Contact and HiL Interaction in Multibody Based Machinery Simulators

Alberto Luaces Fernández

A thesis submitted for the degree of Doctor Ingeniero Industrial

April 18th, 2013, Ferrol, Spain

1. Motivation
2. Multibody Dynamics Formulation
3. Contact Force Models
4. Contact Detection
5. HiL Simulation
6. Example Implementations
7. Conclusions and Future Work

1. Motivation
2. Multibody Dynamics Formulation
3. Contact Force Models
4. Contact Detection
5. HiL Simulation
6. Example Implementations
7. Conclusions and Future Work

Computer simulation is a modern, yet widely used technique.

- System damages and human injuries are reduced at a minimum level.
- Running costs are much lower than the required by system tests.
- Unpredictable or unusual working conditions can be displayed.
- Logging, virtual sensor placement.
- Simulator components are usually non-specific and can be shared.

Multibody techniques automate the process of simulating machines and mechanisms:

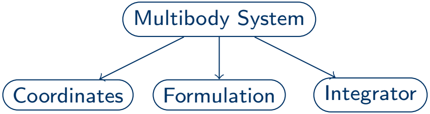
- Motion is computed in a generic fashion, without having to resort to ad-hoc physical models.
- Any parameter in the model can be checked and can be interpreted at run time.

However:

- Not all multibody techniques are suitable for real-time purposes.
- Heterogeneous phenomena (contacts, external hardware) must be incorporated.

Having a common framework is a valuable resource for rapid development of multibody simulators. The described in this work framework includes:

- Real-time capable multibody formulation.
- Contact modelling.
- Contact detection.
- Interaction with external devices and/or human users.



Desired characteristics for a multibody simulator designing system are:

- Robustness: able to cope with a wide scenario ranges.
- Efficiency: capacity of running in real-time.
- Easy mechanism definition: allows rapid development.

- In 3D systems, natural coordinates are sets of points and unit vectors.
- Body position and orientation are expressed easily.
- Constraint equations and their derivatives are simple expressions.
- Mass matrices and gravity forces are constant if 4 entities (or more) are used.
- The points and vectors can be shared between bodies.
- Increased number of coordinates, these coordinates are always dependent.
- Lead to sparse systems.
- Mechanism definition requires human hindsight for best results.

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}})$$
$$\mathbf{\Phi}(\mathbf{q},t) = \mathbf{0}$$
DAE

→

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^* + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi} = \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}})$$
$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* + \alpha\mathbf{\Phi}_{i+1},\ i = 0, 1, 2, \ldots$$
ODE

- Penalty at position level.
- Derivatives of the constraints $(\dot{\mathbf{\Phi}}, \ddot{\mathbf{\Phi}})$ are not enforced.
- Velocities and accelerations must be projected at each time step.

$$\min V = \frac{1}{2}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*\right)^{\mathrm{T}} \mathbf{P} \left(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*\right)$$
$$\text{subject to } c\dot{\mathbf{\Phi}}\left(\mathbf{q},\dot{\mathbf{q}},t\right) = 0$$

$$\min V = \frac{1}{2}\left(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*\right)^{\mathrm{T}} \mathbf{P} \left(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*\right)$$
$$\text{subject to } c\,\ddot{\mathbf{\Phi}}\left(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},t\right) = 0$$

- Stable for constrained systems.
- Energy dissipation is controlled by means of the $\rho_\infty = (0, 1)$ parameter.
- At each iteration, inertia terms and applied forces are interpolated between the states n and $n+1$.
- The system of equations of the integrator is solved by the method of *Newton-Raphson* (1).
- As a result, terms coming from the applied forces must be differentiated: $\mathbf{K}, \mathbf{C}$

$$\mathbf{M}\ddot{\mathbf{q}}_{\delta m} + \left[\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}\right]_{\delta f} = \mathbf{Q}_{\delta f}$$

$$\ddot{\mathbf{q}}_{\delta m} = (1-\delta_m)\,\ddot{\mathbf{q}}_{n+1} + \delta_m\ddot{\mathbf{q}}_n$$

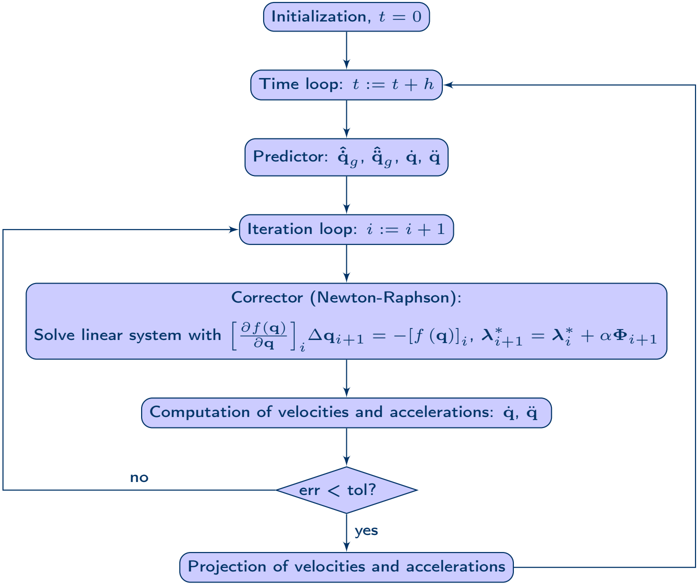
$$\mathbf{Q}_{\delta f} = \left(1-\delta_f\right)\mathbf{Q}_{n+1} + \delta_f\mathbf{Q}_n$$

$$[\ldots]_{\delta f} = \left(1-\delta_f\right)[\ldots]_{n+1} + \delta_f[\ldots]_n$$

$$\left[\frac{\partial f(\mathbf{q})}{\partial \mathbf{q}}\right] \cong (1-\delta_m)\,\mathbf{M} + \left(1-\delta_f\right)\gamma h\mathbf{C}_{n+1} + \left(1-\delta_f\right)\beta h^2\left(\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi}_{\mathbf{q}} + \mathbf{K}\right)_{n+1} \quad (1)$$

By default, the Augmented Lagrangian penalty factor α is set globally for all the system. However, it has to be adjusted for each body if heterogeneous mass distributions are handled.

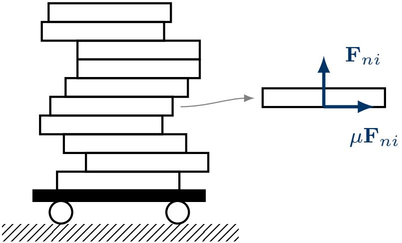
Global penalty factor $\alpha = 5 \times 10^{11}$
Hook penalty factor $\alpha_h = 5 \times 10^{7}$


Initialization, $t = 0$
Time loop: $t := t + h$
Predictor: $\hat{\dot{\mathbf{q}}}_g$, $\hat{\ddot{\mathbf{q}}}_g$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$
Iteration loop: $i := i + 1$
Corrector (Newton-Raphson):
Solve linear system with $\left[\frac{\partial f(\mathbf{q})}{\partial \mathbf{q}}\right]_i \Delta\mathbf{q}_{i+1} = -\left[f(\mathbf{q})\right]_i$, $\boldsymbol{\lambda}^*_{i+1} = \boldsymbol{\lambda}^*_i + \alpha\boldsymbol{\Phi}_{i+1}$
Computation of velocities and accelerations: $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$
err < tol?
no
yes
Projection of velocities and accelerations

- Most common applied forces to a multibody system come from contact phenomena.
- Contact force models are complex and have a high computational cost.
- A real-time solution with admissible precision must be found.

There exists two ways of considering contacts:

- As a instantaneous event in time (unilateral constraints).
- As a continuous event in time (interpenetration forces).



- Hertz-type model + damping model → Hunt-Crossley model.
- Robust enough for overcoming contacts in a reduced number of integration steps.
- Assumes a local deformation only in the contact region.

$$\mathbf{F}_n = k_n \, \delta^e \left(1 + \frac{3\,(1-\epsilon)}{2} \frac{\dot{\delta}}{\dot{\delta}_0} \right) \mathbf{n}$$

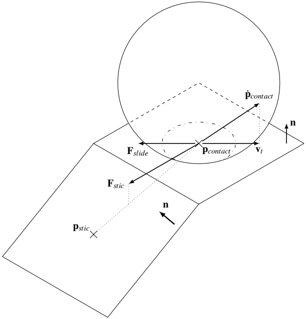
$$k_n = \frac{4}{3\left(\sigma_{sph} + \sigma_{pln}\right)} \sqrt{R_{sph}}$$

$$\sigma_{sph} = \frac{1-\nu_{sph}^2}{E_{sph}}; \quad \sigma_{pln} = \frac{1-\nu_{pln}^2}{E_{pln}}$$

- Stiction force at low velocities.
- Sliding friction force at high velocities.
- Viscous friction force.

$$\mathbf{F}_t = \kappa \ \ \mathbf{F}_{stic} \ + (1-\kappa) \ \ \mathbf{F}_{slide} \ - \ \mu_{visc}\mathbf{v}_t$$

$$\kappa = \begin{cases} 0; & \|\mathbf{v}_t\| >> v_{stic} \\ 1; & \|\mathbf{v}_t\| = 0 \end{cases}$$



- Stiction force at low velocities.
- Sliding friction force at high velocities.
- Viscous friction force.

$$\mathbf{F}_t = \kappa \ \mathbf{F}_{stic} \ + (1-\kappa) \ \mathbf{F}_{slide} \ - \ \mu_{visc}\mathbf{v}_t$$

$$\mathbf{F}_{stic} = \left\{ \begin{array}{cc} 0; & s = 0 \\ \dfrac{f^m_{stic}}{s}\left(\mathbf{I}_3 - \mathbf{n}\mathbf{n}^{\mathrm{T}}\right)(\mathbf{p}_{contact} - \mathbf{p}_{stic}); & s > 0 \end{array} \right\}$$

$$f^m_{stic} = -k_{stic}\, s - c_{stic}\, \dot{s}$$

- Stiction force at low velocities.
- Sliding friction force at high velocities.
- Viscous friction force.

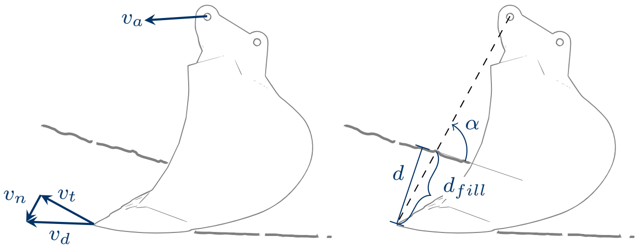
$$\mathbf{F}_t = \kappa \ \mathbf{F}_{stic} \ + (1-\kappa) \ \mathbf{F}_{slide} \ - \ \mu_{visc}\mathbf{v}_t$$

$$\mathbf{F}_{slide} = \left\{ \begin{array}{cc} 0; & \|\mathbf{v}_t\| = 0 \\ -\mu_{din} \|\mathbf{F}_n\| \dfrac{\mathbf{v}_t}{\|\mathbf{v}_t\|}; & \|\mathbf{v}_t\| > 0 \end{array} \right\}$$

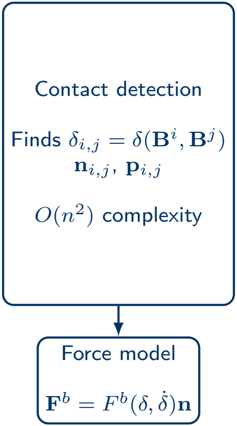
- A terrain-digging model featuring excavation drag based on excavation velocity.
- Neglects brittle failure from the soil.

$$\mathbf{F}_{dig} = -\mu_{dig}(\sigma_1 + \sigma_2 V_b^m) d_b^n \mathbf{v}_t - \mu_{cp} d_b^n \mathbf{v}_n$$

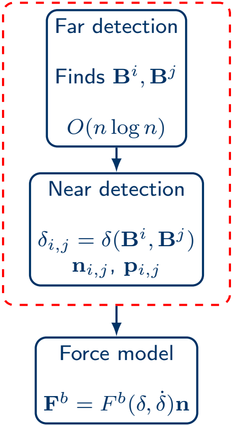
$$\mathbf{P}_{soil} = \rho V_b \mathbf{g}$$



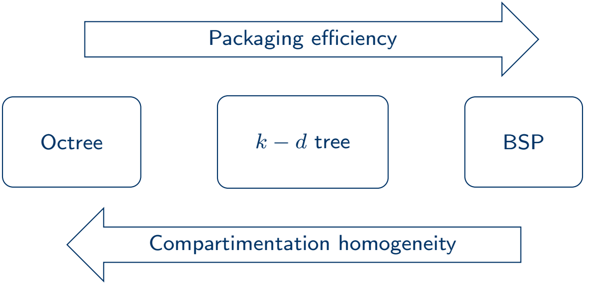
Reaction forces are computed by means of contact force models

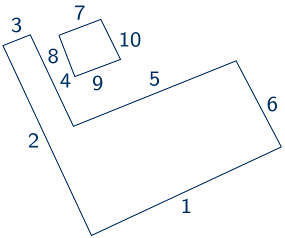


Reaction forces are computed by means of contact force models

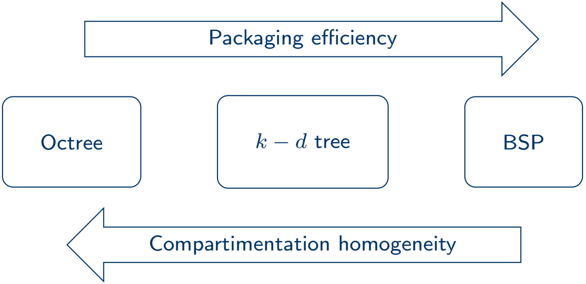


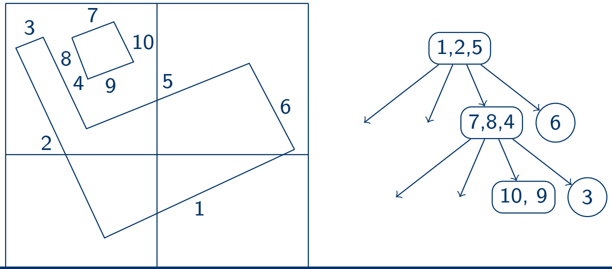
Space partitioning techniques for static geometry:





Space partitioning techniques for static geometry:



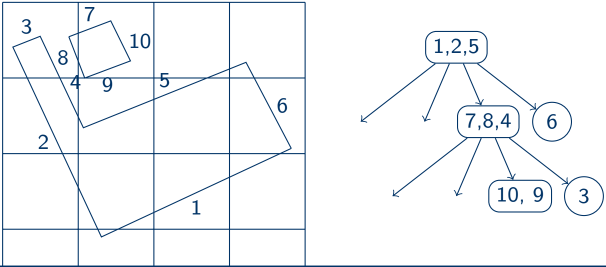


Space partitioning techniques for static geometry:

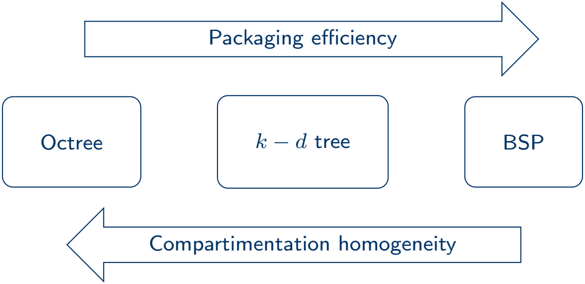
Packaging efficiency →

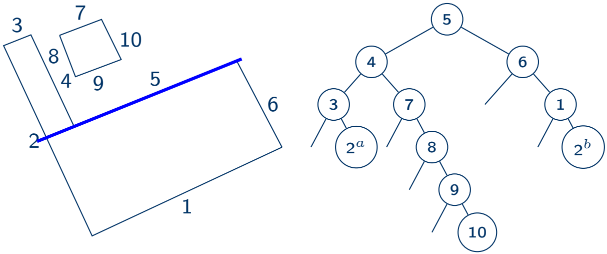
Octree | $k-d$ tree | BSP

← Compartimentation homogeneity

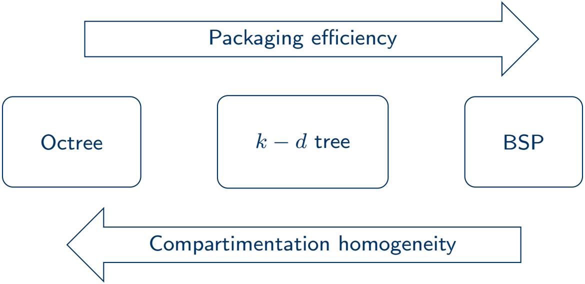


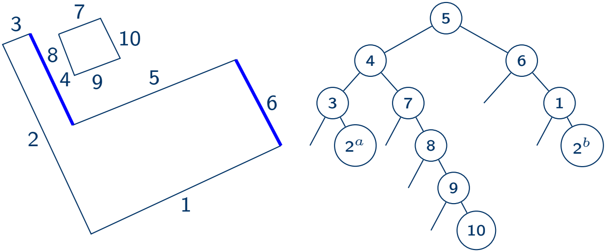
Space partitioning techniques for static geometry:





Space partitioning techniques for static geometry:



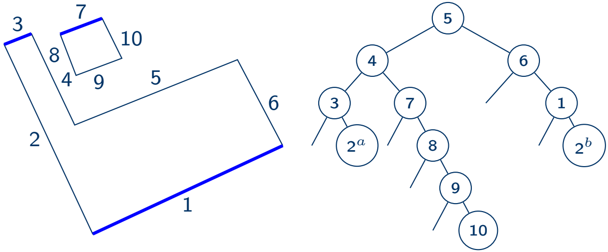


Space partitioning techniques for static geometry:

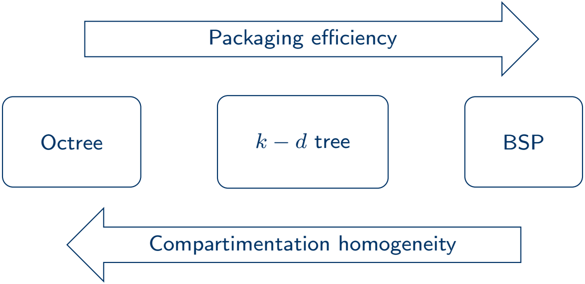
Packaging efficiency →

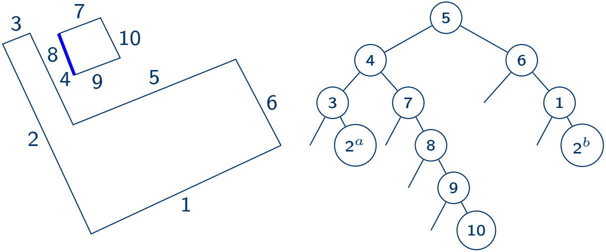
Octree	$k-d$ tree	BSP

← Compartimentation homogeneity

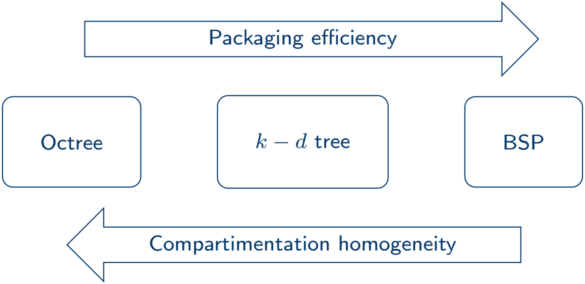


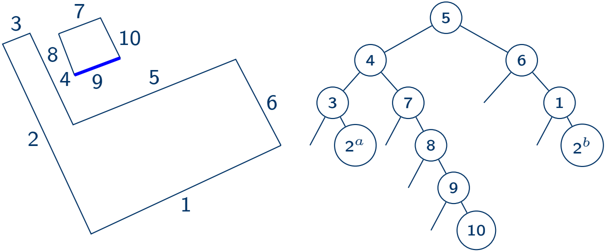
Space partitioning techniques for static geometry:





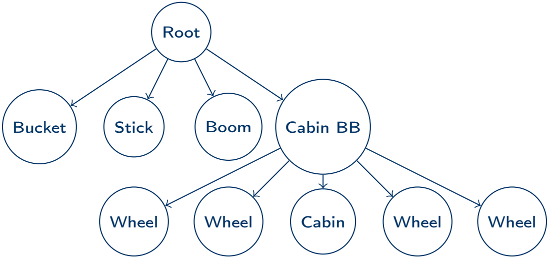
Space partitioning techniques for static geometry:

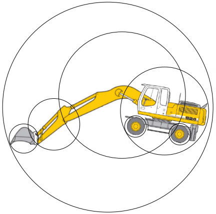


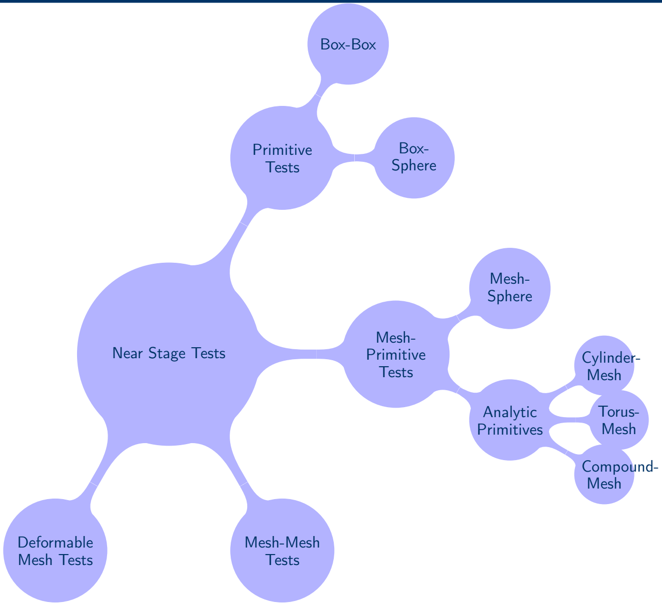


4 – Collision tests for moving objects: Direct Acyclic Graphs

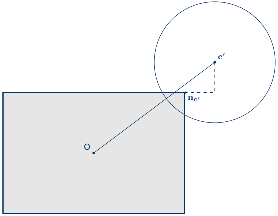
- Overlapping bounding volumes.
- Non-automatic partitioning.



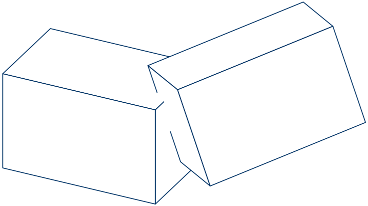



Box-Box
Primitive Tests
Box-Sphere
Mesh-Sphere
Near Stage Tests
Mesh-Primitive Tests
Cylinder-Mesh
Analytic Primitives
Torus-Mesh
Compound-Mesh
Deformable Mesh Tests
Mesh-Mesh Tests

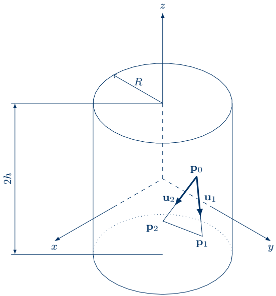
- Simple and fast tests.
- They constitute the basis for more elaborate geometry tests.

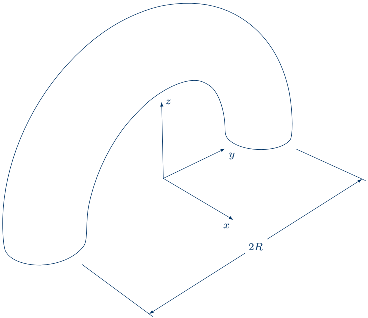


Mesh-Sphere



Box-box





min. $r_p^2 = p_x^2 + p_y^2$

st.

$$\phi_1 = 1 - \frac{\mu_1}{l_1} - \frac{\mu_2}{l_2} \geq 0$$

$$\phi_2 = h - p_z \geq 0$$

$$\phi_3 = h + p_z \geq 0$$

$$\phi_4 = \mu_1 \geq 0$$

min. $r_p^2 = p_x^2 + \left(\sqrt{p_y^2 + p_z^2} - R\right)^2$

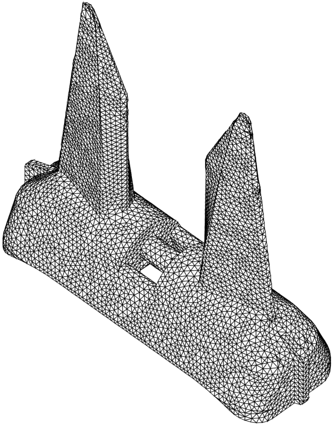
st.

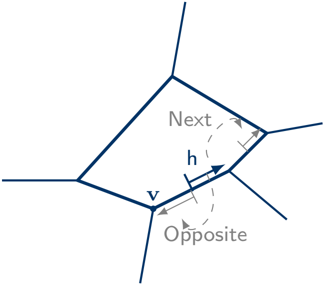
$$\phi_1 = 1 - \frac{\mu_1}{l_1} - \frac{\mu_2}{l_2} \geq 0$$

$$\phi_2 = p_z \geq 0$$

$$\phi_4 = \mu_1 \geq 0$$

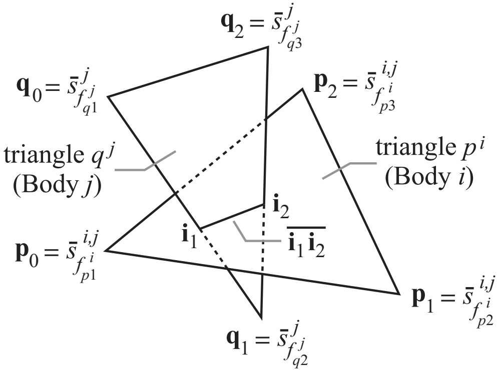
- OBB trees and topology information must be available.



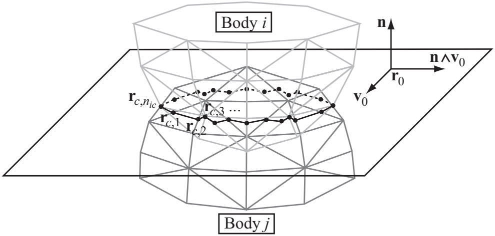


The *half edge* structure.

- OBB trees and topology information must be available.
- A triangle-triangle intersection is performed to find pairs of colliding faces.



- OBB trees and topology information must be available.
- A triangle-triangle intersection is performed to find pairs of colliding faces.
- The pairs are ordered so their intersection segments form closed contours.



- OBB trees and topology information must be available.
- A triangle-triangle intersection is performed to find pairs of colliding faces.
- The pairs are ordered so their intersection segments form closed contours.
- For each contour, a contact plane is computed by a least squares fitting.

$$\begin{bmatrix} \mathbf{r}_{c,1}^{\mathrm{T}} & 1 \\ \mathbf{r}_{c,2}^{\mathrm{T}} & 1 \\ \dots & \dots \\ \mathbf{r}_{c,n_{ic}}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \overline{\mathbf{n}} \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \Rightarrow \mathbf{A}\mathbf{x} = \mathbf{0}$$

$$\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\mathbf{x} = \mathbf{0}$$

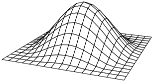
- OBB trees and topology information must be available.
- A triangle-triangle intersection is performed to find pairs of colliding faces.
- The pairs are ordered so their intersection segments form closed contours.
- For each contour, a contact plane is computed by a least squares fitting.
- The centroid of the projected contour defines the application point.

$$\bar{\mathbf{r}}_c^{\triangle} = \frac{1}{6A} \sum_{i=1}^{N} \begin{bmatrix} (x_i + x_{i\oplus 1})(x_i y_{i\oplus 1} - x_{i\oplus 1} y_i) \\ (y_i + y_{i\oplus 1})(x_i y_{i\oplus 1} - x_{i\oplus 1} y_i) \\ 0 \end{bmatrix}$$

$$A = \frac{1}{2} \sum_{i=1}^{N} (x_i y_{i\oplus 1} - x_{i\oplus 1} y_i)$$

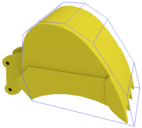
- OBB trees and topology information must be available.
- A triangle-triangle intersection is performed to find pairs of colliding faces.
- The pairs are ordered so their intersection segments form closed contours.
- For each contour, a contact plane is computed by a least squares fitting.
- The centroid of the projected contour defines the application point.
- The maximum indentation δ is the maximum distance from any interior vertex to the plane.

$$\begin{aligned} \delta_{\mathbf{v}_i} &= \mathbf{n}^{\mathrm{T}}(\bar{\mathbf{s}}_i^{\triangle} - \bar{\mathbf{r}}_c^{\triangle}) \\ \delta &= \max(\delta_{\mathbf{v}_i}),\ \forall \mathbf{v}_i \end{aligned}$$



Regular mesh grid

+



$$\forall\ \mathbf{r}_i, \mathbf{r}_j \in \mathcal{P} \Rightarrow \mathbf{r}_\lambda = \mathbf{r}_i + \boldsymbol{\lambda}(\mathbf{r}_j - \mathbf{r}_i) \in \mathcal{P}$$
$$0 \leq \boldsymbol{\lambda} \leq 1$$

Convex hull

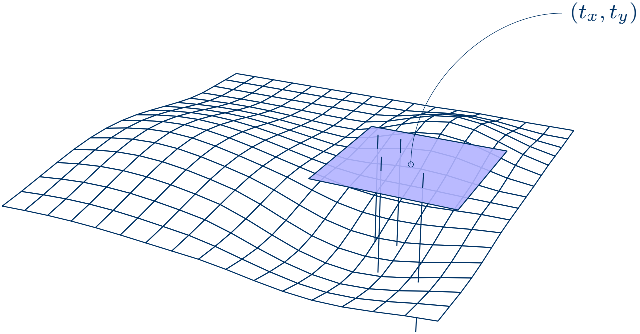
+

$$\forall f_i \in \mathcal{P} \Rightarrow \mathbf{n}_{f_i}^T \mathbf{p} + d \leq 0$$

Inclusion test

Bucket Discharge → Projectile motion → Find or create terrain patch. → Discharge zone →

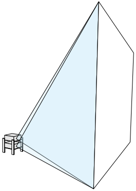
Gaussian distribution
(σ, μ)




Industrial controllers
Game controllers
USB, RS-232
Data acquisition device
PC
VGA, USB
Touchscreen

Hemispheric screen

CAVE

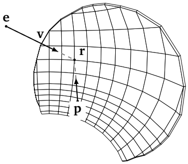


Dual projectors

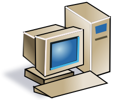
Reflective surfaces

Stereo rendering

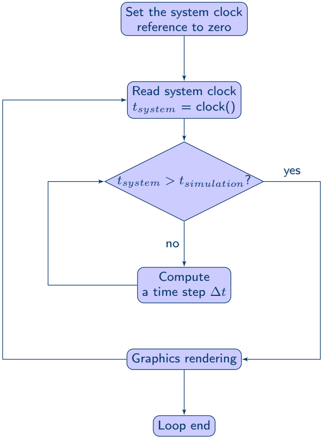
Composition buffers

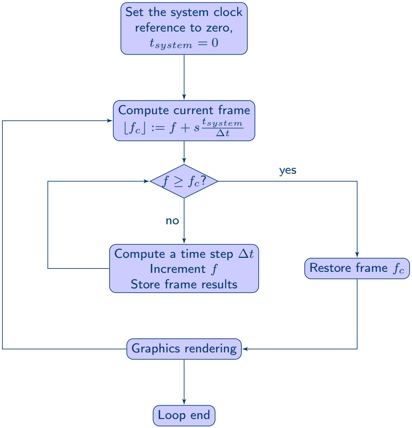


Distortion correction



Graphic display

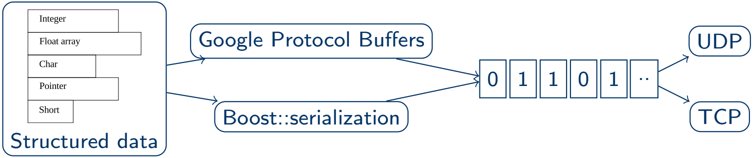

Set the system clock reference to zero
Read system clock
t_{system} = clock()
$t_{system} > t_{simulation}$?
yes
no
Compute a time step Δt
Graphics rendering
Loop end


Set the system clock reference to zero, $t_{system} = 0$
Compute current frame
$\lfloor f_c \rfloor := f + s\frac{t_{system}}{\Delta t}$
$f \geq f_c$?
yes
no
Compute a time step Δt
Increment f
Store frame results
Restore frame f_c
Graphics rendering
Loop end

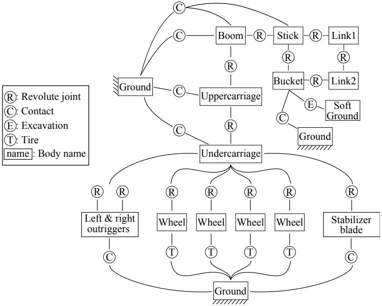
5 – External & internal events

- Simulation events

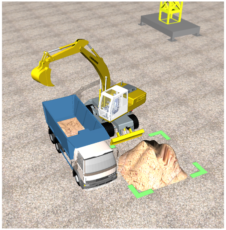
- Network synchronization



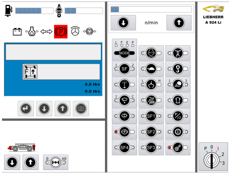
1. Excavator simulator
2. Anchor weighing simulator



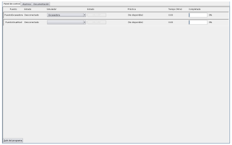
Motion	No.
Actuated degrees of freedom	
Boom, stick and bucket hydraulic cylinders	3
Uppercarriage rotation	1
Steering	1
Stabilizer blade	1
Outriggers	1
TOTAL	7
Non-actuated degrees of freedom	
Undercarriage free motion	6
Wheel rotation	4
TOTAL	10
TOTAL	17

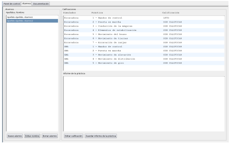


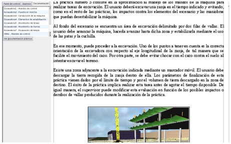
- Tactile touch screen panel
- Lever controllers
- Pedals

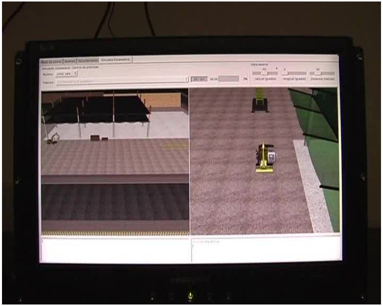


- Control panel
- Student tracking module
- Session documentation reader
- Run-time monitoring tracker

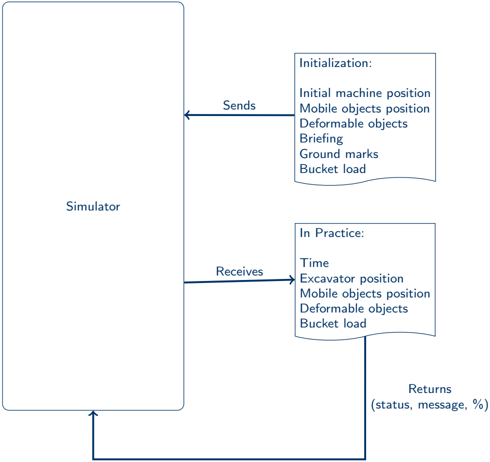




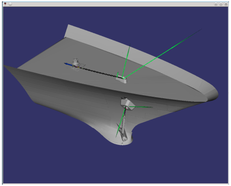


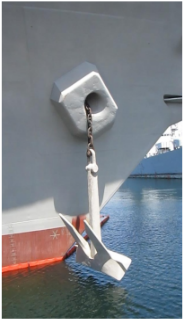


- Session initialization and definition
- Logic processing.
- Return value.



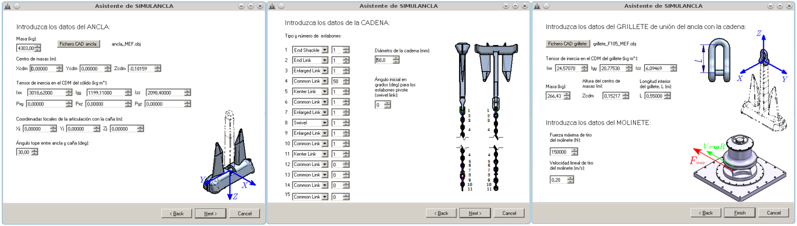
- Problem: designing the geometric design of a ship's hull that guarantees a correct anchor maneuver.
- Large amount of contacts take place between the bodies of the system: not real-time.

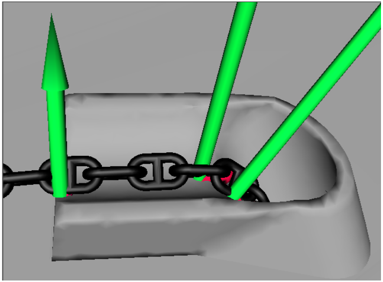




6 – Anchor weighing simulator: 3D geometry

- The user must select different files for anchor, hull and chain geometry definitions.
- Each part is defined by complex, arbitrary surfaces.
- The simulator must resort to compute full mesh-mesh collision detection.
- Considering regular chain links as analytic surfaces improves performance and accuracy.





Conclusions:

- Convenient framework for developing multibody-based machinery simulators.
- Flexible and efficient multibody formulation.
- Contact force models for rigid and deformable bodies.
- Contact detection methods.
- Physical implementation with commercial off the shelf hardware.

Future work:

- Simulation parallelization.
- Granular media simulation.