A fast nonlinear conjugate gradient based method for frictional contact problems

J. Zhao, E.A.H. Vollebregt, C.W. Oosterlee

Delft Institute of Applied Mathematics, TU Delft

2nd Feb., 2015

Delft University of Technology

Outline

- Introduction.
- Formulation of frictional contact.
- TangCG algorithm.
- Numerical results of a Cattaneo shift problem.
- Conclusion.

Introduction

United States

France

Spain

China

Germany

England

Introduction

United States

France

Spain

China

Germany

England

- Speed

Introduction

United States

France

Spain

China

Germany

England

- Speed
- Security (e.g. anti-derailment).

Introduction

United States

France

Spain

China

Germany

England

- Speed
- Security (e.g. anti-derailment).
- Comfort (e.g. less noise, less swaying).

Introduction

United States

France

Spain

China

Germany

England

- Speed
- Security (e.g. anti-derailment).
- Comfort (e.g. less noise, less swaying).

Introduction

United States

France

Spain

China

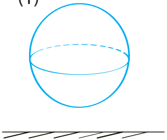
Germany

England

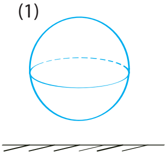
- Speed
- Security (e.g. anti-derailment).
- Comfort (e.g. less noise, less swaying).

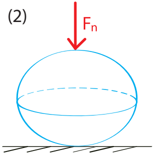
Introduction

(1)



Introduction



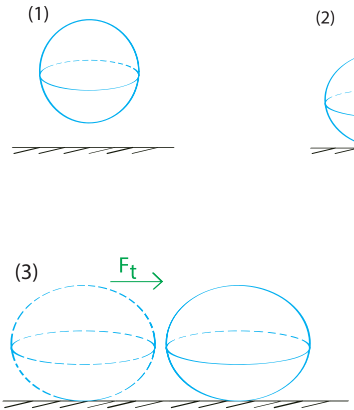


Delft University of Technology

Introduction

Pressure &
Contact area

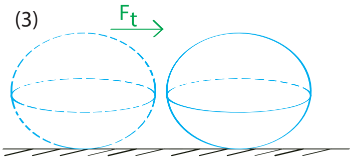
Introduction

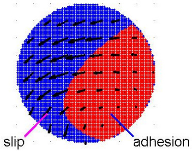


Pressure &
Contact area

Introduction

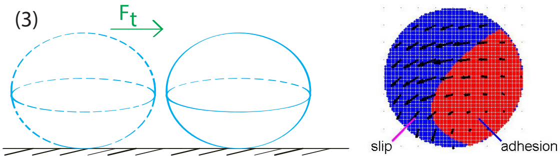
Pressure &
Contact area





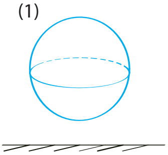
Introduction

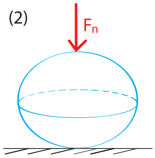
Pressure &
Contact area



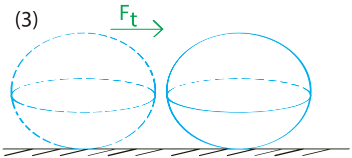
Frictional stress,
adhesion & slip

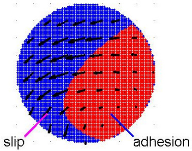
Introduction





Pressure &
Contact area





Frictional stress,
adhesion & slip

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \ \phi = \frac{1}{2}\mathbf{p}^T A\mathbf{p} + \mathbf{w}^T\mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \ \phi = \frac{1}{2}\mathbf{p}^T A \mathbf{p} + \mathbf{w}^T \mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

- Coulomb's friction law: $g_I = \mu p_{nI}$.

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \ \phi = \frac{1}{2}\mathbf{p}^T A \mathbf{p} + \mathbf{w}^T \mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

- Coulomb's friction law: $g_I = \mu p_{nI}$.
- A is symmetric, positive definite (SPD), dense, and block Toeplitz with Toeplitz blocks (BTTB).

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \ \phi = \frac{1}{2}\mathbf{p}^T A \mathbf{p} + \mathbf{w}^T \mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

- Coulomb's friction law: $g_I = \mu p_{nI}$.
- A is symmetric, positive definite (SPD), dense, and block Toeplitz with Toeplitz blocks (BTTB).
- Constraints are nonlinear in 3D.

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \;\; \phi = \frac{1}{2}\mathbf{p}^T A \mathbf{p} + \mathbf{w}^T \mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

- Coulomb's friction law: $g_I = \mu p_{nI}$.
- A is symmetric, positive definite (SPD), dense, and block Toeplitz with Toeplitz blocks (BTTB).
- Constraints are nonlinear in 3D.
- Convex optimization.

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \ \phi = \frac{1}{2}\mathbf{p}^T A \mathbf{p} + \mathbf{w}^T \mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

- Coulomb's friction law: $g_I = \mu p_{nI}$.
- A is symmetric, positive definite (SPD), dense, and block Toeplitz with Toeplitz blocks (BTTB).
- Constraints are nonlinear in 3D.
- Convex optimization.

Existing methods: TANG method, ConvexGS method.

Formulation of frictional contact

Kalker's numerical variational form (after discretization):

$$\min_{\mathbf{p}} \ \phi = \frac{1}{2}\mathbf{p}^T A \mathbf{p} + \mathbf{w}^T \mathbf{p}$$

$$\text{sub } ||\mathbf{p}_I|| \leq g_I, \quad \text{for element } I = 1, ..., N$$

- Coulomb's friction law: $g_I = \mu p_{nI}$.
- A is symmetric, positive definite (SPD), dense, and block Toeplitz with Toeplitz blocks (BTTB).
- Constraints are nonlinear in 3D.
- Convex optimization.

Existing methods: TANG method, ConvexGS method.

We aim at a faster solver.

What to solve:

Convex optimization problem:

What to solve:

Convex optimization problem:

1. unique minimum.

What to solve:

Convex optimization problem:

1. unique minimum.
2. the KKT conditions provide both necessary and sufficient conditions.

What to solve:

Convex optimization problem:

1. unique minimum.
2. the KKT conditions provide both necessary and sufficient conditions.

According to KKT condition, the governing equations are:

- $\mathbf{s} = A\mathbf{p} + \mathbf{w}$,
- In adhesion area H:

$$\mathbf{s}_I = \mathbf{0}$$

- In slip area S:

$$\begin{cases} ||\mathbf{p}_I|| = g_I \\ p_{Ix}s_{Iy} - p_{Iy}s_{Ix} = 0 \end{cases}$$

What to solve:

Convex optimization problem:

1. unique minimum.
2. the KKT conditions provide both necessary and sufficient conditions.

According to KKT condition, the governing equations are:

- $\mathbf{s} = A\mathbf{p} + \mathbf{w}$,
- In adhesion area H:

$$\mathbf{s}_I = \mathbf{0}$$

- In slip area S:

$$\begin{cases} ||\mathbf{p}_I|| = g_I \\ p_{Ix}s_{Iy} - p_{Iy}s_{Ix} = 0 \end{cases}$$

What to solve:

Convex optimization problem:

1. unique minimum.
2. the KKT conditions provide both necessary and sufficient conditions.

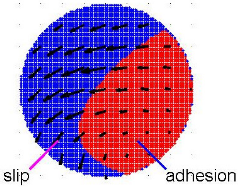
According to KKT condition, the governing equations are:

- $\mathbf{s} = A\mathbf{p} + \mathbf{w}$,
- In adhesion area H:

$$\mathbf{s}_I = \mathbf{0}$$

- In slip area S:

$$\begin{cases} ||\mathbf{p}_I|| = g_I \\ p_{Ix}s_{Iy} - p_{Iy}s_{Ix} = 0 \end{cases}$$



What to solve:

Convex optimization problem:

1. unique minimum.
2. the KKT conditions provide both necessary and sufficient conditions.

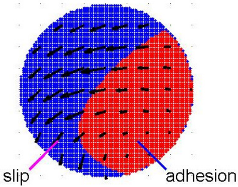
According to KKT condition, the governing equations are:

- $\mathbf{s} = A\mathbf{p} + \mathbf{w}$,
- In adhesion area H:

$$\mathbf{s}_I = \mathbf{0}$$

- In slip area S:

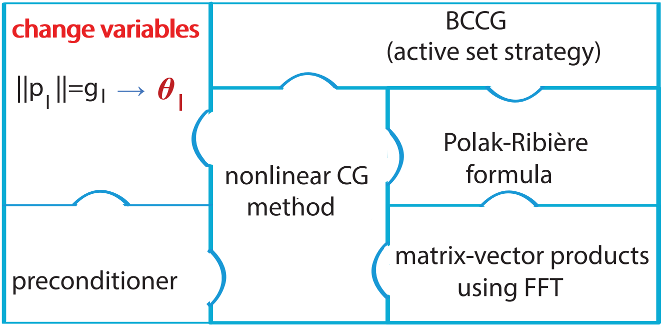
$$\begin{cases} ||\mathbf{p}_I|| = g_I \\ p_{Ix} s_{Iy} - p_{Iy} s_{Ix} = 0 \end{cases}$$



We solve for $\mathbf{p}$, H and S.

TangCG algorithm

Main components of TangCG:



TangCG algorithm

♠ Change variables in slip area

Inspired by:

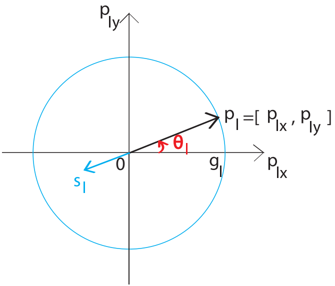
In slip area: $||\mathbf{p}_I|| = g_I$

TangCG algorithm

♠ Change variables in slip area

Inspired by:

In slip area: $||\mathbf{p}_I|| = g_I$



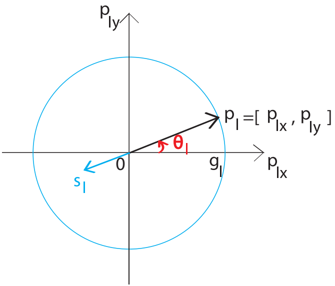
Delft University of Technology

TangCG algorithm

♠ Change variables in slip area

Inspired by:

In slip area: $||\mathbf{p}_I|| = g_I$



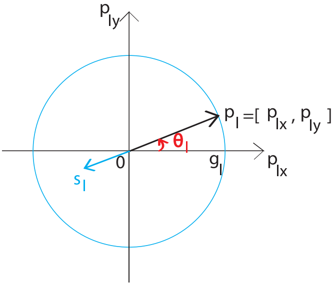
- Conventional variables
  p_{Ix}, p_{Iy}

TangCG algorithm

♠ Change variables in slip area

Inspired by:

In slip area: $||\mathbf{p}_l|| = g_l$



- Conventional variables

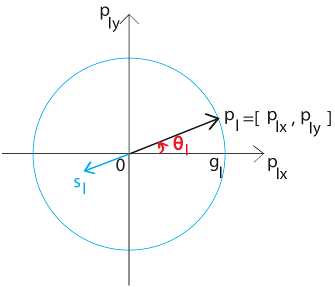
  $p_{lx}, p_{ly} \Rightarrow \quad \theta_l$.

TangCG algorithm

♠ Change variables in slip area

Inspired by:

In slip area: $||\mathbf{p}_I|| = g_I$



- Conventional variables $p_{Ix}, p_{Iy} \Rightarrow \quad \theta_I$.
- Governing equations for slip element are reduced to

$$p_{Ix} s_{Iy} - p_{Iy} s_{Ix} = 0.$$

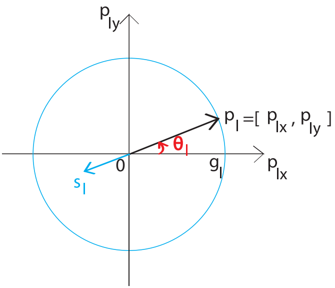
Delft University of Technology

TangCG algorithm

♠ Change variables in slip area

Inspired by:

In slip area: $||\mathbf{p}_I|| = g_I$

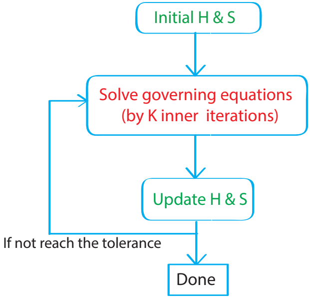


- Conventional variables $p_{Ix}, p_{Iy} \Rightarrow \quad \theta_I$.
- Governing equations for slip element are reduced to

$$p_{Ix}s_{Iy} - p_{Iy}s_{Ix} = 0.$$

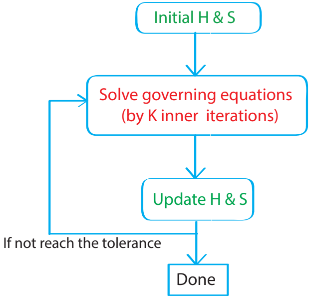
TangCG algorithm

♠ Active set strategy:



TangCG algorithm

♠ Active set strategy:

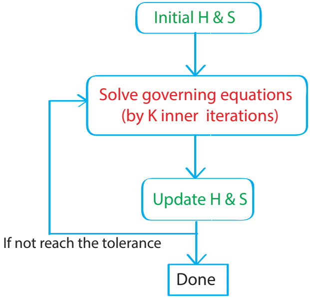


The governing equations:

$$\begin{cases} \mathbf{s} = A\mathbf{p} + \mathbf{w} \\ \mathbf{s}_l = \mathbf{0}, \quad l \in H \\ p_{lx}s_{ly} - p_{ly}s_{lx} = 0, \quad l \in S \end{cases}$$

TangCG algorithm

♠ Active set strategy:



The governing equations:

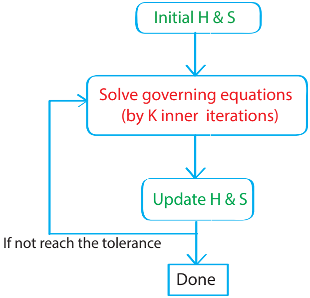
$$\begin{cases} \mathbf{s} = A\mathbf{p} + \mathbf{w} \\ \mathbf{s}_l = \mathbf{0}, \quad l \in H \\ p_{lx} s_{ly} - p_{ly} s_{lx} = 0, \quad l \in S \end{cases}$$

- Nonlinear system.

Delft University of Technology

TangCG algorithm

♠ Active set strategy:



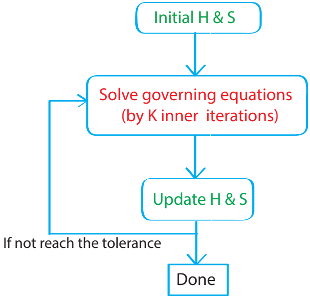
The governing equations:

$$\begin{cases} \mathbf{s} = A\mathbf{p} + \mathbf{w} \\ \mathbf{s}_l = \mathbf{0}, \quad l \in H \\ p_{lx}s_{ly} - p_{ly}s_{lx} = 0, \quad l \in S \end{cases}$$

- Nonlinear system.
- Nonlinear CG method.

TangCG algorithm

♠ Active set strategy:



The governing equations:

$$\begin{cases} \mathbf{s} = A\mathbf{p} + \mathbf{w} \\ \mathbf{s}_l = \mathbf{0}, \quad l \in H \\ p_{lx}s_{ly} - p_{ly}s_{lx} = 0, \quad l \in S \end{cases}$$

- Nonlinear system.
- Nonlinear CG method.

TangCG algorithm

♠ Nonlinear CG: in each iteration:

TangCG algorithm

♠ Nonlinear CG: in each iteration:

1. Linearize at $\mathbf{p}^k$:

$$J^k \cdot \delta\mathbf{p}^k = -F(p^k).$$

TangCG algorithm

♠ Nonlinear CG: in each iteration:

1. Linearize at $\mathbf{p}^k$:

$$J^k \cdot \delta \mathbf{p}^k = -F(p^k).$$

2. Solve this linearized equation:
   2.1 Compute residual $\mathbf{r}^k$.
   2.2 Construct search direction $\mathbf{v}^k$.
   2.3 Perform a line search, and obtain the steplength:

$$\alpha^k = \frac{(\mathbf{v}^k, \mathbf{r}^k)}{(\mathbf{v}^k, \mathbf{q}^k)}$$

where $\mathbf{q}^k = J^k \mathbf{v}^k$.

TangCG algorithm

♠ Nonlinear CG: in each iteration:

1. Linearize at $\mathbf{p}^k$:

$$J^k \cdot \delta \mathbf{p}^k = -F(p^k).$$

2. Solve this linearized equation:
   2.1 Compute residual $\mathbf{r}^k$.
   2.2 Construct search direction $\mathbf{v}^k$.
   2.3 Perform a line search, and obtain the steplength:

$$\alpha^k = \frac{(\mathbf{v}^k, \mathbf{r}^k)}{(\mathbf{v}^k, \mathbf{q}^k)}$$

   where $\mathbf{q}^k = J^k \mathbf{v}^k$.

3. Update:

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \alpha^k \mathbf{v}^k.$$

TangCG algorithm

♠ Nonlinear CG: in each iteration:

1. Linearize at $\mathbf{p}^k$:

$$J^k \cdot \delta \mathbf{p}^k = -F(p^k).$$

2. Solve this linearized equation:
   2.1 Compute residual $\mathbf{r}^k$.
   2.2 Construct search direction $\mathbf{v}^k$.
   2.3 Perform a line search, and obtain the steplength:

$$\alpha^k = \frac{(\mathbf{v}^k, \mathbf{r}^k)}{(\mathbf{v}^k, \mathbf{q}^k)}$$

   where $\mathbf{q}^k = J^k \mathbf{v}^k$.

3. Update:

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \alpha^k \mathbf{v}^k.$$

Delft University of Technology

TangCG algorithm

Construct search direction?

TangCG algorithm

Construct search direction?

- Polak-Ribière formula, to obtain conjugate search directions.

TangCG algorithm

Construct search direction?

- Polak-Ribière formula, to obtain conjugate search directions.
- Use the steepest descent direction when H and S are changed.

TangCG algorithm

Construct search direction?

- Polak-Ribière formula, to obtain conjugate search directions.
- Use the steepest descent direction when H and S are changed.

Compute $\mathbf{q}^k = J^k \mathbf{v}^k$?

TangCG algorithm

Construct search direction?

- Polak-Ribière formula, to obtain conjugate search directions.
- Use the steepest descent direction when H and S are changed.

Compute $\mathbf{q}^k = J^k \mathbf{v}^k$?

- Not necessary to generate J^k explicitly.

TangCG algorithm

Construct search direction?

- Polak-Ribière formula, to obtain conjugate search directions.
- Use the steepest descent direction when H and S are changed.

Compute $\mathbf{q}^k = J^k \mathbf{v}^k$?

- Not necessary to generate J^k explicitly.
- Involves computing the matrix-vector products with a BTTB matrix A.

TangCG algorithm

Construct search direction?

- Polak-Ribière formula, to obtain conjugate search directions.
- Use the steepest descent direction when H and S are changed.

Compute $\mathbf{q}^k = J^k \mathbf{v}^k$?

- Not necessary to generate J^k explicitly.
- Involves computing the matrix-vector products with a BTTB matrix A.
- Apply the fast Fourier transform (FFT) technique, the complexity is reduced to $\mathcal{O}(n \log n)$ from $\mathcal{O}(n^2)$.

TangCG algorithm

One iteration on the slip element *l*:

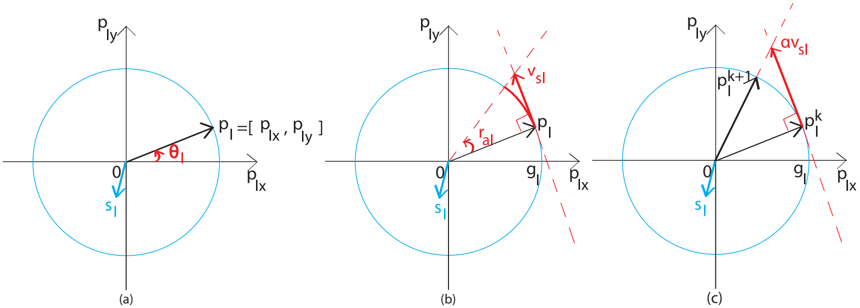


Figure: (a) current iterate. (b) residual and search direction. (c) update.

TangCG algorithm

One iteration on the slip element *l*:

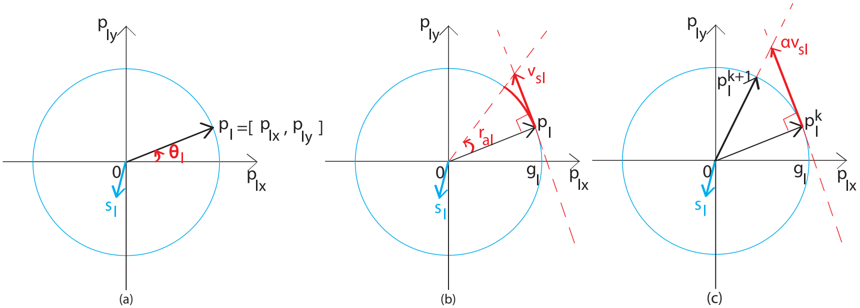


Figure: (a) current iterate. (b) residual and search direction. (c) update.

TangCG algorithm

♠ Preconditioner:

- For what matrix?

TangCG algorithm

♠ Preconditioner:

- For what matrix?
  ⇒ Jacobian matrix in each nonlinear CG iteration.

TangCG algorithm

♠ Preconditioner:

- For what matrix?
  $\Rightarrow$ Jacobian matrix in each nonlinear CG iteration.
- Use what approach?

TangCG algorithm

♠ Preconditioner:

- For what matrix?
  $\Rightarrow$ Jacobian matrix in each nonlinear CG iteration.
- Use what approach?
  $\Rightarrow$ Diagonal Scaling.

TangCG algorithm

♠ Preconditioner:

- For what matrix?
  $\Rightarrow$ Jacobian matrix in each nonlinear CG iteration.
- Use what approach?
  $\Rightarrow$ Diagonal Scaling.
- How to precondition?

TangCG algorithm

♠ Preconditioner:

- For what matrix?
  $\Rightarrow$ Jacobian matrix in each nonlinear CG iteration.
- Use what approach?
  $\Rightarrow$ Diagonal Scaling.
- How to precondition?
  $\Rightarrow$ Make all diagonal elements equal.

TangCG algorithm

♠ Preconditioner:

- For what matrix?
  $\Rightarrow$ Jacobian matrix in each nonlinear CG iteration.
- Use what approach?
  $\Rightarrow$ Diagonal Scaling.
- How to precondition?
  $\Rightarrow$ Make all diagonal elements equal.

Numerical results of a Cattaneo shift problem

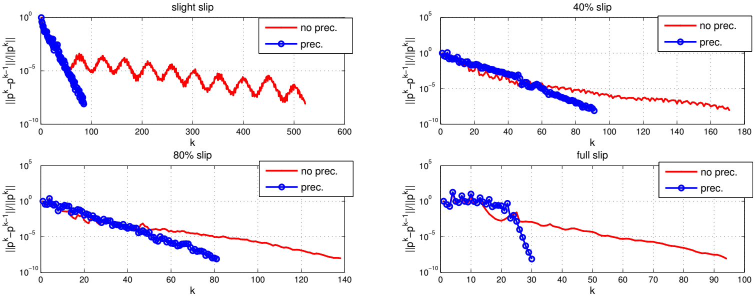
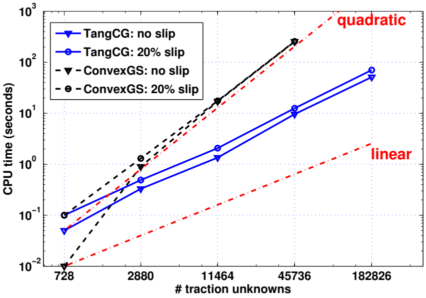
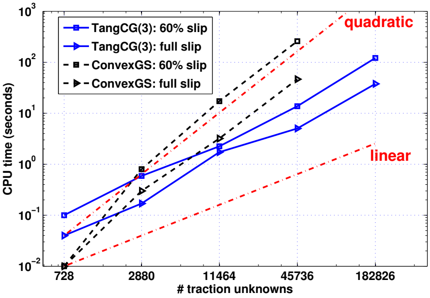


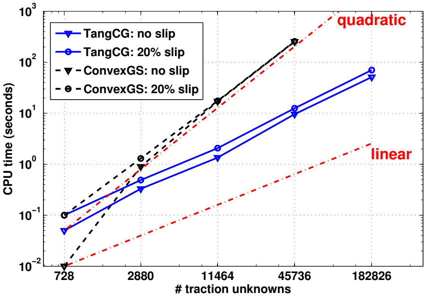
Figure: The convergence behavior of TangCG with and without prec.

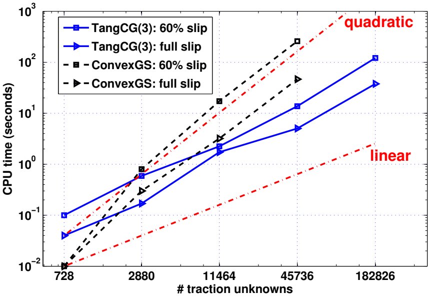
Numerical results and conclusion





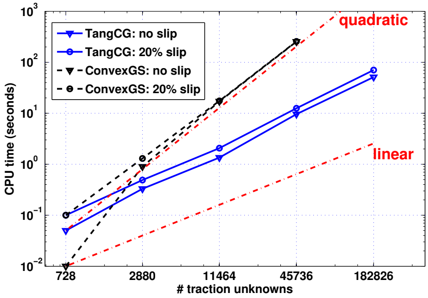
Numerical results and conclusion

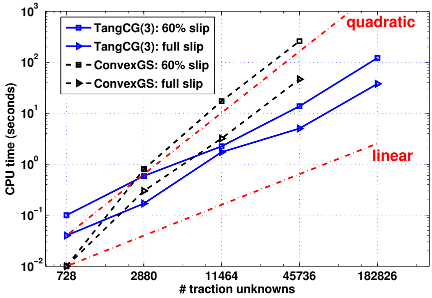




Conclusion about TangCG with prec.:

Numerical results and conclusion

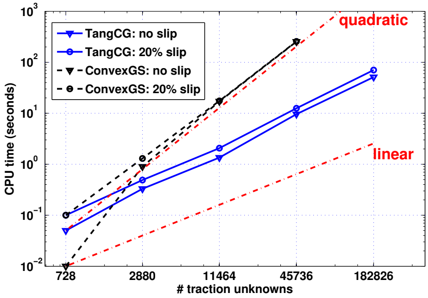


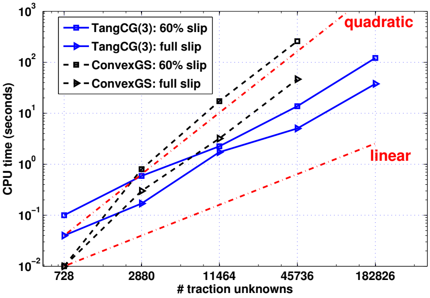


Conclusion about TangCG with prec.:

- robust: it works well for different situation of slip.

Numerical results and conclusion



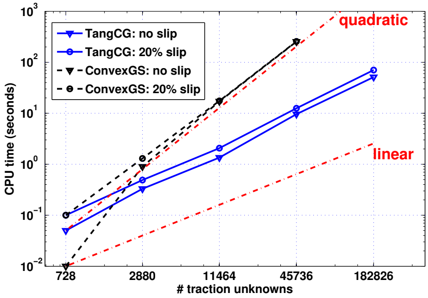


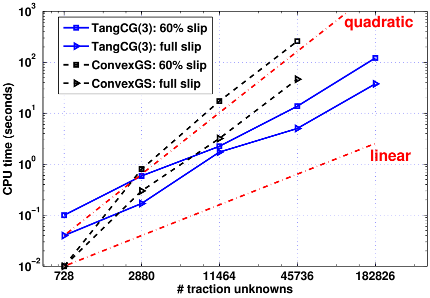
Conclusion about TangCG with prec.:

- robust: it works well for different situation of slip.
- fast: it reduces the computational time dramatically (around $\mathcal{O}(n^{1.7})$).

Delft University of Technology

Numerical results and conclusion





Conclusion about TangCG with prec.:

- robust: it works well for different situation of slip.
- fast: it reduces the computational time dramatically (around $\mathcal{O}(n^{1.7})$).

Thank you for your attention!