\operatorname*{argmin}_{\boldsymbol{u}} \frac{1}{2}\boldsymbol{u}^T\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-T}\boldsymbol{u} - \boldsymbol{u}^T\boldsymbol{L}^{-1}\boldsymbol{b} \quad \text{s.t.} \quad \boldsymbol{l} \leq \boldsymbol{L}^{-T}\boldsymbol{u},$$

with $\boldsymbol{x} = \boldsymbol{L}^{-T}\boldsymbol{u}$.

MPPCG with Preconditioning

Input: $\boldsymbol{A}$, $\boldsymbol{x}_0 \in \Omega$, $\boldsymbol{b}$, $\Gamma > 0$, $\boldsymbol{M}$

$\boldsymbol{g}_0 = \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{b}$, $\boldsymbol{z}_0 = \boldsymbol{M}^{-1}\boldsymbol{g}_0^f$, $\boldsymbol{p}_0 = \boldsymbol{z}_0$, $k = 0$

while $||\boldsymbol{g}_k^P||$ *is not small*:

    if $||\boldsymbol{g}_k^c|| \leq \Gamma||\boldsymbol{g}_k^f||$:

        Projected CG

    else:

        Proportioning step

    $k = k + 1$

Output: $\boldsymbol{x}_k$

Proportioning step:

1. $\alpha_k = \boldsymbol{g}_k^T \boldsymbol{g}_k^c / (\boldsymbol{g}_k^c)^T \boldsymbol{A} \boldsymbol{g}_k^c$
2. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k^c$
3. $\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \alpha_k \boldsymbol{A} \boldsymbol{g}_k^c$
4. $\boldsymbol{z}_{k+1} = \boldsymbol{M}^{-1} \boldsymbol{g}_{k+1}^f$
5. $\boldsymbol{p}_{k+1} = \boldsymbol{z}_{k+1}$

MPPCG with Preconditioning

Input: $\boldsymbol{A}$, $\boldsymbol{x}_0 \in \Omega$, $\boldsymbol{b}$, $\Gamma > 0$, $\boldsymbol{M}$

$\boldsymbol{g}_0 = \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{b}$, $\boldsymbol{z}_0 = \boldsymbol{M}^{-1}\boldsymbol{g}_0^f$, $\boldsymbol{p}_0 = \boldsymbol{z}_0$, $k = 0$

while $||\boldsymbol{g}_k^P||$ *is not small*:

    if $||\boldsymbol{g}_k^c|| \leq \Gamma||\boldsymbol{g}_k^f||$:

        Projected CG

    else:

        Proportioning step

    $k = k + 1$

Output: $\boldsymbol{x}_k$

Projected CG:

$\alpha_k^{feas} = \max\{\alpha : \boldsymbol{x}_k - \alpha\boldsymbol{p}_k \in \Omega\}$

$\alpha_k^{cg} = \boldsymbol{g}_k^T\boldsymbol{z}_k / \boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k^{cg}\boldsymbol{p}_k$

if $\alpha_k^{cg} \leq \alpha^{feas}$:

    $\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \alpha_k^{cg}\boldsymbol{A}\boldsymbol{p}_k$

    $\boldsymbol{z}_{k+1} = \boldsymbol{M}^{-1}\boldsymbol{g}_{k+1}^f$

    $\beta_k = \boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{z}_{k+1} / \boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k$

    $\boldsymbol{p}_{k+1} = \boldsymbol{z}_{k+1} - \beta_k\boldsymbol{p}_k$

else:

    $\boldsymbol{x}_{k+1} = P_\Omega(\boldsymbol{x}_{k+1})$

    $\boldsymbol{g}_{k+1} = \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{b}$

    $\boldsymbol{z}_{k+1} = \boldsymbol{M}^{-1}\boldsymbol{g}_{k+1}^f$

    $\boldsymbol{p}_{k+1} = \boldsymbol{z}_{k+1}$

Preconditioning in face

[O'Leary 1980]

$\mathcal{A}$ is active set, $\mathcal{F}$ is free set

$$\overline{\boldsymbol{M}} = \begin{pmatrix} \boldsymbol{M}_{\mathcal{F}\mathcal{F}} & \boldsymbol{M}_{\mathcal{F}\mathcal{A}} \\ \boldsymbol{M}_{\mathcal{A}\mathcal{F}} & \boldsymbol{M}_{\mathcal{A}\mathcal{A}} \end{pmatrix}$$

Precondition only on the free set

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \boldsymbol{M}^{-1} \begin{pmatrix} \boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}} & \boldsymbol{o} \\ \boldsymbol{o} & \boldsymbol{o} \end{pmatrix} \begin{pmatrix} \boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix}.$$

Approximate Variant

$$\boldsymbol{z} = \begin{pmatrix} \widetilde{\boldsymbol{z}}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \boldsymbol{M}^{-1} \begin{pmatrix} \boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \text{gradientSplit}_{Free}(\overline{\boldsymbol{M}}^{-1} \begin{pmatrix} \boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix})$$

Approximate Variant

$$\boldsymbol{z} = \begin{pmatrix} \widetilde{\boldsymbol{z}}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \boldsymbol{M}^{-1} \begin{pmatrix} \boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \text{gradientSplit}_{Free}(\overline{\boldsymbol{M}}^{-1} \begin{pmatrix} \boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix}) \qquad \text{assuming } \overline{\boldsymbol{M}}^{-1} \text{is inverse}$$

$$= \begin{pmatrix} (\boldsymbol{M}_{\mathcal{F}\mathcal{F}} - \boldsymbol{M}_{\mathcal{F}\mathcal{A}}\boldsymbol{M}^{-1}_{\mathcal{A}\mathcal{A}}\boldsymbol{M}_{\mathcal{A}\mathcal{F}})^{-1}\boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix} = \begin{pmatrix} \boldsymbol{S}^{-1}\boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix}$$

$$= \begin{pmatrix} (\boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}} + \boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}}\boldsymbol{M}_{\mathcal{F}\mathcal{A}}(\boldsymbol{M}_{\mathcal{A}\mathcal{A}} - \boldsymbol{M}_{\mathcal{A}\mathcal{F}}\boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}}\boldsymbol{M}_{\mathcal{F}\mathcal{A}})^{-1}\boldsymbol{M}_{\mathcal{A}\mathcal{F}}\boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}})\boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix}$$

$$= \begin{pmatrix} (\boldsymbol{I} + \boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}}\boldsymbol{M}_{\mathcal{F}\mathcal{A}}(\boldsymbol{M}_{\mathcal{A}\mathcal{A}} - \boldsymbol{M}_{\mathcal{A}\mathcal{F}}\boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}}\boldsymbol{M}_{\mathcal{F}\mathcal{A}})^{-1}\boldsymbol{M}_{\mathcal{A}\mathcal{F}})\boldsymbol{M}^{-1}_{\mathcal{F}\mathcal{F}}\boldsymbol{g}^f_{\mathcal{F}} \\ \boldsymbol{o} \end{pmatrix}$$

Let $\boldsymbol{M} = \boldsymbol{A}$ and $r = \text{rank}(\boldsymbol{M}_{\mathcal{A}\mathcal{F}})$ then the precondioned operator has eigenvalues

$$1 = \lambda_1 = \cdots = \lambda_{n-r} \leq \cdots \leq \lambda_n$$

Laplace 1D (ex1): Centred finite difference discretization of

$$
\begin{aligned}
-u''(x) &= -15, \qquad x \in [0,1] \\
u(0) &= u(1) = 0 \\
\text{s.t.} \quad u(x) &\geq \frac{sin(4\pi x - \frac{\pi}{6})}{2} - 2, \qquad x \in [0,1]
\end{aligned}
$$

Journal Bearing (lubricant pressure distribution): P1 discretization of

$$
\operatorname*{argmin}_{v \in K} \int_{\mathcal{D}} \left(\frac{1}{2} w_q(x) \|\nabla v(x)\|^2 - w_l(x) v(x) \right) dx,
$$

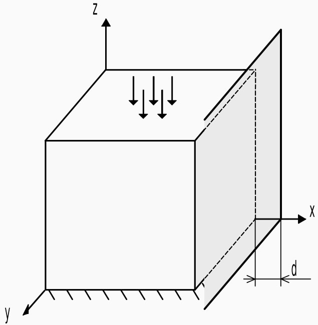
$$
K = \{v \in H_0^1(\mathcal{D})\colon\ v \geq 0\}, \qquad \mathcal{D} = (0, 2\pi) \times (0, 2d),
$$

where $w_q(x_1, x_2) = (1 + \epsilon \cos x_1)^3$, $w_l(x_1, x_2) = \epsilon \sin x_1$, $\epsilon = 0.1$ and $d = 10$.

Benchmarks and Parameters

3D Linear Elasticity Contact Problem:

- bottom fixed
- pushed from above
- obstacle close to the right side
- Q1 discretization
- dual problem is solved

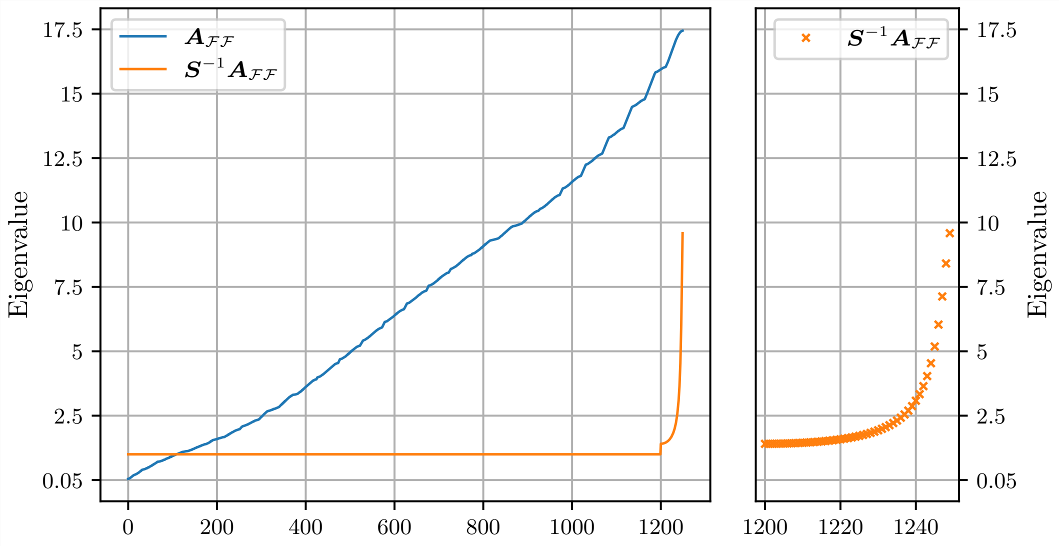


Parameters:

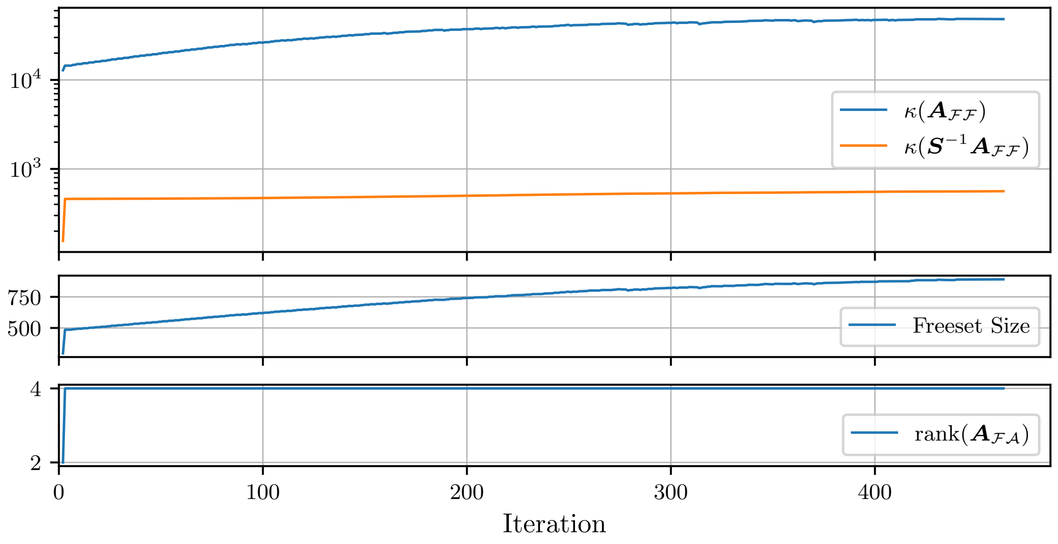
- rtol $1\text{e}{-}10$
- $\overline{\alpha} = 1.9||\boldsymbol{A}||^{-1}$
- $\Gamma = 1$

Computed on LUMI supercomuter, AMD EPYC 7763 @ 2.45 GHz, single core, Cray clang 16 with -O3

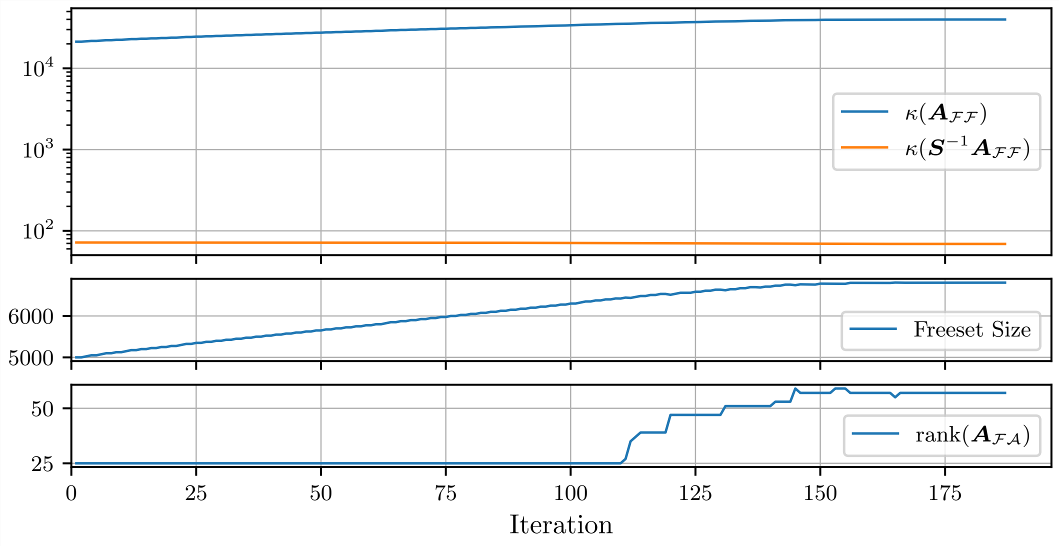
Eigenvalues – Inverse Precond. – Journal Bearing – 50x50 Grid Points (2500 DoFs)



Inverse Preconditioner – 1D Laplace – 1,000 DoFs



Inverse Preconditioner – Journal Bearing – 400x25 Grid Points (10,000 DoFs)



3D Cube Contact Problem with 20x40x80 Finite Elements (209,223 DoFs)

Method	Type	Precond.	Hess.	CG	Exp.	Prop.	Time [s]	S_M
MPRGP	None	None	**6544**	3590	1472	9	88.51	**1.00**
MPRGP	Face	Cholesky	**1818**	6	903	5	14883.00	**0.01**
MPRGP	Approx	Cholesky	**3095**	44	1522	6	439.77	**0.20**
MPRGP	Face	ICC	**4258**	209	2020	8	594.58	**0.15**
MPRGP	Approx	ICC	**5446**	350	2544	7	102.93	**0.86**

3D Cube Contact Problem with 20x40x80 Finite Elements (209,223 DoFs)

Method	Type	Precond.	Hess.	CG	Exp.	Prop.	Time [s]	S_M
MPRGP	None	None	**6544**	3590	1472	9	88.51	**1.00**
MPRGP	Face	Cholesky	**1818**	6	903	5	14883.00	**0.01**
MPRGP	Approx	Cholesky	**3095**	44	1522	6	439.77	**0.20**
MPRGP	Face	ICC	**4258**	209	2020	8	594.58	**0.15**
MPRGP	Approx	ICC	**5446**	350	2544	7	102.93	**0.86**
MPPCG	None	None	**2766**	2269	244	8	37.93	**2.33**
MPPCG	Face	Cholesky	**14**	6	1	5	212.37	**0.42**
MPPCG	Approx	Cholesky	**57**	43	3	7	30.19	**2.93**
MPPCG	Face	ICC	**344**	212	60	11	72.38	**1.22**
MPPCG	Approx	ICC	**473**	297	84	7	10.38	**8.53**

Journal Bearing Problem with 1600x100 Discretization Points (160,000 DoFs)

Method	Type	Precond.	Hess.	CG	Exp.	Prop.	Time [s]	S_M
MPRGP	None	None	**37044**	28703	3844	652	60.14	**1.00**
MPRGP	Face	Cholesky	**617**	308	0	308	317.32	**0.19**
MPRGP	Approx	Cholesky	**3612**	244	1525	317	156.26	**0.38**
MPRGP	Face	ICC	**987**	357	159	311	12.91	**4.66**
MPRGP	Approx	ICC	**6225**	250	2738	498	14.06	**4.28**

Journal Bearing Problem with 1600x100 Discretization Points (160,000 DoFs)

Method	Type	Precond.	Hess.	CG	Exp.	Prop.	Time [s]	S_M
MPRGP	None	None	**37044**	28703	3844	652	60.14	**1.00**
MPRGP	Face	Cholesky	**617**	308	0	308	317.32	**0.19**
MPRGP	Approx	Cholesky	**3612**	244	1525	317	156.26	**0.38**
MPRGP	Face	ICC	**987**	357	159	311	12.91	**4.66**
MPRGP	Approx	ICC	**6225**	250	2738	498	14.06	**4.28**
MPPCG	None	None	**25166**	21632	1509	515	40.40	**1.49**
MPPCG	Face	Cholesky	**617**	308	0	308	317.43	**0.19**
MPPCG	Approx	Cholesky	**887**	379	93	321	59.38	**1.01**
MPPCG	Face	ICC	**776**	368	42	323	11.15	**5.40**
MPPCG	Approx	ICC	**1976**	238	564	609	4.47	**13.46**

Thesis Outcome

Main results:

- Software (1 conf. paper):
  - PERMON improvements, maintenance and user support
  - PCDEFLATION – multilevel deflation preconditioner in PETSc
- Improvements of Algorithms (2 JIMP):
  - MPRGP expansion modifications (MPPCG + fallback, MPSPG, MPSPGf) achieving geometric mean of speedups of 2.9 - 6.25 on suitable benchmarks
  - Approximate preconditioning in face for MPRGP-type methods achieving speedups between 5.13 and 13.46

Thesis Outcome

Other results (collaborations):

- Improvements to FETI-type methods (5 JIMP, 4 conf. papers):
  - Projector-avoiding FETI
  - Scalable strategies for FETI coarse problem
  - Hybrid FETI/BETI method
  - FETI preconditioners for elastoplasticity
  - FETI for slope stability
  - Node renumbering for FETI stiffness matrix factorization
- QP applications (2 JIMP, 1 conf. paper)
  - Contact problems in (hydro)mechanics
  - Linear SVM
- Unconstrained QP and preconditioning (1 JIMP, 1 conf. paper)
  - Inner product free methods and preconditioners for 3×3 block matrices
  - Schwarz DD preconditioners for Darcy flow
- Unrelated (1 JIMP, 4 conf. papers)

Outlook

- Publications:
  - PERMON
  - MPRGP Expansions (MPPCG fallback, MPSPG)
  - Approximate in face preconditioner
- Research:
  - Approximate in face preconditioner for contact problems in FETI
  - SMALE stopping criteria

Thank you for your attention!

Any questions?

Jakub Kruzik

jakub.kruzik@vsb.cz

PERMON

http://permon.vsb.cz

Q: In deflation algorithms, eigenvectors of various matrices are used. I did not find the time needed for their computation. From my point of view, the time of the computation of the eigenvectors has to be included to the total time needed for solution of a problem. Users are interested in the wall clock time of computation.

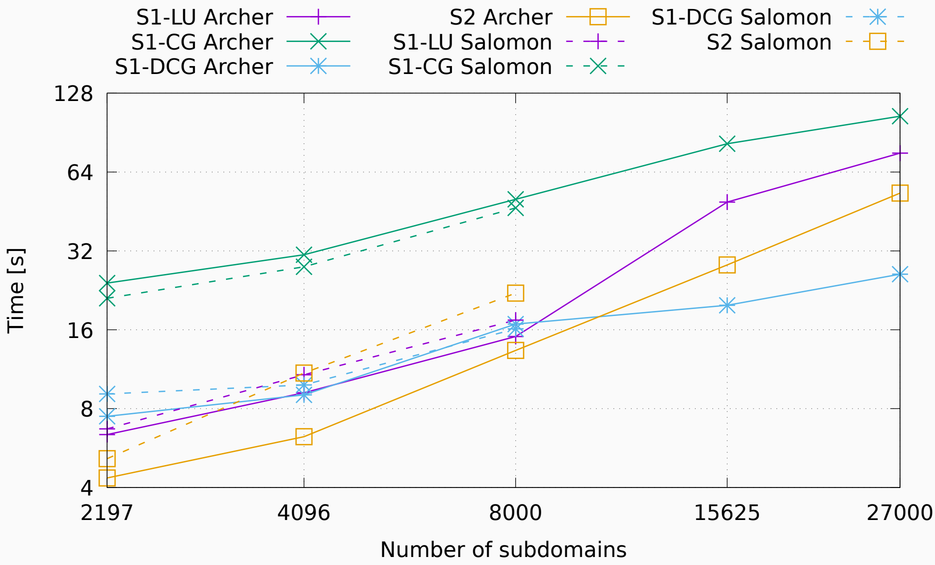


Figure 1: Weak scaling: Coarse problem solver setup + 1,000 coarse problem solves. One subdomain is assigned to one core [?].

graf z clanku choosing strategy

Q: In subsection 4.3.9, 100 eigenvectors are used. In such a case, there is also question about computer memory because the matrix W, which contains the eigenvectors, is not sparse.

A: Yes. However, $\boldsymbol{W}$ storage requirements for the largest presented case are:

$$\frac{\text{subdomains} \cdot \text{rigid body modes} \cdot \text{eigenvectors} \cdot \text{bytes}}{\text{bytes per MiB}} = \frac{27000 \cdot 6 \cdot 100 \cdot 8}{1024^2} \approx 124 \text{ MiB}$$

Q: Str. 37: Vysvětlete minimalizační problém uvedený před koncem stránky. Obsahuje nedefinované symboly y_i a ξ_i.

A: pp. 36: Let us define a training set

$$\{(\boldsymbol{x}_1, y_1)\,,\ (\boldsymbol{x}_2, y_2)\,,\ \ldots,\ (\boldsymbol{x}_m, y_m)\},$$

where m is the number of samples, $\boldsymbol{x}_i \in \mathbb{R}^n$ ($n \in \mathbb{N}$ represents the number of features) is the ith sample, and $y_i \in \{-1, 1\}$ denotes the label (class) of the ith sample.

$$\underset{\boldsymbol{w},\, b,\, \xi_i}{\operatorname{argmin}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\xi_i \quad \text{s.t.} \quad \begin{cases} y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i - b\right) \geq 1 - \xi_i, \\ \xi_i \geq 0, \end{cases}$$

Q: Could you comment on the effect of shifting the eigenvalues in deflation, as described in Section 4.3.4? What are the main advantages of this approach? Can you say when it is important and what is a suitable value of parameter c?

A:

$$\begin{aligned} \boldsymbol{P}_c^T \boldsymbol{A}\boldsymbol{W} &= \boldsymbol{A}\boldsymbol{W} - \boldsymbol{A}\boldsymbol{W}\left(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} + c\boldsymbol{W}\left(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \\ &= c\boldsymbol{W} = \operatorname{diag}(c,\ldots,c)\,\boldsymbol{W}, \end{aligned}$$

Q: In Chapter 5, the performance is evaluated by reporting the number of multiplications with the Hessian. Is this a faithful measure of performance? In other words, is the computational time proportional to the number of matrix multiplications? Could you provide a measurement of time demonstrating that the speedups (e.g., in Tables 5.4 and 5.5) evaluated from the multiplications would stay about the same if they were determined from the computational time? As the author has a high- performance implementation and run his tests on actual supercomputers, it would seem more natural to report time as there can be other effects hidden in the different algorithms that would make the cost more complex than just the number of Hessian multiplications.

Method	Prob.	Hess.	Time [s]	Time [%]			S_H	S_t	S_H / S_t
				MatVec	DotProd	VecUp			
MPRGP	ex1	3907	0.0175	48.83	23.95	27.22	1.0000	1.0000	1.0000
MPPCG	ex1	3586	0.0160	48.49	24.82	26.68	1.0895	1.0875	1.0018
MPRGP	cube	6544	88.51	95.49	1.58	2.93	1.0000	1.0000	1.0000
MPPCG	cube	2766	37.93	95.28	1.94	2.77	2.3659	2.3335	1.0139

- ex1 - 1D Poisson, 1000 unknowns
- cube - 3D elasticity, 20x40x80 finite elements (209,223 unknowns)

Computed on LUMI supercomuter, AMD EPYC 7763 @ 2.45 GHz, single core, Cray clang 16 with -O3

Convex optimization problems inspired by geotechnical stability analysis

Stanislav Sysala

Institute of Geonics of the Czech Academy of Sciences, Ostrava, Czech Republic

stanislav.sysala@ugn.cas.cz

in cooperation with **M. Béreš, S. Béréšová, J. Kružík, T. Luber (Ostrava), J. Haslinger (Prague), M. Neytcheva (Uppsala) and others**

Motivation and aims

Geotechnical stability analysis:

- stability of slopes, foundations, tunnels, etc.
- determination of *factors of safety* (FoS)
- estimation of *failure mechanisms*

Methods based on computational plasticity and finite elements:

- *Limit load (LL) method* – parametrization of external forces
- *Strength reduction (SR) method* – reduction of selected material parameters
- *Limit analysis (LA) method* – optimization variant of the LL method

Aims of the talk:

- to explain ideas of these methods without any knowledge of plasticity and geotechnics
- to introduce convex optimization problems in $\mathbb{R}^n$ inspired by the LL, LA, SR methods
- to suggest convenient abstract assumptions to achieve results expected in geotechnics

Outline

1. Introduction and preliminaries.
2. Abstract problem inspired by the limit load (LL) method.
3. Abstract problem inspired by the limit analysis (LA) method.
4. Abstract problem inspired by the strength reduction (SR) method.
5. Numerical examples from slope stability analysis.

Preliminaries

Convex optimization problem:

$$\text{find } \boldsymbol{u}^* \in \mathbb{R}^n : \quad \mathcal{J}(\boldsymbol{u}^*) \leq \mathcal{J}(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathbb{R}^n, \qquad \mathcal{J}: \mathbb{R}^n \to \mathbb{R} \text{ is convex}$$

1. Solution set $\mathcal{K}$ is closed and convex, possibly empty.
2. Convexity in $\mathbb{R}^n$ implies the continuity.
3. The following statements are equivalent in $\mathbb{R}^n$:
   - $\mathcal{J}$ is coercive, i.e., $\mathcal{J}(\boldsymbol{v}) \to +\infty$ as $\|\boldsymbol{v}\| \to +\infty$.
   - $\mathcal{J}$ has at least linear growth at infinity, i.e.,

     $$\exists c_1 > 0,\ c_2 \geq 0 : \quad \mathcal{J}(\boldsymbol{v}) \geq c_1\|\boldsymbol{v}\| - c_2 \quad \forall \boldsymbol{v} \in \mathbb{R}^n$$

   - $\mathcal{K}$ is nonempty and bounded.
4. If $\mathcal{J}$ is continuously differentiable then $\nabla\mathcal{J}(\boldsymbol{u}^*) = 0$.

Basic problem and assumptions

$$\mathcal{J}(\boldsymbol{v}) := \mathcal{I}(\boldsymbol{v}) - \boldsymbol{b}^\top \boldsymbol{v}$$

Assumptions:

$(\mathcal{A}_1)$ $\mathcal{I}\colon \mathbb{R}^n \to \mathbb{R}$ is convex, continuously differentiable

$(\mathcal{A}_2)$ $\mathcal{I}$ is coercive in $\mathbb{R}^n$, i.e. $\exists\, c_1 > 0,\ c_2 \in \mathbb{R}: \quad \mathcal{I}(\boldsymbol{v}) \geq c_1\|\boldsymbol{v}\| - c_2 \quad \forall \boldsymbol{v} \in \mathbb{R}^n$.

$(\mathcal{A}_3)$ $\boldsymbol{b} \neq 0$, $F(0) = 0$, where $F(\boldsymbol{v}) = \nabla\mathcal{I}(\boldsymbol{v})$ for any $\boldsymbol{v} \in \mathbb{R}^n$

Basic consequences:

- $\mathcal{J}(\boldsymbol{u}^*) \leq \mathcal{J}(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathbb{R}^n \qquad \Longleftrightarrow \qquad F(\boldsymbol{u}^*) = \boldsymbol{b}$
- if $\|\boldsymbol{b}\| < c_1$ then the solution set $\mathcal{K}$ is nonempty and bounded

Proof: $\mathcal{J}(\boldsymbol{v}) = \mathcal{I}(\boldsymbol{v}) - \boldsymbol{b}^\top \boldsymbol{v} \overset{(\mathcal{A}_2)}{\geq} (c_1 - \|\boldsymbol{b}\|)\|\boldsymbol{v}\| - c_2 \quad \forall \boldsymbol{v} \in \mathbb{R}^n$

Basic aims:

- to decide about the solvability (if $\boldsymbol{b}$ is larger)
- to find the factor of safety (FoS):
  1. parametrize the problem, either $\boldsymbol{b}$ or $\mathcal{I}$;
  2. FoS = critical value of the used parameter, solvability condition: FoS > 1

2. Parametrization of b (limit load method)

Parametrized problem:

$$\mathcal{J}_t(\boldsymbol{v}) := \mathcal{I}(\boldsymbol{v}) - t\boldsymbol{b}^\top \boldsymbol{v}$$

$$\text{find } \boldsymbol{u}_t \in \mathbb{R}^n: \quad \mathcal{J}_t(\boldsymbol{u}_t) \le \mathcal{J}_t(\boldsymbol{v}) \text{ for all } \boldsymbol{v} \in \mathbb{R}^n, \quad \text{or} \quad F(\boldsymbol{u}_t) = t\boldsymbol{b}$$

- $\mathcal{J}_t$ is coercive and $\mathcal{K}_t$ is nonempty and bounded for sufficiently small $t > 0$
- If $\mathcal{J}_{\bar{t}}$ is coercive for some $\bar{t} \ge 0$, then $\exists \epsilon > 0$ such that $\mathcal{J}_t$ is coercive for any $t \in [0, \bar{t} + \epsilon]$.

Definition of FoS (limit load factor):

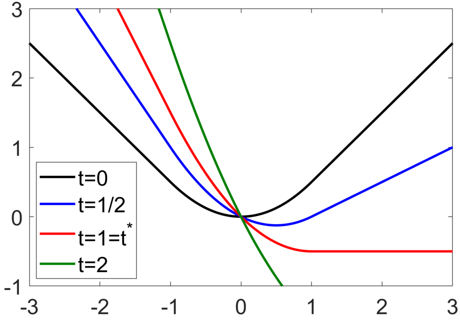
$$t^* \;:=\; \text{supremum of } \; t \ge 0 \; \text{ such that } \mathcal{K}_t \text{ is nonempty}$$

Basic properties under $(\mathcal{A}_1) - (\mathcal{A}_3)$:

- $t^* > 0$, possibly $t^* = +\infty$
- $\mathcal{K}_t$ is nonempty and bounded for any $t < t^*$
- $\mathcal{K}_t$ is empty and $\mathcal{J}_t$ is unbounded from below for any $t > t^*$
- if $t^* < +\infty$ then $\mathcal{K}_{t^*}$ is either empty or unbounded
- if $t^* < +\infty$ then $\mathcal{J}_{t^*}$ is bounded from below under additional assumptions

Illustrative example in $\mathbb{R}^1$ satisfying $(\mathcal{A}_1) - (\mathcal{A}_3)$

$$\mathcal{J}_t(v) := \mathcal{I}(v) - tb^\top v, \quad \text{where} \quad b = 1, \quad \mathcal{I}(v) = \begin{cases} \frac{1}{2}v^2, & |v| \leq 1 \\ |v| - \frac{1}{2}, & |v| \geq 1, \end{cases}$$



- $t < 1 \quad \Longleftrightarrow \quad u_t = t$ – unique solution
- $t = 1 \quad \Longrightarrow \quad \mathcal{K}_t = [1, +\infty)$ – unbounded solution set $\Longrightarrow \quad t^* = 1$
- $t > 1 \quad \Longrightarrow \quad \mathcal{K}_t = \emptyset$ – no solution

Other examples in $\mathbb{R}^1$ satisfying $(\mathcal{A}_1) - (\mathcal{A}_3)$

$$\mathcal{J}_t(v) := \mathcal{I}(v) - tb^\top v, \quad \text{where} \quad \mathcal{I}(v) = e^{-v} + v - 1, \quad b = 1$$

- $t^* = 1$, $\mathcal{K}_{t^*} = \emptyset$, $\mathcal{J}_{t^*}$ is bounded from below

$$\mathcal{J}_t(v) := \mathcal{I}(v) - tb^\top v, \quad \text{where} \quad \mathcal{I}(v) = \begin{cases} v^2 - 2v, & |v| \leq 1 \\ v - \ln v - 2, & |v| \geq 1, \end{cases} \quad b = 1$$

- $t^* = 1$, $\mathcal{K}_{t^*} = \emptyset$, $\mathcal{J}_{t^*}$ is unbounded from below

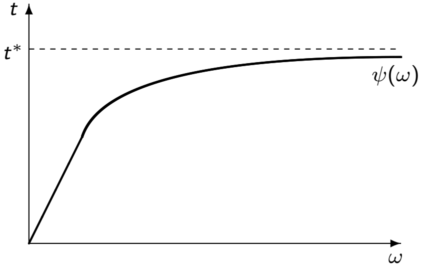
Continuation techniques for finding t^*

Direct continuation:

- t is a directly enlarged up to the unknown t^*
- solvability of $F(\boldsymbol{u}_t) = t\boldsymbol{b}$ is not guaranteed
- non-convergence of a solver is frequent even for $t < t^*$

Advanced (indirect) continuation:

- use another control variable $\omega > 0$ and a mapping $\psi \colon \omega \mapsto t_\omega$
- requirements: ψ is continuous, nondecreasing and $\psi(\omega) \to t^*$ as $\omega \to +\infty$



Construction of the function $\psi : \omega \mapsto t_\omega$

Auxiliary problem for given $\omega > 0$:

$$\mathcal{I}(\boldsymbol{u}_\omega) = \min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \boldsymbol{b}^\top \boldsymbol{v}=\omega}} \mathcal{I}(\boldsymbol{v})$$

- The minimum $\boldsymbol{u}_\omega$ exists for any $\omega > 0$ because $\mathcal{I}$ is coercive, convex and $\boldsymbol{b} \neq 0$.
- Optimality condition:

$$\exists t_\omega \in \mathbb{R}^n : \quad F(\boldsymbol{u}_\omega) = t_\omega \boldsymbol{b}; \quad \boldsymbol{b}^\top \boldsymbol{u}_\omega = \omega$$

- $\omega t_\omega = t_\omega \boldsymbol{b}^\top \boldsymbol{u}_\omega = F(\boldsymbol{u}_\omega)^\top \boldsymbol{u}_\omega = [F(\boldsymbol{u}_\omega) - F(0)]^\top (\boldsymbol{u}_\omega - 0) \geq 0 \quad \Longrightarrow \quad t_\omega \geq 0$
- Uniqueness of t_ω:

$$t_1 < t_2,\ \mathcal{K}_{t_1} \neq \emptyset \neq \mathcal{K}_{t_2} \quad \Longrightarrow \quad \boldsymbol{b}^\top \boldsymbol{u}_1 < \boldsymbol{b}^\top \boldsymbol{u}_2 \quad \forall \boldsymbol{u}_i \in \mathcal{K}_{t_i},\ i = 1, 2$$

Other properties of the function $\psi : \omega \mapsto t_\omega$:

- ψ is continuous, nondecreasing and $\psi(\omega) \to t^*$ as $\omega \to +\infty$

3. Limit analysis (LA) problem: formal derivation

$$\mathcal{I}_\omega : \mathbb{R}^n \to \mathbb{R} : \qquad \mathcal{I}_\omega(\boldsymbol{v}) := \frac{1}{\omega}\mathcal{I}(\omega \boldsymbol{v}), \quad \omega > 0,\ \boldsymbol{v} \in \mathbb{R}^n$$

- $\mathcal{I}_\omega$ is convex, coercive and continuously differentiable for any $\omega > 0$
- $$\mathcal{I}(\boldsymbol{u}_\omega) = \min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \boldsymbol{b}^\top \boldsymbol{v}=\omega}} \mathcal{I}(\boldsymbol{v}) \quad \Longleftrightarrow \quad \mathcal{I}_\omega(\boldsymbol{w}_\omega) = \min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \boldsymbol{b}^\top \boldsymbol{v}=1}} \mathcal{I}_\omega(\boldsymbol{v}), \quad \text{where} \quad \boldsymbol{w}_\omega = \frac{\boldsymbol{u}_\omega}{\omega}$$
- $\omega_1 \leq \omega_2 \implies \mathcal{I}_{\omega_1} \leq \mathcal{I}_{\omega_2}$

$$\mathcal{I}_\infty : \mathbb{R}^n \to \mathbb{R}_+ \cup \{+\infty\} : \qquad \mathcal{I}_\infty(\boldsymbol{v}) = \lim_{\omega\to+\infty} \mathcal{I}_\omega(\boldsymbol{v}), \quad \boldsymbol{v} \in \mathbb{R}^n$$

- $\mathcal{C} := \{\boldsymbol{v} \in \mathbb{R}^n \mid \mathcal{I}_\infty(\boldsymbol{v}) < +\infty\}$ – convex cone, $0 \in \mathcal{C}$
- $\mathcal{I}_\infty$ is convex and coercive in $\mathcal{C}$, it holds $\mathcal{I}_\infty(\boldsymbol{v}) \geq c_1\|\boldsymbol{v}\| \quad \forall \boldsymbol{v} \in \mathcal{C}$
- $\mathcal{I}_\infty$ is *1-positively homogeneous*, i.e. $\mathcal{I}_\infty(\alpha\boldsymbol{v}) = \alpha\mathcal{I}_\infty(\boldsymbol{v})\ \ \forall\alpha \geq 0,\ \forall \boldsymbol{v} \in \mathcal{C}$

$$\min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \boldsymbol{b}^\top \boldsymbol{v}=1}} \mathcal{I}_\omega(\boldsymbol{v}) \quad \xrightarrow{(\omega\to+\infty)} \quad \inf_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \boldsymbol{b}^\top \boldsymbol{v}=1}} \mathcal{I}_\infty(\boldsymbol{v}) = \inf_{\substack{\boldsymbol{v}\in\mathcal{C} \\ \boldsymbol{b}^\top \boldsymbol{v}=1}} \mathcal{I}_\infty(\boldsymbol{v})$$

Limit analysis problem and its investigation

Limit analysis (LA) problem and related FoS:

$$t_\infty = \inf_{\substack{\boldsymbol{v} \in \mathcal{C} \\ \boldsymbol{b}^\top \boldsymbol{v} = 1}} \mathcal{I}_\infty(\boldsymbol{v}), \qquad t_\infty \geq t^*, \quad \text{possibly } t_\infty = +\infty$$

Additional assumptions:

$(\mathcal{A}_4)$ If $\boldsymbol{v}_\omega \to \boldsymbol{v}_\infty$ as $\omega \to +\infty$ and $\{\mathcal{I}_\omega(\boldsymbol{v}_\omega)\}$ is bounded then

$$\boldsymbol{v}_\infty \in \mathcal{C} \quad \text{and} \quad \liminf_{\omega \to +\infty} \mathcal{I}_\omega(\boldsymbol{v}_\omega) \geq \mathcal{I}_\infty(\boldsymbol{v}_\infty).$$

$(\mathcal{A}_5)$ *For any* $\boldsymbol{v} \in \mathcal{C}$ *there exists a constant* $c_v > 0$ *such that*

$$\omega \mathcal{I}_\infty(\boldsymbol{v}) - c_v \leq \mathcal{I}(\omega \boldsymbol{v}) \leq \omega \mathcal{I}_\infty(\boldsymbol{v}) \quad \text{for all } \omega \geq 0.$$

Consequences of $(\mathcal{A}_1) - (\mathcal{A}_5)$:

- $t_\infty = t^*$, if $t^* < +\infty$ then $\{\boldsymbol{w}_\omega\}$ is bounded and its accumulation point solves LA
- $\mathcal{C}$ is closed $\implies$ LA has a minimum if and only if $\{\boldsymbol{v} \in \mathcal{C} \mid \boldsymbol{b}^\top \boldsymbol{v} = 1\} \neq \emptyset$
- if $t^* < +\infty$ then $\mathcal{J}_{t^*}$ is bounded from below

Solvability in convex optimization via LA problem

Original convex optimization problem:

$$\text{find } \boldsymbol{u}^* \in \mathbb{R}^n : \quad \mathcal{J}(\boldsymbol{u}^*) \leq \mathcal{J}(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathbb{R}^n, \qquad \mathcal{J}(\boldsymbol{v}) = \mathcal{I}(\boldsymbol{v}) - \boldsymbol{b}^\top \boldsymbol{v}$$

LA problem:

$$t_\infty = \inf_{\substack{\boldsymbol{v} \in \mathcal{C} \\ \boldsymbol{b}^\top \boldsymbol{v} = 1}} \mathcal{I}_\infty(\boldsymbol{v})$$

Solvability conditions under the assumptions $(\mathcal{A}_1) - (\mathcal{A}_5)$**:**

- $t_\infty > 1 \iff$ the solution set $\mathcal{K}$ is nonempty and bounded
- $t_\infty = 1 \implies$ $\mathcal{K}$ is either empty or unbounded, $\mathcal{J}$ is bounded from below
- $t_\infty < 1 \implies$ $\mathcal{K}$ is empty, $\mathcal{J}$ is unbounded from below

Possible extensions:

- for some noncoercive functions (see the next example)
- for nonsmooth coercive and convex functions
- for selected problems with constraints (e.g. contact problems of elasto-plastic bodies) [Sysala, Haslinger, Hlaváček, Čermák - ZAMM 2015]

Illustrative example

$$\boldsymbol{A}\boldsymbol{u}^* = \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{R}^{n\times n}_{sym} \text{ – positive semidefinite}, \ \boldsymbol{b} \in \mathbb{R}^n, \ \boldsymbol{b} \neq 0,$$

Standard solvability condition:

$$\boldsymbol{b} \perp \mathrm{Ker}\, \boldsymbol{A} \qquad \text{or equivalently} \quad \boldsymbol{b} \in \mathrm{Im}\, \boldsymbol{A}$$

Solvability analysis via LA approach:

$$\mathcal{I}(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v}, \quad \mathcal{I}_\infty(\boldsymbol{v}) = \lim_{\omega\to+\infty} \frac{1}{\omega}\mathcal{I}(\omega\boldsymbol{v}) = \begin{cases} 0, & \boldsymbol{v} \in \mathrm{Ker}\,\boldsymbol{A}, \\ +\infty, & \boldsymbol{v} \notin \mathrm{Ker}\,\boldsymbol{A}, \end{cases} \qquad \mathcal{C} = \mathrm{Ker}\,\boldsymbol{A}$$

$$t_\infty = \inf_{\substack{v\in \mathrm{Ker}\,\boldsymbol{A} \\ \boldsymbol{b}^\top \boldsymbol{v}=1}} \mathcal{I}_\infty(\boldsymbol{v}) = \begin{cases} 0, & \exists \boldsymbol{v} \in \mathrm{Ker}\,\boldsymbol{A} : \ \boldsymbol{b}^\top \boldsymbol{v} = 1 \iff \boldsymbol{b} \not\perp \mathrm{Ker}\,\boldsymbol{A} \\ +\infty, & \forall \boldsymbol{v} \in \mathrm{Ker}\,\boldsymbol{A} : \ \boldsymbol{b}^\top \boldsymbol{v} \neq 1 \iff \boldsymbol{b} \perp \mathrm{Ker}\,\boldsymbol{A} \end{cases}$$

$$t_\infty > 1 \quad \iff \quad \boldsymbol{b} \perp \mathrm{Ker}\,\boldsymbol{A}$$

Remark:

- $\mathcal{I}$ is coercive only if $\boldsymbol{A}$ is positive definite.

4. Parametrization of $\mathcal{I}$ (SR method)

Parametrization inspired by the strength reduction (SR) method:

$$\mathcal{J}_\lambda(\boldsymbol{v}) := \mathcal{I}_\lambda(\boldsymbol{v}) - \boldsymbol{b}^\top \boldsymbol{v}, \quad \lambda \geq \lambda_0, \ \ \lambda_0 \in (0,1]$$

$$\text{find } \boldsymbol{u}_\lambda \in \mathbb{R}^n : \quad \mathcal{J}_\lambda(\boldsymbol{u}_\lambda) \leq \mathcal{J}_\lambda(\boldsymbol{v}) \ \text{ for all } \boldsymbol{v} \in \mathbb{R}^n, \quad \text{or} \quad F_\lambda(\boldsymbol{u}_\lambda) = \boldsymbol{b}$$

Related FoS:

$$\lambda^* \ := \ \text{supremum of } \ \lambda \geq \lambda_0 \ \text{ such that } \mathcal{K}_\lambda \text{ is nonempty}$$

Basic assumptions:

$(\mathcal{B}_1)_\lambda$ $\mathcal{I}_\lambda : \mathbb{R}^n \to \mathbb{R}$ is convex and continuously differentiable for any $\lambda \geq \lambda_0$, $F_\lambda := \nabla \mathcal{I}_\lambda$

$(\mathcal{B}_2)_\lambda$ (a) $\mathcal{J}_{\lambda_0}$ is coercive in $\mathbb{R}^n$.
(b) If $\mathcal{J}_{\bar\lambda}$ is coercive for some $\bar\lambda > \lambda_0$ then $\exists \epsilon > 0$ such that $\mathcal{J}_{\bar\lambda+\epsilon}$ is also coercive.
(c) If $\mathcal{J}_{\bar\lambda}$ is bounded from below for some $\bar\lambda > \lambda_0$, then $\mathcal{J}_\lambda$ is coercive for any $\lambda \in [\lambda_0, \bar\lambda)$.

Remarks:

- $(\mathcal{B}_2)_\lambda$ depends on $\boldsymbol{b}$ – it is natural in geotechnics.
- $(\mathcal{B}_2)_\lambda$(a) does not hold for larger tension forces in geotechnics.
- $(\mathcal{B}_2)_\lambda$ is observed in geotechnics on numerical examples but cannot be verified a priori.
- t-parametrization corresponds to choice $\mathcal{I}_\lambda = \mathcal{I}/\lambda$ and $(\mathcal{A}_1) - (\mathcal{A}_3)$ implies $(\mathcal{B}_2)_\lambda$.
- λ-parametrization is more general than t-parametrization, $(\mathcal{B}_2)_\lambda$ is very general

Basic properties of the λ-parametrization

Consequences of $(\mathcal{B}_1)_\lambda - (\mathcal{B}_2)_\lambda$:

- $\lambda^* > \lambda_0$
- $\mathcal{K}_\lambda$ is nonempty and bounded for any $\lambda < \lambda^*$
- if $\lambda^* < +\infty$ then $\mathcal{K}_{\lambda^*}$ is either empty or unbounded

Next aims within λ-parametrization:

1. Indirect continuation method
2. Limit analysis approach

Other important assumptions (inspired by SSR method):

$(\mathcal{B}_3)_\lambda$ $\boldsymbol{b} \neq 0,\ F_\lambda(0) = \nabla \mathcal{I}_\lambda(0) = 0$

$(\mathcal{B}_4)_\lambda$ $\mathcal{I}_\lambda$ is coercive in $\mathbb{R}^n$ for any $\lambda \geq \lambda_0$

$(\mathcal{B}_5)_\lambda$ For any $\boldsymbol{v} \in \mathbb{R}^n$, the function $\lambda \mapsto \mathcal{I}_\lambda(\boldsymbol{v})$ is nonincreasing.

$(\mathcal{B}_6)_\lambda$ $\lambda_n \to \lambda,\ \ \boldsymbol{v}_n \to \boldsymbol{v} \quad \Longrightarrow \quad \lim_{n \to +\infty} \mathcal{I}_{\lambda_n}(\boldsymbol{v}_n) = \mathcal{I}_\lambda(\boldsymbol{v}).$

Indirect continuation method

Additional assumptions:

$(\mathcal{B}_7)_\lambda$ $\exists\{\boldsymbol{v}_\lambda\} \subset \mathbb{R}^n$: $\lim_{\lambda\to+\infty} \mathcal{J}_\lambda(\boldsymbol{v}_\lambda) = -\infty.$

$(\mathcal{B}_8)_\lambda$ Let $\lambda_1 < \lambda_2$ and $\mathcal{K}_{\lambda_i} \neq \emptyset$, $i = 1, 2$. Then $\boldsymbol{b}^\top \boldsymbol{u}_1 \leq \boldsymbol{b}^\top \boldsymbol{u}_2$ for any $\boldsymbol{u}_1 \in \mathcal{K}_{\lambda_1}$ and $\boldsymbol{u}_2 \in \mathcal{K}_{\lambda_2}$.

$(\mathcal{B}_8)_\lambda^+$ Let $\lambda_1 < \lambda_2$ and $\mathcal{K}_{\lambda_i} \neq \emptyset$, $i = 1, 2$. Then $\boldsymbol{b}^\top \boldsymbol{u}_1 < \boldsymbol{b}^\top \boldsymbol{u}_2$ for any $\boldsymbol{u}_1 \in \mathcal{K}_{\lambda_1}$ and $\boldsymbol{u}_2 \in \mathcal{K}_{\lambda_2}$.

Consequences of $(\mathcal{B}_1)_\lambda - (\mathcal{B}_8)_\lambda$:

$$\left\{\boldsymbol{b}^\top \boldsymbol{u} \;\middle|\; \boldsymbol{u} \in \bigcup_{\lambda\geq\lambda_0} \mathcal{K}_\lambda\right\} = [\omega_0, +\infty), \qquad \omega_0 := \min_{\boldsymbol{u}\in\mathcal{K}_{\lambda_0}} \boldsymbol{b}^\top \boldsymbol{u}$$

$$\forall\omega \geq \omega_0 \;\; \exists\boldsymbol{u}_\omega \in \mathbb{R}^n, \; \exists\lambda_\omega \geq \lambda_0 : \quad F_{\lambda_\omega}(\boldsymbol{u}_\omega) = b, \quad \boldsymbol{b}^\top \boldsymbol{u}_\omega = \omega$$

$$(\mathcal{B}_8)_\lambda^+ \quad \Longrightarrow \quad \lambda_\omega \text{ is unique}$$

$$\psi\colon \omega \mapsto \lambda_\omega \;\text{ is continuous, nondecreasing and }\; \lim_{\omega\to+\infty} \psi(\omega) = \lambda^*$$

LA approach for the λ-parametrization

LA problem for fixed $\lambda \geq \lambda_0$:

$$\mathcal{I}_{\omega,\lambda} : \mathbb{R}^n \to \mathbb{R}, \quad \mathcal{I}_{\omega,\lambda}(\boldsymbol{v}) := \frac{1}{\omega}\mathcal{I}_\lambda(\omega \boldsymbol{v}), \quad \boldsymbol{v} \in \mathbb{R}^n,\ \omega > 0,$$

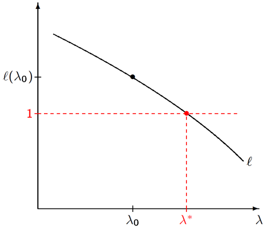
$$\mathcal{I}_{\infty,\lambda} : \mathbb{R}^n \to \mathbb{R}_+ \cup \{+\infty\}, \quad \mathcal{I}_{\infty,\lambda}(\boldsymbol{v}) := \lim_{\omega \to +\infty} \mathcal{I}_{\omega,\lambda}(\boldsymbol{v}), \quad \boldsymbol{v} \in \mathbb{R}^n,$$

$$\mathcal{C}_\lambda := \operatorname{dom} \mathcal{I}_{\infty,\lambda} = \{\boldsymbol{v} \in \mathbb{R}^n \mid \mathcal{I}_{\infty,\lambda}(\boldsymbol{v}) < +\infty\},$$

$$\ell(\lambda) := \inf_{\substack{\boldsymbol{v} \in \mathbb{R}^n \\ \boldsymbol{b}^\top \boldsymbol{v} = 1}} \mathcal{I}_{\infty,\lambda}(\boldsymbol{v}) = \inf_{\substack{\boldsymbol{v} \in \mathcal{C}_\lambda \\ \boldsymbol{b}^\top \boldsymbol{v} = 1}} \mathcal{I}_{\infty,\lambda}(\boldsymbol{v}),$$

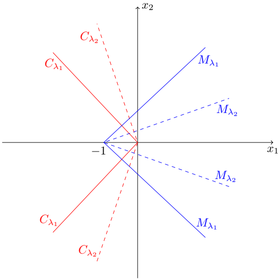
Results under additional assumptions:

- ℓ is decreasing and continuous
- $\ell(\lambda) > 1$ for any $\lambda < \lambda^*$
- $\ell(\lambda) < 1$ for any $\lambda > \lambda^*$
- $\lambda^* < +\infty \quad \Longrightarrow \quad \ell(\lambda^*) = 1$



Example inspired by Mohr-Coulomb plasticity

- $M_\lambda = \{\boldsymbol{x} = (x_1, x_2)^\top \in \mathbb{R}^2 \mid x_1 - \lambda|x_2| + 1 \geq 0\}$
- $\lambda_0 \in (0, 1]$ - given, $\boldsymbol{b} = (b_1, b_2) \in \mathrm{int}\, M_{\lambda_0}$, $b_2 \neq 0$,
- $\mathcal{I}_\lambda(\boldsymbol{v}) = \max_{\boldsymbol{x} \in M_\lambda} \left\{\boldsymbol{x}^\top \boldsymbol{v} - \frac{1}{2}\|\boldsymbol{x}\|^2\right\}$
- $\nabla \mathcal{I}_\lambda = F_\lambda$ – projection of $\mathbb{R}^2$ onto M_λ
- $F_\lambda(\boldsymbol{b}) = \boldsymbol{b}$ iff $\boldsymbol{b} \in M_\lambda$, i.e., $\boldsymbol{u}_\lambda = \boldsymbol{b} \in \mathcal{K}_\lambda$
- $\mathcal{K}_\lambda = \{\boldsymbol{b}\}$ iff $\boldsymbol{b} \in \mathrm{int}\, M_\lambda$, $\mathcal{K}_\lambda = \emptyset$ iff $\boldsymbol{b} \notin M_\lambda$
- if $\boldsymbol{b} \in \partial M_\lambda$ then $\mathcal{K}_\lambda$ is unbounded and $\lambda = \lambda^*$



$$\lambda^* = \frac{1 + b_1}{|b_2|} > \lambda_0, \quad \mathcal{K}_{\lambda^*} = \{\boldsymbol{b} + \alpha \boldsymbol{w} \mid \alpha \geq 0\}, \quad \boldsymbol{w} = \left(-1, \frac{1 + b_1}{b_2}\right)$$

$$\mathcal{I}_{\infty,\lambda}(\boldsymbol{v}) = \begin{cases} -v_1, & v_1 + \frac{1}{\lambda}|v_2| \leq 0 \\ +\infty, & v_1 + \frac{1}{\lambda}|v_2| > 0, \end{cases} \qquad \mathcal{C}_\lambda = \left\{\boldsymbol{v} = (v_1, v_2)^\top \in \mathbb{R}^2 \mid v_1 + \frac{1}{\lambda}|v_2| \leq 0\right\},$$

$$b_2 \neq 0 \quad \Longrightarrow \quad \ell(\lambda) = \frac{1}{\lambda|b_2| - b_1} \quad \forall \lambda \geq \lambda_0 \quad \Longrightarrow \quad \ell(\lambda^*) = 1$$

- if $b_1 < -1$ then the λ-parametrization is not meaningful!

5. Examples from slope stability analysis

Model and methods:

- Elastic-perfectly plastic model with the Mohr-Coulomb yield criterion
- Davis' approximation of nonassociated plastic flow rule
- Finite element method (mostly P2 elements)

Numerical methods:

- indirect continuation technique with adaptive enlargement of ω:

$$\text{either} \quad F(\boldsymbol{u}_\omega) = t_\omega \boldsymbol{b}, \quad \boldsymbol{b}^\top \boldsymbol{u}_\omega = \omega \quad \text{or} \quad F_{\lambda_\omega}(\boldsymbol{u}_\omega) = \boldsymbol{b}, \quad \boldsymbol{b}^\top \boldsymbol{u}_\omega = \omega$$

- Semismooth Newton's method with damping and regularization
- Iterative solvers for linearized systems related to 3D problems:
  – deflated flexible GMRES
  – preconditioner: algebraic multigrid and separate displacement method

Implementation:

- Matlab codes, see https://github.com/sysala/SSRM
- Meshes imported from GMSH

Numerical example I – homogeneous slope in 2D

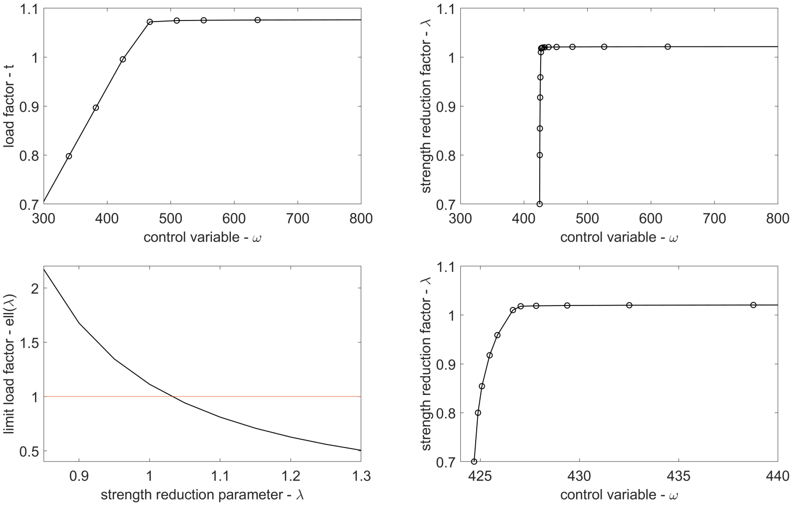
Setting the problem:

- Young's modulus: $E = 40\,\mathrm{MPa}$
- Poisson's ration: $\nu = 0.3$
- cohesion, $c = 6\,\mathrm{kPa}$
- friction and dilatancy angles: $\phi = \psi = 30^\circ$
- slope is loaded by self-weight
- Dirichlet boundary conditions on the left, right and bottom
- uniform mesh with 5,200 elements and 20,840 unknowns

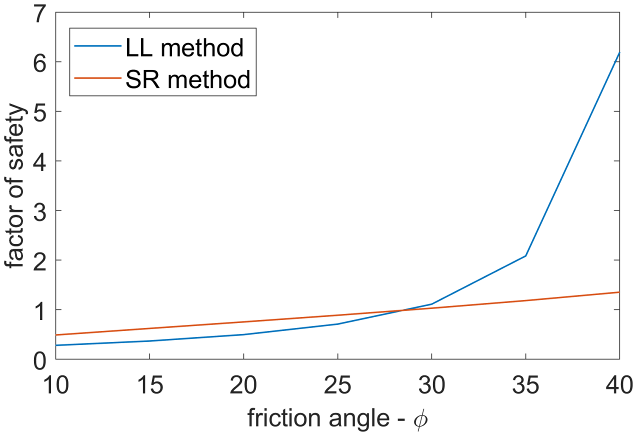
FoS and failure mechanism:

$$t^* = 1.08, \qquad \lambda^* = 1.02$$

Continuation curves and the function ℓ



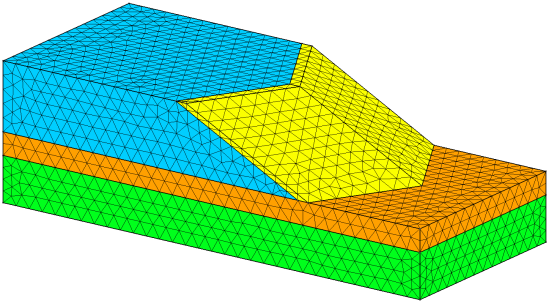
Dependence of FoS on the friction angle



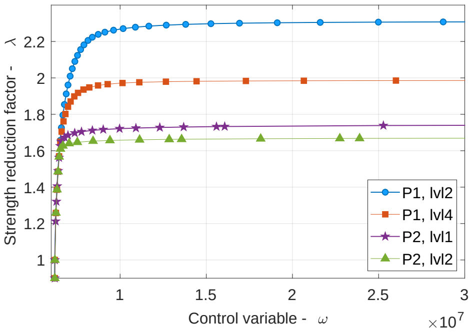
Numerical example II - heterogeneous slope in 3D

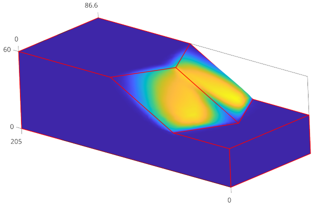
Setting the problem:

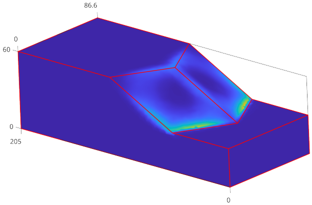
- geometry from [Zhou et al., Water, 2020]
- only SR method
- comparison of linear and quadratic finite elements (P1 and P2 elements)
- meshes with 75–600 thousands DoFs created in SW GMSH



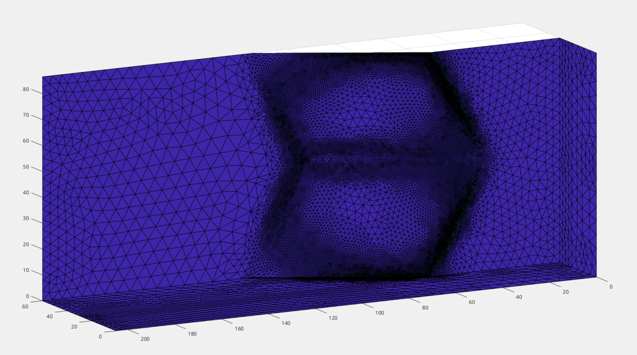
Visualization of the curve $\omega \mapsto \lambda_\omega$ and failure







Local mesh adaptivity



- refinement on basis of the results from the original (coarser) mesh
- more accurate failure zones and FoS

7. Conclusion

Concluding remarks:

- Original analysis of the stability methods on abstract algebraic problems
- Deriving of advanced continuation techniques
- Development of efficient iterative solvers for 3D problems
- Development of in-house codes in Matlab, see https://github.com/sysala/SSRM
- Cooperation with the SW companies FEM Consulting and Dlubal Software

References:

- S. Sysala, J. Haslinger, I. Hlaváček, M. Čermák: Discretization and numerical realization of contact problems for elastic-perfectly plastic bodies. PART I - discretization, limit analysis. ZAMM 95, 2015, pages 333-353.
- M. Čermák, J. Haslinger, T. Kozubek, S. Sysala: Discretization and numerical realization of contact problems for elastic-perfectly plastic bodies. PART II - numerical realization, limit analysis. ZAMM 95, 2015, pages 1348-1371.
- S. Sysala, E. Hrubešová, Z. Michalec, F. Tschuchnigg: Optimization and variational principles for the shear strength reduction method. International Journal for Numerical and Analytical Methods in Geomechanics 45, 2021, pages 2388-2407.

Thank you for your attention!