Discrete and hybrid models: Applications to concrete damage

Václav Šmilauer

2 July 2007
About me

- PhD student, financed by French ministry of research this year.
- Enrolled both in Prague (Milan Jirásek) and Grenoble (Laurent Daudeville) — “doctorat en co-tutelle”.
- Work focusing on Yade:
  - open-source platform for numerical calculations, since 2003;
  - continuing development, funded mostly by laboratory 3S-R in Grenoble.
About me

- PhD student, financed by French ministry of research this year.
- Enrolled both in Prague (Milan Jirásek) and Grenoble (Laurent Daudeville) — “doctorat en co-tutelle”.
- Work focusing on Yade:
  - open-source platform for numerical calculations, since 2003;
  - continuing development, funded mostly by laboratory 3S-R in Grenoble.

http://yade.wikia.com
1 Introduction

2 Discrete element method
   - DEM intricacies
   - DEM and concrete

3 Lattice models

4 Hybrid models
1 Introduction

2 Discrete element method
   - DEM intricacies
   - DEM and concrete

3 Lattice models

4 Hybrid models
1 Introduction

2 Discrete element method
   - DEM intricacies
   - DEM and concrete

3 Lattice models

4 Hybrid models
1. Introduction

2. Discrete element method
   - DEM intricacies
   - DEM and concrete

3. Lattice models

4. Hybrid models
(Primarily) continuous models

- Problem formulated in terms of differential equations – continuum mechanics.
- Displacement function $u$, found by numerical solution of boundary value problem.
- Discontinuities in $u$ are an extension of the method.
- Strain undefined at discontinuity, "awkward" (sophisticated) methods.
- Problem formulated in terms of differential equations – continuum mechanics.
- Displacement function $u$, found by numerical solution of boundary value problem.
- Discontinuities in $u$ are an extension of the method.
- Strain undefined at discontinuity, “awkward” (sophisticated) methods.
(Primarily) discrete models

- Local equations determine global behavior numerically — element interactions.
- No integration necessary, more computationally intensive.
- Discontinuity description trivial.
- Continuity (cohesion) by linking elements.
- Discrete element method (DEM).
- Lattice models.
(Primarily) discrete models

- Local equations determine global behavior numerically — element interactions.
- No integration necessary, more computationally intensive.
- Discontinuity description trivial.
- Continuity (cohesion) by linking elements.
- Discrete element method (DEM).
- Lattice models.
(Primarily) discrete models

- Local equations determine global behavior numerically — element interactions.
- No integration necessary, more computationally intensive.
- Discontinuity description trivial.
- Continuity (cohesion) by linking elements.
- Discrete element method (DEM).
- Lattice models.
DEM background

- elements are rigid bodies, motion governed by Newton’s laws
- explicit integration in time
- “smooth” (pinball) vs. “non-smooth” (overlaps) DEM
- mechanics of granular media — Cundall, 1971 (“distinct element method” in 2D, spherical elements)
- molecular dynamics of gas (1980s)
DEM background

- elements are rigid bodies, motion governed by Newton’s laws
- explicit integration in time
- “smooth” (pinball) vs. “non-smooth” (overlaps) DEM
- mechanics of granular media — Cundall, 1971 (“distinct element method” in 2D, spherical elements)
- molecular dynamics of gas (1980s)
DEM background

- elements are rigid bodies, motion governed by Newton’s laws
- explicit integration in time
- “smooth” (pinball) vs. “non-smooth” (overlaps) DEM
- mechanics of granular media — Cundall, 1971 (“distinct element method” in 2D, spherical elements)
- molecular dynamics of gas (1980s)
DEM simple example

How to calculate spheres falling through a funnel?

- Known element constants \((m, I, \ldots)\)
- and variables at \(t = t_i\) \((x, o, v, \omega, \text{state parameters})\).
- Solve for variables at \(t = t_{i+1} = t_i + \Delta t\).
DEM simple example

How to calculate spheres falling through a funnel?

- Known element constants \((m, I, \ldots)\)
- and variables at \(t = t_i\) \((x, o, v, \omega,\text{ state parameters})\).
- Solve for variables at \(t = t_{i+1} = t_i + \Delta t\).
How to calculate spheres falling through a funnel?

- Known element constants ($m, I, \ldots$)
- and variables at $t = t_i$ ($x, o, v, \omega$, state parameters).
- Solve for variables at $t = t_{i+1} = t_i + \Delta t$. 
DEM simple example

How to calculate spheres falling through a funnel?

- Known element constants \((m, I, \ldots)\)
- and variables at \(t = t_i\) \((x, o, v, \omega, \text{state parameters})\).
- Solve for variables at \(t = t_{i+1} = t_i + \Delta t\).
DEM iteration

1. Calculate forces:
   - independent fields,
   - inter-element links,
   - element collisions.

2. Calculate acceleration from forces.

3. Integrate over $\Delta t$, $t = t_i + \Delta t$.

4. (Adjust $\Delta t$.)

```c
void SphericalDEMSimulator::doOneIteration()
{
  // compute dt
  if (useTimeStepper)
    [dt=computeDt(spheres,contacts);
     // detect potential collision
     sap.action(spheres,contacts);
     // detect real collision
     findRealCollision(spheres,contacts);
     // compute response
     computeResponse(spheres,contacts);
     // add damping
     addDamping(spheres);
     // apply response
     applyResponse(spheres);
     // time integration
     timeIntegration(spheres);
}
```
Discrete and hybrid models: Applications to concrete damage

Václav Šmilauer

Introduction

Discrete element method

DEM intricacies
DEM and concrete

Lattice models

Hybrid models

DEM iteration

```
void SphericalDEMSimulator::doOneIteration()
{
    // compute dt
    if (useTimeStepper)
        [dt=computeDt(spheres,contacts);]
    // detect potential collision
    sap.action(spheres,contacts);
    // detect real collision
    findRealCollision(spheres,contacts);
    // compute response
    computeResponse(spheres,contacts);
    // add damping
    addDamping(spheres);
    // apply response
    applyResponse(spheres);
    // time integration
    timeIntegration(spheres);
}
```

1. Calculate forces:
   - independent fields,
   - inter-element links,
   - element collisions.

2. Calculate acceleration from forces.

3. Integrate over $\Delta t$, $t = t_i + \Delta t$.

4. (Adjust $\Delta t$.)
DEM iteration

1. Calculate forces:
   - independent fields,
   - inter-element links,
   - element collisions.

2. Calculate acceleration from forces.

3. Integrate over $\Delta t$, $t = t_i + \Delta t$.

4. (Adjust $\Delta t$.)

```c
void SphericalDEMSimulator::doOneIteration()
{
    // compute dt
    if (useTimeStepper)
        dt = computeDt(spheres, contacts);
    // detect potential collision
    sap.action(spheres, contacts);
    // detect real collision
    findRealCollision(spheres, contacts);
    // compute response
    computeResponse(spheres, contacts);
    // add damping
    addDamping(spheres);
    // apply response
    applyResponse(spheres);
    // time integration
    timeIntegration(spheres);
}
```
DEM iteration

1. Calculate forces:
   - independent fields,
   - inter-element links,
   - element collisions.

2. Calculate acceleration from forces.

3. Integrate over $\Delta t$, $t = t_i + \Delta t$.

4. (Adjust $\Delta t$.)
Discrete and hybrid models: Applications to concrete damage

Václav Šmilauer

Introduction

Discrete element method

DEM intricacies

DEM and concrete

Lattice models

Hybrid models

DEM iteration

```c
void SphericalDEMSimulator::doOneIteration()
{
    // compute dt
    if (useTimeStepper)
        dt = computeDt(spheres, contacts);
    // detect potential collision
    sap.action(spheres, contacts);
    // detect real collision
    findRealCollision(spheres, contacts);
    // compute response
    computeResponse(spheres, contacts);
    // add damping
    addDamping(spheres);
    // apply response
    applyResponse(spheres);
    // time integration
    timeIntegration(spheres);
}
```

1. Calculate forces:
   - independent fields,
   - inter-element links,
   - element collisions.

2. Calculate acceleration from forces.

3. Integrate over $\Delta t$, $t = t_i + \Delta t$.

4. (Adjust $\Delta t$.)
DEM iteration

1. Calculate forces:
   - independent fields,
   - inter-element links,
   - element collisions.

2. Calculate acceleration from forces.

3. Integrate over $\Delta t$, $t = t_i + \Delta t$.

4. (Adjust $\Delta t$.)

```cpp
void SphericalDEMSimulator::doOneIteration()
{
    // compute dt
    if (useTimeStepper)
        dt=computeDt(spheres,contacts);
    // detect potential collision
    SAP::action(spheres,contacts);
    // detect real collision
    findRealCollision(spheres,contacts);
    // compute response
    computeResponse(spheres,contacts);
    // add damping
    addDamping(spheres);
    // apply response
    applyResponse(spheres);
    // time integration
    timeIntegration(spheres);
}
```
Collision detection

Big research issue in applied mathematics, trivial approach is $O(n^2)$. Better approach:

1. Replace each element by its AABB (Axis-Aligned Bounding Box).
2. Sort $x$, $y$, $z$ min-max arrays independently.
3. Overlaps on all coordinates are collision candidates.
Collision detection

Big research issue in applied mathematics, trivial approach is $O(n^2)$. Better approach:

1. Replace each element by its AABB (Axis-Aligned Bounding Box).
2. Sort $x$, $y$, $z$ min-max arrays independently.
3. Overlaps on all coordinates are collision candidates.
Collision detection

Big research issue in applied mathematics, trivial approach is $O(n^2)$. Better approach:

1. Replace each element by its AABB (Axis-Aligned Bounding Box).
2. Sort $x$, $y$, $z$ min-max arrays independently.
3. Overlaps on all coordinates are collision candidates.
Collision detection

Geometry:

- $P$, $n$, $t$, $d$ for spheres (trivial).
- Complicated for other shapes (e.g. tetrahedra: $C$, $V$, $I$).
- Combinations: sphere with tetrahedron, parallelepiped, …

Forces are yet to be found.
Collision detection

Geometry:
- P, n, t, d for spheres (trivial).
- Complicated for other shapes (e.g. tetrahedra: C, V, I).
- Combinations: sphere with tetrahedron, parallelepiped, ...
Collision detection

Geometry:

- $P, \eta, t, d$ for spheres (trivial).
- Complicated for other shapes (e.g. tetrahedra: $C, V, I$).
- Combinations: sphere with tetrahedron, parallelepiped, ...
Physical laws

The most simple model:

- $F_n = k_n d, \Delta F_s = k_s \Delta u_t$ (incremental).
- Fracture with Coulomb criterion $\max F_s = F_n \tan \phi$.

How to determine $k_n, k_s$ from macroscopic characteristics?

- Simplistically for sphere $F = Ku$, $F = \sigma S = E(1 - d/2r)\pi r^2$ $(d \ll r)$
- Really used formulas: coefficients without physical meaning.

→ Model calibration.
Physical laws

The most simple model:

- \( F_n = k_n d \), \( \Delta F_s = k_s \Delta u_t \) (incremental).
- Fracture with Coulomb criterion \( \max F_s = F_n \tan \phi \).

How to determine \( k_n \), \( k_s \) from macroscopic characteristics?

- Simplistically for sphere \( F = Ku \), \( F = \sigma S = E(1 - d/2r)\pi r^2 \) (\( d \ll r \))
- Really used formulas: coefficients without physical meaning.

→ Model calibration.
Mesh generation

- Requirements: isotropy, high coordination number and compactness, size distribution.
- Regular packing leads to anisotropic behavior.
- Dynamic methods: gravity, growing spheres. Slow.

Jean-François Durier: Geometric method based on tetrahedral mesh:
  - Leverages existing FEM meshers — arbitrary shapes.
  - Very fast with excellent parameters.
Mesh generation

- Requirements: isotropy, high coordination number and compacity, size distribution.
- Regular packing leads to anisotropic behavior.
- Dynamic methods: gravity, growing spheres. Slow.
- Jean-François Durier: Geometric method based on tetrahedral mesh:
  - Leverages existing FEM meshers — arbitrary shapes.
  - Very fast with excellent parameters.
DEM concrete fracture (Hentz)

Modélisation d’une Structure en Béton Armé Soumise à un Choc par la Méthode des Éléments Discrets (PhD thesis of Sebastian Hentz, 2003)

- Dropping reinforced concrete cube on reinforced concrete slab.
- Concrete elements not “physical” (matrix, inclusions).
- Reinforcement modelled by special elements (including plastification).
- Parameters calibrated on basic setups, not ex post — predictive value.
DEM concrete fracture, results

- 80+110 thousand elements (reinforcement + concrete)
- Comparison with instrumented experiment at different limit states.
  - Good results (e.g. displacements ±10%).
  - Long calculation
Lattice overview

- Replaces continuum by arrangement of 1D elements (trusses, rods, beams).
- Nodes may carry inertia mass (dynamic) or not.
- Irregular meshes, less sensitive to degenerate geometry.
- Voronoï tessellation / Delaunay triangulation.

Kozicki 2007
Lattice beam model (Cusatis & co.)

- Meso-scale lattice beam model (matrix, inclusions).
- Constitutive law with damage, fracture, plasticity.
- Elaborate beam properties based on geometry of the respective Voronoï cell.
- Good match in tensile as well as compressive (usual weak point of lattices) loading.
Method considerations

**FEM**

- Efficient and easy for undamaged continuum.
  - Difficult discontinuity description.

→ Undamaged zone.

**Lattice**

- No collision detection necessary.
  - No volumetric information.

→ Fragmenting zone interior.

**DEM**

- Compressive links created during simulation.
  - Collisions: computationally expensive.

→ Highly fragmented, collapsing zones.

→ Colliding boundaries or detached zones.
<table>
<thead>
<tr>
<th>Method considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FEM</strong></td>
</tr>
<tr>
<td>+ Efficient and easy for undamaged continuum.</td>
</tr>
<tr>
<td>- Difficult discontinuity description.</td>
</tr>
<tr>
<td>→ Undamaged zone.</td>
</tr>
<tr>
<td><strong>Lattice</strong></td>
</tr>
<tr>
<td>+ No collision detection necessary.</td>
</tr>
<tr>
<td>- No volumetric information.</td>
</tr>
<tr>
<td>→ Fragmenting zone interior.</td>
</tr>
<tr>
<td><strong>DEM</strong></td>
</tr>
<tr>
<td>+ Compressive links created during simulation.</td>
</tr>
<tr>
<td>- Collisions: computationally expensive.</td>
</tr>
<tr>
<td>→ Highly fragmented, collapsing zones.</td>
</tr>
<tr>
<td>→ Colliding boundaries or detached zones.</td>
</tr>
</tbody>
</table>
## Method considerations

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FEM</strong></td>
<td>Efficient and easy for undamaged continuum.</td>
<td>Difficult discontinuity description.</td>
<td>Undamaged zone.</td>
</tr>
<tr>
<td><strong>Lattice</strong></td>
<td>No collision detection necessary.</td>
<td>No volumetric information.</td>
<td>Fragmenting zone interior.</td>
</tr>
<tr>
<td><strong>DEM</strong></td>
<td>Compressive links created during simulation.</td>
<td>Collisions: computationally expensive.</td>
<td>Highly fragmented, collapsing zones. Colliding boundaries or detached zones.</td>
</tr>
</tbody>
</table>
Method considerations

<table>
<thead>
<tr>
<th>Method</th>
<th>Pros</th>
<th>Cons</th>
<th>Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEM</td>
<td>Efficient and easy for undamaged continuum.</td>
<td>Difficult discontinuity description.</td>
<td>Undamaged zone.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lattice</td>
<td>No collision detection necessary.</td>
<td>No volumetric information.</td>
<td>Fragmenting zone interior.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Colliding boundaries or detached zones.</td>
</tr>
</tbody>
</table>
## Method considerations

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FEM</strong></td>
<td>Efficient and easy for undamaged continuum.</td>
<td>Difficult discontinuity description.</td>
<td>Undamaged zone.</td>
</tr>
<tr>
<td><strong>Lattice</strong></td>
<td>No collision detection necessary.</td>
<td>No volumetric information.</td>
<td>Fragmenting zone interior.</td>
</tr>
<tr>
<td><strong>DEM</strong></td>
<td>Compressive links created during simulation.</td>
<td>Collisions: computationally expensive.</td>
<td>Highly fragmented, collapsing zones.</td>
</tr>
</tbody>
</table>
Method considerations

**FEM**

+ Efficient and easy for undamaged continuum.
  - Difficult discontinuity description.
  → Undamaged zone.

**Lattice**

+ No collision detection necessary.
  - No volumetric information.
  → Fragmenting zone interior.

**DEM**

+ Compressive links created during simulation.
  - Collisions: computationally expensive.
  → Highly fragmented, collapsing zones.
  → Colliding boundaries or detached zones.
## Method considerations

<table>
<thead>
<tr>
<th>Method</th>
<th>Pros</th>
<th>Cons</th>
<th>Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FEM</strong></td>
<td>Efficient and easy for undamaged continuum.</td>
<td>Difficult discontinuity description.</td>
<td>Undamaged zone.</td>
</tr>
<tr>
<td><strong>Lattice</strong></td>
<td>No collision detection necessary.</td>
<td>No volumetric information.</td>
<td>Fragmenting zone interior.</td>
</tr>
<tr>
<td><strong>DEM</strong></td>
<td>Compressive links created during simulation.</td>
<td>Collisions: computationally expensive.</td>
<td>Highly fragmented, collapsing zones.</td>
</tr>
</tbody>
</table>
FEM inside DEM

- Particles in DEM are themselves FEM domains.
- Interest: reduces computational expenses wrt pure DEM for unfractures parts.
- Allows for dynamical states — in pure FEM, leads to statically under-determined states.
- More difficult collision detection (mesh — mesh).
- For non-predetermined fracture, FEM→DEM transition (via crack modeling and tracing) must be provided.
- Part of domain expected to break is DEM, the rest is FEM.
- Reduces computation wrt pure DEM.
- Must know fracturing (DEM) domain beforehand.
- Parameters must be tuned to have similar elastic behavior in both domains.
- The domain interface reflects waves (remedy: overlap zone — E. Frangin).
DEM with lattice

- Nodes are also DEM elements.
- Or: boundary nodes are DEM (collision), insert equivalent DEM element as needed.
- Preserves volume when fractured; pure lattice collapses.

Sun & al, 2003
References


