Title

TOPAS: An innovative proton Monte Carlo platform for research and clinical applications

Permalink

https://escholarship.org/uc/item/4jr0f35f

Journal

Medical Physics, 39(11)

ISSN

0094-2405

Authors

Perl, J
Shin, J
Schümann, J
et al.

Publication Date

2012-10-19

DOI

10.1118/1.4758060

Peer reviewed

TOPAS: An innovative proton Monte Carlo platform for research and clinical applications

J. Perl[a)]
SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025

J. Shin
University of California San Francisco Comprehensive Cancer Center, 1600 Divisadero Street,
San Francisco, California 94143-1708

J. Schümann
Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School,
Boston, Massachusetts 02114

B. Faddegon
University of California San Francisco Comprehensive Cancer Center, 1600 Divisadero Street,
San Francisco, California 94143-1708

H. Paganetti
Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School,
Boston, Massachusetts 02114

Purpose: While Monte Carlo particle transport has proven useful in many areas (treatment head design, dose calculation, shielding design, and imaging studies) and has been particularly important for proton therapy (due to the conformal dose distributions and a finite beam range in the patient), the available general purpose Monte Carlo codes in proton therapy have been overly complex for most clinical medical physicists. The learning process has large costs not only in time but also in reliability. To address this issue, we developed an innovative proton Monte Carlo platform and tested the tool in a variety of proton therapy applications.
Methods: Our approach was to take one of the already-established general purpose Monte Carlo codes and wrap and extend it to create a specialized user-friendly tool for proton therapy. The resulting tool, TOol for PArticle Simulation (TOPAS), should make Monte Carlo simulation more readily available for research and clinical physicists. TOPAS can model a passive scattering or scanning beam treatment head, model a patient geometry based on computed tomography (CT) images, score dose, fluence, etc., save and restart a phase space, provides advanced graphics, and is fully four-dimensional (4D) to handle variations in beam delivery and patient geometry during treatment. A custom-designed TOPAS parameter control system was placed at the heart of the code to meet requirements for ease of use, reliability, and repeatability without sacrificing flexibility.
Results: We built and tested the TOPAS code. We have shown that the TOPAS parameter system provides easy yet flexible control over all key simulation areas such as geometry setup, particle source setup, scoring setup, etc. Through design consistency, we have insured that user experience gained in configuring one component, scorer or filter applies equally well to configuring any other component, scorer or filter. We have incorporated key lessons from safety management, proactively removing possible sources of user error such as line-ordering mistakes. We have modeled proton therapy treatment examples including the UCSF eye treatment head, the MGH stereotactic alignment in radiosurgery treatment head and the MGH gantry treatment heads in passive scattering and scanning modes, and we have demonstrated dose calculation based on patient-specific CT data. Initial validation results show agreement with measured data and demonstrate the capabilities of TOPAS in simulating beam delivery in 3D and 4D.
Conclusions: We have demonstrated TOPAS accuracy and usability in a variety of proton therapy setups. As we are preparing to make this tool freely available for researchers in medical physics, we anticipate widespread use of this tool in the growing proton therapy community. © *2012 American Association of Physicists in Medicine*. [http://dx.doi.org/10.1118/1.4758060]

I. INTRODUCTION

Monte Carlo dose calculation is considered a benchmark for other dose calculation algorithms. The potential impact of Monte Carlo dose calculation in radiation therapy may be bigger in proton therapy than in conventional therapy with x-rays because of the sharp distal dose falloff. Higher conformity to the region encompassing the patient's tumor causes the delivered dose distribution to be more affected by uncertainties in beam delivery, patient setup/immobilization, tissue heterogeneities, organ motion, and dose calculation. The effect of tissue heterogeneities on proton range is profound. In addition, the Monte Carlo method can account for both electromagnetic and nuclear interactions. It has been shown that the routine use of Monte Carlo for clinical dose calculation in proton therapy could potentially allow a reduction of uncertainty margins in treatment planning, which in turn might significantly reduce side effects.[1] To achieve this, we need to introduce user-friendly Monte Carlo to proton clinics.

In spite of their value, Monte Carlo codes are currently underutilized. Simulation projects in proton therapy are generally undertaken only by research groups with deep Monte Carlo expertise, limited to a handful of institutions focusing on research. Proton Monte Carlo is not yet widely used among clinical medical physicists. The learning process has large costs not only in time, but also in reliability. *Ad hoc* applications are passed from user to user along with potentially incorrect or outdated habits.

There are Monte Carlo codes in proton therapy that focus on specific applications. One example is VMCpro.[2] This code is extremely fast and is the basis of some commercial applications. However, it is optimized for patient dose calculation and does not allow complex geometries or tracking of all secondary particles. Its physics implementations are based on approximations which limit its general applicability. For general purpose applications, various Monte Carlo codes have been used in proton therapy, such as MCNPX,[3] FLUKA,[4] and Geant4.[5,6] While the accuracy of these codes can be expected to be comparable, Geant4 supports a higher level of geometrical complexity. Geant4 is an all-particle code able to handle motion and magnetic fields (including time-dependent fields). By virtue of its open source design and use of a modern programming language (C++), Geant4 allows easy access to particle information at any level of the simulation. Geant4 already has a long history in proton therapy.[7,8]

It is the very flexibility of Geant4 that leads to its complexity. It takes a long time for a user to achieve the expertise generally required for medical physics applications, and there are many ways in which the inexpert user may misuse the code. Users often find it difficult to identify the appropriate physics settings or to handle complex geometries in a CPU and memory-efficient manner. Geant4's functionality to handle time-dependent changes in position, fields, beam current, etc., escapes all but the most expert users. And while Geant4 supports powerful concepts such as parallel world geometry, to allow users to score in arbitrary regions and perform useful tasks such as propagating particles in the computed tomo-graphic (CT) grid while calculating dose in a coarser planning grid, in practice few users master these abilities.

Accordingly, we launched a project to layer on top of the Geant4 Monte Carlo toolkit a specialized proton simulation tool that would:

1. preserve the underlying Geant4 code,
2. deliver the full capabilities of Geant4 in terms of speed, accuracy, and flexibility,
3. offer a well benchmarked underlying physics,
4. support users who have limited or no programming expertise, and
5. introduce the routine use of Monte Carlo to the proton therapy community.

Since TOPAS will be freely available for researchers in medical physics, the tool is expected to boost proton Monte Carlo related research in medical physics and the clinical use of Monte Carlo simulation. A watershed moment in the history of Monte Carlo simulation in medical physics was the release of BEAM (Ref. 9) (now BEAMnrc), a user code written for EGSnrc (Ref. 10) to simulate electron and x-ray beams through radiotherapy treatment heads. A testament to the impact of BEAM is that its general paper, published in 1995,[9] has over 600 citations. TOPAS is in a sense a BEAMnrc (and DOSXYZnrc) for all particles (although our current implementation focuses on proton therapy). We anticipate that TOPAS will become the most used Monte Carlo tool in proton therapy. To achieve this ambitious goal the TOPAS design is based on a well-structured software architecture that anticipates the majority of scenarios that a user might want to simulate (including all existing proton therapy facilities), and is user-friendly and accurate.

II. MATERIALS AND METHODS

II.A. Overall structure of TOPAS

TOPAS is designed as a "user code" layered on top of Geant4. TOPAS includes the standard Geant4 toolkit, plus additional code to make Geant4 easier to control and to extend Geant4 functionality. Because TOPAS is layered neatly on top of Geant4, it can evolve as Geant4 evolves. With TOPAS, whatever new features, improved accuracy and calculation efficiency become available to the general Geant4 user community become easily available to the proton therapy community without obstructing any of the flexibility of Geant4. For the rest of this paper, when we refer to "native Geant4" we mean the toolkit as it comes directly from the Geant4 Collaboration, as opposed to with TOPAS extensions.

TOPAS aims to make proton simulation both "reliable" and "repeatable." "Reliable" means both accurate physics and a high likelihood to simulate precisely what the user intended to simulate, reducing issues of wrong units, wrong materials, wrong scoring locations, etc. "Repeatable" means not just getting the same result from one simulation to another, but being able to easily restore a previously used setup and reducing sources of error when a setup is passed from one user to another.

We first discuss TOPAS' custom-designed control layer, the "TOPAS parameter control system." Next, we discuss the additional parts of TOPAS that make it possible for users to develop sophisticated treatment head models, import patient geometry, and control particle sources, physics settings, scoring and graphical output, all without the need to write C++ user code. This paper is not meant to be a user handbook (a handbook will become available for online use soon). Nevertheless, we do show detailed examples even on the syntax level to demonstrate the capabilities of TOPAS and to give the reader a clear understanding how this tool would be used in specific applications.

II.B. Parameter control system

Key to TOPAS reliability is that entirely different simulation projects, such as those shown in the results section, are all built with the exact same compiled code, built by the TOPAS Collaboration, tested formally, and tested by many users. The user does not have to write this code or even have detailed understanding of Geant4. What is different from one example to the next is the set of user input "parameter files" that specify everything: geometry, particle source, fields, motion, scoring, graphical output, and physics settings.

One runs TOPAS as a command-line program with the name of the top level parameter file. That file includes whatever other parameter files are needed (Fig. 1). Each parameter file is a simple text file consisting of one or more lines, specifying either an include file or a parameter definition. Each parameter definition line has the same easily mastered format that specifies a parameter type, parameter name, parameter value, and has an optional comment:

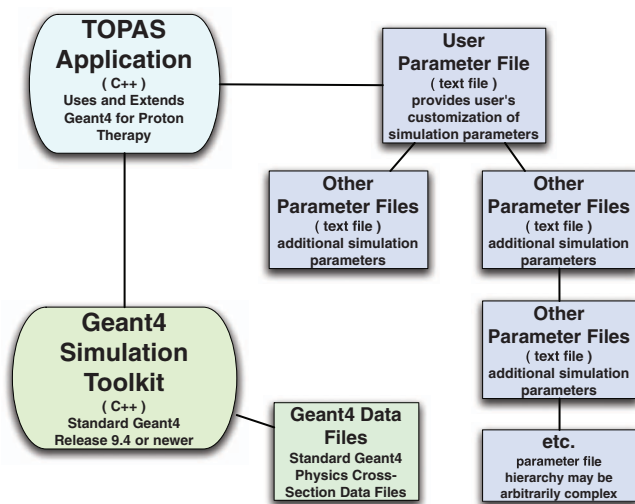- Parameter_Type : Parameter_Name = Parameter_Value # Optional comment

FIG. 1. TOPAS application uses and extends the standard Geant4 simulation toolkit. The only element that the user needs to write is the user parameter file, a simple text file that controls the simulation. The user parameter file may in turn include additional parameter files that the user may write or may obtain from other users at their own institution, from colleagues at other institutions, or from hardware vendors.

The order of lines within a parameter file does not matter, removing a potential source of user error. Where a parameter file includes more than one other file, the order of that inclusion does not matter. Throughout TOPAS a guiding principle is to "engineer-out" sources of user error.

We require specification of a "Parameter_Type", "s", "b", "i" or "d", meaning string, boolean, integer or double, to catch the next level of possible user errors. TOPAS performs strict type checking, checking that the Parameter_Value is appropriate and complete for the given Parameter_Type. Parameters of type "double" will not be accepted without a unit (or an explicit statement that this particular double should be unitless). This protects against users making the wrong assumption that some number was in cm versus mm. Parameters of type "integer" will not allow any decimal point. This protects against users making the wrong assumption about whether decimal portions are rounded versus truncated.

The parameter names are organized with a set of prefixes corresponding to major parts of the code: Ge/ for geometry components, So/ for particle sources, Ph/ for physics, Sc/ for scoring, Gr/ for graphics, Tf/ for time features (time dependent behaviors), and Ts/ for TOPAS (overall control). Example parameter settings given below are: a string to define a material, a boolean to initiate dose scoring, an integer to specify a number of scoring bins, and a double to set the size of a phantom:

- s:Ge/Phantom/Material = "Water" # filling phantom with water
- b:Sc/DoseScorer/Active = "True"
- i: Sc/DoseScorer/NBinsZ = 100
- d:Ge/Phantom/HLX = 10. cm

Because there are many cases where definition of a single shape or motion requires multiple numeric values, we provide special parameter types called "vectors" (the standard term for such structures in modern programming languages). The following example shows how one would specify four angles that define a single shape, where "dv:" indicates this will be a vector of double values:

- dv:Ge/Some_Set_Of_Angles = 4 69.11 92.29 111.04 126.02 deg

Note that TOPAS requires the length of this vector (the first number to the right of the equals sign above), another example of strict type checking. TOPAS similarly supports vectors of strings, booleans, and integers.

All parameter files are read into memory at the beginning of the simulation. This is important to support maximum flexibility on distributed processing systems where file access may not remain constant throughout a simulation. This procedure also protects against changes to parameter files after the simulation has begun.

II.B.1. Relative parameters

TOPAS supports "relative parameters," wherein one parameter may be set relative to another. The many uses of this syntax become more clear once one more fully understands

the TOPAS design, including hierarchical control files and time features (discussed below). The relative parameter handling is again engineered for safety. For example, with relative double parameters, we protect against the case of a user setting a parameter relative to some other parameter that does not have appropriate units. The solution is to insist that a unit be included on the right side of the expression:

- d:Ge/Phantom/HLX = SomeOtherParameterName cm

In this example, the unit of "cm" indicates that SomeOtherParameter must itself have units of length. If that other parameter's unit is for a different physical quantity (mass, angle, etc.), TOPAS will refuse to run. If the unit is for the same physical quantity but the unit is different (m, mm, etc.), TOPAS will perform unit conversion.

In simulation modeling, one often needs to set one parameter from a sum or product of some other parameters. TOPAS has a grammar for operations such as adding or multiplying parameters:

- Ge/Compensator/ZTrans = Ge/Aperture/DistalEdge + Ge/Compensator/HLZ mm

Certain other operations are explicitly forbidden to avoid potentially ambiguous situations, for example:

1. TOPAS does not allow setting a double parameter from an integer parameter, since integer parameters have no units.
2. TOPAS omits situations where the user may need to understand subtleties of order of precedence (hence, for example, there are no parentheses in the operations).
3. TOPAS does not allow the multiplication of two doubles since units then become difficult to check.
4. TOPAS does not allow the division of integers since rounding rules become an issue.

II.B.2. Hierarchical control

A Monte Carlo simulation using TOPAS is controlled by a hierarchy of parameter files for ease of use, reliability, and repeatability. The file adjusted on a daily basis will typically be very short because one can inherit most of the simulation parameters from other parameter files. All the parameters from the included files become part of the simulation, but one can override any of those parameters from this top file (Fig. 2).

This hierarchical structure makes it easy for the user to vary just a few parameters for a given simulation. With this design it is trivial, for example, to change the thickness of a foil by 20%. The top parameter file could simply contain a single definition:

- Ge/IonChamber/Layer2/Foil/HLZ = inheritedValue mm * 1.2

A complex apparatus may involve many separate parameter files (for a typical proton therapy treatment head this may add up to thousands of lines). But most end users will write a very simple file, overriding just a few of these parameters
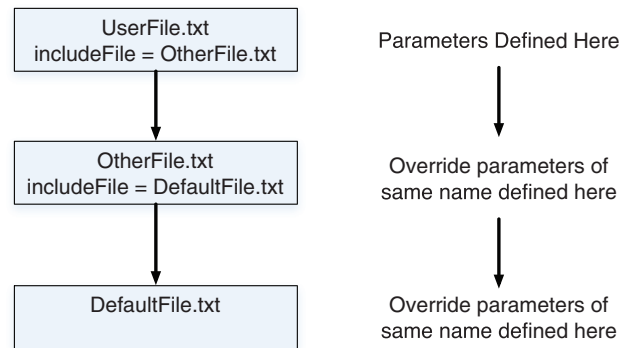
FIG. 2. By the include mechanism, the UserFile pulls in additional parameters defined in the OtherFile which in turn pulls in parameters defined in the DefaultFile. If the same parameter is in more than one file, the value from UserFile overrides the value from the OtherFile, the value from the OtherFile overrides the DefaultFile.

and inheriting the rest from files provided by others. Tools are provided to export the contents of the full set of parameter files into a useful chart, showing all parameters in all files and how settings from one file may override those from another.

II.B.3. Multiple independent workgroups

TOPAS is designed to facilitate independent workgroups focused on separate aspects of proton therapy such as treatment head design, patient handling, and imaging. To this end, a parameter file may inherit settings from more than one other parameter file. Let us consider a full clinical proton therapy setup as an example. One parameter file describes the geometry of the treatment head. Any user file that inherits from this treatment head definition file inherits all the standard (default) treatment head parameters. To simulate a specific treatment head setting, one has to override only a small set of these parameters (Fig. 3). Thus, the user input file would potentially consist of only a few lines.

Settings that differ from one gantry to another (such as which range modulator wheels have been installed) fit in a gantry parameter file layer. The user just inherits from one or the other gantry files to get generic gantry settings. To
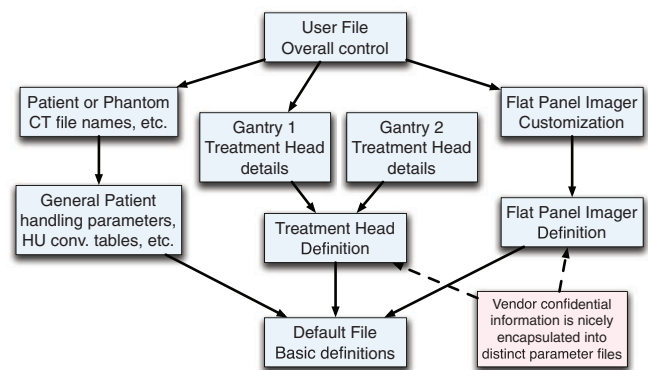

FIG. 3. Multiple chains of parameter files. The UserFile pulls in parameters from patient, gantry and imager files. Values from the UserFile override values from the other files.

simulate a patient being treated, the user file includes the patient file. That file in turn inherits some standard patient settings for the user's institutions, such as Hounsfield unit (HU) conversion tables for CT data. Using the same methodology one can also model, for example, an imaging system installed in the treatment room defined in another set of parameter files. Settings from independent workgroups are in separate "parameter file chains." TOPAS checks to make sure that no two parameter file chains modify the same parameter in a way that is ambiguous. If, for example, the material MySpecialTungstenAlloy has been defined in the imaging chain, it cannot also be defined differently in the treatment head chain (unless the top level file, the user file, itself defines this parameter).

The TOPAS parameters system provides an ideal solution for another common issue facing medical physicists—that highly accurate device details necessary for accurate simulation often constitute vendor confidential information. The TOPAS code itself can be freely shared as it contains none of the device details. The confidential information can be neatly encapsulated into just a few parameter files.

II.C. Geometry

TOPAS provides simple geometry components such as boxes, cylinders, etc., which can be combined to make more complicated structures. These simple components match the standard library of constructive geometry solids provided by Geant4. In addition, TOPAS provides components for complex devices such as a compensator, aperture, range modulator wheel (RMW), range modulator propeller, steering magnet, voxelized structure to model a patient's CT, etc. Parameters are available to set the geometry, materials, and other characteristics of the device. Time features, discussed later, make it possible to set different parts of the geometry in motion.

II.C.1. The geometry components

While Geant4 already has a layered geometry structure, in which "solids" define shapes, "logical volumes" add material information to solids and "physical volumes" place a given logical volume one or more times in space, the "Component" exists in TOPAS as a fundamental fourth geometry layer. The TOPAS geometry component provides functionality not present in the other layers. Components are the basic blocks with which the user builds a setup. A component may be a simple, single volume structure such as a box or cylinder, or it may be a complex structure created from many different volumes, such as a RMW. Each component has a "parent" component in which it is placed, with a translation and rotation relative to the center of that parent. All functionality of TOPAS interacts at the component level. Particle source positions are placed relative to components, scoring occurs in the volumes or surfaces of components, graphical features are controlled at the level of components, and so forth.

Unlike native Geant4 volumes, TOPAS components "know" what division scheme is reasonable within them. So, for example, the TsBox component "knows" it can be seg-

mented into x, y, and z slices. The TsCylinder "knows" it can be segmented into rho, phi, and z. A segmented component may have different materials assigned to each segment. The TsCylinder and TsSphere can represent cut solids (such as a specific phi segment of a cylinder). Additional simple components cover the full set of Geant4 shapes, from simple shapes like torus, polycone or tetrahedron, to elaborate shapes such as a twisted box.

The segmented components make efficient use of memory by means of Geant4's "replica" and "parameterized volume" capabilities.[5,6] This allows a large number of divided parts of a geometry to be represented internally in computer memory as a single geometrical solid whose position, size, and orientation vary from one division to the next (for example, allowing a patient CT structure that would otherwise require 20 GB to be stored in 50 MB). These powerful capabilities have proven extremely difficult to utilize correctly in Geant4. The TOPAS user gets these features simply by adding a few parameters in the parameter file defining the geometry.

II.C.2. The group component

The component layer allows another valuable function unavailable in native Geant4, the "Group Component." If one wants to have several volumes all move together as a larger unit, such as when one rotates the entire treatment head on a gantry, native Geant4 requires that these all be placed into an overall mother volume. This mother volume does not represent any actual physical reality, yet its boundaries will cause the Monte Carlo stepping action to have to take extra navigation steps, and a collision between this overall envelope and any other volume could potentially cause a boundary overlap error. In TOPAS, one can instead place the many daughter components into a group component. One can then rotate this group just as one would have rotated a mother volume. The group rotation influences the placement of the group's child components, but no mother volume is ever created, thus we eliminate unnecessary navigation steps and potential boundary collisions.

Group components also provide a solution when one wishes to rotate some other component around a point other than its center. Create a group centered on the desired rotation point, place the component into that group, and apply rotation to the group rather than the component. An example of this is the motion that brings retractable scatterer components into and out of the beam path. These scatterers might be on arms that rotate around a point other than the center.

II.C.3. Specialized components

There are various predefined specialized components. The TsRangeModulator specifies a RMW. To accommodate commercially available wheel designs, parameters allow one to specify the number of tracks (including double sided wheels), angular divisions within a track, material compositions, etc. Time features make it possible to rotate the wheel over time. The following parameter file fragment using arbitrary numbers shows some of the settings for a RMW:

The TsAperture specifies an aperture. Parameters allow to specify a simple cylindrical cutout or complex cutout that takes information from a specified aperture file that might be defined by a treatment planning program. The TsCompensator specifies a compensator. Parameters allow one to specify a simple shape, or a complex cutout that takes information from a specified compensator file. The TsDipole-Magnet specifies a perfect dipole magnet using the material properties of the magnet (useful to accurately account for the shielding effects of the magnet mass) and the field properties of the magnet, including time varying magnetic fields for beam scanning, wobbling, etc. (discussed later). Similarly, the TsQuadrupoleMagnet specifies a quadrupole magnet. The Tabulated3DField specifies a complex field map. Parameters allow one to specify the field in the OPERA-3d format,[11] commonly used to describe complex magnetic fields. The TsMultiWireChamber specifies a multiwire chamber using the number of wires, their positions, compositions, etc. The TsPropeller specifies a range modulation propeller using the number of blades, angular positions, compositions, etc. The propellor may be rotated over time.

The TsXIOPatient and TsDICOMPatient components specify patients. The former imports patient CT data from the XIO planning system (Computerized Medical System Inc.), while the latter imports DICOM. Conversion from HU is performed using a standard or user defined method, from a HU conversion table read from a TOPAS parameter file. Methods of modeling a voxelized geometry include modeling a separate box for each individual voxel, and the parameterized approach, where the entire voxel structure is represented as a single voxel in computer memory, whose position and material can vary during program execution. The choice of model has implications both for memory use and for simulation efficiency. We performed a comprehensive study of different Geant4 voxel models.[12] Based on the conclusions of that study, our TsXIOPatient and TsDICOMPatient components model CT data internally by the Geant4 voxelization method known as "nested parameterization." Variable slice thicknesses are supported. The import file may be changed over time to handle 4D patient simulations.

II.C.4. Geometries not requiring specialized components

Because simple components can be combined one within another with great flexibility, many complex structures do not require specialized TOPAS components. Examples are ridge filter, scatterer, collimator, mirror, water tank, pin diode chamber, flat panel imaging device, cylindrical ion chamber, segmented ion chamber, and Faraday cup. That such devices are modeled as a combination of simple components does not mean that users who need these devices are "on their own." Such a device can be provided to the user as a single, ready-made parameter file containing the entire description of that device. The user simply includes this parameter file into their simulation to include the device. The hierarchical parameter control system then allows the user to customize this device by overriding some parameters in their file just as they would customize a specialized component.

II.C.5. User-supplied components

If the user wants an unforeseen complex component that cannot be built from what is provided, they still have recourse to the full Geant4 C++ geometry library. The user can model any special component by creating a single C++ class and can then combine this with the TOPAS prebuilt components. User-contributed components are automatically able to take advantage of other TOPAS features, such as scoring, visualization, time features, etc. The user's coding work is only to specify the basic geometry of the component, and they can read customization information into their component through parameters. Users can share components with other TOPAS users by just sharing this one extra class. We envision an open marketplace of components, with diverse users building on one-another's accomplishments.

II.D. Particle sources

TOPAS supports parameterized sources and phase space input. The source location can be anywhere in the overall geometry by tying it to any TOPAS geometry component, and the location will move along with any motion of that geometry component. Time features make it possible to vary the source parameters during the simulation.

Source particles can be of any particle type, energy, direction, etc. One can specify arbitrary angular distribution, energy, and spatial distribution. The parameterized source defaults to a simple Gaussian energy distribution. Additional correlation factors, such as the Twiss parameterization,[13] may be included to account for scanned beams where spatial, angular, and energy distributions can no longer be treated independently. The following parameter file fragment shows some of the settings for a parameterized source that has been placed relative to a component named ExitWindow:

- s:So/Default/Type = "Beam"
- s:So/Default/Component = "ExitWindow"
- s:So/Default/BeamParticle = "proton"
- d:So/Default/BeamEnergy = 169.23 MeV
- d:So/Default/BeamEnergySpread = 0.757504 unitless # FWHM, a percentage, thus unitless
- s:So/Default/BeamShape = "Ellipse"
- d:So/Default/BeamAngularSpreadX = 0.0032 rad
- d:So/Default/BeamAngularSpreadY = So/Default/BeamAngularSpreadX rad

A phase space file contains the particle information for a set of particles. TOPAS recognizes the IAEA phase space format.[14] This file may be from some other simulation program or may have been saved from a previous TOPAS simulation (as described under scoring below). Optional extra information for timing or history information may be included. Phase space data may be recycled multiple times. The source may be rotated, allowing it to be recycled without using exactly the same particle positions.

TOPAS allows use of more than one particle source at a time. For example, one may mix effects from the proton beam with effects from residual material activation. Another example is the simultaneous simulation of a therapeutic beam and an imaging beam. With the source particle type set to an optical photon, one can simulate the mirror and light source included in many treatment heads for patient setup (Geant4 and thus TOPAS includes optical physics). The resulting light field can be visualized to confirm the setup.

Source filtering is available. This feature is useful for performing simulations with subsets of particles from a saved phase space. TOPAS uses a consistent set of filtering options to filter sources, scoring, and graphics. Filtering is discussed more fully in Sec. II.F on scoring.

II.E. Physics settings

In native Geant4, the user specifies their choice of physics models and settings in a piece of C++ code called a "physics list." This could be one of several "reference" physics lists, already provided by Geant4, or may be the user's own customized physics list. There is no single default physics. The variety of physics models and many adjustable settings within each model leads to the current dilemma that various groups in the field are using homemade physics lists which may or may not be properly validated. TOPAS provides a default physics list that has been carefully validated for clinical proton beam simulations at Massachusetts General Hospital.[15] The included models handle protons and all secondary particles (neutrons, helium ions, deuterons, tritons, photons, electrons, etc.). Only minor changes were made to migrate this physics list from the referenced paper's Geant4 version 8.1 to the current TOPAS Geant4 version 9.4. Users can further adjust settings from their parameter file, turning off various processes, adjusting the step size or the range cut, etc. Adjustments may be set independently in different geometry components, allowing, for example, more detailed simulation in an ionization chamber. Parameters also allow the user to replace the TOPAS physics list with any of the reference physics lists included in Geant4.

Scattering is considered by a condensed history algorithm[16] with functions to calculate angular and spatial distributions that match the Lewis theory,[17] validated with an electron scatter benchmark.[18] While remaining discrepancies in multiple Coulomb scattering implementations might be negligible, the uncertainties in the mean excitation energy are not. The two main parameters in the Bethe-Bloch equation that determine the range are the density and the mean excitation energy. Recommended values for the mean excitation energy of water range from 75 to 82 eV.[19–22] Similarly, for tissues, the uncertainty in I-value is potentially 10%–15%, resulting in a range uncertainty of ∼1.5%.[23] The TOPAS proton default values, which can be changed by the user, are taken from the ICRU 49 report (ICRU 1994). Many settings have been tailored for proton therapy simulations.[15,24,25] Uncertainties in nuclear interaction probabilities can be substantial, because the experimental data on which many models rely are for thin targets analyzed

for a limited set of energies. Total inelastic cross sections have been validated using a multilayer Faraday cup.[8]

II.F. Scoring

TOPAS offers options to score dose to medium or other specified material (through stopping power conversion), fluence, energy, surface current, charge, linear energy transfer (LET), etc., all with calculation of statistical uncertainties. Some of these scoring capabilities would require extensive C++ programming if a user wanted to use them in the native Geant4 environment. LET can be track or dose-averaged, and might be used for calculations of biological effectiveness.[26, 27] Scoring output comes as columns of data in comma separated values (CSV) format, in binary format, or alternatively in the AIDA (Ref. 28) or Root (Ref. 29) data analysis formats. To score dose to medium in a component named Phantom, one only needs:

- s:Sc/DoseAtPhantom/Quantity = "DoseToMedium"
- s:Sc/DoseAtPhantom/Component = "Phantom"

The TOPAS scoring system also handles phase space output. Output may be written to ASCII or binary files in the standard IAEA format. One may optionally add information not included in the IAEA standard, such as timing information (e.g., when studying the gamma emission for *in vivo* dose verification[30–32]), or history and track ID information useful to correlate phase space information from multiple planes (e.g., when studying proton computed tomography[33]). These rich phase space output options exploit Geant4's ability to control how much track history is carried throughout a simulation. It is nontrivial to learn how to control this in native Geant4, while it is easily controlled in TOPAS.

The scoring location (volume or surface) is defined by associating the scorer to a geometry component. This may be a real world component (part of the treatment head, patient, imager, etc.) or may be a component that has no material, a "parallel world component," that merely exists as a scoring region. Multiple geometries may be overlaid, with one being the mass world, in which the basic physics occurs, and others overlaying this mass world for scoring purposes. Parallel world geometry is one of the powerful Geant4 features that can be difficult to correctly exploit. Such parallel world components may lie anywhere, even overlapping real world components. More than one scorer may be associated with the same component, and each may have its own binning and filtering options.

Scorers are of two general types, "volume scorers" and "surface scorers." Volume scorers accumulate information within the associated component's volume (dose, fluence, etc.). Surface scorers accumulate information at one of the associated component's surfaces (surface current, phase space, etc.). Surface scorers can be attached to any surface of a component (face of a box, end face of a cylinder, curved section of a cylinder, etc.) and can be set sensitive to particles going into the component, out of the component or both. Scorers may be activated/deactivated in a time-dependent manner over the course of the simulation run, providing the ability for

gated scoring. Such gating could be useful, for example, to accumulate only dose deposited during a particular breathing phase.

Some components, such as the box, cylinder, and sphere, allow one to slice the scoring (in x,y,z coordinates, rho, phi, z coordinates, or rho, phi, theta coordinates, respectively). Scoring for divided components may be binned the same or differently from the geometry (material) segmentation of the component. This is an important feature for patient dose calculation. One may import a patient in a specific fine CT grid, but score in a more coarse grid. Another example would be scoring on an artificial CT grid after image registration. The design is such that any specialized component could have its own scheme to break into scoring divisions.

The TOPAS user can filter what is scored based on particle type, energy, particle emission physics process, origin volume, etc. Different scorers may use different filters. The following parameters would change the above scorer to only count dose from particles that originated in a component named collimator and having an energy above 1. MeV:

- s:Sc/DoseInTank/OnlyIncludeParticlesFromComponent = "Collimator"
- d:Sc/DoseInTank/OnlyIncludeParticlesAbove = 1. MeV

The above filter syntax is an example of how TOPAS is designed to help users avoid mistakes. The filter is named "OnlyIncludeParticlesFromComponent" rather something brief like "ComponentFilter," so that the user does not need to consult a manual to know that the filter is inclusive rather than exclusive.

It can be useful to filter scoring on some aspect of the particle's ancestors. For example, when assessing scattered dose from neutrons produced in a specific component, one may want to include not just dose from those neutrons but dose from other particles subsequently produced by those neutrons, i.e., a "neutron chain."[34] While this is cumbersome to set up in native Geant4, TOPAS users have such filters readily available.

TOPAS does not make use of the native Geant4 scoring system due to limitations in that system. Native Geant4 scorers do not have sufficient detailed knowledge of the geometry in which they are scoring to account for the sorts of complex scoring divisions that TOPAS accommodates. TOPAS scorers cooperate closely with geometry components to delegate all geometry issues to the relevant component. This close relationship allows, for example, that dose scored in a segmented sphere (segmented in rho, phi, and theta) correctly accounts for the volume of the sphere segment rather than the volume of the entire sphere. Native Geant4 provides only a few filtering options and allows a scorer to filter only on a single quantity. TOPAS provides a greater range of options and allows any number of filters to be applied to a single scorer (a daisy-chain design).

TOPAS scoring includes additional capabilities beyond native Geant4 such as statistical handling (options to report mean, min/max, variance, etc.). TOPAS includes logic to automatically determine whether the user's scoring division requirements can be accomplished by scoring directly in the component or whether it will be necessary to instead score in a parallel world copy of that component. Such a copy is needed, for example, to score patient dose in a different grid than the CT grid. When a parallel world copy is needed, TOPAS creates it automatically. Notwithstanding these many advances over native Geant4, the individual TOPAS scoring and filtering C++ classes are simpler than Geant4 scoring and filtering classes since the TOPAS classes delegate most of the work to base classes the user need never study. The user who wishes to extend TOPAS by writing their own scorer or filter need only write a few lines of code.

The TOPAS phase-space output system is integrated into the scoring architecture. This allows the phase-space output system to share as much code as possible with the scorers. For example, the same filters that control what particles are accumulated in a dose plot can control what particles are included in phase space. Such code sharing simplifies the user experience and tends towards more robust code.

II.G. Graphics

TOPAS provides the full range of graphics already available from Geant4.[35] This includes many different visualization tools from OpenGL to VRML to HepRApp.[36] Some of these tools provide highly interactive visualizations for geometry checking, others provide high resolution graphics suitable for publication. All graphics options (visibility, transparency, color, etc.) are controllable at the component level. The same filtering options available to TOPAS scoring can also apply to TOPAS graphics, so that one can choose to visualize just a certain kind of trajectory (e.g., protons), or trajectories that originated at a given component, etc. For volume rendering TOPAS supports the gMocren visualization system[37] developed by the PTSim collaboration.[38, 39] This allows one to overlay CT data, dose distributions, treatment head geometry, and particle trajectories.

II.H. Time features

Most modern treatment technologies involve time-dependent beam delivery and delivery geometries. On the proton therapy side, this involves, for example, a rotating modulator wheel (see above) or changing magnetic fields in beam scanning. TOPAS provides consistent, comprehensive handling of all time-dependent quantities. These include component motion, beam current modulation, electric and magnetic fields, gated scoring, and even patient or organ motion. This "time feature" handling is important to provide accurate simulation in the presence of possible interplay effects of proton therapy's many different time-dependent behaviors, especially in the area of pencil beam scanning (Fig. 4).

The TOPAS time feature environment supports various parameter types:

1. A time-dependent double may be used for rotational and translational movement, particle energy, magnetic fields, etc.
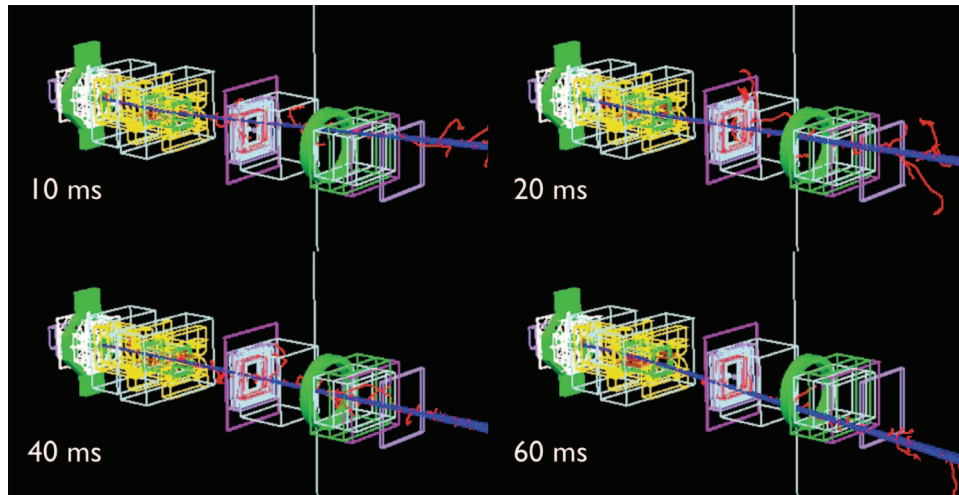
FIG. 4. The treatment head for scanning beam delivery at MGH shown at four different times during a scan. The proton trajectories through the treatment head are shown along with the much shorter delta ray tracks.

2. A time-dependent string may be used for changing the name of the patient CT file to use over time, thus supporting patient studies based on time-resolved 4D CT.
3. A time-dependent boolean may be used for gated scoring.

A user may set any parameter relative to some other parameter, such as:

- d:Ge/ModulatorWheel/RotZ   =   Tf/WheelRotation/Value deg

To make a parameter time-dependent, the user need only have the parameter on the right be of a special kind called a time feature value ("Tf"). Then, elsewhere in a parameter file, a "time feature" is defined that sets the initial value of this "time feature value" and the value's behavior over time.

The TOPAS system is designed to handle time-dependent simulations anticipating a variety of user requirements. To our knowledge this is the most versatile Monte Carlo system for time-dependent structures currently available. Because the capabilities and subtleties of the system are too extensive to treat completely in the present paper, we will highlight just a few details. The internal design of the time feature system has been described elsewhere including several application examples.[40]

A time feature may be set from a single function, such as a linear function:

- s:Tf/WheelRotation/Function = "TsLinear deg"
- d:Tf/WheelRotation/Rate = 1. deg/sec
- d:Tf/WheelRotation/StartValue = 0. deg
- d:Tf/WheelRotation/StartTime = 0. ms
- d:Tf/WheelRotation/EndTime = 50. ms

The above parameters will cause creation of a parameter called "Tf/WheelRotation/Value" that will start at 0°, then increase linearly by 1° per second for 50 ms, then repeat again from 0. It might be used when simulating a modulator wheel. Other kinds of functions are sine, cosine, discrete (which changes from one discrete value to another at specified times). A time feature may be set from just one such "time function" or from a combination of time functions. For example, beam current modulation[41] is typically generated as the multiplication of several other modulating functions, a beam pulse on/off signal, a varying intensity, and a normalization factor. A time feature may also result from a set of "consecutive" time functions. In such a case, the value is first generated by one function (e.g., sweeping a scanning beam to the right), and then, when the end time of that first function is reached, the time value is generated by a second function (e.g., sweeping the beam back to the left). Any number of time functions can be arranged consecutively.

The user does not need to take any specific action to tell TOPAS when to update the affected geometry component, particle source, etc. They only need to place a time feature value on the right side of the relevant parameter equation. TOPAS handles the rest of the work automatically by watching which components, particle sources, etc., make use of a time feature value. Whenever that time feature value changes, TOPAS signals the relevant object to update. By minimizing the work the user must do to set up a time-dependent simulation, TOPAS minimizes opportunities for user error.

Internally, time feature values are handled by an additional layer of parameters in memory, the "transient parameter" Values found in this top level file supersede those found anywhere else in the parameter system. The software design insures that to the rest of TOPAS, a transient parameter value behaves exactly the same as any other parameter value—they just happen to have the highest precedence. These transient parameter values are updated as needed by the time feature system without ever overwriting the user's original default starting values, stored elsewhere in the system.

TOPAS provides two different modes for time-dependent simulation, "sequential" and "random." In sequential mode, TOPAS starts simulation with one specified time value and then advances this time value in a linear fashion, simulating one or more histories at each time interval. Sequential is the
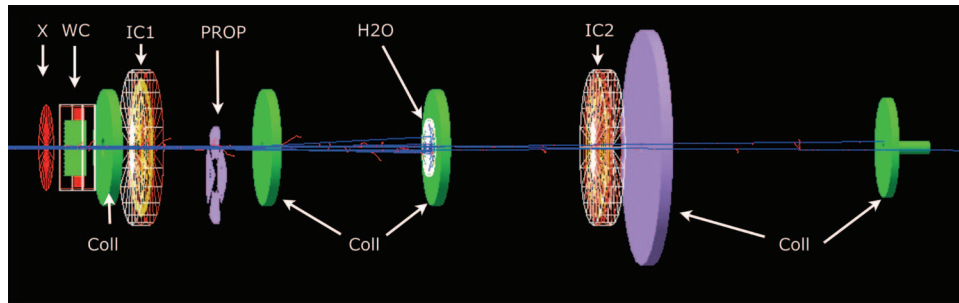
FIG. 5. UCSF proton beam line used for eye treatment as built in TOPAS. Shown are the exit window (X), wire chamber (WC), ion chambers (IC), rotating propeller (Prop), collimators (Coll), the position of the water column (H2O). The proton trajectories through the treatment head are shown along with the much shorter delta ray tracks.

default mode as its behavior matches the normal time behavior of a therapy system. However, an accurate final dose representation can be obtained only after the entire time sequence is complete. Terminating the simulation earlier will not give an accurate dose if there is any time-dependent behavior. The result may, for example, contain only part of the modulator wheel rotation sequence, or only part of the beam scanning sequence. In random mode, TOPAS randomizes its internal time value at every new history. A modulator wheel seen in this mode would be seen to randomly jump from one rotation to another to another. The advantage of random mode is that a reduced simulation time will still sample over the full range of time phases of the setup. This can be useful in doing a quick preview of an overall dose distribution. But if the full simulation is to be run, sequential mode is likely a better choice as the frequent geometry changes of random mode can hinder performance.

III. RESULTS

As a general purpose tool, TOPAS must be validated in a wide variety of configurations. We have applied "unit testing" to validate individual parts of TOPAS ensuring, for example, that beam models are correctly controlled, that dose and fluence are correctly recorded in various scoring configurations, etc. We have applied "end-to-end" testing to model complex treatment heads and to calculate dose in patient geometries. Treatment heads studied with TOPAS so far include the MGH gantry treatment delivery system installed by IBA (Ion Beam Applications) in both double-scattering and beam scanning mode, the MGH STereotactic Alignment in Radiosurgery (STAR) delivery system and the UC Davis eye treatment delivery system. Geant4 has been extensively tested for various applications in proton beam therapy.[8,15,42–47] The available data serve as additional validation of TOPAS, being layered on top of Geant4.

Each of the following TOPAS examples was built from the exact same version of the TOPAS software application. They demonstrate how the TOPAS user can accomplish complex Geant4 simulations without needing to write custom Geant4 code. No C++ programming was involved. All four examples demonstrate end-to-end validation, showing that a user can correctly set up and control a TOPAS simulation by writing only parameter files. This is a substantial improvement compared to native Geant4.

III.A. The UCSF eye treatment beamline

The University of California at San Francisco (UCSF) maintains an eye treatment facility at the Crocker Nuclear Lab at UC Davis.[48] The facility provides therapeutic proton beams for the treatment of ocular melanoma. Protons of 67.5 MeV from the cyclotron pass through the exit window, a wire chamber, a collimator, an ion chamber, a rotating range modulation propeller, a second collimator, a water column of variable thickness, a third collimator, a second ion chamber, and a patient shield (Fig. 5). To model the system in TOPAS, the geometry of the UC Davis system (dimensions and material compositions) was obtained from measurements. The residual range of proton beams can be set by changing the thickness of the water column. The water column is constructed from two TsCylinder components (one for water and one for PMMA). Four propellers are available for treatment, depending on the prescribed width of the spread out Bragg peak (SOBP). The rotating propellers were built from the TsPropeller component (see Sec. II.C.3).

Figure 6 shows the hierarchy of TOPAS parameter chains used for the UCSF eye treatment simulation (the dosimetry parameter chains are not shown). Two parameter chains, for Bragg peak and SOBP simulation, are available depending on whether a propeller is used. In user parameter files (top of each chain), the thickness of the water column is given to set the distal falloff position of a depth dose curve, and scoring parameters may be specified such as phase-spaces, a water phantom setup, etc.

Protons of 67.5 MeV with a Gaussian energy distribution of FWHM 0.7% were simulated normally incident on the exit window with a spot size of 10 mm. The different thicknesses of material traversed by the proton beam as the propeller rotated converted the monoenergetic protons into an energy spectrum, producing a SOBP. An angular velocity time feature was assigned to the propeller rotation angle. Since the beam current is constant, the propeller geometry determines the flatness of the SOBP. Thus, a deconvolution of depth dose curves in time may be useful to design propellers for better flatness. For simulating time-varying energy modulation correlated with depth dose distributions, phase-space data
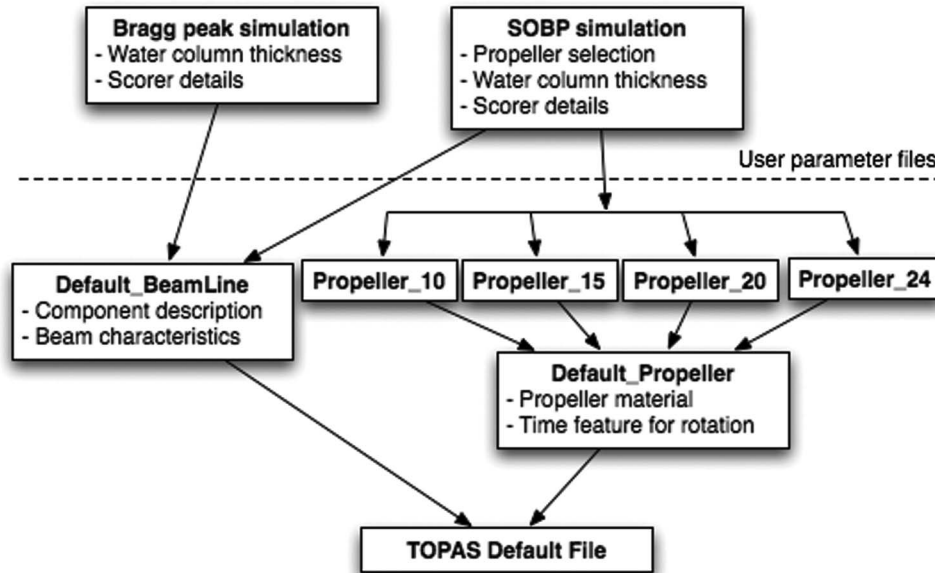
FIG. 6.   The TOPAS parameter chain for UCSF eye treatment simulation. Default_BeamLine parameter file includes initial beam characteristics and all component description except rotating propellers, which are implemented in separate parameter files, i.e., Propeller_10, Propeller_15, Propeller_20, and Propeller_24. A user parameter file for SOBP simulation needs to include Default_BeamLine and one of those propeller implementations while a user parameter file Bragg peak simulation needs only Default_BeamLine.

including time information can be accumulated at different positions in the beam path.

TOPAS has been applied to simulate the device used as a routine check of beam energy at the UC Davis facility, a water tank that one fills over time to measure the Bragg Peak.[49] TOPAS can fill this tank, "live," during the simulation run, to get the entire result from one simulation pass (Fig. 7).

As shown in Fig. 8, time features allow us to visualize how the average kinetic energy of primary protons decreases with distance from the exit window. The average energy downstream of the propeller fluctuates with time, the four separate peaks reflecting the change in propeller thickness with time for the four propeller blades. Figure 9 shows the energy spectra of primary protons at the isocenter and depth dose curves separated in four groups as follows. The 150 ms rotation period was divided into 32 equal time intervals then averaged

over the 8 time intervals having the same mean energy: group A, at 0–4.7 ms, 32.8–42.1 ms, etc., group B at 4.7–9.4 ms, 28.1–32.8 ms, etc., group C at 9.4–14.1, 23.4–28.1 ms, etc., group D at 14.1–23.4 ms, etc. The mean energy is the lowest and the width of the energy spectrum is the greatest for group A, showing the beam traverses the thickest layer in this case and is spread out over more propeller layers than for the other groups. This results in the shallowest depth penetration of the partial SOBP. The summed result gives the full SOBP. Such studies are of immense value for treatment head optimization for passive scattering and beam scanning.

III.B. The MGH STAR radiosurgery beamline

The Francis H. Burr Proton Therapy Center (FHBPTC) at MGH maintains a treatment delivery system based on
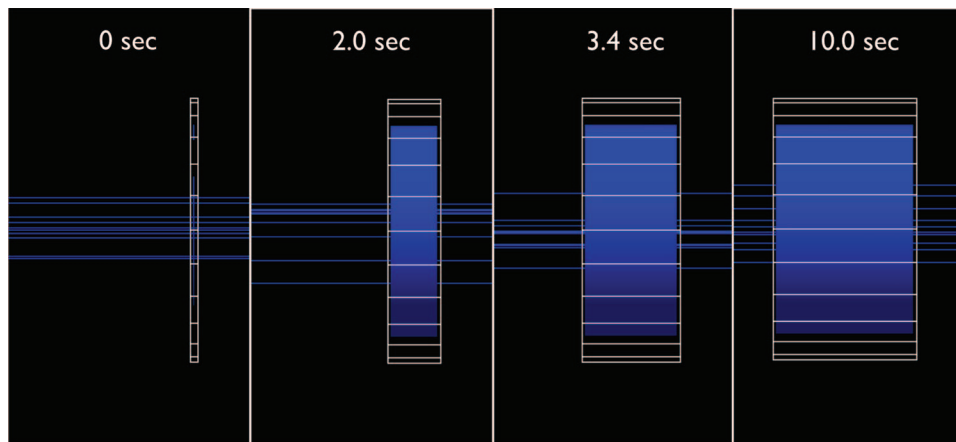

FIG. 7.   Water tank expanding over time to facilitate measurement of Bragg peak. The thicknesses are 0.01, 1.0, 1.7, and 5.0 cm, respectively, and the water phantom expands at 5 mm/s in one direction.
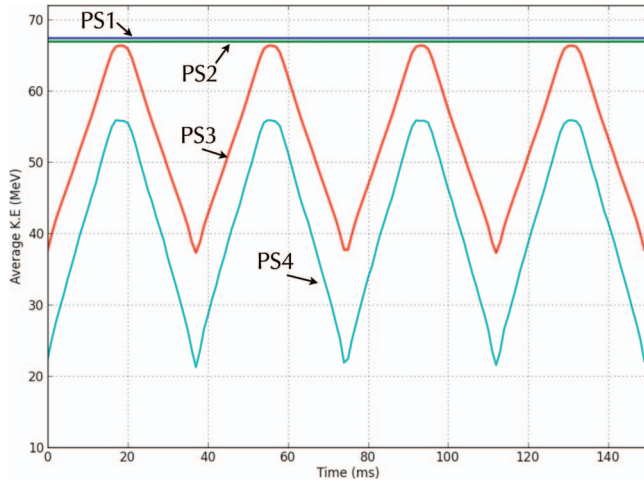
FIG. 8. Time versus average kinetic energy of primary protons at four positions along the beam path; PS1 at downstream exit window, PS2 in between the wire chamber and the first collimator, PS3 downstream of the propeller, PS4 at the isocenter. The propeller rotates once every 150 ms.

single-scattering for small-field intracranial radiosurgery treatments called STAR.[50] The treatment head was designed and built inhouse. The system is not based on a gantry system. Instead, the patient can be rotated relative to the beam isocenter. The treatment head involves a binary absorber system consisting of five lead and ten lexan blocks to modulate the energy. Brass apertures upstream and downstream of the absorber confine the beam laterally. STAR produces SOBP's by consecutive delivery of pristine Bragg peaks. A TOPAS model of the STAR treatment head has been developed (Fig. 10). The model provides a good example of how a complex system can be implemented using only basic TOPAS geometry components because the entire system is composed of the basic components TsBox and TsCylinder.

The depth dose curve for a SOBP in water is shown in Fig. 10. TOPAS simulations are compared with data measured in a water tank with an ion chamber. The distal and proximal ranges agree within the measurement uncertainly
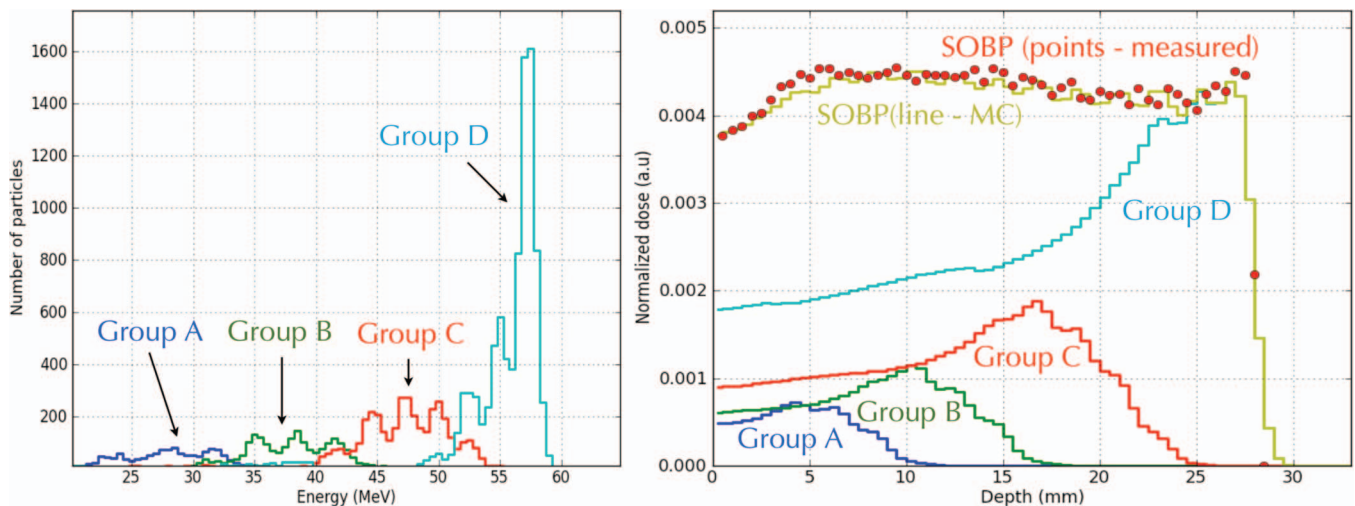
(within 1 mm) and the shape of the simulated SOBP follows the measured data. Measurements started at 7.8 mm depth due to physical limitations of the experimental setup.

III.C. The MGH gantry treatment head in passive scattering mode

The FHBPTC at MGH houses two full 360° rotational gantries with IBA universal (i.e., allowing passive scattered and scanned beam delivery) treatment heads for treatment of a wide variety of tumors. Protons with an energy of up to 230 MeV pass through a treatment head consisting of an exit window and ion chamber, a first scatterer, gantry-specific RMWs and second scatterers, scanning magnets, x and y movable collimators, a second ion chamber and a snout including patient-specific aperture and compensator.[42] All features from both gantries have been modeled using TOPAS parameter files. The geometry information and material compositions were obtained by original documents from IBA.

Figure 11 shows the FHBPTC gantry 1 treatment head as modeled in TOPAS in double scattering mode. All treatment head devices listed above are included in the model. For completeness, the model also includes a simple implementation of an x-ray tube and field mirror. TOPAS time features allow the RMW to rotate during the simulation and allow the beam current to vary. The synchronization of the RMW with beam current modulation achieves the desired spread out Bragg peak distribution. The model includes all three of the RMWs, with the user being able to control the rotation of the RMW through the parameter file. The model includes all the second scatterer options that are present in the clinical system. The user can select the snout used for the treatment, which depends on the field size. The snout is retractable to ensure that the air gap between the beam shaping devices and the patient can be minimized to reduce effects of scattering in air.

To use just one example of the geometry implementation, the TOPAS relative parameter assignment feature makes it easy for the user to get the right second scatterer settings.

FIG. 9. Energy spectra of primary protons at isocenter grouped according to time as described in the text, averaged over a 1×1 cm^2 area (left). The partial SOBP for the full phase space from each time group is shown along with the summed SOBP for the 24 mm propeller (right). The dose was averaged over a 1 cm $\times$ 1 cm $\times$ 0.05 cm volume. The published measurement for this propeller is also shown (Ref. 48).
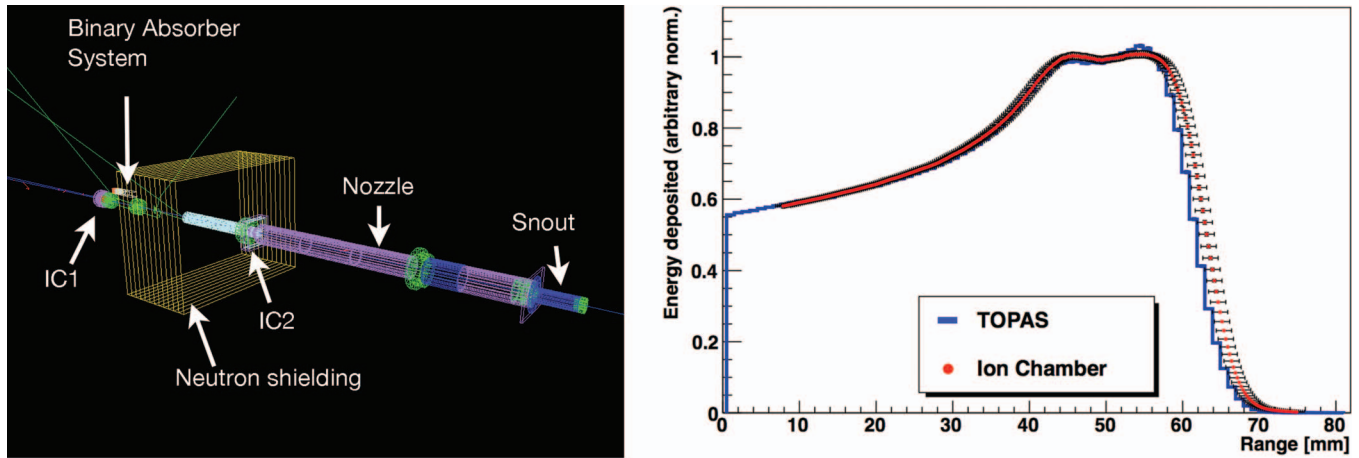
FIG. 10. (Left) STAR Radiosurgery Beamline at MGH (proton beam enters from the left). (Right) SOBP as measured (circles) and simulated (histogram) for the STAR beamline in a water tank.

Each of the gantries at MGH has three second scatterers mounted in a scatterer holder (which also has one open slot for scanned delivery without a scatterer). The holder is rotated to bring one or another scatterer into the beam. We predefined parameters to encapsulate knowledge of which scatterer is installed at which angle, such as:

- d:Ge/ScattererHolder/RotationForSS3inGantry1 = 270. deg # angle of SS3

The user then sets the holder rotation not by remembering which scatterer is where but by setting the angle to the appropriate predefined parameter:

- d:Ge/ScattererHolder/RotationZ = Ge/ScattererHolder/ RotationForSS3InGantry1 deg

Note that at MGH the TOPAS aperture and compensator components import their patient-specific description from the XiO planning system as described previously.[42]

TOPAS simulations of SOBP from the MGH gantry treatment heads in double-scattering mode were compared to water tank measurements for depth dose curves and lateral beam profiles. Note that the simulations used the TOPAS time features, as the modulator wheel is rotating with the beam current modulation being changed at the same time. The beam current modulation data were those used by the treatment control system. Figure 12 shows three depth dose curves representing the range of agreement observed between simulation and measurement using an ionization chamber. The experimental data are in cGy/MU (where MU refers to counts in a treatment head transmission ionization chamber) while our Monte Carlo
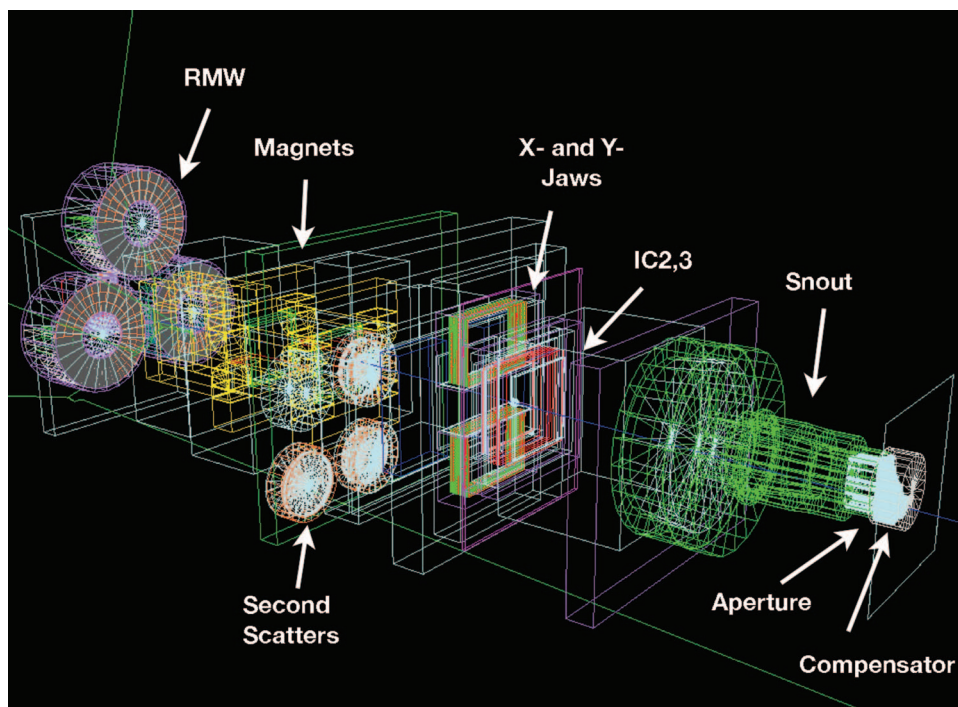

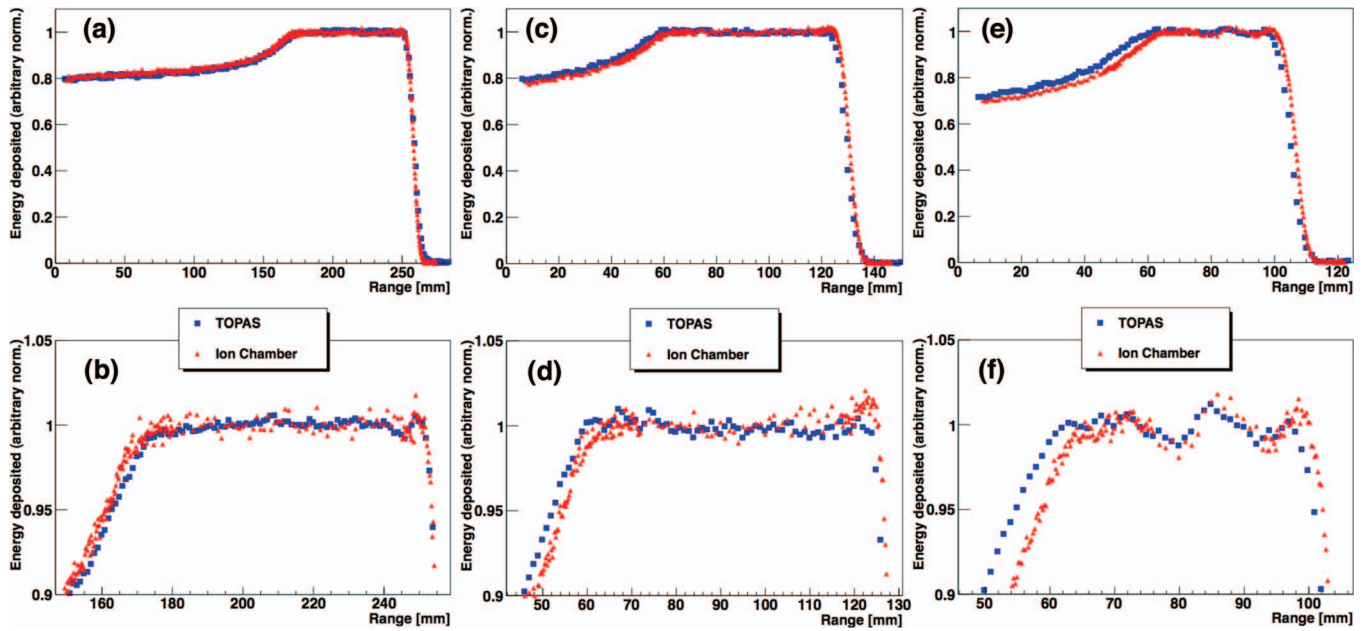FIG. 11. One of the IBA gantry treatment heads at MGH in double-scattering mode.

FIG. 12. Spread out Bragg peak in water for three different range and modulation width options. The TOPAS simulation (squares) is compared with ion chamber measurement (triangles) from the MGH gantry treatment delivery systems in double-scattering mode. A total of 5×10^6 histories were simulated and the energy deposited in a 3 cm radius around the center of the beam was scored. The mean of the SOBP dose was normalized to unity. The SOBP region is shown in a zoomed view in the lower plots. From left to right, one of the best matches (a) and (b), an average match (c) and (d), and the worst case (e) and (f).

calculation is in cGy/particle-at-treatment-head-entrance (to avoid bringing in simulation uncertainties from the ionization chamber's small scoring volume and complex geometry). As the MU is facility-dependent and varies with range and modulation width, we specify the dose here in relative units. In Fig. 12(a), the distribution shows excellent agreement, as can also be seen in a Fig. 12(b), which shows an enhanced view of the SOBP region. In Figs. 12(c) and 12(d), the TOPAS dose distribution has a slightly larger modulation width than the measurement but the result is well within clinical required accuracy, range being within $+1/-1.5$ mm and modulation being within ± 3 mm of requested specifications. There are very few settings where the agreement is not as good. Figures 12(e) and 12(f) show the combination of scattering system and modulator wheel where we obtained the worst agreement. For this combination, TOPAS correctly reproduces the wiggles in the central SOBP region, but the range is shifted by ~ 1 mm and the width of the SOBP (modulation) is ~ 2.5 mm larger than measured. This suggests that either the water equivalent thickness of a treatment head component or the time-dependent angular step position of the

RMW deviates slightly from design parameters. For all settings of the treatment head simulated with TOPAS (more than 40 have been tested), the TOPAS simulation is within clinical requirements.

Discrepancies between range, modulation width or SOBP flatness between measured data and Monte Carlo simulations can be caused by inaccurate knowledge of the actual materials used in the delivery system, which sometimes vary from the blueprints. In this case, adjustments in the beam current modulation system can be used for fine tuning.[51] Note that we have not done this "optimization" in the results shown in Fig. 12, i.e., they are based entirely on first principles.

Comparison was performed between measured and simulated dose profiles. A compensator consisting of just a uniform half block of Lexan was placed in the beam path upstream of a water phantom leaving an air gap of 8 cm (Fig. 13). Data were accumulated with an ion chamber. TOPAS correctly predicts the lateral width of the beam (Fig. 14). The oscillation of dose around X = 0 cm is caused by multiple Coulomb scattering at the compensator interface. TOPAS shows very good agreement with the ion chamber
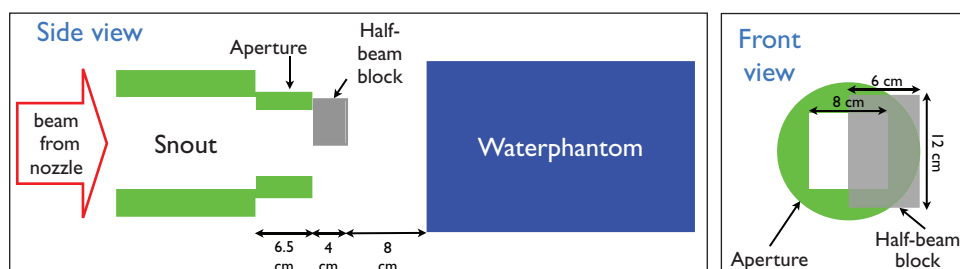

FIG. 13. A compensator consisting of just a uniform half block of Lexan was placed in the beam path upstream of a water phantom.
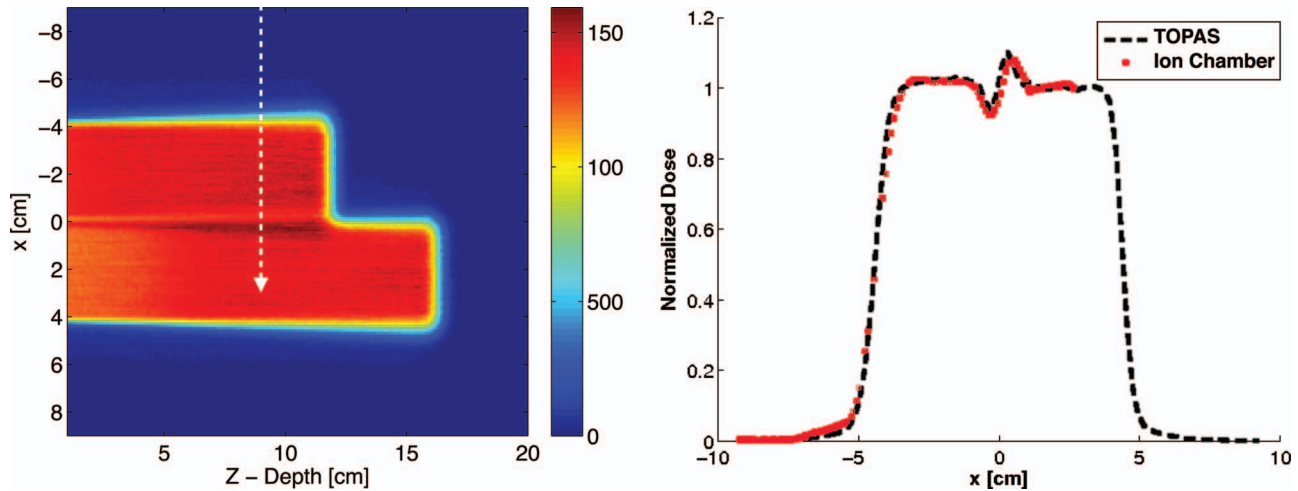
FIG. 14.   Profile simulated with TOPAS and measured with an ion chamber. (Left) The simulated dose distribution in the water tank, the white arrow indicates the path of the measurements. (Right) Normalized profiles at Z = 9 cm.

measurements, including in the central region where this oscillation occurs.

III.D. The MGH treatment head in scanning mode

TOPAS was used to model and simulate the MGH treatment head in beam scanning mode (Fig. 4). In this mode, the RMWs are replaced with two quadrupole magnets mounted on a slider. The parameter chain for the scanning mode includes time features for handling magnetic fields. Synchronization of beam current and energy modulation with magnetic field strength is crucial to simulate scanned beam delivery. Time features were assigned to the x and y magnet fields, beam energy and current, and the other quantities that vary with time.[40] The fluence map from the treatment planning system (TPS), consisting of the position and intensity of the proton beam with respect to the isocenter, was converted to a TOPAS parameter file. The real time flow of the treatment was included in the simulation, e.g., beam

pause times for setting the magnetic field to move a spot to the next position and next layer.

One advantage of the time features embedded in TOPAS is the ability to easily track a moving target. As an example, the treatment of a square field is modulated to include a linear oscillation of the target, as seen in Fig. 15. To deliver the same fluence distribution to the moving target as the stationary target, the time feature for the motion compensated field is simply added to the time feature of the X field. The particle number and Y field are not changed. The resulting fluence maps ($2 \times 2 \times 5$ mm^3 voxels) are shown in Fig. 16. The 20×10^6 histories were simulated giving 2% statistical uncertainty. In this case, the protons were scanned in a raster pattern. Examples of additional scanning patterns available with TOPAS are presented in Ref. 40.

Nonrigid registration and dose summation have not yet been implemented in TOPAS, but the time feature system was designed with such techniques in mind. Time-varying deformation maps can be handled by reading in a series of
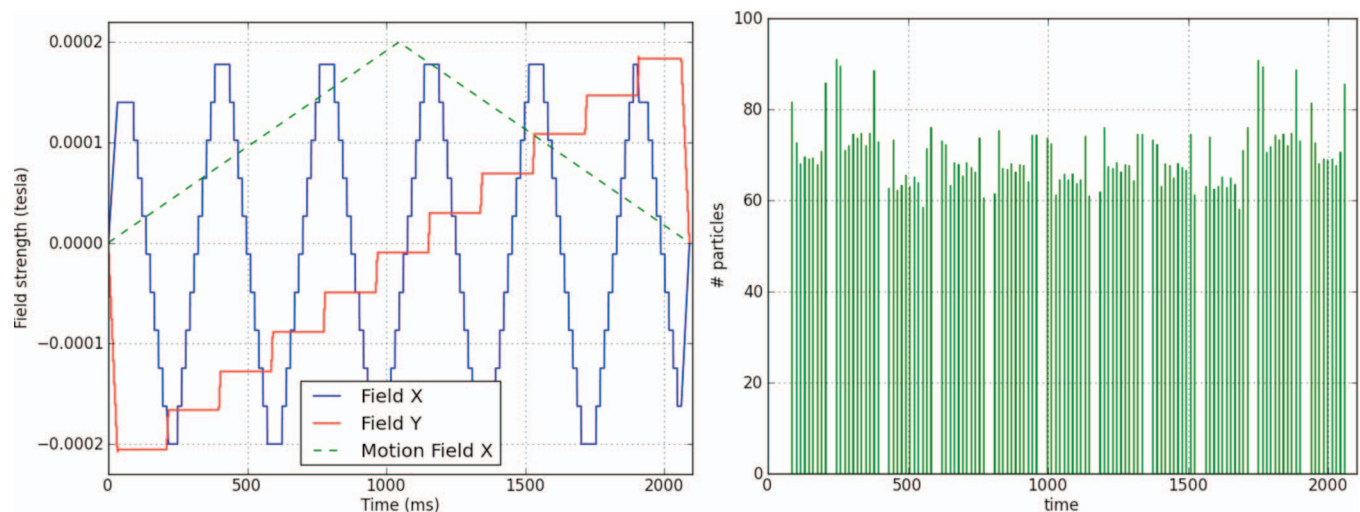

FIG. 15.   Time dependence of X and Y dipole magnet fields, and triangular modulation of X field to compensate for patient motion (left). Number of particles simulated each time interval (right).
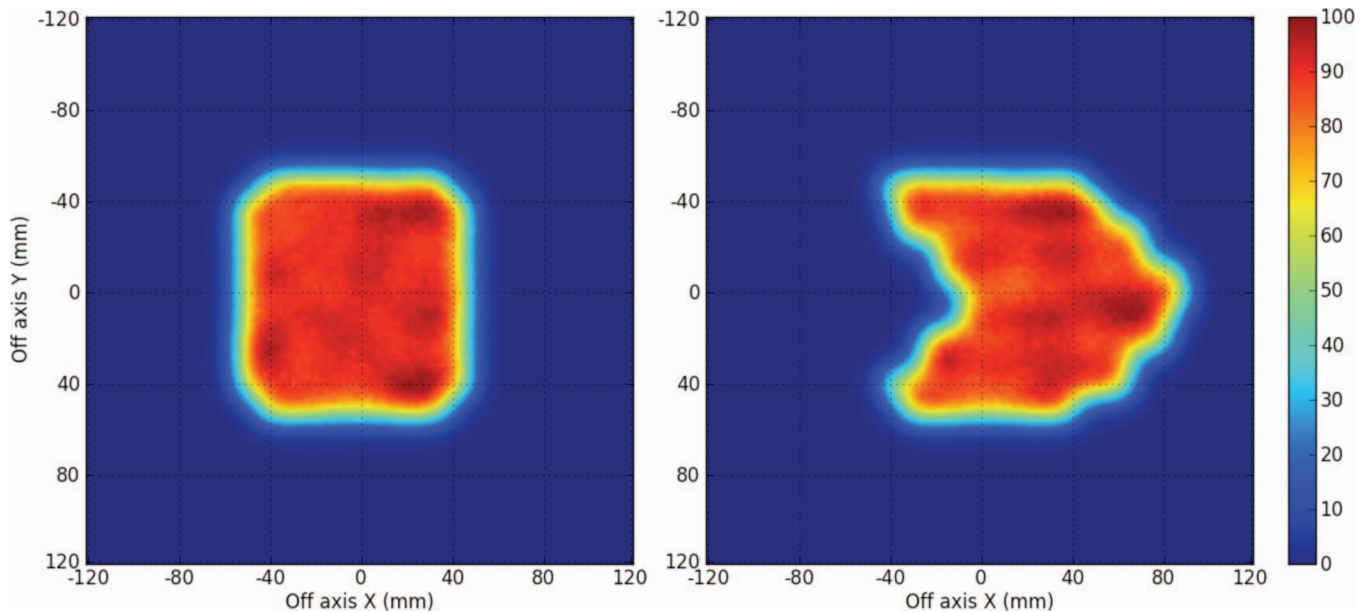
FIG. 16. The fluence distribution at treatment system isocenter. (Left) Applying only the Field X and Field Y time features from Fig. 15. (Right) Applying also the Motion Field X time feature.

deformation maps over time (just as one can currently read in a series of breathing phase patient CT images). Alternatively, deformation can be represented by mathematical models whose parameters are set from TOPAS time features. A complete solution will include both prebuilt deformation functions plus a way for users to link their own C++ deformation code. In either case, TOPAS parameters will drive the deformations, backed by the full flexibility of the time feature system.

III.E. Dose calculation based on patient-specific CT data

Example CTs from patients treated at the MGH facilities were imported into TOPAS using the specialized patient component TsXIOPatient discussed above.

Because TOPAS is controlled from a clear set of files, it becomes a good candidate to incorporate as the dose calculation engine within a TPS. One can write a script to convert TPS settings to TOPAS parameter files. TOPAS, run on these files, produces a dose map which can then be returned to the TPS. Because TPS are highly customized to individual proton centers, the job of developing such scripts is left to experts at the relevant center. This has already happened at the MGH where a user simply runs a script to drive TOPAS from the planning systems—XiO used for passive scattering and ASTROID,[52] an inhouse development, used for beam scanning. Both systems use an inhouse developed pencil-beam algorithm for dose calculation based on Hong.[53] Patient CT data, patient setup parameters, couch position, aperture and compensator definitions, beam settings, etc., are read from the planning systems and translated into a TOPAS parameter file. A patient is selected and the TOPAS jobs are then submitted to a computer cluster. The passive scattering simulations include the entire treatment head while the scanning simulations are based on a beam parameterization at treatment head

exit. Results from TOPAS are exported in DICOM format. One might expect that as TOPAS becomes more established in the field of proton therapy, users might contribute scripts for other planning systems.

A detailed comparison of Monte Carlo and analytical dose calculation methods is beyond the scope of this paper, however, examples shown in Fig. 17 demonstrate how TOPAS is currently used at MGH. The figure shows a comparison between TOPAS simulations and XiO planned dose for two patients treated in passive scattering mode. Dose distributions for one CT slice in the clinical target volume are shown together with dose difference distribution. A good agreement between the pencil beam algorithm and the Monte Carlo simulations can be seen over much of the treated region, especially for the clinical target volumes. There are significant differences as well. For this prostate patient, the planning system predicts a longer range, resulting in considerable dose differences at the distal edge of the beams. Note that the dose is simulated in absolute values. No normalization to the planning system dose was done. A TOPAS scorer giving ionization chamber charge deposit allows for output factor determination. Doses in proton therapy are reported in Gy(RBE) using a generic RBE of 1.1. TOPAS supports the calculation of LET which in turn can be used to calculate dose and endpoint dependent RBE values. Biophysical model algorithms to do this are not currently part of TOPAS, but an LET scorer already included in the scoring system provides the necessary inputs to such calculations.

IV. DISCUSSION

IV.A. Lessons from safety management

TOPAS builds in lessons from safety management. The most important of these is that engineering controls, forcing
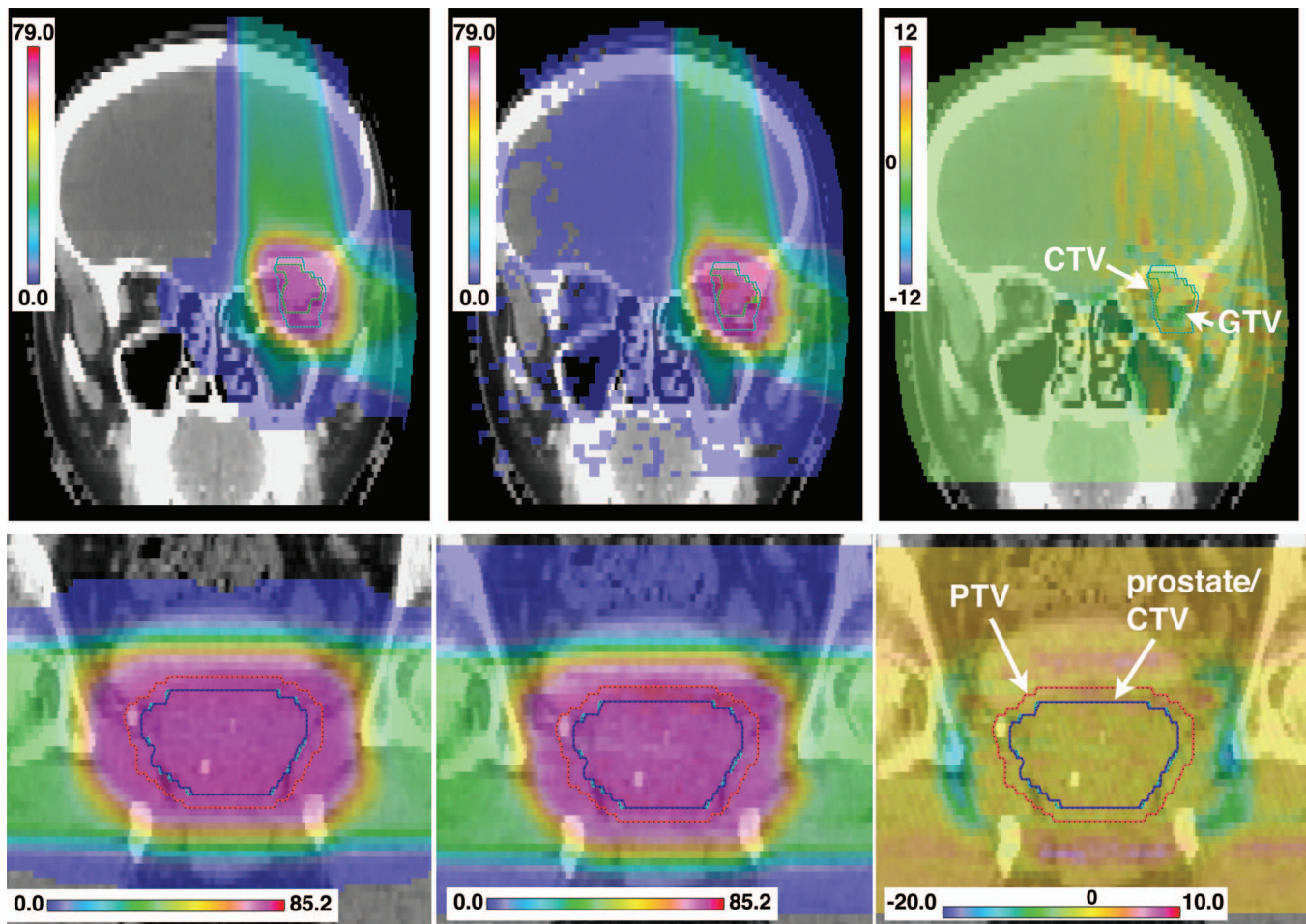
FIG. 17. Comparison between XiO planned dose (left), TOPAS dose calculation (middle), and dose difference distribution (right) for one CT slice in the CTV for two patients, a head and neck (top) and a prostate patient (bottom). Shown are complete plans including all fields. Doses are given in Gy.

one to do things, are better than administrative controls, telling one to do things. So TOPAS gets rid of line-ordering mistakes because in TOPAS, line order does not matter. TOPAS strictly enforces the type and units of all variables (string, boolean, integer or double). When a value in one file overrides the value set in another file, TOPAS insists that the variable types and unit categories (length, mass, etc.) have to match appropriately.

In the larger software world, users like software to make reasonable guesses and go ahead, but for medical physics, we want software to enforce precision. The goal is not just to get an accurate simulation, but to get an accurate simulation of the setup the user actually intended to simulate. By making it both easier to run a simulation and more difficult to make mistakes, we facilitate wider application of Monte Carlo to a variety of complex problems and to a variety of user types.

IV.B. Advantages of TOPAS software design

TOPAS is written in pure C++, the same language as the underlying Geant4 toolkit. TOPAS requires absolutely no external libraries other than Geant4. These two decisions insure that TOPAS can run on the largest possible set of computing systems—any system that can run Geant4 can run TOPAS,

which includes systems based on Linux, MAC-OS as well as MS-Windows.

The TOPAS software architecture makes extensive use of object oriented software techniques such as base classes and inheritance to avoid code duplication from one class to another. All geometry components inherit most of their functionality from a base geometry component, all scorers inherit most of their functionality from a base scorer, all filters inherit most of their functionality from a base filter, etc. Such architecture keeps software compact and enhances reliability. This code reuse also insures that user experience gained in configuring one component, scorer or filter will apply equally well to configuring any other component, scorer or filter.

The TOPAS design intentionally limits interactive control. A key characteristic of Monte Carlo simulation is that one spends a relatively small amount of time setting up parameters and relatively much more time running a simulation with those parameters. During most of the simulation time, the parameters are not to be changed. In cases where parameters are to be changed, for time-dependent behaviors, the changes should be carefully controlled, automated, not modified through interactive commands. Thus, while interactive features may seem attractive, they generally increase the risk of unrepeatable or incorrect results.

The TOPAS parameter system is not just used as an input structure but is the structure from which all other parts of TOPAS draw variables throughout the simulation run. Parameters are never copied into other local variables (except for a few cases where performance considerations cannot allow even the small amount of time involved in retrieving a value from the parameters system). By avoiding duplication of parameter values into local "cached" variables, TOPAS reduces the possibility that some local variable may fall out of synch with the intended parameter (a serious issue with time-dependent behaviors).

TOPAS is not inherently faster or slower than native Geant4. TOPAS makes it easier to set up the job, but once the event loop begins, most time is spent, as in any Geant4-based simulation, in Geant4 navigation and Geant4 physics. A well-engineered native Geant4 simulation will perform just as fast as the equivalent TOPAS simulation. But TOPAS makes it much easier to get to that well-engineered state.

The Geant4 collaboration issues a new code release roughly once per year. In doing so, they are very cautious about changing any part of Geant4 that would require updates to user codes. Because TOPAS is cleanly layered on top of the standard Geant4 release, TOPAS is, to Geant4, just another user code. At each new Geant4 release, the TOPAS collaboration will make any required adjustments and will perform regression tests to assess any changes in accuracy. TOPAS will also maintain backwards compatibility with the last several Geant4 versions, as some users may need to remain on a release that they have already themselves validated rather than migrate to a new one.

IV.C. Future work

TOPAS is meant to be freely available for researchers in medical physics. At the time of this writing, users from a variety of cancer centers in North America, Europe, and Asia are testing the code. Testers have web access to code updates and user guides. There is also a user forum where announcements are made, bugs are reported, and new features requests are discussed. This content will become more widely available when TOPAS makes its full public release in June 2013. Until then, a more limited view of the work can be seen at: https://sites.google.com/site/topasmc/

Interested users can apply there to join our test community or to stay advised of upcoming releases.

A major ongoing effort is to improve the efficiency of the code through careful code profiling and software engineering and to build in options for variance reduction that will be validated in the proton therapy context. Furthermore, TOPAS will eventually include a graphical user interface (GUI), but internally, TOPAS will always be driven by parameter files.

V. CONCLUSION

We have shown how TOPAS takes a comprehensive approach to ease of use, reliability, and repeatability. A powerful parameter control system coupled with a flexible geometry component model enables great flexibility in a manner perfectly suited to the real world of scientific collaboration in which the medical physicist lives. TOPAS, by its core architecture, enables collaboration. All users share a common, well built and well tested software application. The hierarchical nature of the parameter system makes it possible for users to hold many variables constant while at any one time adjusting just a few. Through the exchange of parameter files, users can exchange simulation setups, sharing each other's validated solutions between institutions or in translation from research to clinic. TOPAS enables an ecology of shared solutions.

We have built the basics of TOPAS and demonstrated its accuracy and usability in a variety of proton therapy setups at our own proton centers. Through the examples that we have tested, we have demonstrated that particle sources, physics options, scoring and graphics are indeed easily configurable, and that time-dependent behaviors are well supported. To use Monte Carlo transport for proton therapy today, one must be both an expert in Monte Carlo coding details and an expert in medical physics. With TOPAS, it will be sufficient to be an expert in medical physics.

[a] Electronic mail: perl@slac.stanford.edu

[1] H. Paganetti, "Range uncertainties in proton therapy and the role of Monte Carlo simulations," Phys. Med. Biol. **57**, R99–R117 (2012).

[2] M. Fippel and M. Soukup, "A Monte Carlo dose calculation algorithm for proton therapy," Med. Phys. **31**, 2263–2273 (2004).

[3] L. Waters, "MCNPX user's manual," Los Alamos National Laboratory, 2002.

[4] A. Ferrari, P. R. Sala, A. Fasso, and J. Ranft, "FLUKA: A multi-particle transport code," CERN Yellow Report No. CERN 2005-10; INFN/TC 05/11, SLAC-R-773 (CERN, Geneva, 2005).

[5] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J. J. Gomez Cadenas, I. Gonzalez, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger,

R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F. W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampen, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M. G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, P. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J. P. Wellisch, T. Wenaus, D. C. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche, "Geant4: A simulation toolkit," Nucl. Instrum. Methods Phys. Res. A **506**, 250–303 (2003).

[6]J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce Dubois, M. Asai, G. Barrand, R. Capra, S. Chauvie, R. Chytracek, G. A. P. Cirrone, G. Cooperman, G. Cosmo, G. Cuttone, G. G. Daquino, M. Donszelmann, M. Dressel, G. Folger, F. Foppiano, J. Generowicz, V. Grichine, S. Guatelli, P. Gumplinger, A. Heikkinen, I. Hrivnacova, A. Howard, S. Incerti, V. Ivanchenko, T. Johnson, F. Jones, T. Koi, R. Kokoulin, M. Kossov, H. Kurashige, V. Lara, S. Larsson, F. Lei, O. Link, F. Longo, M. Maire, A. Mantero, B. Mascialino, I. McLaren, P. Mendez Lorenzo, K. Minamimoto, K. Murakami, P. Nieminen, L. Pandola, S. Parlati, L. Peralta, J. Perl, A. Pfeiffer, M. G. Pia, A. Ribon, P. Rodrigues, G. Russo, S. Sadilov, G. Santin, T. Sasaki, D. Smith, N. Starkov, S. Tanaka, E. Tcherniaev, B. Tomé, A. Trindade, P. Truscott, L. Urban, M. Verderi, A. Walkden, J. P. Wellisch, D. C. Williams, D. Wright, and H. Yoshida, "Geant4: Developments and Applications," IEEE Trans. Nucl. Sci. **53**, 270–278 (2006).

[7]R. Gotta, "Monte Carlo simulation with Geant4 of a 3D dosimeter for therapeutical proton beams," MS thesis, University of Torino, 1999.

[8]H. Paganetti and B. Gottschalk, "Test of Geant3 and Geant4 nuclear models for 160 MeV protons stopping in CH_2," Med. Phys. **30**, 1926–1931 (2003).

[9]D. W. O. Rogers, B. A. Faddegon, G. X. Ding, C.-M. Ma, J. Wei, and T. R. Mackie, "BEAM: A Monte Carlo code to simulate radiotherapy treatment units," Med. Phys. **22**, 503–524 (1995).

[10]I. Kawrakow, "Accurate condensed history Monte Carlo simulation of electron transport: I. EGSnrc, the new EGS4 version," Med. Phys. **27**, 485–498 (2000).

[11]"OPERA-3D Reference Manual," Vector Fields Limited, Oxford, England, 2004.

[12]J. Schümann, H. Paganetti, J. Shin, B. Faddegon, and J. Perl, "Efficient voxel navigation for proton therapy dose calculation in TOPAS and Geant4," Phys. Med. Biol. **57**, 3281-3293 (2012).

[13]K. L. Brown and R. V. Servranckx, "First- and second-order charged particle optics," SLAC Report No. SLAC-PUB-3381 (Stanford Linear Accelerator Center, Menlo Park, California, 1984).

[14]International Atomic Energy Agency, "Phase-space database for external beam radiotherapy," IAEA Technical Report No. INDC (NDS)-0484 (IAEA, Vienna, Austria, 2006).

[15]C. Zacharatou Jarlskog and H. Paganetti, "Physics settings for using the Geant4 toolkit in proton therapy," IEEE Trans. Nucl. Sci. **55**, 1018–1025 (2008).

[16]L. Urban, "Multiple scattering model in Geant4," CERN Report No. CERN-OPEN-2002-070 (CERN, Geneva, 2002).

[17]H. W. Lewis, "Multiple scattering in an infinite medium," Phys. Rev. **78**, 526–529 (1950).

[18]B. A. Faddegon, I. Kawrakow, Y. Kubyshin, J. Perl, J. Sempau, and L. Urban, "The accuracy of EGSnrc, Geant4 and PENELOPE Monte Carlo systems for the simulation of electron scatter in external beam radiotherapy," Phys. Med. Biol. **54**, 6151–6163 (2009).

[19]J. F. Janni, "Proton range energy tables, 1 keV–10 GeV," At. Data Nucl. Data Tables **27**, 147–529 (1982).

[20]H. Bichsel and T. Hiraoka, "Energy loss of 70 MeV protons in elements," Nucl. Instrum. Methods Phys. Res. B **66**, 345–351 (1992).

[21]ICRU, "Stopping powers and ranges for protons and alpha particles," ICRU Report No. 49 (International Commission on Radiation Units and Measurements, Bethesda, MD, 1993).

[22]Y. Kumazaki, T. Akagi, T. Yanou, D. Suga, Y. Hishikawa, and T. Teshima, "Determination of the mean excitation energy of water from proton beam ranges," Radiat. Meas. **42**, 1683–1691 (2007).

[23]P. Andreo, "On the clinical spatial resolution achievable with protons and heavier charged particle radiotherapy beams," Phys. Med. Biol. **54**, N205–N215 (2009).

[24]J. Herault, N. Iborra, B. Serrano, and P. Chauvel, "Monte Carlo simulation of a proton therapy platform devoted to ocular melanoma," Med. Phys. **32**, 910–919 (2005).

[25]A. Stankovskiy, S. Kerhoas-Cavata, R. Ferrand, C. Nauraye, and L. Demarzi, "Monte Carlo modelling of the treatment line of the Proton Therapy Center in Orsay," Phys. Med. Biol. **54**, 2377–2394 (2009).

[26]C. Grassberger, A. Trofimov, A. Lomax, and H. Paganetti, "Variations in linear energy transfer within clinical proton therapy fields and the potential for biological treatment planning," Int. J. Radiat. Oncol. Biol. Phys. **80**, 1559–1566 (2011).

[27]C. Grassberger and H. Paganetti, "Elevated LET components in clinical proton beams," Phys. Med. Biol. **56**, 6677–6691 (2011).

[28]G. Barrand, T. Johnson, and A. Pfeiffer, "AIDA abstract interfaces for data analysis," 2001 [available URL: http://aida.freehep.org/].

[29]R. Brun and F. Rademakers, "ROOT: An object oriented data analysis framework," Nucl. Instrum. Methods. **389**, 81–86 (1996).

[30]F. Sommerer, F. Cerutti, K. Parodi, A. Ferrari, W. Enghardt, and H. Aiginger, "In-beam PET monitoring of mono-energetic (16)O and (12)C beams: Experiments and FLUKA simulations for homogeneous targets," Phys. Med. Biol. **54**, 3979–3996 (2009).

[31]J. C. Polf, S. Peterson, M. McCleskey, B. T. Roeder, A. Spiridon, S. Beddar, and L. Trache, "Measurement and calculation of characteristic prompt gamma ray spectra emitted during proton irradiation," Phys. Med. Biol. **54**, N519–N527 (2009).

[32]M. Moteabbed, S. Espana, and H. Paganetti, "Monte Carlo patient study on the comparison of prompt gamma and PET imaging for range verification in proton therapy," Phys. Med. Biol. **56**, 1063–1082 (2011).

[33]T. Li, Z. Liang, J. V. Singanallur, T. J. Satogata, D. C. Williams, and R. W. Schulte, "Reconstruction for proton computed tomography by tracing proton trajectories: A Monte Carlo study," Med. Phys. **33**, 699–706 (2006).

[34]C. Z. Jarlskog and H. Paganetti, "Sensitivity of different dose scoring methods on organ specific neutron doses calculations in proton therapy," Phys. Med. Biol. **53**, 4523–4532 (2008).

[35]J. Allison, M. Asai, G. Barrand, M. Donszelmann, K. Minamimoto, J. Perl, S. Tanaka, E. Tcherniaev, and J. Tinslay, "The Geant4 visualisation system," Comput. Phys. Commun. **178**, 331–365 (2008).

[36]J. Perl, "HepRApp: The original HepRep data browsing application," 2007 [available URL: http://www.slac.stanford.edu/~perl/HepRApp].

[37]A. Saitoh, A. Kimura, S. Tanaka, and T. Sasaki, "gMocren: High-quality volume visualization tool for Geant4 simulation," *Nuclear Science Symposium Conference Record*, NSS '07 (IEEE, Honolulu, Hawaii, 2007), Vol. 1, pp. 888–891.

[38]T. Aso, T. Yamashita, T. Akagi, S. Kameoka, T. Nishio, K. Murakami, C. Omachi, T. Ssasaki, K. Amako, A. Kimura, H. Yoshida, H. Kurashige, and M. Kaburagi, "Validation of PTSIM for clinical usage," in *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Knoxville, Tennessee, 2010* (IEEE, New York, New York, 2010), pp. 158–160.

[39]T. Aso, A. Kimura, S. Tanaka, H. Yoshida, N. Kanematsu, T. Sasaki, and T. Akagi, "Verification of the dose distributions with GEANT4 simulation for proton therapy," IEEE Trans. Nucl. Sci. **52**, 896–901 (2005).

[40]L. Shin, J. Perl, J. Schümann, H. Paganetti, and B. Faddegon, "A consistent, modular framework to handle multiple time-dependent quantities in Monte Carlo simulations as implemented in TOPAS," Phys. Med. Biol. **57**, 3295-3308 (2012).

[41]H. M. Lu and H. Kooy, "Optimization of current modulation function for proton spread-out Bragg peak fields," Med. Phys. **33**, 1281–1287 (2006).

[42]H. Paganetti, H. Jiang, S.-Y. Lee, and H. Kooy, "Accurate Monte Carlo for nozzle design, commissioning, and quality assurance in proton therapy," Med. Phys. **31**, 2107–2118 (2004).

[43]H. Paganetti, H. Jiang, K. Parodi, R. Slopsema, and M. Engelsman, "Clinical implementation of full Monte Carlo dose calculation in proton beam therapy," Phys. Med. Biol. **53**, 4825–4853 (2008).

[44]B. Clasie, A. Wroe, H. Kooy, N. Depauw, J. Flanz, H. Paganetti, and A. Rosenfeld, "Assessment of out-of-field absorbed dose and equivalent dose in proton fields," Med. Phys. **37**, 311–321 (2010).

[45]B. Gottschalk, R. Platais, and H. Paganetti, "Nuclear interactions of 160 MeV protons stopping in copper: A test of Monte Carlo nuclear models," Med. Phys. **26**, 2597–2601 (1999).

[46] H. Paganetti, "Monte Carlo calculations for absolute dosimetry to determine output factors for proton therapy treatments," Phys. Med. Biol. **51**, 2801–2812 (2006).

[47] S. W. Peterson, J. Polf, M. Bues, G. Ciangaru, L. Archambault, S. Beddar, and A. Smith, "Experimental validation of a Monte Carlo proton therapy nozzle model incorporating magnetically steered protons," Phys. Med. Biol. **54**, 3217–3229 (2009).

[48] I. K. Daftari, T. R. Renner, L. J. Verhey, R. P. Singh, M. Nyman, P. L. Petti, and J. R. Castro, "New UCSF proton ocular beam facility at the Crocker Nuclear Laboratory Cyclotron (UC Davis)," Nucl. Instrum. Methods Phys. Res. A **380**, 597–612 (1996).

[49] W. T. Chu, B. A. Ludewig, and T. R. Renner, "Instrumentation for treatment of cancer using proton and light-ion beams," Rev. Sci. Instrum. **64**, 2055–2122 (1993).

[50] J. Daartz, M. Engelsman, H. Paganetti, and M. R. Bussiere, "Field size dependence of the output factor in passively scattered proton therapy: Influence of range, modulation, air gap, and machine settings," Med. Phys. **36**, 3205–3210 (2009).

[51] B. Bednarz, H. M. Lu, M. Engelsman, and H. Paganetti, "Uncertainties and correction methods when modeling passive scattering proton therapy treatment heads with Monte Carlo," Phys. Med. Biol. **56**, 2837–2854 (2011).

[52] H. M. Kooy, B. M. Clasie, H. M. Lu, T. M. Madden, H. Bentefour, N. Depauw, J. A. Adams, A. V. Trofimov, D. Demaret, T. F. Delaney, and J. B. Flanz, "A case study in proton pencil-beam scanning delivery," Int. J. Radiat. Oncol., Biol., Phys. **76**, 624–630 (2010).

[53] L. Hong, M. Goitein, M. Bucciolini, R. Comiskey, B. Gottschalk, S. Rosenthal, C. Serago, and M. Urie, "A pencil beam algorithm for proton dose calculations," Phys. Med. Biol. **41**, 1305–1330 (1996).