The use of CONTACT in dynamical simulations

Hugo de Looij

Faculty EEMCS, Applied Mathematics

November 26th 2015
About VORtech

- Founded in 1996 in Delft.
- Specialized in mathematical consultancy and development of high performance scientific software.
- Broad range of customers.
CONTACT

- Software that solves contact problems between two objects (e.g. train wheel & rails).
- Main problem: simulation of a train over a bridge.
- Research question: how can CONTACT be used for dynamical contact problems?
Rigid body motion

- A rigid ball is dropped on an elastic surface.
- Height ball $z(t)$ measured from a reference point.
- How to compute $z(t)$?

Gravitational force $F_g = mg$

Normal force exerted by the surface $F_n = 4/3 E^* \sqrt{\delta(t)}^{3/2}$

The use of CONTACT in dynamical simulations November 26th 2015
Rigid body motion

- A rigid ball is dropped on an elastic surface.
- Height ball $z(t)$ measured from a reference point.
- How to compute $z(t)$?

Gravitational force

$$F_g = mg$$
A rigid ball is dropped on an elastic surface.

Height ball $z(t)$ measured from a reference point.

How to compute $z(t)$?

Gravitational force

$$F_g = mg$$

Normal force exerted by the surface

$$F_n = \frac{4}{3} E^* \sqrt{R} \delta(t)^{3/2}$$
Possibilities CONTACT

CONTACT is capable of

- Computing the normal force $F_n$.
- Computing the distribution of the pressure $p_n$ in the contact area.
- Computing the (quasi-)stationary elastic deformation of the surface.

Can be done for arbitrarily shaped objects by supplying the penetration.
Rigid body motion

- The resulting force is $F(z) = F_n(z) - F_g$.
- By Newton’s second law:

$$m \ddot{z} = F(z) = F_n(z) - mg$$

Can be solved using a time integration scheme.
Time integration schemes

Different kind of integration schemes:

- Runge-Kutta schemes, (Forward Euler, RK4, ...)
- Radau schemes, (Backward Euler, Radau5, ...)
- Verlet,
- Leapfrog,
- Adams methods,
- Backward differentiation formulas,
- Newmark-beta,
- HHT, and
- Generalized-α integration.
Numerical results
Computing the deformation

The deformation of the surface is described by:

\[
\begin{align*}
\frac{\rho}{\partial t^2} \frac{\partial^2 u_i}{\partial t^2} &= \sum_{j=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_j} + F_i & i = 1, 2, 3 \\
\sigma_{ij} &= 2Ge_{ij} + \lambda \delta_{ij} \sum_{k=1}^{3} e_{kk} & i, j = 1, 2, 3 \\
e_{ij} &= \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) & i, j = 1, 2, 3
\end{align*}
\]

This can be solved using a Finite Element approach.

- Expensive, need to discretise w.r.t. \( z \) direction.
- Accurate, inertia is taken into account.
The quasi-static deformation

Using CONTACT

- Computational inexpensive.
- Quasi-static, inertia at the surface elements is ignored.
Global deformation of a bridge

The (1D) Euler-Bernoulli beam equation:

\[
\frac{\partial^2}{\partial x^2} \left( E(x) I(x) \frac{\partial^2 u}{\partial x^2} \right) = -\rho(x) \frac{\partial^2 u}{\partial t^2} + p(x, t)
\]
Global deformation of a bridge

The (1D) Euler-Bernoulli beam equation:

\[
\frac{\partial^2}{\partial x^2} \left( E(x) I(x) \frac{\partial^2 u}{\partial x^2} \right) = -\rho(x) \frac{\partial^2 u}{\partial t^2} + p(x, t)
\]

\[
EI \frac{\partial^4 u}{\partial x^4} = -\rho \frac{\partial^2 u}{\partial t^2} + p(x, t)
\]
Modal analysis

Mode shapes are natural vibrations of the beam.

Substitute $u(x, t) = e^{i\lambda t}w(x)$ into

$$EI \frac{\partial^4 u}{\partial x^4} = -\rho \frac{\partial^2 u}{\partial t^2}$$

$$\Rightarrow EI w^{(4)}(x) = \rho \lambda^2 w(x)$$
Modal analysis

Mode shapes are natural vibrations of the beam.

Substitute $u(x, t) = e^{i\lambda t}w(x)$ into

$$EI \frac{\partial^4 u}{\partial x^4} = -\rho \frac{\partial^2 u}{\partial t^2}$$

$$\implies EI w^{(4)}(x) = \rho \lambda^2 w(x)$$

$$\implies w(x) = c_1 \cos(\beta x) + c_2 \sin(\beta x)$$
$$+ c_3 \cosh(\beta x) + c_4 \sinh(\beta x)$$

where $\beta = \left(\frac{\rho \lambda^2}{EI}\right)^{1/4}$
Modal analysis

The mode shapes for a clamped beam satisfy

$$\cos(\beta L) \cosh(\beta L) = 1$$
Modal analysis

The solution can be written as

\[ u(x, t) = \sum_{i=1}^{\infty} c_i(t) w_i(x) \]
Modal analysis

The solution can be written as

\[ u(x, t) = \sum_{i=1}^{\infty} c_i(t)w_i(x) \]

Idea: approximate \( u \) by

\[ u(x, t) \approx u_m(x, t) = \sum_{i=1}^{m} c_i(t)w_i(x) \]

How do we compute \( c_i(t) \)?
Modal analysis

\[ EI \sum_{i=1}^{\infty} c_i(t)w_i^{(4)}(x) = -\rho \sum_{i=1}^{\infty} c_i''(t)w_i(x) + p(x, t) \]
Modal analysis

\[ EI \sum_{i=1}^{\infty} c_i(t) w_i^{(4)}(x) = -\rho \sum_{i=1}^{\infty} c_i''(t) w_i(x) + p(x, t) \]

We have \( w_i^{(4)}(x) = \beta_i^4 w(x) \), so that

\[ \sum_{i=1}^{\infty} w_i(x) \left[ EI \beta_i^4 c_i(t) + \rho c_i''(t) \right] = p(x, t) \]
Modal analysis

\[ EI \sum_{i=1}^{\infty} c_i(t) w_i^{(4)}(x) = -\rho \sum_{i=1}^{\infty} c_i''(t) w_i(x) + p(x, t) \]

We have \( w_i^{(4)}(x) = \beta_i^4 w(x) \), so that

\[ \sum_{i=1}^{\infty} w_i(x) \left[ EI \beta_i^4 c_i(t) + \rho c_i''(t) \right] = p(x, t) \]

Multiplying by \( w_j(x) \) and integrating over \([0, L] \) yields

\[ \sum_{i=1}^{\infty} \left( [EI \beta_i^4 c_i(t) + \rho c_i''(t)] \int_0^L w_i(x) w_j(x) dx \right) = \int_0^L p(x, t) w_j(x) dx \]
Modal analysis

We arrive at the differential equation:

\[
\rho c_j''(t) = \int_0^L p(x, t)w_j(x)dx - EI \beta_j^A c_j(t)
\]
Modal analysis

We arrive at the differential equation:

\[ \rho c_j''(t) = \int_0^L p(x, t)w_j(x)dx - EI \beta_j^4 c_j(t) \]

The modal coefficients \( c_i \) are independent of each other.
Modal analysis

We arrive at the differential equation:

$$\rho c_j''(t) = \int_0^L p(x, t)w_j(x)dx - EI\beta_j^4 c_j(t)$$

The modal coefficients $c_i$ are independent of each other.

Can be solved by combining
Modal analysis

We arrive at the differential equation:

$$\rho c_j''(t) = \int_0^L p(x, t) w_j(x) \, dx - EI \beta_j^4 c_j(t)$$

The modal coefficients $c_i$ are independent of each other.

Can be solved by combining

- A numerical integrator (such as a Newton-Cotes formula),
Modal analysis

We arrive at the differential equation:

\[ \rho c_j''(t) = \int_0^L p(x, t)w_j(x)dx - EI\beta_j^4 c_j(t) \]

The modal coefficients \( c_i \) are independent of each other.

Can be solved by combining

- A numerical integrator (such as a Newton-Cotes formula),
- A time integration scheme (such as Newmark-beta), and
Modal analysis

We arrive at the differential equation:

$$\rho c''_j(t) = \int_0^L p(x, t)w_j(x)dx - EI\beta_j^4 c_j(t)$$

The modal coefficients $c_i$ are independent of each other.

Can be solved by combining

- A numerical integrator (such as a Newton-Cotes formula),
- A time integration scheme (such as Newmark-beta), and
- An iterative solver (such as Picard iteration).
The error of the stationary modal solution satisfies

\[ \|u - u_m\|_2 \leq \frac{L^4 \|p\|_2}{3\pi^4 EI m^3} = \mathcal{O}(m^{-3}) \]
Combining local and global deformation

We described two different phenomena:

- Local deformation (occurring around the contact area), and
- Global deformation.

\[
\text{u}_{\text{tot}}(x,t) = \text{u}(x,t) + \text{l}(x,t)
\]
Combining local and global deformation

We described two different phenomena:

- Local deformation (occurring around the contact area), and
- Global deformation.

In reality, a bridge can deform both globally as locally:

\[ u_{\text{tot}}(x, t) = u(x, t) + l(x, t) \]
Combining local and global deformation
Combining local and global deformation

- The global deformation can be derived by superposing the modal coefficients that satisfy

\[ \rho c''_j(t) = \int_0^L p(x, t)w_j(x)dx - EI\beta^4_j c_j(t) \]

- The rigid height \( z(t) \) of the wheel can be derived by solving

\[ m_w \frac{d^2 z}{dt^2} = F(t) \]
Combining local and global deformation

- The global deformation can be derived by superposing the modal coefficients that satisfy
  \[ \rho c''_j(t) = \int_0^L p(x, t)w_j(x)dx - EI\beta^4_j c_j(t) \]

- The rigid height \( z(t) \) of the wheel can be derived by solving
  \[ m_w \frac{d^2z}{dt^2} = F(t) \]

- Both \( p(x, t) \) and \( F(t) \) are derived using CONTACT by supplying the penetration
  \[ \delta(x, t) = \sum_{j=1}^{m} c_j(t)w_j(x) - [z(t) + g(x - st)] \]
Combining local and global deformation

Applied to Backward Euler:

\[
\begin{align*}
c_{k+1}^1 &= c_1^k + \Delta t \dot{c}_1^{k+1} \\
\dot{c}_1^{k+1} &= \dot{c}_1^k + \frac{\Delta t}{\rho} \left[ \int_0^L p(x, c_{k+1}^1, z_{k+1}^1, t_{k+1}^1) w_1(x) \, dx - EI \beta_1^4 c_{k+1}^1 \right] \\
&\vdots \\
c_{m+1}^1 &= c_m^k + \Delta t \dot{c}_m^{k+1} \\
\dot{c}_m^{k+1} &= \dot{c}_m^k + \frac{\Delta t}{\rho} \left[ \int_0^L p(x, c_{k+1}^m, z_{k+1}^m, t_{k+1}^m) w_m(x) \, dx - EI \beta_m^4 c_{m+1}^1 \right] \\
z_{k+1} &= z_k + \Delta t \dot{z}_{k+1} \\
\dot{z}_{k+1} &= \dot{z}_k + \Delta t \left[ \frac{F(c_{k+1}^1, z_{k+1}^1, t_{k+1}^1)}{m_c} - g \right]
\end{align*}
\]
Combining local and global deformation

This can be written as

\[
\begin{pmatrix}
1 - \Delta t \\
\Delta t \frac{EI}{\rho} \beta_1 \\
\vdots \\
1 - \Delta t \\
0 1
\end{pmatrix}
\begin{pmatrix}
c_1 \\
\dot{c}_1 \\
\vdots \\
z \\
\dot{z}
\end{pmatrix}^{k+1}
= \begin{pmatrix}
c_1 \\
\dot{c}_1 \\
\vdots \\
z \\
\dot{z}
\end{pmatrix}^k + \Delta t
\begin{pmatrix}
1 \\
\frac{1}{\rho} \int_0^L p(x, c^{k+1}, z^{k+1}, t^{k+1}) w_1(x) dx \\
\vdots \\
\frac{1}{\rho} \int_0^L p(x, c^{k+1}, z^{k+1}, t^{k+1}) w_m(x) dx \\
\frac{1}{m_c} \int F(c^{k+1}, z^{k+1}, t^{k+1}) - g
\end{pmatrix}
\]

or

\[A y^{k+1} = y^k + \Delta t f^{k+1}\]
Combining local and global deformation

This can be written as

\[
\begin{pmatrix}
\frac{1}{\Delta t} - \Delta t \\
\frac{1}{\Delta t} - \Delta t \\
\frac{1}{\Delta t} - \Delta t \\
\frac{1}{\rho} \int_0^L p(x, c^{k+1}, z^{k+1}, t^{k+1}) w_1(x) dx \\
\frac{1}{\rho} \int_0^L p(x, c^{k+1}, z^{k+1}, t^{k+1}) w_m(x) dx \\
\frac{1}{m_c} - g
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_1 \\
c_m \\
c_m \\
c_m \\
a_{c^{k+1}, z^{k+1}, t^{k+1}} - g
\end{pmatrix}
\begin{pmatrix}
k+1 \\
k+1 \\
k+1 \\
k+1 \\
k+1 \\
k+1
\end{pmatrix}
\]

or

\[
Ay^{k+1} = y^k + \Delta tf^{k+1}
\]

Picard approach:

\[
y^{k+1} = A^{-1}(y^k + \Delta tf^{k})
\]

For stiff materials, this doesn’t converge!
The Quasi-Newton approach

We linearise $F(c_j^{k+1}, z_j^{k+1}, t^{k+1})$ at each iteration $j$.

- Cannot be done analytically, but instead we can set

$$\frac{\partial F_j}{\partial z_j^{k+1}} \approx \frac{F(c_j^{k+1}, z_j^{k+1}, t^{k+1}) - F(c_j^{k+1}, z_j^{k+1} - \alpha, t^{k+1})}{\alpha}$$
The Quasi-Newton approach

We linearise $F(c_j^{k+1}, z_j^{k+1}, t^{k+1})$ at each iteration $j$.

- Cannot be done analytically, but instead we can set
  
  $$\frac{\partial F_j}{\partial z_j^{k+1}} \approx \frac{F(c_j^{k+1}, z_j^{k+1}, t^{k+1}) - F(c_j^{k+1}, z_j^{k+1} - \alpha, t^{k+1})}{\alpha}$$

- To achieve stability, we should also linearise w.r.t to $c_j^{k+1}$. 

The use of CONTACT in dynamical simulations November 26th 2015
The Quasi-Newton approach

We linearise $F(c_j^{k+1}, z_j^{k+1}, t^{k+1})$ at each iteration $j$.

- Cannot be done analytically, but instead we can set

$$
\frac{\partial F_j}{\partial z_j^{k+1}} \approx \frac{F(c_j^{k+1}, z_j^{k+1}, t^{k+1}) - F(c_j^{k+1}, z_j^{k+1} - \alpha, t^{k+1})}{\alpha}
$$

- To achieve stability, we should also linearise w.r.t to $c_j^{k+1}$.

- This is computationally very expensive.
The Quasi-Newton approach

We linearise $F(c_{j}^{k+1}, z_{j}^{k+1}, t^{k+1})$ at each iteration $j$.

- Cannot be done analytically, but instead we can set

$$\frac{\partial F}{\partial z_{j}^{k+1}} \approx \frac{F(c_{j}^{k+1}, z_{j}^{k+1}, t^{k+1}) - F(c_{j}^{k+1}, z_{j}^{k+1} - \alpha, t^{k+1})}{\alpha}$$

- To achieve stability, we should also linearise w.r.t to $c_{j}^{k+1}$.
- This is computationally very expensive.

Idea: we only linearise with respect to the approach $\delta_{j}^{k+1}$.

$$F_{j+1}^{k+1} = F_{j}^{k+1} + \left(\frac{\partial F}{\partial \delta}\right)_{j}^{k+1} \cdot (\delta_{j+1}^{k+1} - \delta_{j}^{k+1})$$
The Quasi-Newton approach

\[ \delta(t) = \max_{0 \leq x \leq L} [u(x, t) - w(x, t)] \]

\[ = \max_{0 \leq x \leq L} \left[ \sum_{i=1}^{m} c_i(t)w_i(x) - g(x - st) \right] - z(t) \]
The Quasi-Newton approach

\[
\delta(t) = \max_{0 \leq x \leq L} [u(x, t) - w(x, t)]
\]

\[
= \max_{0 \leq x \leq L} \left[ \sum_{i=1}^{m} c_i(t) w_i(x) - g(x - st) \right] - z(t)
\]

After iteration \( j \) of time step \( k + 1 \), we have

\[
\delta_{j+1}^{k+1} = \max_{0 \leq x \leq L} \left[ \sum_{i=1}^{m} c_{i,j+1}^{k+1} w_i(x) - g(x - st_{k+1}^{j+1}) \right] - z_{j+1}^{k+1}
\]
The Quasi-Newton approach

\[ \delta(t) = \max_{0 \leq x \leq L} \left[ u(x, t) - w(x, t) \right] \]

\[ = \max_{0 \leq x \leq L} \left[ \sum_{i=1}^{m} c_i(t) w_i(x) - g(x - st) \right] - z(t) \]

After iteration \( j \) of time step \( k + 1 \), we have

\[ \delta_{j+1}^{k+1} = \max_{0 \leq x \leq L} \left[ \sum_{i=1}^{m} c_{i,j+1}^{k+1} w_i(x) - g(x - st_{k+1}^{j+1}) \right] - z_{j+1}^{k+1} \]

\[ \approx \sum_{i=1}^{m} c_{i,j+1}^{k+1} w_i(x_{j}^{k+1}) - g(x_{j}^{k+1} - st_{k+1}^{j+1}) - z_{j+1}^{k+1} \]
The Quasi-Newton approach

After linearising, we arrive at

\[
\begin{align*}
\dot{c}_{i,j}^{k+1} + \frac{\Delta t}{\rho} \left[ EI\beta_i^4 c_{i,j}^{k+1} + \left( \frac{\partial I}{\partial \delta} \right)_{i,j}^{k+1} \cdot \left( z_{j}^{k+1} - \sum_{l=1}^{m} c_{l,j}^{k+1} w_l(x_j^{k+1}) \right) \right] \\
= \dot{c}_i + \frac{\Delta t}{\rho} \left[ I_{i,j}^{k+1} + \left( \frac{\partial I}{\partial \delta} \right)_{i,j}^{k+1} \cdot \left( z_{j}^{k+1} - \sum_{l=1}^{m} c_{l,j}^{k+1} w_l(x_j^{k+1}) \right) \right] \\
\end{align*}
\]

\[
\begin{align*}
\dot{z}_{j}^{k+1} + \frac{\Delta t}{m_c} \cdot \frac{\partial}{\partial \delta} F_{j}^{k+1} \cdot \left( z_{j}^{k+1} - \sum_{i=1}^{m} c_{i,j}^{k+1} w_i(x_j^{k+1}) \right) \\
= \dot{z}^k - \Delta t g + \frac{\Delta t}{m_c} \left[ F_{j}^{k+1} + \frac{\partial}{\partial \delta} F_{j}^{k+1} \cdot \left( z_{j}^{k+1} - \sum_{i=1}^{m} c_{i,j}^{k+1} w_i(x_j^{k+1}) \right) \right]
\end{align*}
\]
The Quasi-Newton approach

This can be written as

\[ \dot{x}_{j+1}^{k+1} + \Delta t A_{j}^{k+1} x_{j}^{k+1} = \dot{x}^{k} + \Delta t g_{j}^{k+1}, \]

where

\[ A_{j}^{k+1} = \begin{pmatrix} \frac{EI}{\rho} \beta_{1}^{4} - \frac{w_{1}(x_{j}^{k+1})}{\rho} \cdot \frac{\partial}{\partial \delta} I_{1,j}^{k+1} & \cdots & -\frac{w_{m}(x_{j}^{k+1})}{\rho} \cdot \frac{\partial}{\partial \delta} I_{1,j}^{k+1} & \frac{1}{\rho} \cdot \frac{\partial}{\partial \delta} I_{1,j}^{k+1} \\ \vdots & \ddots & \vdots & \vdots \\ -\frac{w_{1}(x_{j}^{k+1})}{\rho} \cdot \frac{\partial}{\partial \delta} I_{m,j}^{k+1} & \cdots & \frac{EI}{\rho} \beta_{m}^{4} - \frac{w_{m}(x_{j}^{k+1})}{\rho} \cdot \frac{\partial}{\partial \delta} I_{m,j}^{k+1} & \frac{1}{\rho} \cdot \frac{\partial}{\partial \delta} I_{m,j}^{k+1} \\ -\frac{w_{1}(x_{j}^{k+1})}{m_{c}} \cdot \frac{\partial}{\partial \delta} F_{j}^{k+1} & \cdots & -\frac{w_{m}(x_{j}^{k+1})}{m_{c}} \cdot \frac{\partial}{\partial \delta} F_{j}^{k+1} & \frac{1}{m_{c}} \cdot \frac{\partial}{\partial \delta} F_{j}^{k+1} \end{pmatrix} \]
The Quasi-Newton approach

This can be written as

\[ \dot{x}_{j+1}^{k+1} + \Delta t A_{j}^{k+1} x_{j}^{k+1} = \dot{x}^{k} + \Delta t g_{j}^{k+1}, \]

where

\[
A_{j}^{k+1} = \begin{pmatrix}
\frac{EI}{\rho} \beta_{1} - \frac{w_{1}(x_{j}^{k+1})}{\rho} & \frac{\partial}{\partial \delta} F_{1,j}^{k+1} & \cdots & - \frac{w_{m}(x_{j}^{k+1})}{\rho} & \frac{\partial}{\partial \delta} F_{1,j}^{k+1} & 1 & \frac{\partial}{\partial \delta} I_{1,j}^{k+1} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-w_{1}(x_{j}^{k+1}) \frac{\rho}{m_{c}} & \frac{\partial}{\partial \delta} F_{j}^{k+1} & \cdots & - w_{m}(x_{j}^{k+1}) \frac{\rho}{m_{c}} & \frac{\partial}{\partial \delta} F_{j}^{k+1} & 1 & \frac{\partial}{\partial \delta} I_{m,j}^{k+1}
\end{pmatrix}
\]

The matrix \( A_{j}^{k+1} \) is dense; the mode shapes \( c_{i} \) and the rigid height \( z \) are dependent on each other!
The Quasi-Newton approach

We arrive at the system

\[
\begin{pmatrix}
I & -\Delta t I \\
A_j^{k+1} & I
\end{pmatrix}
\begin{pmatrix}
\dot{x}
\end{pmatrix}_{j+1}^{k+1} =
\begin{pmatrix}
\dot{x}
\end{pmatrix}_j^k + \Delta t
\begin{pmatrix}
0 \\
g_j^{k+1}
\end{pmatrix}
\]
The Quasi-Newton approach

We arrive at the system

\[
\begin{pmatrix}
I & -\Delta t I \\
A_{j+1} & I
\end{pmatrix}
\begin{pmatrix}
x^{k+1} \\
\dot{x}_{k+1}
\end{pmatrix}
=
\begin{pmatrix}
x^k \\
\dot{x}_k
\end{pmatrix}
+ \Delta t
\begin{pmatrix}
0 \\
g_{j+1}^k
\end{pmatrix}
\]

Theorem

The solution of the system is given by

\[
\begin{pmatrix}
x^{k+1} \\
\dot{x}_{k+1}
\end{pmatrix}
=
\begin{pmatrix}
Y_{j+1}^{k+1} (x_j^{k+1} + \Delta t \dot{x}_j^{k+1} + (\Delta t)^2 g_j^{k+1}) \\
Y_j^{k+1} (\dot{x}_j^{k+1} + \Delta t g_j^{k+1} + \frac{1}{\Delta t} x_j^{k+1}) - x_j^{k+1}
\end{pmatrix}
\]

where

\[
Y_{j+1}^{k+1} = D - \frac{\Delta t D e_j^{k+1} (f_j^{k+1})^T D}{1 + \Delta t (f_j^{k+1})^T D e_j^{k+1}}
\]
Numerical results
Conclusion & Further Research

- Quasi-static deformation is computed using CONTACT.
- Combined with a time integration scheme such as Newmark-beta or Radau5.
- Global deformation is solved using modal analysis.
- The total deformation is solved using a Quasi-Newton approach.

Further research: taking friction into account as the result of rolling and sliding of wheels.
Conclusion & Further Research

- Quasi-static deformation is computed using CONTACT.
- Combined with a time integration scheme such as Newmark-beta or Radau5.
- Global deformation is solved using modal analysis.
- The total deformation is solved using a Quasi-Newton approach.

- Further research: taking friction into account as the result of rolling and sliding of wheels.
Any questions?