CUDA Geant4-based Monte Carlo Simulation for Radiation Therapy

N. Henderson & K. Murakami

GTC 2013
The collaboration

Makoto Asai, SLAC
Joseph Perl, SLAC
Koichi Murakami, KEK-SLAC
Takashi Sasaki, KEK
Margot Gerritsen, ICME
Nick Henderson, ICME

Special thanks to the CUDA Center of Excellence Program
20,000,000

radiotherapy treatments per year in US
Simulation methods

Analytic
• time: seconds to minutes
• accurate within 3-5%
• used in treatment planning

Monte Carlo
• time: several hours to days of CPU time
• accurate within 1-2%
• used to verify treatment plans in certain cases
Geant4 Toolkit

• Enables Monte Carlo simulation of particles travelling through and interacting with matter
• Allows modeling of complex geometries
• Covers all elementary particles and nuclei for a wide energy range
Geant4 Applications

High Energy Physics

ATLAS

Space & Radiation

LISA

Medical Physics

gMocren

Images from: Geant4 gallery and gMocren
Geant4 101

• Geant4 simulates particles travelling through and interacting with matter
• Example: photoelectric effect

Image: CC-SA by Wolfmankurd on Wikipedia
Parallelization challenges in Geant4

• Large and complex code base
• Sophisticated geometry framework
• Elaborate physics models
• Branching, look-up tables, single-thread optimizations
Thank goodness

The simulation is embarrassingly parallel!
(the particles are independent)
Requirements for X-ray Radiotherapy

• Geometry is a voxelized box
• Physics is limited to low-energy electromagnetics
• Material is modeled as water with different densities
What makes a physics process?

• Sample for angular and energy distributions of secondary particles
• Deposit energy to material
• Produce secondary particles that must be tracked at a later point
Low energy electromagnetics

Gamma
- Compton scattering
- Photoelectric effect
- Gamma conversion

Electron/positron
- Ionization
- Bremsstrahlung
- Todo: Multiple scattering
- Positron annihilation
Tracking algorithm

- Particles are tracked through space
- Each discrete move is called a step
- Physics process may occur along step, after the step, or both
How are processes selected?

• Each process has an “interaction length” (dx)
• Process with shortest dx is selected
• After the step dx is decreased or resampled
Energy deposition

• Happens along the step for the ionization process
• May happen at the end of the step
• Secondary particles with low energy are not generated, but treated as point-like energy depositions
G4CU: CUDA-base MC for RT

• Important data structures
• Algorithm summary
• Details
  – Parallel stack
  – Look-up tables
Struct-of-arrays data pattern

```c
struct ParticleArray {
    // length of arrays
    int length;
    // kind of particle
    ParticleKind *kind;
    // position
    float *x, *y, *z;
    // direction
    float *dx, *dy, *dz;
    // particle energy
    float *energy;
    // voxel index
    int *vx, *vy, *vz, *vid;
};
```

- Common pattern in CUDA to allow for coalesced memory access
- Experiments with transport showed this to be 3-4x faster than AOS
G4CU: algorithm

main loop

- query all processes to select step size
- apply all continuous processes
- decrease interaction lengths
- apply limiting discrete process
- check termination conditions
- reduce dose stacks to global array
- pop secondaries from stacks
- generate primary particles

management
Some details

• A process does a few things:
  1. Changes direction and momentum of primary particle
  2. Generates secondary particles
  3. Deposits energy to the material
• We use thread local stacks to handle 2 and 3
Parallel stacks

```
template<typename T>
struct pstack {
  // number of stacks
  int stack_num;
  // size of each stack
  int stack_size;
  // starts and ends of stacks
  int *start, *end;
  // stack positions
  int *pos;
  // stack data array
  T *data;
};
```
Dose reduction & storage

• Energy dose is stored in a thread local variable
• When particle moves to new voxel, dose is pushed onto stack
• Dose reduction is performed periodically
• We use **Thrust**
Look-up tables

- Used for interaction length computation
- 40 bins, log spaced
- Linear and spline interpolation
- Bremsstrahlung also uses 2D interpolation
Benchmarks

Configuration:
• Geometry: 512 x 512 x 256 voxels
• Dose reduction frequency: every 200 iterations
• 128 blocks with 256 threads
• Primary particle is a 6 MeV gamma
• Tesla C2070
Simulation time

- 100 million primary particles
- Time: 72 minutes
- ~23.1 primary particles per ms
- ~50-60x speedup over Geant4 on 1 CPU
## Profile

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage of overall time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physics processes</td>
<td>50</td>
</tr>
<tr>
<td>Energy dose reduction</td>
<td>30</td>
</tr>
<tr>
<td>Interaction length</td>
<td>18</td>
</tr>
<tr>
<td>Run management</td>
<td>2</td>
</tr>
</tbody>
</table>
# Physics process breakdown

<table>
<thead>
<tr>
<th>Process</th>
<th>Particle</th>
<th>Percentage of overall time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bremsstrahlung</td>
<td>e- / e+</td>
<td>23</td>
</tr>
<tr>
<td>Pair production</td>
<td>γ</td>
<td>7.5</td>
</tr>
<tr>
<td>Transport</td>
<td>all</td>
<td>7</td>
</tr>
<tr>
<td>Photoelectric effect</td>
<td>γ</td>
<td>7</td>
</tr>
<tr>
<td>Ionization</td>
<td>e- / e+</td>
<td>3</td>
</tr>
<tr>
<td>Compton scattering</td>
<td>γ</td>
<td>1.5</td>
</tr>
<tr>
<td>Positron annihilation</td>
<td>e+</td>
<td>1</td>
</tr>
</tbody>
</table>
Dose distribution
Acknowledgements

Geant4 Collaboration, see


NVIDIA & CCOE Program

People:

- Koichi Murakami, KEK
- Makoto Asai, SLAC
- Joseph Perl, SLAC
- Takashi Sasaki, KEK
- Margot Gerritsen, ICME