Revision of the CheckOverlaps() algorithm

Evgueni Tcherniaev
Overlap checking in Geant4

• The overlap checking in Geant4 is based on the MC method - randomly distributed points are generated on the surface of a volume and then it is checked that the points are not placed outside the mother volume or inside any of the sister volumes.

• In complex geometry descriptions like CMS and ATLAS many volumes have regular sub-structure with multiple elements, it means that:
  a) in average volumes contain many daughter volumes;
  b) the positions of sister volumes inside their mother volume differ only by translation;
  c) most of sister volumes have exactly the same shape;

• These factors have been taken into account during revision of the algorithm of the CheckOverlaps() method.

• Benchmark testing was done using two GDML files: cms2023D17.gdml (27.2 Mb) and atlas2016.gdml (172.9 Mb)

• Total time measured in the benchmark can be subdivided in three parts:
  • Time of operations not related to the overlap checking, it includes reading GDML file (marked as \textit{gdml}, will see on slide 5, Statistics and timing)
  • Time of the random point generation (marked as \textit{pnts})
  • Time of the algorithm itself (marked as \textit{alg})
Former algorithm

Loop by number of random points  
{  // default is 1000 points
  
  Generate a random point on the surface of the solid
  Create transformation defining position of the solid in the mother volume
  Transform the point to the mother coordinate system
  Check overlap with the mother volume

  Loop along sister volumes  
  {  
    Create transformation defining position of the sister in the mother volume
    Transform the point to the sister coordinate system
    Check overlap with the sister volume
  }
}

• The highlighted operation is the source of inefficiency - average number of checked daughter volumes in one mother volume is 220 (max 791) for CMS and 6370 (max 61440) for ATLAS
Restructured algorithm

Create std::vector of random points { // default is 1000 points

// Check that the solid is inside mother
Create transformation defining position of the solid in the mother volume
Loop along points {
    Transform current point to the mother coordinate system
    Check overlap with the mother volume
}

// Check overlaps with sisters
Loop along sister volumes {
    Create transformation defining position of current sister in the mother volume
    Loop along points {
        Transform current point to the sister coordinate system
        Check overlap with the sister volume
    }
}

• Such restructuring gave big improvement in the performance
Further improvement, bounding box check

Create `std::vector` of random points  
// default is 1000 points

// Check that the solid is inside mother
Create transformation defining position of the solid in the mother volume
Loop along points  
  Transform current point to the mother coordinate system
  Check overlap with the mother volume
  Calculate bounding box of the points

// Check overlaps with sisters
Loop along sister volumes  
  Create transformation defining position of current sister in the mother volume
  Get bounding box of the sister, recalculate it in the mother coordinate system
  Compare the bounding boxes, if they do not intersect go to next sister
  Loop along points  
    Transform current point to the sister coordinate system
    Check overlap with the sister volume

• This modification gave remarkable speed up in the algorithm
## Statistics and timing

### cms2023D17.gdml (27.2 Mb)

- **# of volumes:** 87.842 \( (81.754 \text{ have same shape as previous volume}) \)
- **# of checks with sister volumes:** 19.304.310 \( \text{(average: 219 per volume)} \)
- **# of bounding box intersections:** 182.414 \( (< 1\%) \)

<table>
<thead>
<tr>
<th></th>
<th>Former alg. (slide 2)</th>
<th>Restructured (slide 3)</th>
<th>Bounding Box (slide 4)</th>
<th>Potential improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gdml + pnts + alg = total</td>
<td>gdml + pnts + alg = total</td>
<td>gdml + pnts + alg = total</td>
<td>gdml + pnts + alg = total</td>
</tr>
<tr>
<td>Geant4 (Ranlux)</td>
<td>8 + 182 + 325 = 515</td>
<td>8 + 182 + 133 = 323</td>
<td>8 + 182 + 5 = 195 ( \text{(MixMax) 69} )</td>
<td>8 + 52 + 5 = 65</td>
</tr>
<tr>
<td>Geant4+VecGeom</td>
<td>8 + 59 + 324 = 391</td>
<td>8 + 59 + 139 = 206</td>
<td>8 + 59 + 9 = 76</td>
<td>8 + 14 + 9 = 31</td>
</tr>
</tbody>
</table>

### atlas2016.gdml (172.9 Mb)

- **# of volumes:** 341.172 \( (221.832 \text{ have same shape as previous volume}) \)
- **# of checks with sister volumes:** 2.160.653.339 \( \text{(average: 6333 per volume)} \)
- **# of bounding box intersections:** 698.292 \( (0.03\%) \)

<table>
<thead>
<tr>
<th></th>
<th>Former alg. (slide 2)</th>
<th>Restructured (slide 3)</th>
<th>Bounding Box (slide 4)</th>
<th>Potential improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gdml + pnts + alg = total</td>
<td>gdml + pnts + alg = total</td>
<td>gdml + pnts + alg = total</td>
<td>gdml + pnts + alg = total</td>
</tr>
<tr>
<td>Geant4 (Ranlux)</td>
<td>~ 2 days</td>
<td>360+3130+16900=20390</td>
<td>360 +3130 + 250 = 3740 ( \text{(MixMax) 1212} )</td>
<td>360 +2970 + 250 = 3580</td>
</tr>
<tr>
<td>Geant4+VecGeom</td>
<td>---</td>
<td>---</td>
<td>360 +1120 + 380 = 1860</td>
<td>360 +1090 + 380 = 1830</td>
</tr>
</tbody>
</table>
Yet another potential modification

• As we can see on previous slide, in a loop along volumes the probability that two consecutive volumes have exactly the same shape is very high. In such a case the random points generated for one volume can be also used for another volume. Such approach works very well in case of CMS, but not for ATLAS.

• In case of ATLAS main time is spent in the generation of random points for last 50,000 volumes (of 340,000 volumes in total). Those volumes belong to the Tile calorimeter, most of them differ from each other, have small size, and constructed as a Boolean subtraction of rather big solids. It causes very big number of calls to the random point generator – between 10 and 100 thousand unsuccessful tries preceding one successful attempt to generate a point on the surface.

• A solution could be a redefinition of those volumes as extruded solids.
Warning messages

Former algorithm could print three types of warning messages:

1) Overlap with mother volume!
   - Overlap is detected for volume YBSepar2_w0_10x7f69e05eea00_PV:9
   - with its mother volume MBWheel_00x7f69e068a3e0
   - at mother local point (-1657.16, -5132.52, 1268.1), overlapping by at least: 100 um

2) Overlap with volume already placed!
   - Overlap is detected for volume YB2_w0_b10x7f69e05eeb40_PV:3
   - with YBSepar2_w0_10x7f69e0673770:3 volume's
   - local point (-17.108, 222.5, -625.311), overlapping by at least: 120.219 nm

3) Overlap with volume already placed!
   - Overlap is detected for volume YB2_w0_b40x7f69e05eebe0_PV:10
   - apparently fully encapsulating volume YBSepar2_w0_20x7f69e0674350:10 at the same level!
**Improved warning messages**

- All warning messages have been extended with the type of the solid. For example, message 2) now looks like:

  Overlap with volume already placed!

  Overlap is detected for volume MBCables_Ext0x7f5fb3235ba0_PV:3 (G4Tubs) with Wall0x7f5fb3248ce0:1 (G4SubtractionSolid) volume's local point (7195.98,3608.38,3923.5), overlapping by at least: 6 cm

- Message 3) was misleading – in most of cases it appeared at ordinary overlaps. Now it is printed only in the case if one volume really encapsulate another volume

- Additional check and corresponding warning message has been introduced:

  Sample point is not on the surface!

  The issue is detected for volume Tile_Scintillator0x2fec2fd0_PV:0 (G4SubtractionSolid) generated point (215.373,563.885,-108.719) is outside

This message means that

- either the volume is empty, for example as a result of a Boolean operation; there are 120 such volumes in the ATLAS geometry description;

- or… there is a problem in sampling of random points; in case of VecGeom it was observed for Polycone and Polyhedra; the problem reported by CMS still persists!
Possible enhancements

• The size of an overlap can be better estimated by using multiple points.

• CheckOverlap() could return references to the overlapping volumes as well as information on the size and type (overlap with mother or with sister) of the detected overlaps.

• At present time CheckOverlaps() does the overlap checking with the mother volume (one level up) and the sister volumes (same level). Alternative approach could be to do the overlap checking for the daughter volumes (one level down). Such an approach would allow:
  - Reduce the number of checks volume-vs-volume almost by two time, in current approach implicitly exists double check, A-vs-B and then B-vs-A;
  - Make the sampling of random points in more optimal way, taking into account that most of the daughter volumes have exactly the same shape;
Summary

- Restructuring and applying the bounding box check has substantially improved the performance of the algorithm in CheckOverlaps()

- Migration from Ranlux to MixMax has also given big contribution to the speed up of the overlap checking

- At present time the execution time of the overlap check benchmark has decreased from 500 secs to 60 secs in case of cms2023D17.gdml and from >10 hours to 20 mins in case of atlas2016.gdml

- Still there is large room for further enhancement of the algorithm

- Sampling random points can also be improved by:
  - Applying faster random point generators
  - Optimizing the code in the solids