Published in final edited form as:

Phys Med Biol. 2012 June 7; 57(11): 3281–3293. doi:10.1088/0031-9155/57/11/3281.

Efficient voxel navigation for proton therapy dose calculation in TOPAS and Geant4

J. Schümann¹, H. Paganetti¹, J. Shin², B. Faddegon², and J. Perl³

¹Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114, USA

²University of California, San Francisco, CA 94143, USA

³SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

Abstract

A key task within all Monte Carlo particle transport codes is Navigation, the calculation to determine at each particle step what volume the particle may be leaving and what volume the particle may be entering. Navigation should be optimized to the specific geometry at hand. For patient dose calculation, this geometry generally involves voxelized computed tomography (CT) data. We investigated the efficiency of navigation algorithms on currently available voxel geometry parameterizations in the Monte Carlo simulation package Geant4: G4VPVParameterisation, G4VNestedParameterisation and G4PhantomParameterisation, the latter with and without boundary skipping, a method where neighboring voxels with the same Hounsfield Unit are combined into one larger voxel. A fourth parameterization approach (MGHPParameterization), developed in-house before the latter two parameterizations became available in Geant4, was also included in this study.

All simulations were performed using TOPAS, a TOOl for PArticle Simulations layered on top of Geant4. Runtime comparisons were performed on three distinct patient CT data sets: A head and neck, a liver, and a prostate patient. We included an additional version of these three patients where all voxels, including the air voxels outside of the patient, were uniformly set to water in the runtime study.

The G4VPVParameterisation offers two optimization options. One option has a 60-150 times slower simulation speed. The other is compatible in speed but requires 15-19 times more memory compared to the other parameterizations. We found the average CPU time used for the simulation relative to G4VNestedParameterisation to be 1.014 for G4PhantomParameterisation without boundary skipping and 1.015 for MGHPParameterization. The average run time ratio for G4PhantomParameterisation with and without boundary skipping for our heterogeneous data was = 0.97 : 1. The calculated dose distributions agreed with the reference distribution for all but the G4PhantomParameterisation with boundary skipping for the head and neck patient. Maximum memory usage ranged from 0.8 to 1.8 GB depending on the CT volume independent of parameterizations, except for the 15-19 times greater memory usage with the G4VPVparameterisation when using the option with higher simulation speed. The G4VNestedParameterisation was selected as the preferred choice for the patient geometries and treatment plans studied.

1. Introduction

Proton therapy has been successfully simulated with the Geant4 Monte Carlo (MC) simulation package (Agostinelli et al. 2003, Allison et al. 2006) at Massachusetts General Hospital (MGH) for over 10 years (see for example (Paganetti et al. 2008, Jarlskog and Paganetti 2008)). Beamlines, treatment nozzles and patient geometries have been modeled

accurately. In general, the coding needed is complex when using a MC simulation package to simulate a treatment nozzle and calculate dose distributions in patients or water phantoms, and systems such as Geant4 require specialized programming knowledge. Additionally, once the programming work has been achieved, a dedicated commissioning of the simulation must be performed. This makes it difficult to share MC code for proton therapy simulation and verification of treatment plans between institutions. In order to overcome these hurdles, a MC simulation system is needed that requires minimal or no coding and provides a straightforward method of simulating the beam delivery system, patient and detectors with results generated in a form that can easily be shared between institutions. This study is part of an effort to provide an easy-to-use MC simulation tool for proton therapy. In order to provide insight into the preferred implementation of patient geometries, the performance of available navigations for parameterizations of CT volumes was investigated.

1.1. TOPAS

TOPAS stands for TOol for PArticle Simulation. The goal of the TOPAS project is to make MC simulation more readily available for research and clinical physicists. In TOPAS, the entire therapy geometry, from nozzle entrance to patient is controlled by text-based parameter files, removing the requirement for physicists to be experts in a programming language to perform MC simulations. An entire nozzle can be constructed, patient information imported, and scoring volumes, surfaces and quantities defined through parameter text files. Reliability and repeatability is a special focus during the development. TOPAS is layered on top of the Geant4 Simulation Toolkit and designed to add key functionality for particle therapy. The TOPAS toolkit has been introduced in (Perl et al. 2011). There are other Geant4-based codes addressing some of the aims of TOPAS as well, examples are PTsim (Aso et al. 2010), GATE (Santin et al. 2003), GAMOS (Arce et al. 2008a), and others. Our conclusions should apply equally well to all such codes, as our work treats fundamental voxel navigation issues rather than details of the individual codes.

TOPAS was successfully applied to simulate the MGH proton therapy passive scattering and scanning gantry treatment heads, the MGH radiosurgery treatment head, and the treatment head for eye treatment at the University of California Davis (see separate paper in this issue).

1.2. Voxel Navigation in Geant4

A key task within all MC particle transport codes is “navigation”, the calculation to determine at each particle step what volume (beam modifier component, patient voxel, etc.) the particle may be leaving and what volume the particle may be entering. A patient anatomy is generally described by a CT image, which in geometrical terms is a large box (a rectangular parallelepiped) that is divided into many smaller boxes (voxels). Each voxel has a Hounsfield Unit (HU) assigned, which describes the linear attenuation coefficient of the voxel. The HUs from the CT image are converted into a material and density for the voxel. Typically, the size of a voxel is in the order of $1\text{ mm} \times 1\text{ mm} \times 2.5\text{ mm}$. For patient geometries, this results in millions of voxels that are necessary to construct a patient, thus efficient and fast navigation between these voxels is crucial to make MC calculations for patient geometries produce results in a reasonable time.

In Geant4 a CT volume can be constructed by creating an instance of a box with a given material and density for each voxel. Practically, constructing a separate volume for each voxel would require very large memory and, consequently, the simulation would, on a currently standard computer, spend a lot of time on memory swapping. One solution to this problem is the use of parameterized volumes. The navigation in a parameterization of the CT volume takes advantage of the well-defined geometrical arrangement of the voxels by

filling the volume with copies of a single voxel, thus reducing memory usage and improving computing speed. Geant4 offers three methods to parameterize patient geometries: the G4VPVParameterised, G4VNestedParameterisation and G4PhantomParameterisation classes. Additionally, a voxel parameterization was developed in-house at MGH before Geant4 provided the latter two parameterizations. Each of these parameterizations has a different approach to navigate between voxels. For this paper, the relative performance of these four different navigation approaches was analyzed in terms of speed, accuracy and memory usage.

2. Materials and Methods

All MC simulations were performed using TOPAS layered on top of Geant4.9.4p01. Three patients were used to test the performance of the voxel navigation, with one proton field for each patient. Patients undergoing treatment for head and neck, liver and prostate cancer were used as an indication of the effect of patient geometry and treatment plan when using the different navigation algorithms. Details of each patient CT and field parameters are displayed in Table 1. Patient data was read in from a binary file obtained from the CT scan containing a HU for each voxel. The HUs were then converted to a material and density. In our implementation based on (Schneider et al. 2000), 26 material compositions are used to describe the patient anatomy, and each material is given a varying density as a function of the HU, where HU=-1000 is air and HU=3061 is titanium. The material composition and implementation into Geant4 follows the setup previously implemented at MGH (Paganetti et al. 2008).

2.1. Parameterizations

In the most basic geometrical construction technique, Geant4 would require a patient CT to be represented by about fifty million voxels for our largest case ($512 \times 512 \times 172$ voxels). To improve memory usage, Geant4 provides “parameterized volumes”, wherein just a single volume is created in memory, but appears, at run time, to be in many different places and to have many different materials (allowing different CT materials for different voxels). Some parameterizations are quite general, allowing each placement of the solid to differ not only in location and material but in rotation, size and even shape of the solid. Other parameterizations are more restrictive, working only for sets of boxes that neatly pack the space.

Intimately connected to the parameterization choice is the navigation speed. If no special knowledge were used, a track leaving one voxel would have to interrogate all fifty million other voxels (plus whatever other geometry volumes are included in the simulation setup) to find out which geometry volume would be the next to be hit. Given the regularity of the voxel structure, more efficient navigation can be employed. Geant4 provides various navigation optimizations for various parameterizations.

We investigated four Geant4 parameterizations, where three of them are specifically appropriate to CT geometries. Their geometrical implementations and navigation is shown in Figure 1.

2.1.1. G4VPVParameterisation—This is the most general class for parameterizations in Geant4. It allows each placement of the solid to differ not only in location and material but in rotation, size and even shape of the solid. The user must provide their own “concrete class” based on G4VPVParameterisation (Geant4 Virtual class for Parameterized Volumes) to specify exactly what should vary from one placement to the next. For our CT use, we create a G4VPVParameterisation that varies only location and material (Figure 1a).

Because G4VPVParameterisation has no specific restrictions on volume placement (volumes need not be packed such that the voxels of the parameterization fill the entire space but can be scattered arbitrarily throughout space), the user must indicate an optimization scheme. Before we can clarify this further, we must explain Geant4's "smart voxel" optimization system.

At the beginning of the simulation, just before the first particle is tracked, Geant4 optimizes the simulation geometry. The entire simulation space is sliced along one axis, dividing the space into smart voxels that each contain only a limited set of geometry volumes. Smart voxel refers here not to the CT voxels, but to a set of divisions of the overall simulation space. If the 3D distribution of volumes is such that each smart voxel still has too many volumes, parts of the space may be sliced along a second and third axis. The required granularity level is automatically calculated by Geant4 depending on the geometry, but may be set up by the user. A map is retained associating each geometry volume to its encompassing smart voxel. Navigation can then move more quickly, since upon exiting any one volume, it is only necessary to interrogate the positions of the other volumes in the given smart voxel. When a particle exits one smart voxel, navigation knows which other smart voxel has been entered.

When using a G4VPVParameterisation, the user is required to give the smart voxel system an instruction about how to slice the parameterized space. The user can indicate that slicing happens in just a single direction (passing the flag "kXAxis", "kYAxis" or "kZAxis"), or that slicing should proceed with no limitations, using as many cartesian axes as the smart voxel system finds necessary (passing the flag "kUndefined"). For a CT geometry, kUndefined will always result in slicing in all three directions, since only with three dimensions of slicing can the number of CT voxels per smart voxel be brought to a small number. We will refer to G4VPVParameterisation with single direction slicing as G4VPV-1D and to G4VPVParameterisation with three direction slicing as G4VPV-3D.

G4VPV-1D does not have very fast navigation since the one directional slicing still leaves a large number of CT voxels in each smart voxel. Slicing in the z direction, Geant4 will form as many smart voxels as there are CT slices in the z direction, but each of these slices still has 512×512 x and y divisions, so that a particle leaving one CT voxel will have to interrogate $512 \times 512 = 262,144$ other CT voxels to find the next one hit. G4VPV-3D has faster navigation, since it breaks the CT into many smaller smart voxels. However the smart voxel map that stores the optimization information itself becomes problematic. It has to maintain so many pointers from smart voxels to CT voxels that it consumes large amounts of system memory.

Consequently, neither of the G4VPVParameterisation cases are recommended for use in a patient geometry. Only for completeness of our study, we simulated a smaller number of protons at nozzle entrance (10^5) for G4VPV-1D and -3D to study the timing and memory usage.

2.1.2. G4VNestedParameterisation—G4VNestedParameterisation (G4Nested) is one of two parameterizations recommended by Geant4 for use in tightly packed three dimensional layouts such as we find in a CT geometry. It is a special parameterization class derived from G4VPV. Instead of directly parameterizing in three-dimensions, G4Nested uses another Geant4 geometry device, the G4Replica, for two axes, and then uses a one-dimensional parameterization for the third axis. This is schematically shown in Figure 1b, where the green boxes indicate the parameterization along one axis, followed by a replication of this parameterization along another axis and another replication of the entire slice along the third axis.

Like the Geant4 parameterizations, the G4Replica is a way to represent many copies of a volume as one single volume. But whereas parameterizations allow copies to have arbitrary shape, placement or material, G4Replica copies must be of uniform shape, must be placed immediately adjacent to each other (completely filling their mother volume) and have no mechanism for assigning different materials to different copies. The navigator takes advantage of the Replicas guaranteed simple arrangement to quickly calculate paths from one copy to the next. By placing one layer of replicas (along one axis) within another layer of replicas (along the other axis), this navigator can efficiently solve paths in those directions. We cannot, however, use replicas for the third layer since there would then be no mechanism to assign different materials to different copies. G4Nested solves this by making just the third layer be a parameterization. For every voxel, the G4VNestedParameterisation's ComputeMaterial method is called. This method has access not just to the G4VPVParameterisation copy number (the copy number in z) but to the copy numbers of its parent (the y replica) and grandparent (the x replica). It can use all three indices to calculate the material. Smart voxel optimization is set along the axis of the inner parameterization, making navigation fast in that direction. Because that parameterization is just a one dimensional layout of voxels, the smart voxel map does not consume significant memory (as it would for a 3D voxel layout). The use of replicas makes navigation in the other two directions fast.

2.1.3. G4PhantomParameterisation—G4PhantomParameterisation (G4Phantom) was specifically developed for the use with CT volumes. It is a special implementation of G4VPVParameterisation that uses knowledge about CT dimensions and voxel numbers directly for its navigator to enable faster navigation, as opposed to the smart voxel approach. Like G4VPV, it represents the entire CT as a single parameterized volume. What makes G4Phantom special is that it has its own dedicated Geant4 navigator. Where the generalized navigator of G4VPV relies on the Geant4 smart voxel system to efficiently find the next voxel, G4Phantom's navigator uses knowledge of a voxel's position within the CT to predict the next voxel to be hit. Each voxel has a "copy number" that represents its volume. From the number of voxels of the CT volume in x, y and z, the dimensions of the voxels and the position of the CT volume, the position of the particle determines the copy number of the voxels with simple mathematical equations (Figure 1c).

An additional feature of G4Phantom is the option to combine neighboring voxels of same material (same HU) into one larger voxel. This option is called "boundary skipping". For homogeneous material, such as a voxelized water phantom, this results effectively in the entire volume being replaced with one big box. This simplification of the geometry is intended to improve the speed of the simulation (Arce et al. 2008). When several voxels are combined and a particle traverses them in one step, the energy deposited is only calculated once. Consequently, an algorithm to distribute energy deposit has been introduced that utilizes the step length (the length that a particle travels between two interactions). The dose deposited is distributed among the voxels along this path proportionally with an additional iterative correction considering the kinetic energy of the particle at each voxel entrance (Arce et al. 2008).

2.1.4. MGHPParameterization—MGHPParameterization (G4MGH) is a fourth method to navigate between voxels developed in-house at MGH. It is a direct implementation of the G4VPV parameterization with revised navigation. For each voxel the six neighboring voxel numbers are saved and only these six voxels are considered for determining the next voxel that a particle will enter (Figure 1d).

CT scans can contain different voxel sizes in the z direction. Such CT volumes are used to reduce imaging dose to regions outside the main region of interest. Only the

MGHParameterization can directly construct such a volume with varying slice thickness. To accommodate varying slice thicknesses with any other parameterization, one must use the workaround of constructing separate parameterized volume blocks for each of the different slice thickness regions. All CT volumes used in this study have constant slice thickness.

2.2. Timing

The speed of the four parameterizations were determined with the same patient geometries and treatment fields. Treatment plans based on spread-out Bragg peak (SOBP) fields were generated for each patient by the planning system XiO (by Computerized Medical Systems), used to generate proton treatment plans at MGH. Protons were simulated through the IBA double scattering nozzle at MGH and a phase space file was produced at nozzle exit. Phase space files were produced for 2 and 10 million protons at nozzle entrance, yielding between 2×10^5 and 4×10^5 protons (plus a number of neutrons, photons, electrons and positrons) at nozzle exit, depending on the field size, scatterers, range modulator option and the initial energy of the protons. For the prostate patient, only the 2 million proton run was used since it had an adequate number of protons in the phase space file, having the largest percentage of protons reaching nozzle exit (i.e. highest nozzle efficiency) compared to the other patients. For a given patient site, the same phase space file was used as particle source for all four parameterization methods. For better statistics, the phase spaces were re-used three times for the timing analysis and ten times to obtain dose distributions. Since Geant4 uses seeds for the random number generator to determine the proton interactions, and the seed is different for each re-sampling over the phase space, this re-sampling does not result in ten times the exact same dose deposition.

In order to study possible effects of different types of particles on the parameterization speed, we used two phase spaces, one containing protons, neutrons, electrons, positrons and photons, and one containing only protons. Both phase spaces are used as particle source for the patient simulations. A composition of particles for the phase space containing multiple particles is listed in Table 2.

In order to minimize timing effects from disc access, the entire phase space was read in at the beginning of the simulation run, resulting in no disc access to read the phase space during the navigation part of the simulation. A separate simulation was used to obtain dose distributions to eliminate all disk access during the timing runs. The total run time from start of the program until it finished was recorded. The time taken for an additional run with a single proton event was taken as the setup-offset and subtracted from the total run time to eliminate possible effects that are not related to the navigation for the different parameterizations. This single event run showed us how much time TOPAS spent setting up the patient geometry, performing smart voxel optimization, calculating physics cross-section tables, and reading phase space from disk (the entire phase space was read in even though only one event was actually used). This setup time is typically short compared to the time used for particle tracking. Timing information was recorded as total CPU time, which was divided into the time used by the system and the user. For this analysis we used the corrected CPU time, i.e. set-up time subtracted from the total CPU time. The corrected CPU time thus effectively included only the time necessary for particle creation and particle transport. The first consumes very little time and is identical for all parameterizations, thus it did not result in a timing difference. The latter is the time used for the navigation of particles through the parameterized volumes and physics processes. Since we used the same material compositions and physics lists, the time for physics processes should be the same, within statistical fluctuations.

Each simulation was submitted as a “job” to the research computing cluster at MGH on a dedicated queue to ensure that the same computer hardware was used. Each job ran on a

distinct computing node. The MC simulations were performed on a node with 2 Intel® Xenon® E5450 CPU with 3.00 GHz, 6MB of cache and 16 GB of RAM.

2.3. Memory usage

The main focus of this study was simulation time. Additionally, the maximum memory and swap memory used in each simulation were recorded. These were used to investigate possible differences in memory usage between the parameterization methods. No detailed studies were performed to optimize either of these values.

2.4. Dose analysis

For a quantitative analysis, we studied differences between the dose distributions with a gamma-plot with 3 mm and 3% of the maximum dose as criterion. Voxels within 10% of the maximum dose were considered for this analysis. This limit focuses the gamma index analysis on the “in-field” dose distribution by removing voxels of very low dose. In low dose voxels statistical fluctuations, such as neutron dose deposits, become important. This is not relevant for this study. We generated gamma index with G4MGH as reference dose, which provided the same dose distribution as a well validated code previously used at MGH (Jarlskog and Paganetti 2008, Paganetti et al. 2004, Bednarz et al. 2011).

3. Results and Discussions

3.1. Timing

The two options for the standard G4VPVParameterisation parameterization were only investigated for a small number of protons to accommodate increased memory usage of G4VPV-3D. As described in section 2.1.1, the standard G4VPVParameterisation parameterization is not recommended for volumes with a large number of parametrized voxels such as patient CT. We found G4VPV-1D to be a factor 66 (liver patient) to 154 (head and neck) slower than other parameterizations. G4VPV-3D, while only a factor 2 (head and neck) to 10 (prostate) slower than the other parameterizations, needed a factor of 15.2 (head and neck) to 18.5 (prostate) more memory to run. The 3D optimization for the largest CT volume, the liver patient, was not tested as there was insufficient memory, and the simulation would not run. In the case of a prostate patient with $512 \times 512 \times 119$ CT voxels, the G4VPV-3D optimization method requires 25 GB of memory. G4VPV-1D and -3D were thus excluded from all further studies.

A list of timing information for the different patients, the system time, the setup time and the corrected run time normalized to 10^6 particles is shown in Table 3. The processing time of protons depends strongly on their energy (range), which varies substantially between the three treatment cases. For the other particles this dependency is much weaker. The set-up times for each parameterization were around 200 seconds. If boundary skipping was switched on for G4Phantom, the set-up time doubled. We corrected the CPU times for the set-up times. Note that for a typical patient simulation at MGH, the calculations are distributed to several phase spaces (usually 20) that are run in parallel on a dedicated computing cluster. A complete patient simulation with this set-up starting at nozzle exit, i.e. after acquiring the phase spaces, takes 20-30 minutes. The dose distributions from each of those simulations are then added to produce the final dose distribution that is used for comparison with pencil beam scanning algorithms. All simulations were performed with both a phase space consisting only of protons at nozzle exit and a phase space that included all particles (protons, electrons, positrons, photons and neutrons). The percent difference in run time normalized to the G4Nested result is displayed in Figure 2 for all particles and for protons only. We found that all parameterizations take approximately the same time to run in a heterogeneous CT volume (see Figure 2). The difference in timing for the simulations

was less than 3%. The average ratio of the CPU time used for the simulation was found to be G4VNestedParameterisation : G4PhantomParameterisation : MGHPParameterisation = 1 : 1:014 : 1:015. In addition to the exact patient geometries, the same three CT cubes were used with all voxels, including the air voxels outside of the patient, set to be made of water. This resulted in a CPU time reduction of 5-40%, however, the differences between the three parameterization methods were within 5.5%, see Figure 2. This was the optimal setting for the use of the boundary skipping method for G4Phantom. It effectively removed all boundaries of the voxels. When using the boundary skipping option for this set-up, the CPU time was reduced by 27-32%. This option is not included in the results figure.

Switching on boundary skipping for G4Phantom, where the voxels had to have exactly the same HU, improved the speed performance by less than 5% compared to the G4Phantom parameterization without boundary skipping in our heterogeneous data sets, the average ratio being = 1 : 0.97. Note that this number would improve if fewer different materials were considered, as shown in the case of the entire CT being replaced by water, where we found a speed up of up to 32%.

Random differences in timing, estimated by rerunning the same simulation twice, were found to be within 0.5%.

Overall, G4Nested parameterization had the fastest parameterization navigation. The large timing differences for different patient sites can be explained by the following considerations. Protons from the phase space for the prostate site have energies between 140 and 210 MeV, for the head and neck site between 40 and 110 MeV and for the liver site between 30 and 90 MeV, resulting in different ranges in the patient. Additionally, the head and neck CT volume had air cavities, which result in an extended range of the protons. Higher beam energy causes more steps per proton in the patient and we see a higher percentage of nuclear interactions with a higher range. The probability of a proton undergoing a nuclear interaction in a Bragg curve is about 1% per cm range. The voxel sizes vary with patient and site. The variation of tissues differ by treatment site, the number of HUs used in the CT volumes being 2605 for liver, 3002 for prostate and 3694 for the head and neck case. In addition, timing depends on the treatment volume, because the bigger the volume the more protons are needed to reach the prescription dose.

3.2. Dose distributions

Figure 3 shows the gamma index with a criterion of 3% and 3 mm for G4Nested and G4Phantom with and without boundary skipping for all three patient sites, dose distributions obtained with G4MGH were used as reference. A distribution is typically considered to pass the gamma criterion if less than 5% of the voxels have a gamma value above 1, as indicated by the dashed light blue line and solid red line in the figure. This is the cut-off value used for clinical dose comparisons at MGH. All dose distributions agree within statistical fluctuations, see Figure 3. This study was optimized for the timing study, resulting in low statistics for the dose distribution, the statistical precision was estimated to be 7.8% (head and neck), 6.8% (liver) and 8.0% (prostate). Dose difference plots have also been investigated to look for systematic shifts. The dose difference plot (4a, b) of the head and neck patient shows the area where the dose distribution for G4Phantom deviates from the reference dose distribution. This effect is seen inside an air cavity of the nose.

While a study of the boundary-skipping technique for the G4Phantom in an earlier publication (Arce et al. 2008) showed good agreement for the depth dose in a simple water phantom, our analysis shows the technique to be risky in heterogeneous patient data sets. The approximations of the dose deposits seem to fail when volumes of same HU, where the mean free path of the protons is larger than the voxel size, are surrounded by high density

materials such as bone. To investigate this behavior further, we performed an additional study where we used an artificial CT volume that was composed of repeat sequences of 20 voxels in a row where 4 voxels with HU=3000 are followed by 16 voxels of air. This was constructed to emphasize differences in the dose distribution calculated with the G4Phantom parameterization navigation with boundary skipping. The dose difference distributions for G4Nested and G4Phantom with boundary skipping are shown in Figure 4c, d. G4MGH was used as the reference distribution.

By default, boundary skipping is switched on in Geant4. Since the no-boundary skipping option shows no significant advantage in speed for heterogeneous CT data sets while the skipping option shows the danger of wrong dose deposition, we do not recommend the use of this parameterization in heterogeneous geometries.

3.3. Memory

The maximum memory used for all simulations was between 0.9 and 1.9 GB, the maximum swap memory between 1.1 and 2.2GB, depending on the patient site. Maximum memory and maximum swap memory usage are given in Table 4 for the prostate patient. For a single patient CT, the memory usage was similar across the parameterizations. G4Phantom without boundary skipping used the least memory, and depending on the patient site, either G4Phantom with boundary skipping or G4MGH required the most memory.

Memory usage can effect the timing of the simulations. When the faster RAM memory is filled, the program starts to use swap memory. Since disk access is slower than RAM access, frequent and extensive use of the swap memory will increase the time a program runs. Especially the G4VPV-3D option would run faster on a computer with larger RAM since it uses very large memory. A comparison between G4VPV-3D and G4Nested would be interesting with computers with more RAM. For all other parameterizations, the difference of memory usage is small and the total memory used less than the available RAM, thus we do not anticipate that the availability of more RAM would influence the timing results.

4. Conclusion

No significant difference was found between the three Geant4 parameterization methods in timing, memory usage or calculated dose distributions for the three distinct patient treatment sites. The MGH parameterization is the most flexible for the implementation of CT images with varying slice thickness. However, this is a specific modification of the Geant4 source code developed at MGH. The MGH parameterization is not generally available and there is currently no plan to develop this parameterization for general release as part of Geant4. The G4PhantomParameterisation when used with boundary skipping incorrectly calculated the dose distribution in an albeit contrived geometry. We therefore recommend the G4VNestedParameterisation for heterogeneous voxel CT geometries, given that it is accurate, is already a well-tested part of Geant4, and is about 3% faster than all other methods.

This study was performed as part of the TOPAS project, a tool to make particle simulations for proton therapy easy-to-use, which is layered on top of Geant4. Following the results of this study, patient geometries in TOPAS currently use the G4VNested-parameterisation.

Acknowledgments

This work was supported by NIH/NCI under R01 CA 140735-01.

We would like to thank Makoto Asai for providing many helpful discussions as well as the illustration of G4Nested in Figure 1, Nicolas Depauw for his help with the gamma index and Clemens Grassberger for helpful comments and discussions.

References

- Agostinelli S, et al. GEANT4 - a simulation toolkit. Nucl Instrum Methods Phys Res A. 2003; 506:250–303.
- Allison J, et al. GEANT4 developments and applications. IEEE Trans Nucl Sci. 2006; 53:270–278.
- Arce P, Apostolakis J, Cosmo G. A technique for optimised navigation in regular geometries. IEEE Nucl Sc Symp Conf Record NSS '08. 2008:857–859.
- Arce P, Rato P, Canadas M, Lagares JI. GAMOS: A Geant4-based easy and flexible framework for nuclear medicine applications. 2008 IEEE Nuclear Science Symposium and Medical Imaging conference NSS/MIC. 2008:3162–3168.
- Bednarz B, Lu HM, Engelsman M, Paganetti H. Uncertainties and correction methods when modeling passive scattering proton therapy treatment heads with Monte Carlo. Phys Med Biol. 2011; 56:2837–2854. [PubMed: 21478569]
- Zacharatou Jarlskog C, Paganetti H. Physics settings for using the Geant4 toolkit in proton therapy. IEEE Trans Nucl Sci. 2008; 55:1018–1025.
- Paganetti H, Jiang H, Lee SY, Kooy HM. Accurate Monte Carlo simulations for nozzle design, commissioning and quality assurance for a proton radiation therapy facility. Med Phys. 2004; 31:2107–2118. [PubMed: 15305464]
- Paganetti H, Jiang H, Parodi K, Slopsema R, Engelsman M. Clinical implementation of full Monte Carlo dose calculation in proton beam therapy. Phys Med Biol. 2008; 53(17):4825–4853. [PubMed: 18701772]
- Perl J, Schümann J, Shin J, Faddegon B, Paganetti H. TOPAS: A fast and easy to use tool for particle simulation. TU-C-BRB-8, Med Phys. 2011; 38(6):3754–3754.
- Santin G, et al. Gate: A Geant4-based simulation platform for PET and SPECT integrating movement and time management. IEEE Transactions on Nuclear Science. 2003; 50(5):1516–1521.
- Aso T, et al. Validation of PTSIM for clinical usage. 2010 IEEE Nuclear Science Symposium and Medical Imaging Conference NSS/MIC. 2010:158–160.
- Schneider W, Bortfeld T, Schlegel W. Correlation between CT numbers and tissue parameters needed for monte carlo simulations of clinical dose distributions. Phys Med Biol. 2000; 45(2):459–478. [PubMed: 10701515]

Figure 1.

The four parameterization methods in a schematic display. For a detailed description see section 2.1. a) is a parameterization where each voxel is created one by one without knowledge of the intrinsic geometry patterns of CT volumes, b) a parameterization along one axis (green) followed by replication of this line of voxels along another axis and a replication of the whole slice along the third axis, c) indicates how G4Phantom calculates the position of each voxel by taking advantage of the knowledge of the pattern of CT volumes, where N is the “copy number”, the index of the voxel, dx , dy and dz (nx , ny and nz) are the dimensions (number) of a voxel in x , y and z , respectively, and d) shows the voxel numbers that the MGHParameterization associates with each voxel to find the next voxel a particle would enter when traversing the CT volume.

Figure 2.

Percent difference of the CPU timing results compared to G4VNestedParameterisation (G4Nested). Shown are G4PhantomParameterisation (G4Phantom) and MGHPParameterization (G4MGH) for the nominal CT volume and the CT volume with all voxels set to be made of water (water). Timing results for a liver, a head and neck, and a prostate patient are shown. CPU times are for a) 1 million particles (γ , e^- , e^+ , p and n, for a list of the percentages refer to Table 2), and b) for 1 million protons.

Figure 3.

Gamma index distributions with G4MGH as the reference dose distribution. The distributions are for a) head and neck, b) prostate and c) liver patients. Shown are G4Nested (blue, solid), G4Phantom without (green, dashed) and with (red, thin dashes) boundary skipping. The light blue dashed line indicates the 5% line. The numbers in the legend are the values of each parameterization at a gamma value of unity.

Figure 4.

Dose difference plots. a-b: Head and neck patient, slice with air cavities from the nose. Dose difference plots are obtained by subtracting the dose distribution from G4MGH from G4Nested (a) and G4Phantom (b). Statistical fluctuations can be seen together with the systematic lower dose in the air cavities in (b). c-d: Fabricated head and neck CT that was composed of arrays of 20 voxels where 4 voxels with HU=3000 are followed by 16 voxels of air. Differences between G4MGH and G4Nested (c) and G4Phantom (d). The distribution in (d) is systematically different.

Table 1

Patient setup information for the three patients used in this study: head and neck, liver and prostate. The field size is defined by the aperture opening, the energy is the energy at nozzle entrance.

Patient site	Number of voxels ($n_x \times n_y \times n_z$)	Voxel size [mm^3]	Field size [cm^2]	Energy [MeV]
Head and Neck	$512 \times 512 \times 76$	$0.781 \times 0.781 \times 2.5$	49.6	164
Liver	$512 \times 512 \times 172$	$0.977 \times 0.977 \times 2.5$	32.9	155
Prostate	$512 \times 512 \times 119$	$0.977 \times 0.977 \times 2.5$	30.0	222

Table 2
Percentage composition of particles in the phase spaces used for the simulations for Figure 2.

Patient site	protons	neutrons	photons	electrons	positrons
Head and Neck	60.7	17.9	17.8	3.6	0.01
Liver	58.9	18.4	19.3	3.4	0.01
Prostate	34.6	31.3	31.7	2.4	0.01

CPU run time results for the navigation for three parameterizations for all three patients for the nominal CT volume and the CT volume set to be filled with water and for phase spaces containing p, n, γ , $e^\pm$ or only protons. The CPU time is normalized to 1M particles in the phase space. The system time was not used to correct the CPU time. The setup time (unnormalized) is subtracted from the total run time before normalizing to 1M particles. All values are given in seconds.

Table 3

Parameterization and patient site	System time	Set-up time	CPU time	nominal p,n, γ , $e^\pm$			CPU time	CPU time	CPU time
CT volume phase space				water p,n, γ , $e^\pm$	nominal protons	water protons			
G4Nestod									
Head and Neck	4.5	183	6914	6622	6497	4500			
Liver	6.0	210	13336	9909	4737	3061			
Prostate	6.6	196	18427	11399	26613	16199			
G4Phantom									
Head and Neck	4.4	191	7001	6806	6594	4751			
Liver	4.6	215	13384	9940	4772	3218			
Prostate	7.3	198	18824	11767	27272	16913			
G4MGH									
Head and Neck	4.6	205	6899	6636	6573	4608			
Liver	6.1	248	13647	9933	4798	3138			
Prostate	8.4	218	18857	11588	27119	16403			

Table 4

Maximum memory and maximum swap memory used. The values are for the prostate (Prost.), liver (liver) and head and neck (H&N) patient. For G4Phantom, values with and without the boundary skipping option are given.

Parameterization	Max. memory [MB]	Max. swap memory [MB]
	Prost. / Liver / H&N	Prost. / Liver / H&N
G4Nested	1871 / 1352 / 1049	2243 / 1652 / 1338
G4Phantom w/o boundary skipping	1753 / 1189 / 1007	2125 / 1489 / 1297
G4Phantom w/ boundary skipping	1915 / 1339 / 1159	2286 / 1639 / 1450
G4MGH	1870 / 1570 / 1077	2242 / 1874 / 1368