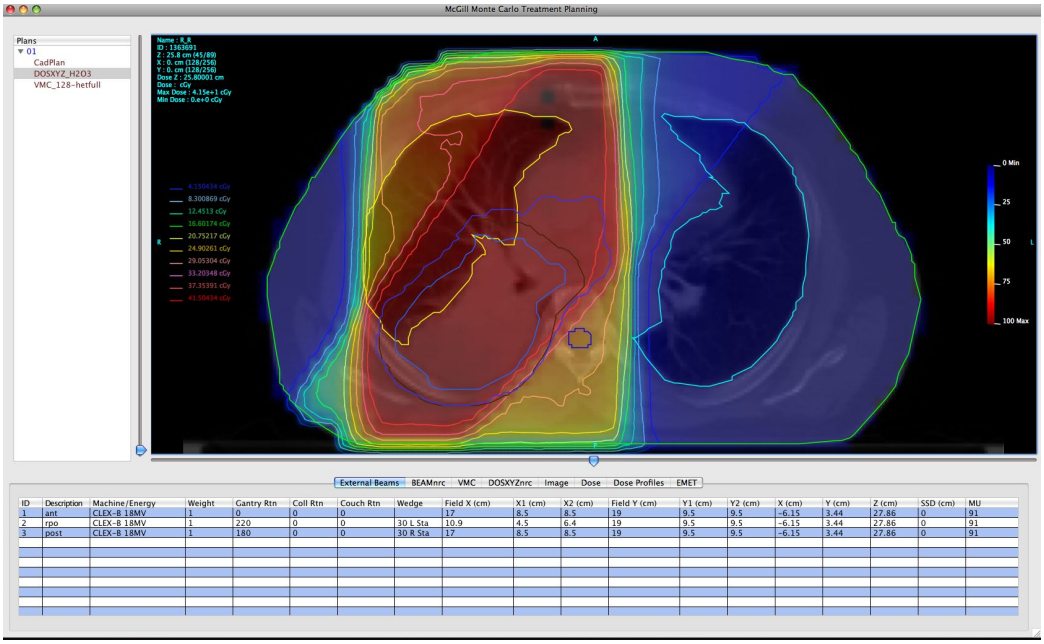
McGill Monte Carlo Treatment Planning Users Manual

Andrew Alexander
Medical Physics Unit, McGill University
Montreal General Hospital, L5-113
1650 av Cedar Montreal, QC
H3G 1A4, CANADA
andrew.alexander@mail.mcgill.ca



June 5, 2011

END-USER SOFTWARE LICENSE AGREEMENT
(the "Agreement")

This end-user software license for the program package entitled MMCTP: McGill Monte Carlo Treatment Planning (the Software), is granted by McGill University (McGill), of 845 Sherbrooke Street West, Montreal, QC, H3A 2T5, Canada, to Licensee, on the following terms and conditions:

1. **Grant of License**
   McGill hereby grants to the Licensee the non-exclusive non-transferable right to use the Software and accompanying manual (the Documentation) for its internal research use and only in the field of prospective and retrospective study of radiotherapy treatment planning, on a single computer at a single location (which may include a multi-processor super computer or a distributed computer system) on the terms and conditions specified herein. This Agreement shall not be interpreted or construed as granting to Licensee any rights, express or implied, not specifically licensed under this Agreement, regardless of whether any technology or patent right shall be dominant or subordinate to the Software, and all rights not expressly granted to Licensee by this Agreement are expressly reserved by McGill. This Agreement does not confer upon the Licensee any rights to any third party software titles or licenses that may be required to run the Software. The Licensee shall be responsible for obtaining and maintaining licenses to any third-party software.

2. **License Fee**
   McGill grants a fully paid-up license to the Licensee for the amount of $xxx.00 (xxx dollars) (the Fee). The parties hereby acknowledge that this fee does not necessarily reflect the full value of the Software and Documentation. McGill will provide the Software in object code and the Documentation to Licensee only upon execution of this Agreement and upon payment of the Fee in full.

3. **Ownership**
   McGill retains all property, right, title, and interest in and to the

Software and Documentation, including, without limitation, all trademarks, trade names, copyrights, patents and other intellectual property rights in the Software and Documentation. McGill reserves the right to use and practice the Software for its purposes, including teaching, research, continuing research, development, and testing and all other practise or utilization of the Software as well as collaborations with other institutions.

4. **Transfer Restriction**
   The Software and Documentation is licensed only to the Licensee, and may not be transferred, and this Agreement may not be assigned to anyone without the prior written permission of McGill, except by operation of law in the case of merger, acquisition, or consolidation.

5. **Copy and Use Restriction**
   The Licensee is permitted to make one backup copy and/or to make a single copy of the Software program on a hard disk, and any such copy together with the original must be kept in the Licensee's possession or control. The Licensee shall reproduce and include the copyright notice of McGill on any and all copies, whether in whole or in part, in any form. The Licensee may not modify, adapt, translate, reverse engineer, decompile, disassemble or create derivative work(s) based upon the Software or Documentation. The Software and/or Documentation may not be rented or leased to others or used to render commercial services. The Software and Documentation may not be used on a computer network unless only one person can use the Software and/or Documentation at a time.

6. **Publications and Presentations**
   In the event that the Licensee makes any publications or presentations with respect to research conducted using the Software, the Licensee shall acknowledge in such publication or presentation that the Software has been used and that the Software was developed at and licensed from McGill.

7. **Termination**
   This Agreement is in force until terminated. This Agreement will terminate automatically upon reception of a written notice to that effect, if the Licensee breaches the provisions 4, 5 or 9 of this Agreement, or if

any petition, assignment or proposal is made against or by the Licensee under the Bankruptcy Act or if the Licensee is adjudged bankrupt, or makes an assignment for the benefit of his/its creditors. The Licensee will be given written notice of default for breaches of other provisions of this Agreement. If Licensee fails to cure the breaches within thirty (30) days, this Agreement shall terminate immediately. Upon termination, Licensee shall a) cease all form of use of the Documentation and Software; and b) certify to McGill within one month after termination that Licensee has destroyed or has returned to McGill the Documentation, the Software, and all copies. This requirement applies to copies in all forms, partial and complete, in all types of media and computer memory, and whether or not modified or merged into other materials. Notwithstanding the termination of this Agreement, Article 9 herein shall remain in full force and effect.

8. **Update Policy**
   McGill may produce, from time to time, modifications and/or updates of the Software and/or Documentation. Such modifications and updates may be made available to the Licensee subject to payment of an appropriate fee. However, any such modifications and/or updates will only become available as they are developed, and McGill cannot and does not commit to the development of such modifications and/or updates on any particular schedule. McGill is under no obligation to correct Software errors that arise in any version of the Software program. McGill does not warrant that Software can or will be corrected or that McGill will develop or provide Licensee with any operations, capabilities or features not present in the version of the Software delivered to Licensee under this agreement.

9. **Confidentiality**
   All information, documentation, or object code which McGill may have imparted, or may from time to time impart to the Licensee relating to the Software or Documentation is proprietary and confidential. Licensee agrees that it shall keep such information, such documentation or such object code strictly confidential and that it shall not at any time during or after expiry or termination of this Agreement disclose the same, whether directly or indirectly, to any third party without McGill's prior written consent.

10. **Absence of Warranty**

    (a) The Software is licensed "as is".

    (b) McGill does not warrant that the Software will meet Licensees requirements, that the Software will operate in the combinations that the Licensee may select for use, or that the operation of the Software will be uninterrupted or error-free.

    (c) Licensee acknowledges that software in general is not error-free and agrees that the existence of such errors in the Software or the Documentation shall not constitute a breach of this Agreement.

    (d) There are no warranties, whether express or implied, and McGill disclaims all warranties of merchantability and fitness for a particular purpose, and any other warranty implied at law or in equity. McGill makes no representations regarding the use or the results of use of the Software and/or Documentation in terms of corrections, accuracy, reliability, or otherwise.

11. **Limitation of Liability**
    In no event shall McGill be liable for any special, indirect or consequential damages, including those damages arising from loss of profits, loss of revenues, loss of data, or downtime, even if McGill has been advised of the possibility of such damages. McGill's liability on any other claim for loss or liability, including negligence, arising out of or connected with this Agreement, or the Software or Documentation, shall in no event exceed the smallest of the amount set forth in Article 2 or $100.00 CDN (one-hundred Canadian dollars).

12. **Miscellaneous**

    (a) This Agreement shall be governed by and construed in accordance with the laws of the Province of Quebec.

    (b) This Agreement is drawn up in English at the request of the parties. Le présent contrat est rédigé en anglais à la demande des parties.

    (c) Neither party shall have the right to use the name of the other party without the specific written permission of the authorized representative of the other party.

(d) This Agreement does not establish a joint venture, agency or partnership between the parties, nor created an employer-employee relationship.

(e) Any waiver of any right under this Agreement must be in writing and signed by an authorised representative of McGill. Failure of McGill to enforce a right or strict performance under this Agreement shall not be deemed to prevent McGill from subsequently asserting or exercising any right or from requiring strict performance. Waiver or failure to enforce shall not affect the validity of this Agreement.

(f) This Agreement contains the entire agreement and understanding between the Licensee and McGill with respect to the subject matter hereof and supersedes any prior agreement or understanding whether oral or written, relating to the subject matter of this Agreement. This Agreement may only be modified by a written agreement signed by McGill.

Abstract

The McGill Monte Carlo Treatment Planning system (MMCTP) [1] is a software environment for research development of patient specific treatment planning using Monte Carlo (MC) dose calculations. MMCTP is suitable for large-scale prospective and retrospective treatment-planning studies with standard treatment planning tools for analysis. The design includes a workstation GUI for treatment planning tools, and anonymous access to standard low cost hardware for MC dose calculation.

Contents

List of Figures

1 Introduction

The McGill Monte Carlo Treatment Planning system (MMCTP) is a software environment with the aim of integrating new patient specific treatment-planning with Monte Carlo (MC) dose calculations suitable for large-scale prospective and retrospective treatment-planning studies. MMCTP features are summarized in table 1.

Table 1: MMCTP version 1, features.

Features	Summary
Import formats	-DICOM_RT, RTOG, CADPlan CART.
Export formats	-DICOM Dose
Visualization options	-2D axial, sagittal or coronal view for images, contours and dose distributions. 3D beam's eye view and room's eye view for images, contours, and beam geometry settings (e.g., jaws, MLC, couch, table and gantry rotation).
Treatment planning	-Add and delete external beams, edit beam properties (e.g., treatment unit, energy, jaw settings and MLC leafs). -Structure contouring and editing tools.
Monte Carlo	-Generation of input files and simulation submission for BEAMnrc, DOSXYZnrc and XVMC.
Dose analysis	-DVH calculator, dose profile graphs and dose distribution operations (e.g., addition, subtraction, multiplication and division of dose distributions).

The design includes a workstation GUI with treatment planning tools, and anonymous access to standard low cost hardware for MC dose calculations. For each MC simulation, MMCTP uses template input files to generate a specific input file based on the treatment plan. The input files are then uploaded to the cluster for MC simulations. Upon completion, the GUI downloads individual dose files to the workstation. Files are saved under the McGill RT format, which was designed for reading and writing radiotherapy data. MMCTP is compatible with Windows, Linux and Macintosh operating systems and as such, it can be used as a convenient DICOM_RT viewer and DVH calculator for radiotherapy data. MMCTP is a valuable

Table 2: MMCTP System Requirements

	MMCTP GUI	Cluster
OS	OS X, WINXP, Linux	Linux, OS X
RAM	1 GB	NA
CPU	2 GHz	NA
Connection Protocols	NA	SSH, FTP or SFTP

tool for the research development of MC treatment-planning studies for the reason that it minimizes the labor-intensive tasks of MC treatment-planning. The visualization, dose matrix operation and DVH tools offer possibilities for plan analysis and comparison to plans imported from commercial treatment-planning systems.

1.1 System Requirements

The preferred system requirements would be a macintosh computer running OSX. However, MMCTP has been designed to be cross-compatible with Windows and Linux operating systems. In general, the MMCTP should be run on a computer with at least 2 GB of ram and a processor speed of at least 2 GHz. However, these specifications may not be sufficient depending on the data sets which the user is using. High resolution images, and dose distributions will quickly use up system resources. There are also system requirements for the cluster which MMCTP connects to. MMCTP connects using SSH and FTP. These two connection protocols must be enabled.

1.2 Monte Carlo Treatment Planning

Monte Carlo treatment planning systems (MCTP) require proper experimental verification of the MC dose algorithms for validity of the system. MCTP commissioning falls outside the goals of MMCTP or this manual, however MMCTP can be used to generate PDDs, profiles and output tables. Please refer to AAPM task group 105 [2] for a overview of MCTP systems. MC validation should follow the process shown in figure 1.

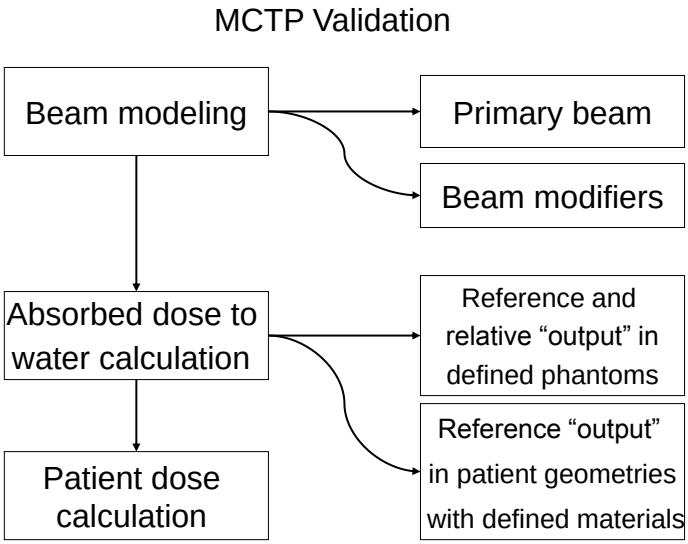


Figure 1: MCTP validation process

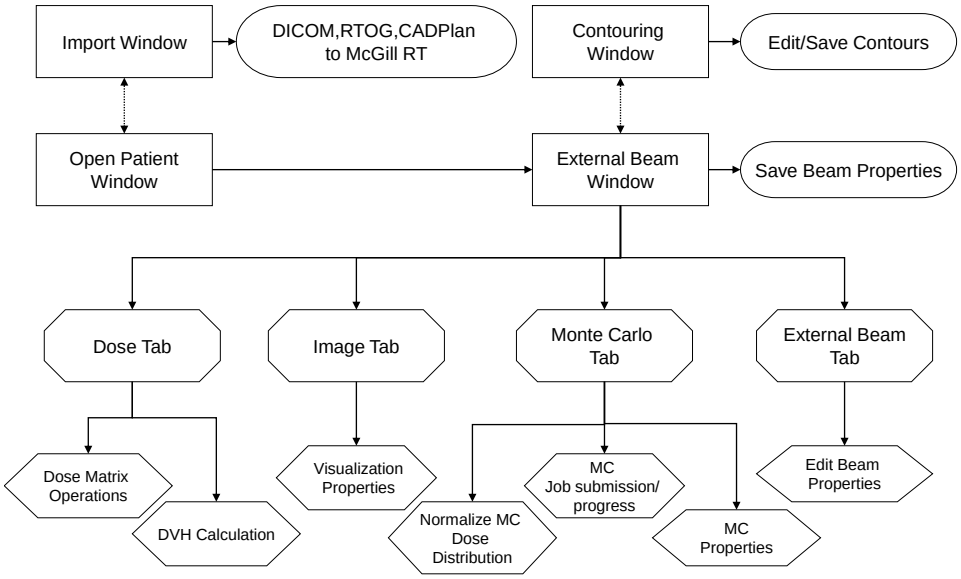


Figure 2: Work flow of MMCTP

1.3 MMCTP Flow

The MMCTP work flow diagram is shown in figure 2. Users should become familiar with this diagram in-order to properly navigate within MMCTP.

2 Installation and Configuration

2.1 Installation

Windows and MAC uses are provided with a self extracting file to install MMCTP system files. These files include the application, configurations folder, user manual, Quesa, Plink, PSCP and MMCTP-Settings folder.

2.1.1 Quesa

Quesa is a high-level, open source, 3D graphics library which must be installed. Quesa can be downloaded from http://www.quesa.org/ or http://www.realsoftware.com under additional utilities. MAC installation : Place the Quese.framework folder into /Library/Frameworks/

2.1.2 Plink and PSCP - Windows

Windows users use plink and PSCP to enable SSH and SCP connections. Plink and PSCP are a command line clients. MMCTP will make use of these program when connecting to remote computers. The shell path of these programs are displayed within the Configurations window under that shell login tab. In the event that these programs are moved, the user must update the shell paths manually.

2.1.3 Configurations folder

The configuration folder stores MMCTP system files. These files are global to all uses and platforms. The location of the configuration folder will be at the same level as the program excitable or source code. MAC : /Applications/MMCTP/

2.1.4 MMCTP-Settings folder

The MMCTP-Setting folder stores user specific MMCTP setting files. These files are unique to each user and are edited within MMCTP. The location of the folder, MAC : /Users/.../Library/Application Support/MMCTP/MMCTP-Settings/

2.2 User Configurations

Once the installation is complete, the user is required to populate user specific settings. These settings include the creation of treatment machines, CT to density curves, MLC properties, remote cluster login properties, and Monte Carlo settings. These settings are accessible within the Configuration Window and are essential for proper MMCTP use.

2.2.1 MMCTP Preferences

The preference tab as shown in figure 3 allows the user to modify the location of various user folders. In addition to the folder path options, the tab includes two options, Export Dose Plane options, and DVH calculation options.

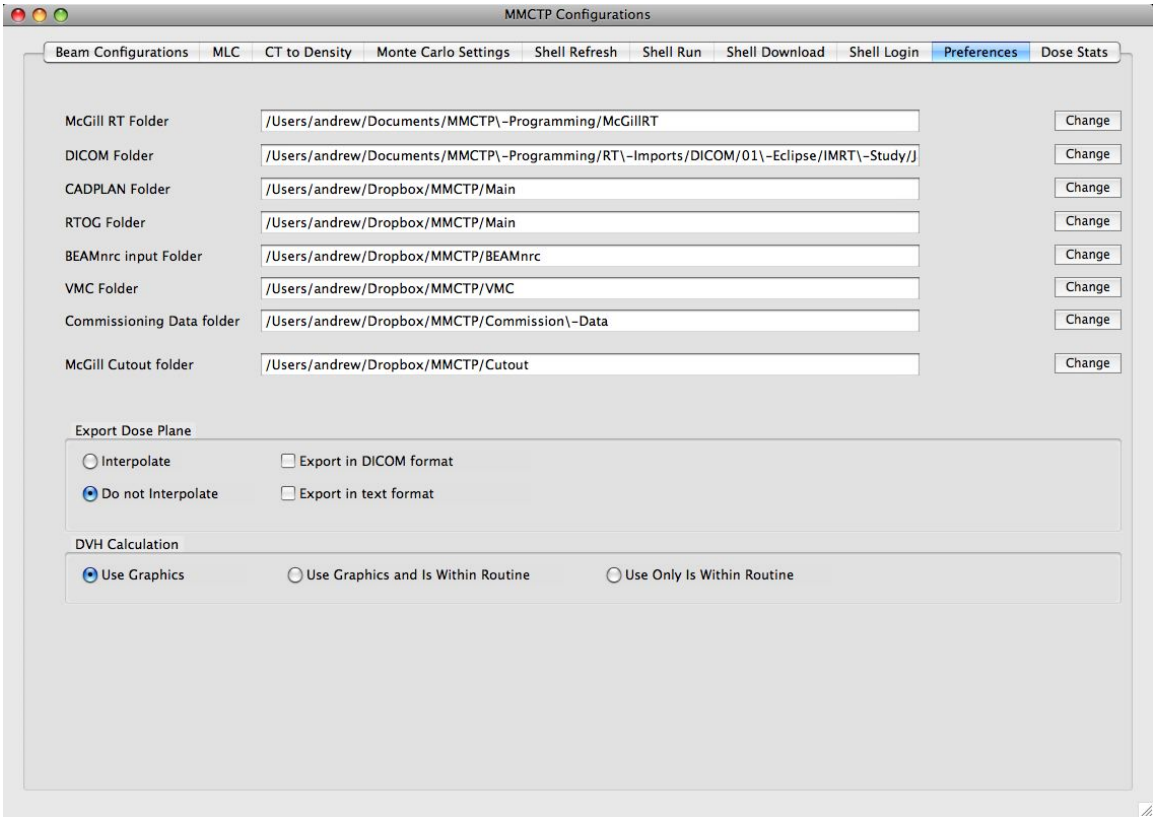


Figure 3: MMCTP Preference tab

McGill RT Folder houses the McGill RT patient folders. The user should choose a hard drive path with sufficient storage.

DICOM Folder is the folder MMCTP browses during DICOM import. This folder can be changed within the import window of MMCTP. In-order to speed-up the import of DICOM files, the user should limit the contents of a folder to one patient.

CADPLAN Folder is the folder MMCTP browses during CADPLAN Cart import. This folder can be changed within the import window of MMCTP.

RTOG Folder is the folder MMCTP browses during RTOG import. This folder can be changed within the import window of MMCTP.

BEAMnrc Folder houses all BEAMnrc user files. These files include: template linac input files, input files for various component modules, the MMCTP BEAMnrc settings file, the MMCTP DOSXYZnrc settings file, the DOSXYZnrc template input file and the DOSXYZnrc material file.

VMC Folder houses the VMC template input file and the MMCTP VMC settings file.

Commissioning Data Folder houses the MMCTP commissioning data. Profiles and output tables are saved within this folder.

McGill Cutout Folder houses the cutout preference file and template cutout input file.

2.2.2 Shell Login

The Shell Login tab as shown in figure 4 allows the user to add, edit and delete shell hosts profiles. These explicit properties are specific to the remote host and require attention to detail. Once configured, these properties allow MMCTP to connect to remote clusters running linux, Unix or MAC.

Title Title is used to determine which login shell to access

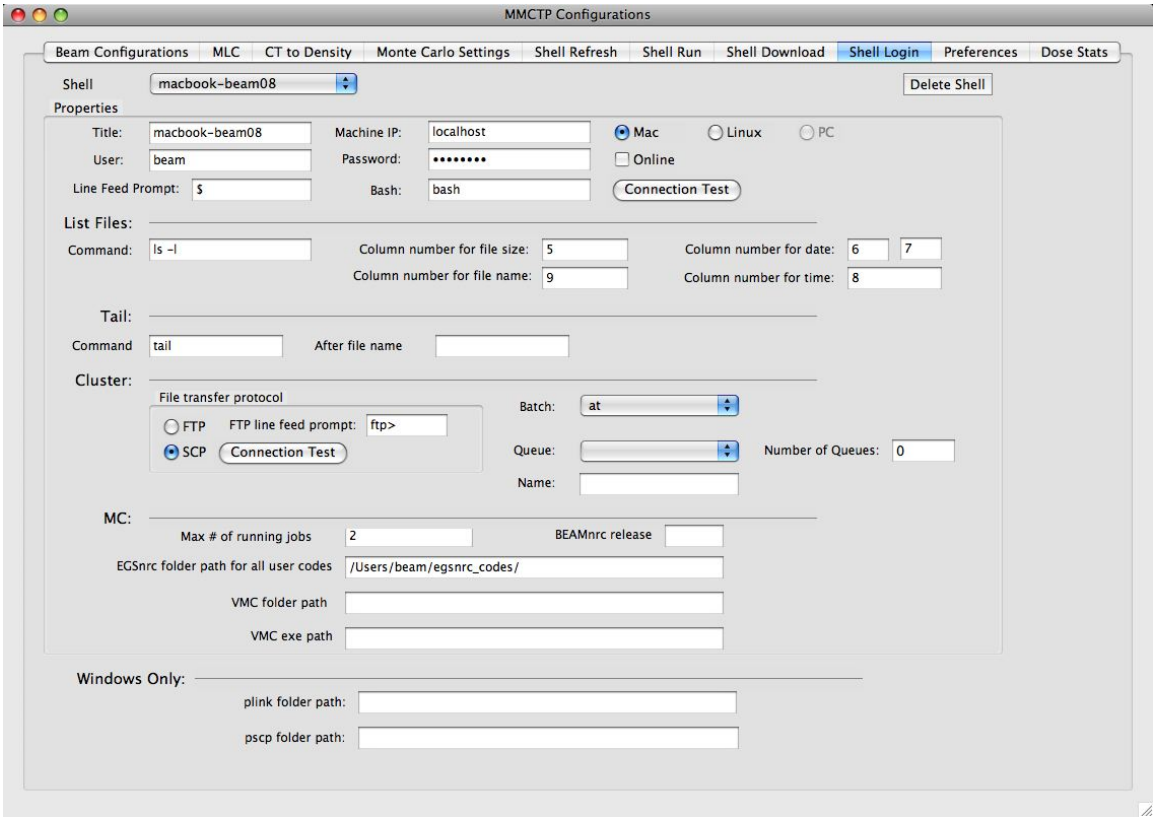


Figure 4: MMCTP Shell Login tab

Machine IP IP address of the host

Users Login user name

Password Login password

Line Feed Prompt Should be a unique set of text to identify the completion of the previous command. If this value is incorrect, MMCTP will not be able to finish one command and move on to the next. Thus, the MMCTP shell will be stuck in an infinite loop searching for a non existent line feed string. If the user name is egsnrc, set PS1="\h:\w \u$ " in your bachrc file on the remote host and set egsnrc$ as the Line Feed Prompt in MMCTP.

Bash Sets the bash environment, if required, (ex, "bash") else leave blank.

List Files This section sets the list files command and the delimiter spacing for the file size, name, date and time. The delimiter is assumed to be a blank space.

Batch Batch type (ex, "at" or "NQS")

Queue Options for adding a queue to job submissions.

File Transfer Protocol FTP or SCP. The FTP line feed prompt, most often "ftp>" This field is used to determine when an FTP command has finished.

Max # of running jobs This value limits the number of concurrent jobs MMCTP will submit to this host.

EGSnrc folder path Sets the EGS_Home folder path

VMC folder/exe path Sets the VMC directory and exe path.

Windows Only: MMCTP requires the use of plink and pscp to establish connections between computers, when installed on Windows platforms. The full folder paths for plink and pscp are required for MMCTP to make use of these executables.

2.2.3 Beam Configurations

The beam configuration tab shown in figure 5 allows the user to add, edit and delete the available treatment machines within MMCTP. The main properties of a treatment machine are beam mode, linac name and beam energy. The beam mode is defined within the first pulldown menu and can Photon or Electron. The linac name and beam energy are customized within their text box. The linac properties are populated once the linac mode, name and energy have been selected.

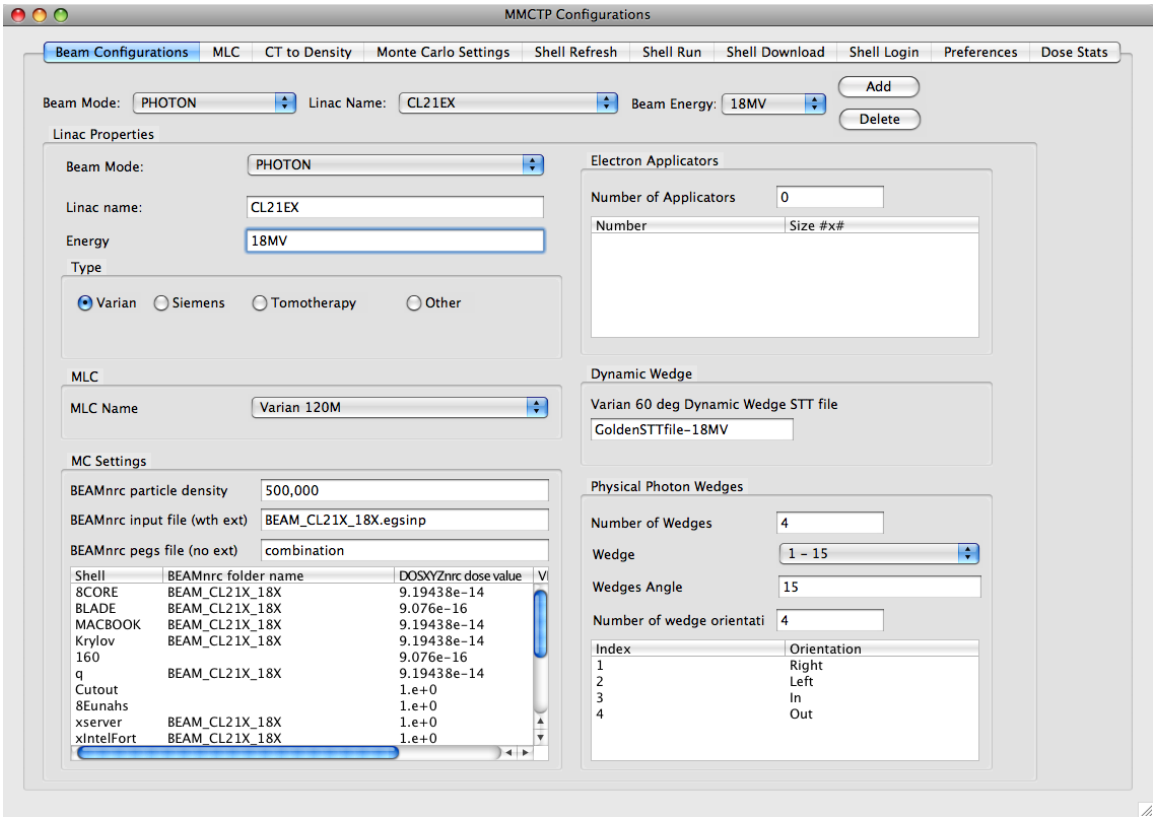


Figure 5: Beam configuration tab, creation/modification of treatment machines

Type The linac type is a flag used within the MMCTP Monte Carlo scripts to specify which method should be called in the creation of new Monte Carlo input files.

MLC The MLC group assigns a specific user defined MLC to the selected linac. The user has the option to define new MLCs within the MLC tab.

Electron Applicators This group assigns electron or any applicators to the linac. The listbox is used to pair an applicator name to a BEAMnrc applicator input file. For example, if an applicator size of 15×15 is defined and the user is running a BEAMnrc simulation with an APPLICAT CM defined

within the BEAMnrc input file, then MMCTP will update the APPLICAT CM with the applicator input file named APP-15×15.egsinp. The applicator file must only contain the APPLICAT CM text. An example applicator file:

```
********** start of CM APPLICAT with identifier applicat  **********
27, RMAX
5x5 applicator
95.01, ZBACK
3, 0, #SCRAPERS, SQUARE
63.75, 1.75, 4.84, 7.137, 0, 8
76.5, 2.025, 4.83, 4.76, 0, 8
93.02, 1.80, 2.375, 7.475, 0, 9
0.7, 0.01, 0, 8,
EAPPLICRU521
EAPPLICRU521
CERROBEND521
```

Dynamic Wedge A dynamic wedge in MMCTP is generated from a STT file. The text box contains the file name for a Varian 60 degree STT file. The STT file must be placed within the MMCTP-Settings folder.

Physical Photon Wedges This group defines the available photon wedges within MMCTP for the selected linac. Each wedge may be defined in multiple orientations. For example, if a 15 degree RIGHT wedge is defined and the user is running a BEAMnrc simulation with the WEDGE CM defined within the BEAMnrc input file, then MMCTP will update the WEDGE CM with the wedge input file named 15RIGHT.egsinp. The wedge input file must only contain the WEDGE CM text. An example wedge file:

```
********** start of CM WEDGE with identifier WEDG  **********
15.0000,
15 RIGHT
1, 1, 0,
57.6,
58.6, 0.14999,
0.7, 0.01, 0, 0, 0.0,            ECUT,PCUT etc
PMMA700ICRU
0, 0, 58.75, 63,       WDIR,WOR,WPOS(2) general wedge data
```

```
0.7, 0.01, 0, 0,            interior of CM wedge (assumed AIR)
2, 4,                       NPROF and NPOINTS for wedge geometry
-9.14, -6.83, 5.16, 9.14, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  profile points for wed
-12.65, 0.12, 0.55, 1.37, 1.42, 0.0, 0.0, 0.0, 0.0, 0.0,
12.65, 0.12, 0.55, 1.37, 1.42, 0.0, 0.0, 0.0, 0.0, 0.0,
5.5,0.01,2,9,0.0,            ECUT,PCUT etc for wedge
STEELLESS700ICRU                  steel
```

MC Settings This group defines the Monte Carlo parameters associated with the selected linac. These include the default BEAMnrc particle density, the template BEAMnrc input file, which is used in the creation of new BEAMnrc simulations, the pegs file and listbox defining the BEAMnrc accelerator folder path, DOSXYZnrc normalization value and VMC normalization value per shell.

2.2.4 MLC

The MLC tab allows the user to create delete and modify MLC properties. MMCTP is compatible with two types of MLC leaves, leaf-pairs and binary leaves. The listbox lists the boundary of each leaf-pair/binary leaf. The leaf boundary is only used in the 2D and 3D viewer to display the MLC pattern.

2.2.5 CT to Density

The CT to Density tab allows the user to create modify and delete CT curves. The CT curve associates a HU range with a material name and density range. During the construction of the egsphant file, the user has the option of using a linear CT model to generate the egsphant file from the image HUs. The user may define more than one CT curve. Each curve is defined with a unique name, pegs file and the number of linear points or material. For example, the CT curve defined in figure 7 is defined by 4 lines. As a rule, the HU range must increase from lowest (1st material) to highest (last material). In addition, the HU high value should equal the HU low value on the following line. The density values do not need to be continuous. Lastly, the material names do not need to be unique. For example, the user can define a CT curve with 4 lines and only one material.

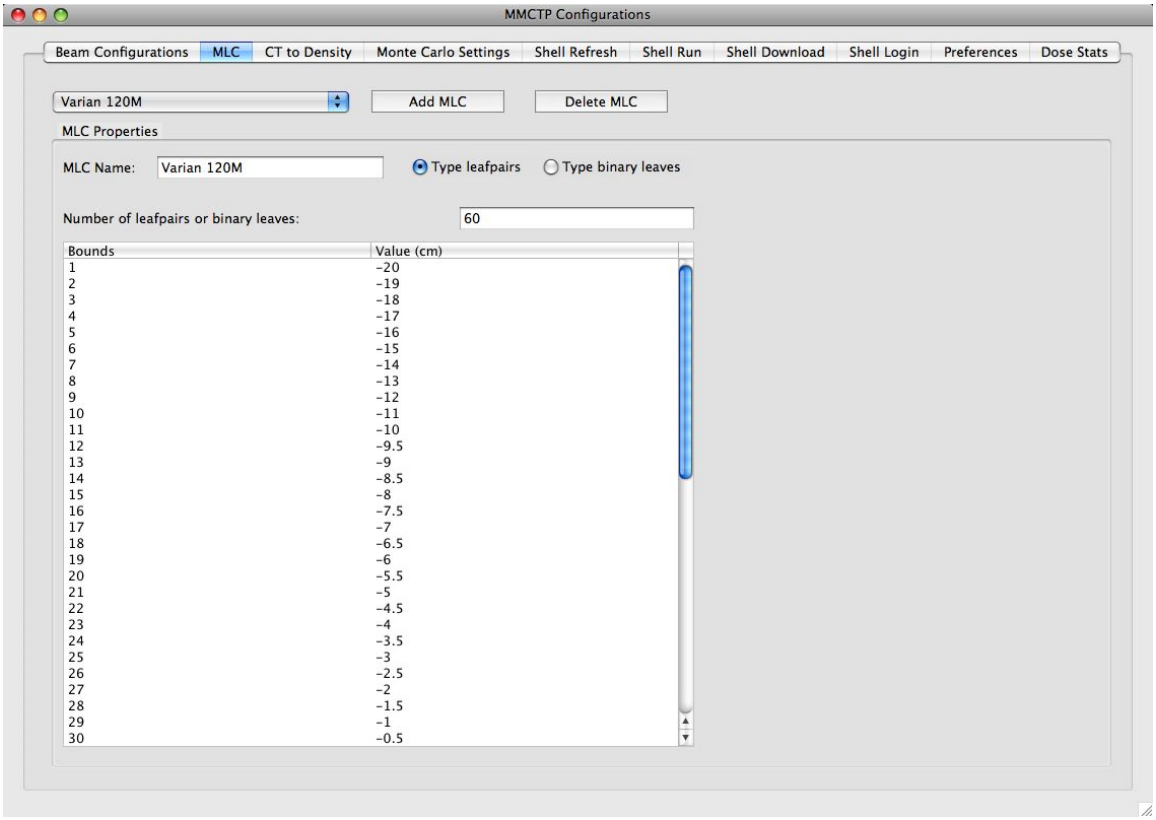


Figure 6: MLC configuration tab, creation/modification of MLC properties

2.2.6 Monte Carlo Settings

These variables are controlled within the Configuration Window, user the tab Monte Carlo Settings. This tab contains five sub groups, general MC settings, current plan settings, DOSXYZnrc settings, BEAMnrc settings and VMC settings.

General MC Settings This group includes: Auto Refresh check box, Auto Refresh period, Auto MC Run check box, Auto MC Run period. These variables control the global shell refresh and shell run processes. These variable take precedence over the Auto Run/Refresh variables for BEAMnrc/DOSXYZnrc and VMC

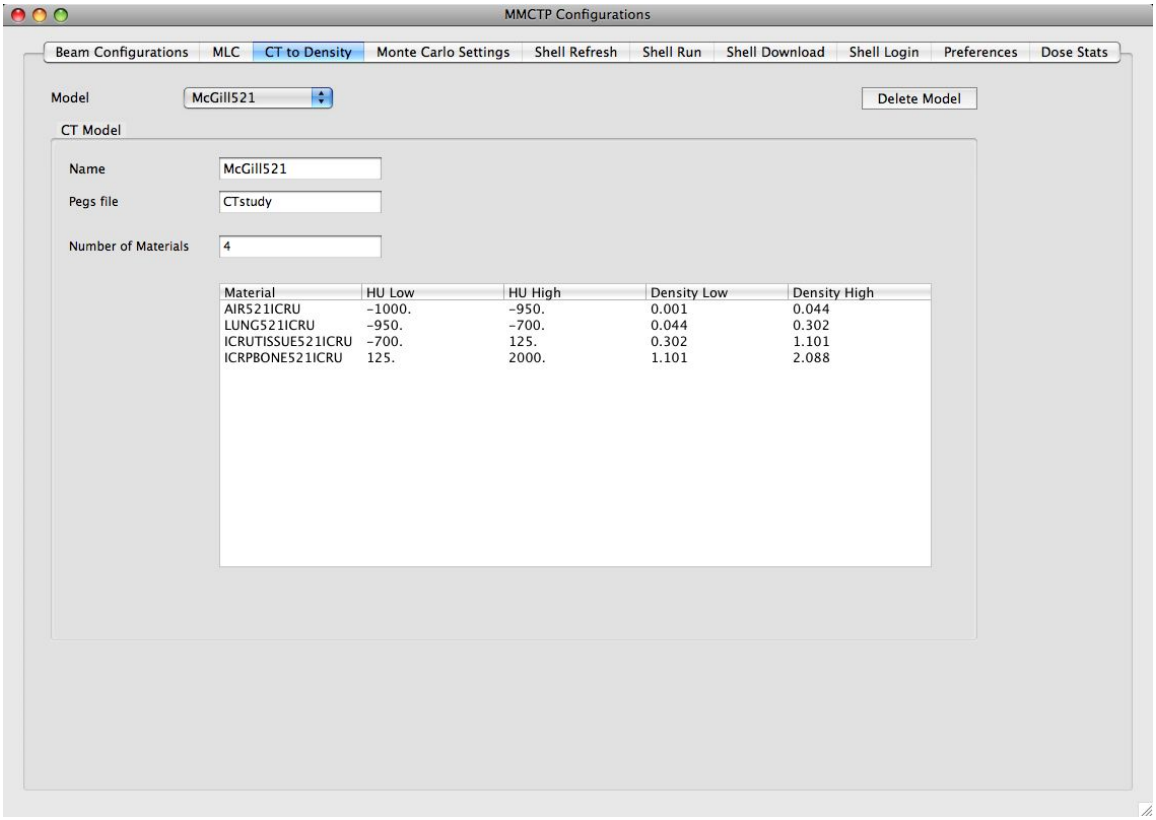


Figure 7: CT to density configuration tab, creation/modification of CT to density curves

Current Plan Settings This group summaries the MC job status of the current plan. The number of BEAMnrc, DOSXYZnrc and VMC jobs per shell and the online status of each shell is displayed in a listbox. In the event that a shell become unavailable, MMCTP will update the online status of a shell. The user has the option of re-enabling the online status.

BEAMnrc Settings This group sets the auto refresh/run values, option to remove w files after the simulation has completed, the option to automatically determine which shell to run a new BEAMnrc simulation, the option to automatically link a new BEAMnrc simulation with a pervious run, the default job splitting number and default options for IDAT and IZLAST.

The default options for IDAT and IZLAST take precedence over the options within the template input file.

DOSXYZnrc Settings This group sets the auto refresh/run values, option to link DOSXYZnrc with BEAMnrc/Cutout/none, the default average percent uncertainty, default job splitting number, the minimum job splitting number, the default number of test histories, the option to import all 3ddose files to McGill RT format, option to import the total plan 3ddose files to McGill RT, the option to load the EGSPhant voxels into memory for viewing, the option to use the MU backscatter correction for dose normalization, the option for removal of w files after the simulation has completed, the default isource number this number takes precedence over the isouce within the template input file.

In addition there are two group boxes, 1) Dosxyznrc material and density values and 2) EGSPhant file options. The user has the option of adding new materials and density values for use within MMCTP. These materials maybe used within the creation of new EGSPhant files and within the DOSXYZnrc input file.

The EGSPhant options specify the creation of a default EGSPhant file. The user may define more than one file. For each file the options include: EGSPhant name, CT Ramp index, outside material name, clean contour label, resolution in X,Y,Z and the auto make option. Auto make option is triggered once MMCTP loads a patient. A new EGSPhant file is generated if the patient dose not already include an EGSPhant file with the same name.

VMC Settings This group sets the VMC auto refresh/run values, option to automatically import all d3d files to McGill RT format, the default dose to water correction ratio, the default DMX resolution for X,Y,Z and the default DMX name and the character limit for a VMC file name.

2.3 Advanced configuration files/folders

Once the installation is complete, the user will be required to populate user specific settings, as described in section 2.2. This section describes the files/folders used to store these settings settings. Most text files stored within the MMCTP-Settings folder. In addition to the MMCTP-Setting files,

there are specific folders which are to be defined by the user to store files for BEAMnrc, DOSXYZnrc, VMC, Cutout and Commission-Data.

2.3.1 MMCTP-Settings files

This group of files include; BEAM.pref, CT_Calibration.txt, login.txt, MLC.txt, GoldenSTTfile, MMCTP-Settings.txt, and pref.txt.

External Beams - BEAM.pref, The BEAM.pref file stores variables for each linac defined within MMCTP. The user must define each linac within the Beam Configurations tab of the Configurations Window. The variables for each linac include; linac name, energy, beam mode, MLC type, wedge properties, applicator properties, BEAMnrc variables and DOSXYZnrc/VMC calibration dose. Dose distributions from VMC or DOSXYZ simulations are first normalized with a calibration factor which relates the Monte Carlo output to the dose per MU. The calibration file contains a list of linacs, their energies and the calibration factor for VMC and DOSXYZ. For more information on this section, refer to section 2.5. The structure of the BEAM.pref file is shown in figure 8.

```
PHOTON, CL21, 6MV, 120 MLC, 15(LEFT-RIGHT) , , /egsnrc/BEAM_CL21X_6/, BEAM_CL21_6X.egsinp, BLADE
PHOTON, CL21, 18MV, 120 MLC, 15(LEFT-RIGHT), , /egsnrc/BEAM_CL21X_18/, BEAM_CL21_18X.egsinp, BLADE
ELECTRON, CL21, 9MeV,  , ,20x20 , /egsnrc/BEAM_CL21X/, BEAM_CL21X_9MeV_20x20.egsinp, BLADE
```

Figure 8: BEAM.pref file links the MMCTP linacs with virtual linacs. The format is as follows; beam mode, linac name, linac energy, MLC name, wedges, applicators, location of BEAMnrc accelerator, template BEAMnrc input file.

CT calibration file The CT calibration file shown in figure 9 associates a HU range with a material name and density range. During the construction of the egsphant file, the user has the option of using a linear CT model to generate the egsphant file from the image HUs. This file can contain more than one model. Modifications to the file are made within the Configurations window under the CT to Density tab. The HU range must increase from lowest (1st material) to highest (last material).

```
Material,HU Range,Density

Model, McGill CT1
AIR521ICRU,-1000:-950,0.001:0.044
LUNG521ICRU,-950:-700,0.044:0.302
ICRUTISSUE521ICRU,-700:125,0.302:1.101
ICRPBONE521ICRU,125:2000,1.101:2.088
```

Figure 9: CT calibration file. Each model name is followed by the a list of materials, HU range and density range. A blank line is used to separate different models

Login Settings - login.txt This file stores specific variables for MMCTP to use during remote connection and job submission tasks. Individual computers (clusters) are stored as profiles within the file. These variables are edited within the Configurations window under the tab Shell Login. The login.txt file stores the login parameters such as IP address, user name, password, etc to all configured computers. These parameters must be carefully entered as there is little feed back to detect errors. For instance, the line feed values depend on the value of the command prompt. These values are used to determine when a command has finished. Thus if the line feed value is incorect, the shell could exit before the command complete or may never exit and stay within an infinite loop.

MLC file - MLC.txt, The MLC file (MLC.txt) is used to configure MLC types for visualization within the GUI. The format of the file is shown in figure 10. These values are not used for in the generation of MC simulations. MC simulations use template input files to determine MLC leaf properties. MLC properties can modified within the MLC tab of the configurations window.

Preference file - pref.txt The pref.txt file stores the folder locations for various user specific folders. These folders include the McGill RT folder, the DICOM import folder, RTOG folder, CADPlan folder, BEAMnrc folder, VMC folder, Commissioning Data folder, and McGill Cutout folder. All folder locations can be changed within the Preferences tab of the configura-

```
//MLC LEAF THICKNESS (cm)
*120
1-10  = 1
11-50 = 0.5
51-60 = 1
*52
1-26  = 1
```

Figure 10: MLC file, where MLCs are labeled by an asterisk (*) and the total number of leaves. The following lines define the leaf thickness for a group or individual leaves.

tions window by clicking on the folder path.

GoldenSTTfile Varian dynamic wedge control point file for reference field.

MMCTP-Settings.txt This file stores global variables for automatic job submission, job status update, and connectivity of remote shells. These variables are controlled within the Configuration Window, user the tab Monte Carlo Settings and include: Auto Refresh check box, Auto Refresh period, Auto MC Run check box, Auto MC Run period, and the online status of shells.

2.3.2 BEAMnrc folder

This folder stores all BEAMnrc input files. Input files are specific to an individually complied linac. MMCTP links each input file to linac as defined within the Beam Configurations tab. This folder also contains files associated to DOSXYZnrc simulations.

DOSXYZnrc Material file The material file 11 contains a list of material names and densities. During the construction of the egsphant file, these materials may be assigned to specific structure volumes. The user must manual edit the file to add or delete materials.

DOSXYZnrc DOSXYZ.egsinp file DOSXYZ input files are build in-part by using a DOSXYZ template file. This file provides MMCTP with the

```
Material,Density
AIR521ICRU,0.0012
H2O521ICRU,1
LUNG521ICRU,2.6000E-01
PMMA521ICRU,1.190
POLYETH521ICRU,9.3000E-01
POLYSTY521ICRU,1.0600E+00
BRAIN521ICRU,1.0400E+00
```

Figure 11: DOSXYZnrc materials file

structure of DOSXYZ input files and sets a standard for DOSXYZ simulations. Before each simulation, MMCTP complements the template file with beam specific properties. Adjustments to specific DOSXYZ inputs can be made prior to a simulation.

DOSXYZnrc MC_WaterPhantom_Output input file DOSXYZ input files are build in-part by using a DOSXYZ template file. This file provides MMCTP with the structure of non-CT based DOSXYZ input files and sets a standard for DOSXYZ simulations. Before each simulation, MMCTP complements the template file with beam specific properties. Adjustments to specific DOSXYZ inputs can be made prior to a simulation.

2.3.3 VMC folder

MMCTP requires a template VMC input file named vmc.inp for VMC simulations. Patient specific VMC input files are build in-part by using the VMC template file. An example of the file is shown in figure 12.

2.4 Virtual Cluster Configurations

The design philosophy behind MMCTP was to separate the resource intensive Monte Carlo calculations from the MMCTP workstation. From the MMCTP workstation, the user may create, transfer and monitor MC simulations on remote clusters or cloud computers. A remote cluster must comply with the following list :

```
*GLOBAL-DATA     |
-WRITE-3D-DOSE   | 1
-NORM-TYPE       | 1
-RANDOM-SET      | 23  45  67  89
!
*BEAM-PARAMETERS|
-BEAM-WEIGHT     | 100.0
-DEVICE-TYPE     |
-DEVICE-KEY      |
-EVENT-NUMBER    |
-ISOCENTER-DIST  |
-ISOCENTER       |
-CHANGE-SAD      | 70
-GANTRY-ANGLE    |
-TABLE-ANGLE     |
-COLL-ANGLE      |
!
*END-INPUT       |
```

Figure 12: Template VMC input file

Operating System: Linux, Unix or Mac OS X.

Connection Protocols: SSH and SFTP or SCP or FTP connection must be enabled.

Queueing System: For batch submissions, MMCTP has been configured for "at", keg, MOAB and NQS. Additional batch systems may be included in future versions.

MC Codes: Successful installation of EGSnrc, BEAMnrc, DOSXYZnrc and XVMC with environment variables.

2.4.1 EGSnrc & BEAMnrc Installation

Current version of MMCTP is compatible with 2008 and above release of EGSnrc and BEAMnrc. Please refer to the NRC web page for detailed EGS and BEAMnrc installation instructions.

DOSXYZnrc DOSXYZnrc must be compiled with the following options. MMCTP uses the Y axis within DOSXYZnrc as the number of slices, and the Z axis as the number of rows. This requires the user to change the default DOSXYZnrc settings for KMAX from 100 to 512 and JMAX may be decreased from 512 to roughly 120. These changes are made within the dosxyznrc_user_macros.mortran file located within the DOSXYZnrc folder.

```
REPLACE {$IMAX}    WITH {512} "Maximum number of x cells
REPLACE {$JMAX}    WITH {120} "Maximum number of y cells
REPLACE {$KMAX}    WITH {512} "Maximum number of z cells
```

Mac OS X (Intel) EGSnrc and BEAMnrc installation requires, xcode, g77 (g77-intel-bin.tar.gz), qt (qt-mac-free-3.3.7.tar) and motif (motif-bin.tar.gz). There files are freely distributed. However, the installation of motif is not straight forward. DOSXYZnrc must be compiled with the following options, make FOPT="-O1 -fforce-address -ffast-math -funroll-all-loops -fexpensive-optimizations". OS X natively supports the "at" queueing system.

2.5 Normalization of MC Dose

For comparison purposes, the dose stated from Monte Carlo calculations should be in the same format, dose per MU, as conventional treatment planning systems. A calibration factor is needed to relate the Monte Carlo dose to the dose per MU. Clinical accelerators at McGill are calibrated to deliver 101 cGy in tissue at the calibration depth, z_{max} from 100 MUs. While the dose reported by Monte Carlo calculations reports dose per incident particle from the source. The calibration factor is determined by performing a Monte Carlo simulation in water under the clinical calibration conditions. These conditions are; 10×10 cm^2 field size, $SSD = 100$ cm, depth at z_{max}. The Monte Carlo calibration reference dose to water per particle is written as:

$$[\frac{D}{particle}]_{calibration} = \frac{D_w^{MC}(10 \times 10, SSD = 100, z_{max})}{particle}. \quad (1)$$

Finally, the patient dose $[\frac{D}{particle}]_{patient}$ from Monte Carlo calculations can be expressed in terms of absorbed dose per monitor unit as follows:

$$\frac{D_{patient}(\text{cGy})}{\text{MU}} = \frac{[\frac{D}{particle}]_{patient}}{[\frac{D}{particle}]_{calibration}} \times \frac{1.01}{1\text{MU}}. \quad (2)$$

This equation ignores the known effects of backscattering into the monitor chamber for Varian machines at small field sizes [9] [3]. This influences the output of the accelerator and has to be taken into account when output factors are derived from Monte Carlo simulations. Published studies examined the influence of backscatter to the monitor chamber. One backscatter graph for a 10 MV photon beam is shown in Figure 13. It is concluded that these effects are only significant for photon beams with small field sizes [9]. Since patient treatment plans use large MLC-shaped fields, the backscattered monitor chamber fluence can be ignored and Equation 2 is valid.

It is important to realize that the calibration dose per particle changes with beam energy. A particle form a 18 MV beam will have a higher dose per incident particle than a particle from a lower energy. Thus, there is a calibration dose per particle for each beam energy. Equation 2 then transforms the dose per particle from Monte Carlo into absolute absorbed dose. With the dose distribution in units of absolute dose, one can sum dose distributions from various beam energies.

2.6 MMCTP Shell

The MMCTP shell was designed to establish connections to remote computers. These connections are needed to preform tasks such as: reading files, deleting files, running programs, and transferring files. The status of the shell can be viewed within the Configurations window under Shell Refresh, Shell Run, Shell Download. All shell connections follow the same process: connection attempt, password, set environment, change directory, command. Please note, a shell connection can only proceed if the shell login online settings is enabled.

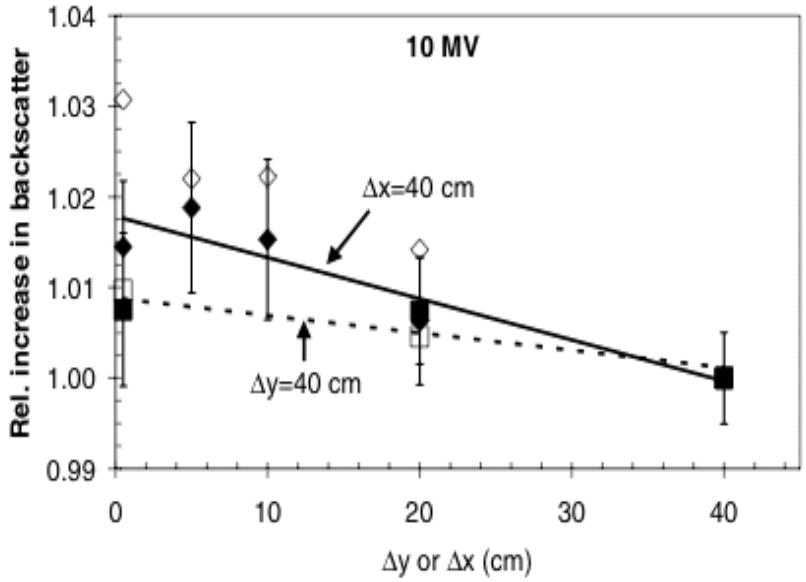


Figure 13: The relative increase in backscatter with decreasing field size for rectangular fields, with respect to a 40×40 cm^2 field for 10 MV photon beams. Results are given for measurements (filled) and Monte Carlo simulations (open) for lower X jaws (squares) fixed at a 40 cm field and upper Y jaws (diamonds) fixed at 40 cm [9]

Shell Output Each time the shell is used to connect to a remote connection, the output is displayed within the shell text box. This is a feed back mechanism to provide information to the user about what the shell is currently doing. Errors can be identified by examining the shell output.

Restart Shell Task The user has the option of restarting a shell task which has hung.

Kill Shell The kill shell button will close any open connection. If the shell can not complete its task, there is no way to automatically close the connection. The shell could be stuck in an infinite loop. The kill shell button was designed to terminate the shell under this situation. At times it may be difficult to determine if the shell is trapped within an infinite loop,

especially if the shell task is to transfer files (input files, dmx files, egsphant files). Examining the shell output should clarify when the shell is trapped or transferring a file over a slow connection.

2.6.1 Shell Run

Shell run is triggered for uploading input files, running test simulations, and submitting full simulations to the queueing system. This shell maybe triggered by the user or automatically, if the Auto MC Run check box is enabled.

2.6.2 Shell Refresh

Shell Refresh is triggered for reading lock files, reading the output of BEAMDP, checking for the existence 3ddose or d3d files. These checks are periodic, with a user defined interval time. The interval time is set within the Configuration Window, user the tab Monte Carlo Settings, Auto Refresh text box. The shell is only triggered if the Auto Refresh check box is enabled.

2.6.3 Shell Download

Shell Download is triggered for downloading 3ddose or d3d files and log-files. The shell is automatically triggered by the return status of Shell Refresh.

3 McGill RT format

A file format called McGill RT was designed for saving patient plans on the MMCTP workstation. The file system resembles the RTOG format but is more suited towards a dynamic patient data set. The RTOG format was designed for exporting and importing patient plans. Therefore, there is no simple method for saving small changes. Within the RTOG format, save changes require a complete rewrite of all patient data. The McGill RT format was developed to minimize redundant information and minimize the number of files edited for save changes. Figure 14 shows a schematic diagram of McGill RT file format. The format uses a combination of folders and files which are either binary or text. Images and dose distributions are stored as binary files while structures and beam information is stored as a text file. The file structure of the McGill RT format contains a main McGill folder, under which lie folders for each patient. The patient folders are titled with the patient name and ID number. Within a patient's folder there can be multiple image sets (CT, MRI, ultrasound (US)) each set with it's own folder. Under each image set, there are folders for multiple plans as well as a folder for the images and structures called McGill RT. The plan folders store beams, doses, and Monte Carlo control files.

3.1 McGill RT folder

3.1.1 Image files

Images are stored individually under the McGill RT folder. The image files are named sequentially with the file type as *.img*. The files are binary, containing a slice position and the pixel data. In order to properly read these files, there is an image properties text file called *RT.dir* as shown in Figure 15.

The properties file stores properties for reading and displaying the images. The dimension size and the bytes per pixel are essential for correctly reading a binary image file. The (X, Y) origin of the patient coordinate system (Figure 16) is at the exact centre of the top left pixel. The X and Y offsets, stated in cm, permit a displacement from the origin.

In order to properly visualize the image, the grid units are used to adjust the image dimension. These units relate the pixels to a physical dimension for proper visualization scaling. The grid unit states the width and height in

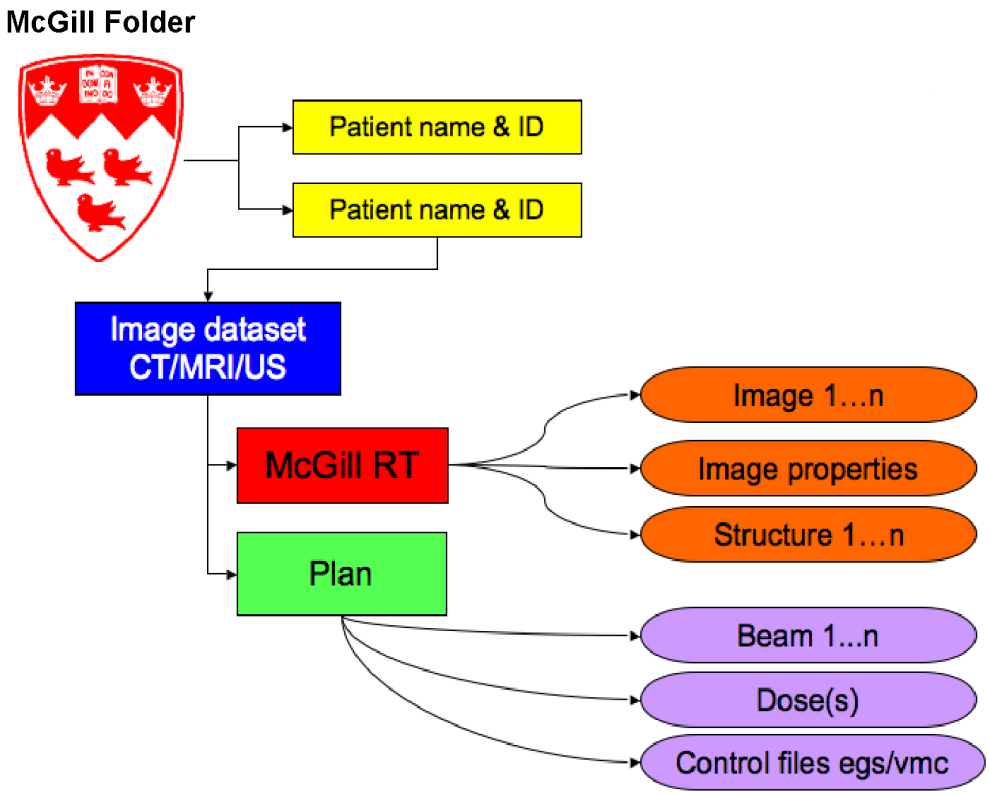


Figure 14: McGill RT file structure, folders and files

Patient Name	:= (string)
Patient ID	:= (integer)
IMAGE MODALITY	:= (string)
IMAGE TYPE	:= (string)
SCAN TYPE	:= (string)
CT offset	:= (integer)
Grid unit width	:= (double)
Grid unit height	:= (double)
Number representation	:= (string)
Bytes per pixel	:= (integer)
Size of Dimensions 1	:= (integer)
Size of Dimensions 2	:= (integer)
X offset	:= (integer)
Y offset	:= (integer)
CT-AIR	:= (integer)
CT-WATER	:= (integer)
Slice thickness	:= (cm)
Date	:= (string)
WINDOW LEVEL	:= (integer)
WINDOW WIDTH	:= (integer)

Figure 15: McGill RT image properties file: properties and their variable type.

cm of one pixel.

After reading the properties file, MMCTP has all the information required to read in each *.img* file. Within each image file, the first 4 bytes represent the Z image position followed by the pixel data. The pixel data is read in row by row with the first pixel corresponding to the top left corner of the image.

3.1.2 Structure files

The structures are a sequence of three-dimensional coordinates which define a volume of interest. A volume can include target volumes and organs at risk. The coordinates are grouped together in planes, which coincide with planes on the CT image slices. A given structure does not have to be defined

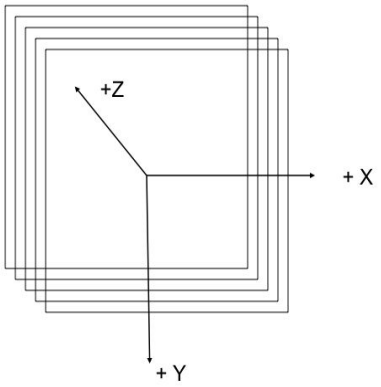


Figure 16: McGill RT patient coordinate system: The (X, Y) origin is located at the exact centre of the top left pixel. The positive X, Y directions are indicated. The $+Z$ direction goes into the page.

on all planes. Within a given plane, a structure may consist of one or more segments. Where each segment is a sequence of at least four points which define a closed curve on the surface of the volume. The structure files are text files named sequentially with the file type *.struct*. These files are found under the McGill RT folder because they are associated to the image files. Each structure file contains a header section followed by the structure coordinates. The specific format of these files is shown in Figure 17.

The header information contains the structure name, colour and number of scans. The number representation and format are always character and scan-based. The structure points follow the header information. The structure points are arranged in order per scan number. Each scan number is listed followed by the number of segments per scan. If there exits a segment, the next line reports the number of points, and the subsequent lines report the points. The coordinate points are in centimeters relative to the patient coordinate system.

STRUCTURE NAME	:= PTV2
NUMBER REPRESENTATION	:= CHARACTER
STRUCTURE FORMAT	:= SCAN-BASED
STRUCTURE COLOUR RGB	:= 252/18/41
NUMBER OF SCANS	:= 3
"NUMBER OF LEVELS" 3	(total number of scans)
"SCAN NUMBER" 1	(=1 for first scan, etc)
"# OF SEGMENTS" 0	(number of segments in this level/scan)
"SCAN NUMBER" 2	
"# OF SEGMENTS" 1	
"# OF POINTS" 4	(number of points in first segment)
0, 1.459, 20.86305	(X, Y, Z coordinates of each point)
-0.56, 1.387, 20.86305	
-0.776, 1.387, 20.86305	
-0.848, 1.315, 20.86305	
"SCAN NUMBER" 3	
"# OF SEGMENTS" 0	

Figure 17: McGill RT structure file: Example structure file, header information followed by struture points.

3.2 Plan folder

Patient plans are assigned folders within an image set folder. The files found within a patient plan folder include:

- Beam files
- MLC files
- Dose distribution files
- Monte Carlo control files

The following sections describe these files.

3.2.1 Beam Geometry files

A beam geometry file contains the information defining an external radiation beam. These include various properties such as: treatment unit, beam energy, beam applicators, number of MUs, number of fractions, aperture type, wedge angle and orientation, collimator gantry and couch angle, isocentre distance, isocentre coordinates (x, y, z), and X, Y jaw positions. Each beam has it's own beam geometry file with *.Beam* as the file type. The specific format of these files is shown in Figure 18

The MLC leaf positions are not included within the beam file. Instead, there is a MLC file (shown in Figure 19) which contains the MLC fields for each beam of a plan. The MLC pattern maybe static, dynamic or step and shoot. For MLC patterns with multiple fields, an index value is used to determine the relative weight of each field. The MLC field index for field i is a number in the range [0,1], where $index(i) > index(i-1)$. This means that $[index(i) - index(i-1)]$ is the probability of field i.

3.2.2 Dose Distribution files

The dose distribution is a matrix of dose values at one or more points throughout a patient. Typically, the distribution is calculated on a three dimensional grid. The format allows for a regular spaced grid, one in which a two dimensional array of points is defined on one or more evenly spaced parallel planes. This format is ideal for the computation of dose on a two-dimensional array of points on each CT scan.

The coordinate system for the array of dose points is defined with the patient coordinate system. Within one plane, a two-dimensional array of points is defined with the x, y position of the top left hand corner point, the number of dimensions in x, y and the grid spacing in x, y. Each axial plane is identified with its z position, which normally corresponds to that of the axial CT image. The dose data is typically in units of Grays.

The dose files are written in binary format with extension *.dose*. Each file contains a header block followed by the dose distribution matrix. The header contains the properties of the dose distribution, which are written as single values. These properties are shown in Figure 20. The dose distribution begins at the 100th byte where dose values are written as singles (4 bytes) per axial plane.

BEAM #	:= (beam number in plan)
TREATMENT UNIT	:= (treatment unit name ex CL21EXA)
BEAM MODALITY	:= (electron, photon)
BEAM ENERGY	:= (energy ex 18 MV)
BEAM APPLICATOR	:= (for electrons only)
BEAM DESCRIPTION	:= (text description of beam)
RX DOSE PER TX (GY)	:= (fraction dose)
MUs	:= (number of MUs)
NUMBER OF TX	:= (number of fractions)
FRACTION GROUP ID	:= (id to group beams of common fraction)
BEAM TYPE	:= (static or arc)
COLLIMATOR TYPE	:= (symmetric or asymmetric)
APERTURE TYPE	:= (block or MLC)
WEDGE ANGLE	:= (wedge angle in degrees ex 15)
WEDGE ORIENTATION	:= (wedge orientation ex in,out,left,right)
WEDGE DYNAMIC	:= (dynamic wedge boolean ex true/false)
COLLIMATOR ANGLE	:= (angle in degrees)
GANTRY ANGLE	:= (angle in degrees)
COUCH ANGLE	:= (angle in degrees)
NOMINAL ISOCENTER DIST	:= (isocentre distance in cm)
NUMBER REPRESENTATION	:= (character)
PLAN ID OF ORIGIN	:= (plan ID of beam for grouping beams)

Figure 18: McGill RT beam geometry file

Treatment	:= STATIC
Number of Fields	:= 1
Tolerance	:=
ModelName	:= Varian 120M
ModelType	:= 0
LeafDirection	:= 0
Field	:= 1
Index	:= 100.
Carriage Group	:=
Operator	:=
Collimator	:=
Leaf 1A	:= 0.
Leaf 2A	:= 0.
Leaf 3A	:= 0.
Leaf 4A	:= 0.
Leaf 5A	:= 0.
Leaf 6A	:= 1.089
Leaf 7A	:= 1.785
Leaf 8A	:= 2.164
Leaf 9A	:= 2.384
Leaf 10A	:= 2.487
.	
.	
.	

Figure 19: McGill RT MLC position file

Coordinate X 1st point	:= (cm)
Coordinate Y 1st point	:= (cm)
Coordinate Z 1st point	:= (cm)
Size of Dimension 1	:= (# horizontal points)
Size of Dimension 2	:= (# vertical points)
Size of Dimension 3	:= (# planes)
Horizontal Grid	:= (cm >0)
Vertical Grid	:= (cm >0)
Depth Grid	:= (cm >0)
Dmax	:= (Gy)
Dmin	:= (Gy)
Dose Units	:= (string)

Figure 20: McGill RT dose header block

3.2.3 Monte Carlo control files

Aside from the input files required to run BEAMnrc, DOSXYZnrc and VMC, there are additional files created called *BEAM.txt*, *DOSXYZ.txt* and *VMC.txt* to track the Monte Carlo progress for all beams of a plan and store simulation properties specific to each beam.

BEAM.txt This file includes the job split number, number of histories, queue type, PEGS file and the simulation status. The job split number has no effect on the simulation result but will greatly affect the simulation time. Computer clusters split jobs to multiple processors for quick calculations. The split number is the number of processors utilized for one simulation. The simulation status stores the percent progress of each beam simulation. This file also includes properties of the simulation which include the CPU time, the desired number of particles in the scoring plane, phsp size, number of phsp particles, number of phsp photons, and the phsp name.

DOSXYZ.txt For each egsphant file, named x, there is a DOSXYZ_x.txt file which stores the properties for all simulations. These properties include: PEGS File, Dose to Water, Dose, Progress, Queue, Number of Histories, Number of Jobs and CPU time per History.

VMC.txt For each DMX file, named x, there is a VMC_x.txt file which stores the properties for all simulations. These properties include: Dose to Water, Dose, Progress, Queue and Number of Histories.

4 Importing Data

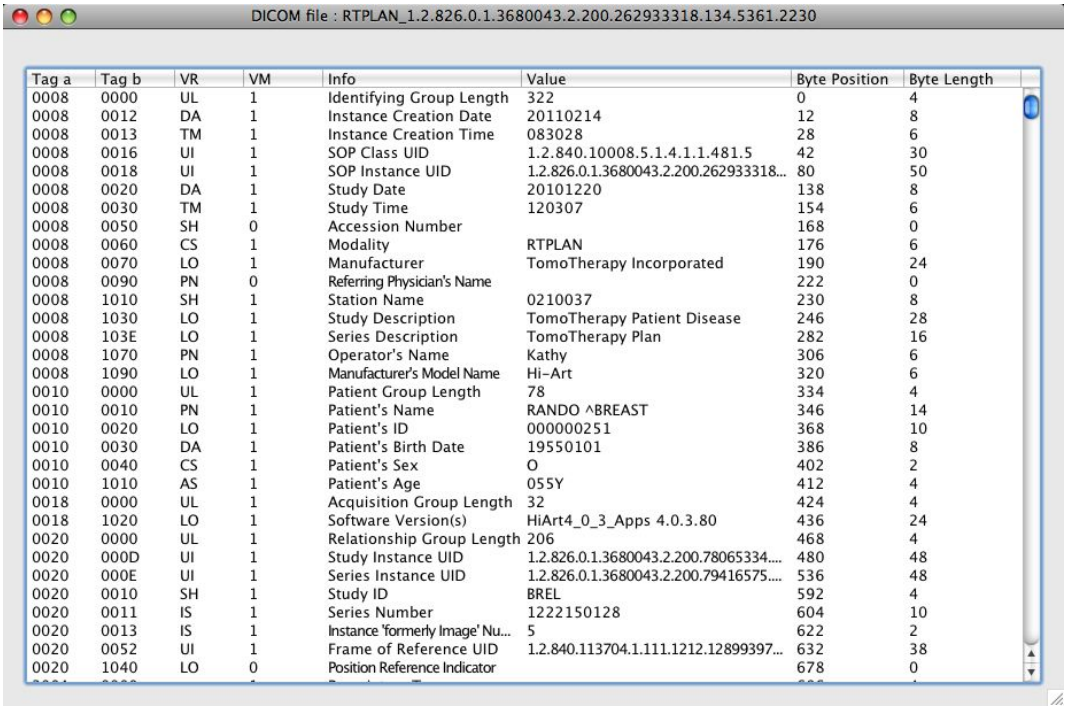
MMCTP uses an internal file format called McGill RT for reading and writing data. The import conversion module reads patient information from standard formats (DICOM_RT, RTOG, CADPlan) and rewrites this information into the McGill RT format. Converting patient information to McGill RT format is the first step in using MMCTP.

4.1 DICOM Import

DICOM import is the preferred method of importing data into the McGill RT classes. MMCTP will import images, plans, beams and doses. The DICOM filter may not read-in all types of DICOM files. However, every effort has been made to create a filter as universal as possible. MMCTP is capable of importing four types of DICOM files: RT images, RT structures, RT plans and RT doses.

DICOM import was the last import format added to MMCTP. DICOM import is split into three methods: DICOM to text parser, DICOM class reader, DICOM to McGill RT converter. The DICOM parser was written to read all types of DICOM files. The parser populates an array of DICOM elements. Each element contains two tags, element type (VR), size and data set value. The complication with DICOM is that data sets can contain other nested data sets which are encoded as sequences. In radiotherapy, these nested data sets can reach five or six layers. Elements contain a layer variable which defines the sequence level of each element. Upon completion of the DICOM parser these elements are viewable within the MMCTP DICOM listbox viewer. The first section of a Tomotherapy DICOM plan file is shown in figure 21.

The DICOM class reader filters the generalized element parser into specific radiotherapy classes. MMCTP contains four DICOM RT classes: image, structure, plan and dose. Each class contains variables pertinent to the function of each type. Sequences and nested sequences are contained within their own class. With a total of twenty seven unique classes, MMCTP is able to store DICOM RT data into native DICOM class structures. These DICOM classes greatly simplified the transformation of data from DICOM to McGill RT. For example there are two common DICOM techniques to define the Z slice thickness and Z slice position for axial dose planes. A serial DICOM reader and transcriber would assume the first read technique to be correct.



Tag a	Tag b	VR	VM	Info	Value	Byte Position	Byte Length
0008	0000	UL	1	Identifying Group Length	322	0	4
0008	0012	DA	1	Instance Creation Date	20110214	12	8
0008	0013	TM	1	Instance Creation Time	083028	28	6
0008	0016	UI	1	SOP Class UID	1.2.840.10008.5.1.4.1.1.481.5	42	30
0008	0018	UI	1	SOP Instance UID	1.2.826.0.1.3680043.2.200.262933318...	80	50
0008	0020	DA	1	Study Date	20101220	138	8
0008	0030	TM	1	Study Time	120307	154	6
0008	0050	SH	0	Accession Number		168	0
0008	0060	CS	1	Modality	RTPLAN	176	6
0008	0070	LO	1	Manufacturer	TomoTherapy Incorporated	190	24
0008	0090	PN	0	Referring Physician's Name		222	0
0008	1010	SH	1	Station Name	0210037	230	8
0008	1030	LO	1	Study Description	TomoTherapy Patient Disease	246	28
0008	103E	LO	1	Series Description	TomoTherapy Plan	282	16
0008	1070	PN	1	Operator's Name	Kathy	306	6
0008	1090	LO	1	Manufacturer's Model Name	Hi-Art	320	6
0010	0000	UL	1	Patient Group Length	78	334	4
0010	0010	PN	1	Patient's Name	RANDO ^BREAST	346	14
0010	0020	LO	1	Patient's ID	000000251	368	10
0010	0030	DA	1	Patient's Birth Date	19550101	386	8
0010	0040	CS	1	Patient's Sex	O	402	2
0010	1010	AS	1	Patient's Age	055Y	412	4
0018	0000	UL	1	Acquisition Group Length	32	424	4
0018	1020	LO	1	Software Version(s)	HiArt4_0_3_Apps 4.0.3.80	436	24
0020	0000	UL	1	Relationship Group Length	206	468	4
0020	000D	UI	1	Study Instance UID	1.2.826.0.1.3680043.2.200.78065334....	480	48
0020	000E	UI	1	Series Instance UID	1.2.826.0.1.3680043.2.200.79416575....	536	48
0020	0010	SH	1	Study ID	BREL	592	4
0020	0011	IS	1	Series Number	1222150128	604	10
0020	0013	IS	1	Instance 'formerly Image' Nu...	5	622	2
0020	0052	UI	1	Frame of Reference UID	1.2.840.113704.1.111.1212.12899397...	632	38
0020	1040	LO	0	Position Reference Indicator		678	0

Figure 21: MMCTP DICOM element viewer.

With the use of a DICOM class, interchangeable elements can be evaluated simultaneously to determine which is the correct Z slice thickness. In addition, DICOM classes also simplify the transformation of McGill RT data back into DICOM format. This transformation is split into three steps: creation of DICOM class from McGill RT, creation of DICOM elements from DICOM class, writing binary DICOM elements to disk which is universal to all classes. DICOM dose export is currently the only DICOM export filter within MMCTP.

4.1.1 DICOM Images

MMCTP will filter out all non-axial images, and DDR images. Images are sorted by Z (slice) value. The DICOM X, Y image offsets are used to define the MMCTP patient coordinate system.

4.1.2 DICOM RT Structures

The RT structures are imported into MMCTP using the image X, Y offsets, in addition to the DICOM image cosine direction. The structure set is sorted by Z (slice) value.

4.1.3 DICOM RT Plans

The plan file contains information such as the plan name, study name, plan beams and the reference structure set. The reference structure set, together with the patient ID is used to group a plan to a specific set of images and structures.

Beams include the following parameters: MLC fields (static/dynamic), electron cutouts, wedges (static/dynamic), applicators, jaw positions, beam mode, linac name, beam energy, SSD, isocenter, MUs, number of fractions.

MLC properties are defined within the beam limiting properties of the plan file. In addition to the MLC fields, MMCTP will attempt to import these properties if they are not already defined within the MLC.txt file.

4.1.4 DICOM RT Dose

Dose distributions are imported to McGill RT plans using the patient ID and DICOM ReferencedSOPInstanceUID. Distributions are sorted by Z (slice) value.

4.2 RTOG Import

This import includes images, structures, plan information and dose distributions. Due to the limited use of RTOG, the import filter is not complete and may lead to incomplete data.

4.3 CADPlan Import

This import includes images, structures and dose distributions.

4.4 Import Window

The import window, shown in figure 22, allows the user to transfer patient data into the McGill RT format. A pulldown menu selects the importing format which can be RTOG, DICOM or CADPlan. Once the format is selected, the program will scan the specific folder associated to that format and populate the import patient listbox with patient information. The import folder path may be changed by clicking on the path (located at the bottom of the import listbox) and choosing a new folder. This change is then saved in the preference file. Once the patient is displayed in the import listbox, the user must select the patient and press the transfer button. The program will then read in the patient information and rewrite in McGill RT format. Upon completion, the patient will be listed in the McGill listbox.

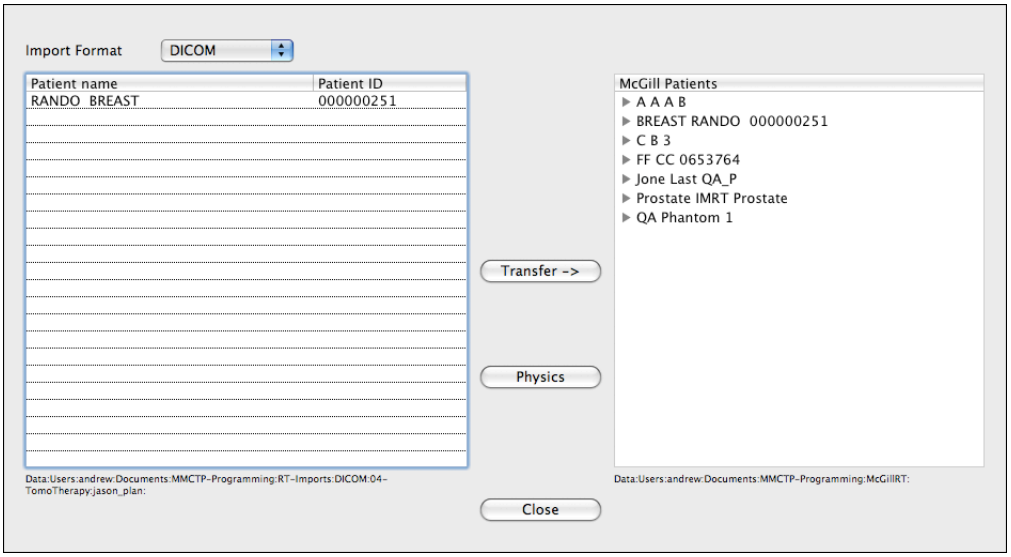


Figure 22: Import window, pulldown window selects the import format. Left listbox displays the available import patient, and right listbox displays the imported patients

It is useful to organize your import patients in folders. This will minimize the time required to scan an import folder and the time required to filter the import patient.

The physics option allows the user to generate a uniform CT number dataset. The user can choose the number and size of voxels per direction.

5 Exporting Data

Various types of data can be exported from MMCTP, precluding a complete patient data-set export. The main exporting options are accessible within the File menu.

5.1 DVH Export

DVHs calculated within MMCTP are saved within the McGill RT folder as text files. There is an option within the DVH Window to export these files as text or as xmgr text. These file can be imported into Excel, Matlab or Grace for analysis or plotting purposes.

5.2 DICOM RT Dose Export

There is an option with the treatment planning window to export a selected MMCTP dose distribution to DICOM RT Dose. The DICOM export filter within MMCTP may-not be universally accepted by all treatment planning systems. The filter has been successfully tested with the Eclipse treatment planning system.

5.3 Plane Export

There is an option with the treatment planning window to export a selected dose plane as a text file. MMCTP will export the selected plane as defined within the top canvas of the treatment planning window. This file can be imported into Excel, Matlab or Grace for analysis or plotting purposes.

5.4 Profile Export

There are two types of profiles within MMCTP. 1) Dose profiles generated by the ruler within the treatment planning window. 2) Commissioning profiles, generated from imported measurement data or non-CT defined 3ddose files.

5.5 Dose Statistics

These points are defined within the Configuration window and are used in the creation of a dose volume index file. These files can be imported into Excel

or Matlab for efficient analysis of dose volume trends over a large patient data-set.

6 Treatment Planning Window

The Treatment Planning window is the main program window that opens once a patient has been loaded. The window contains three canvas plots, a listbox for patient plans and dose distributions and a tab menu for specific tasks. By default it will open with the "External Beams" tab active.

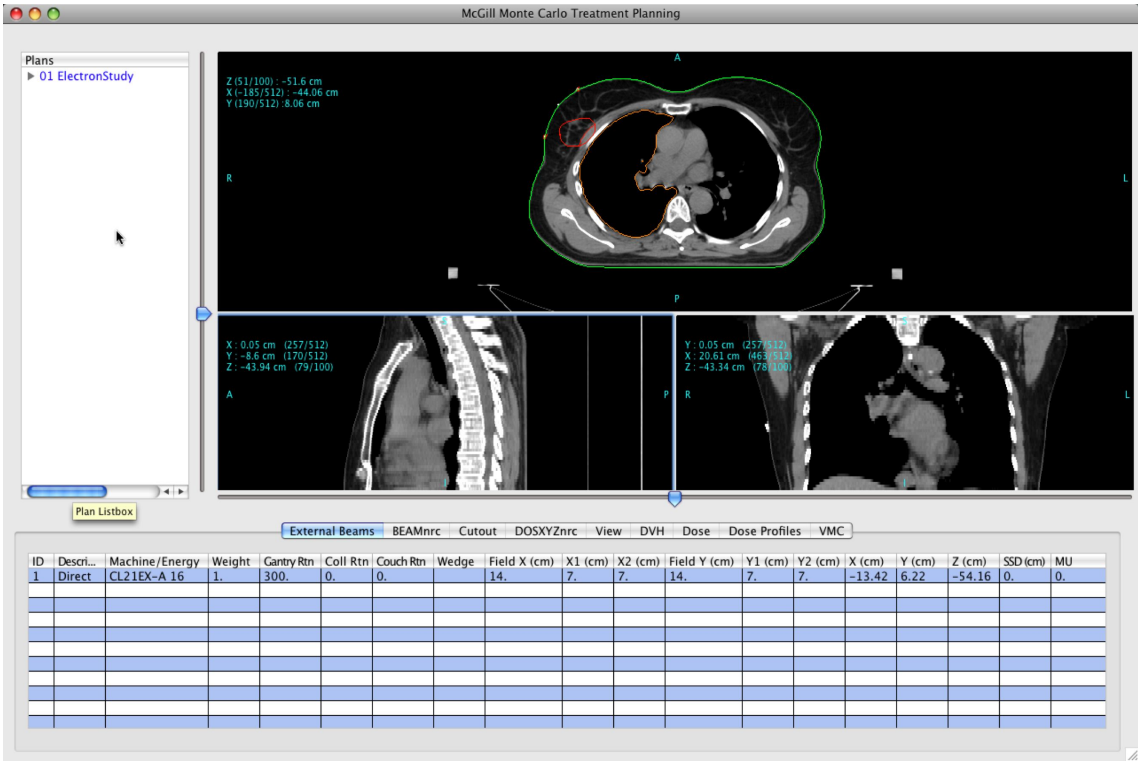


Figure 23: Treatment Planning window. Plans are listed within the planning listbox along the left side of the window. Three main canvases, Top, Left, Right. The tab menu runs below the canvases, for selecting various features. The system opens with the "External Beams" active.

6.1 Canvases

Each of the three canvas plots (Top, Left, Right) are capable of displaying patient images, structures and dose distributions in axial, saggital and coronal views, respectively. Within one canvas, views are changed by right clicking on the image and selecting a different view. The (x,y,z) coordinates of each mouse click are displayed at the top right of each canvas. The images can be changed by using the right and left arrow keys, or by using the crosshairs. Assuming the patient plan has a beam defined, the isocenter is displayed as a red cross. If a dose distribution is selected in the planning list box on the left, the maximum, minimum and dose point are displayed in the top left corner. The uncertainty will also be displayed and is only relevant in case the distribution viewed is an MC distribution. The dose point is updated each time the user clicks a new point. Patient information as present in the DICOM files, is indicated as well. What is shown in the canvases depends on the features turned on or off in the different Tabs. This will be discussed below (see section 6.3).

Independent from this is the Panning feature of the canvases. To Pan, select an image, by clicking within the canvases. Holding down on the space bar key and left mouse button while dragging the mouse will pan the image. The image can be re-centered by right clicking on the canvas.

6.2 Plan Listbox

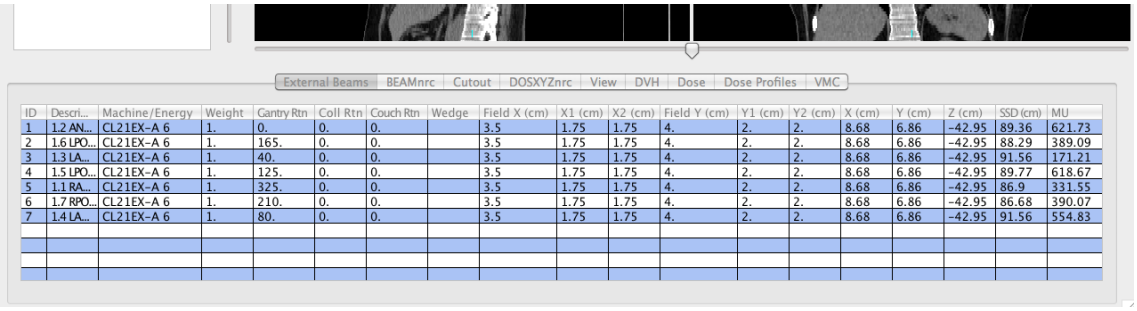
The plan listbox displays the plans associated to a patient. Plans are separated by the plan number and the plan name. The plan number must be unique as this number is used in the generation of MC input files. Dose distributions are listed within their corresponding plan. Once the user selects a plan, or dose distribution, the canvases and tab menu is updated for the specific plan. Plans and dose distributions can be changed with a mouse click only if the BEAMnrc and DOSXYZnrc shell scripts, that may be running in the background, are inactive. Right clicking on a plan or dose distribution, enables a list of options which include: adding or renaming plans or dose distributions. To delete a dose distribution or plan simply press the delete button and confirm you really want to delete the distribution or plan.

6.3 Tab Menu

The tab menu runs along the bottom of the treatment planning window and contains 90% of all MMCTP features. BEAMnrc and DOSXYZnrc are discussed in section 7.

6.3.1 External Beams Tab

The external beam listbox (Fig. 24) displays the basic beam properties for



ID	Descri...	Machine/Energy	Weight	Gantry Rtn	Coll Rtn	Couch Rtn	Wedge	Field X (cm)	X1 (cm)	X2 (cm)	Field Y (cm)	Y1 (cm)	Y2 (cm)	X (cm)	Y (cm)	Z (cm)	SSD (cm)	MU
1	1.2 AN...	CL21EX-A 6	1.	0.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	89.36	621.73
2	1.6 LPO...	CL21EX-A 6	1.	165.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	88.29	389.09
3	1.3 LA...	CL21EX-A 6	1.	40.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	91.56	171.21
4	1.5 LPO...	CL21EX-A 6	1.	125.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	89.77	618.67
5	1.1 RA...	CL21EX-A 6	1.	325.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	86.9	331.55
6	1.7 RPO...	CL21EX-A 6	1.	210.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	86.68	390.07
7	1.4 LA...	CL21EX-A 6	1.	80.	0.	0.		3.5	1.75	1.75	4.	2.	2.	8.68	6.86	-42.95	91.56	554.83

Figure 24: Treatment Planning Window - External Beams Tab. The External Beams tab displays the basic beam properties for the plan selected in the Plans Listbox. The information can either be created newly (right-buttom click, Add beam), copied from another plan/beam (right-buttom click, Copy beam) or imported from a patient Dicom-RT structure.

the selected plan. Beams are separated by row and the user has the option of adding or deleting a selected beam with a right click. The beam information present in a Dicom-imported patient will automatically populate the external beam listbox. A few of these properties can be changed within the listbox and others must be changed within the beam properties window which is opened by double clicking on a beam.

Beam Properties Window This window is used to adjust all beam parameters which includes the treatment mode, linac name, energy, MLC file, and wedge properties. The General Tab in this window summarizes the Modality, Treatment Machine, Beam Energy, Beam Applicator, Applicator Type, Beam Description, Beam Type and prescription information (beam

weight, dose rate, MUs, and number of fractions). The Geometry Tab contains information about beam angles, collimator angle, couch angle as well as isocenter information, SSD and Source-to-isocentre distance ("Nominal Isocenter Distance"). Jaw settings are also included. MLC fields are displayed within the accessory tab. The shape and number of MLC leaves is determined by the configurations of the linac and the leaf thicknesses defined within the MLC.txt file. The field index is considered the weight of the MLC opening. For static fields this number should be 1, while for dynamic fields this number varies between 0 to 1.

6.3.2 View Tab

The image tab contains options for viewing images, structures and dose distributions. This includes a window and level slide bar for adjusting the image window and level. Adjustments are immediately applied to the current axial slice and then looped throughout the image set. Once the window and level has been calculated on all slices, the sagittal and coronal canvases are updated. There is a slide bar for scaling all three canvases. The default setting is set for the "Top" canvas, however this can be changed to the "Left" and "Right" canvas but right clicking on the slide bar. The check boxes turn on/off individual display options. There is also a transparency setting for structure and colourwash dose distributions. The isocenter is marked as a small red x. The hot point point which is defined as the point of highest dose at the current slice, is marked as an asterisk. The listbox on the left is for adjusting the structure display. Each structure can be individually shown as a contour of points or a filled in area. Keep in mind that the contour of points is only visible within the axial display because structures are defined within the axial slices. To view structures in the sagittal and coronal views, one has to fill in the structure. Contour filling is memory intensive and will slow down transitions from one slice to another. Structure colours can be changed by clicking on the colour box. To make permanent changes to the structures one has to edit and save them within the contour window 9

Scaling: All three canvases can be scaled with the scale bar under the image tab. The scale bar acts on one canvass at a time. Right click on the scale bar to change the activated canvass.

Window and Level: Window and Level values for imaging information displayed on the canvases can be modified using these parameters.

Crosshairs: Cross hairs can be activated under the "View" tab. When activated, the user must first select a crosshair by clicking on it with the mouse and then dragging the crosshair to the new location. The crosshair manipulation becomes more difficult, once the dose distributions are loaded. In some cases it is advisable for speed reasons to deactivate the "Isodose Lines" (in the "View" tab) temporarily when changing the position of the crosshairs.

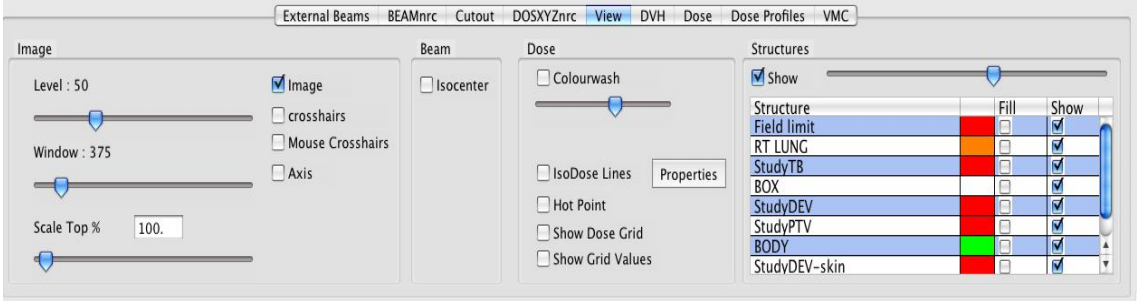


Figure 25: Treatment Planning Window - View Tab. The view tab features various display settings for images, dose distributions, structures, and beam properties.

Isodose Properties The isodose properties button opens a window to edit the isodose lines. Isodose line are recalculated on the fly and thus slow down the functional speed of MMCTP. It would be wise to only use isodose lines once the desired image slice has already been displayed. Isodose lines can be normalized to a percent of the maximum dose, a percent of the isocenter dose, a percent of a dose value or in absolute dose. For plans with multiple beams, the isocenter value is determine from the first beam isocenter position.

6.3.3 DVH Tab

The dose tab includes the DVH calculator and the dose matrix operations. Once a dose distribution has been selected within the plan listbox, the DVH

for each structure can be calculated by checking the calculate option and pressing the calculate button. Calculated DVHs will show-up in additional columns within the DVH listbox. A calculated DVH is displayed by clicking the appropriate column and row check box and pressing the show button. All checked DVHs will be displayed within the DVH Window. The dose matrix operations are addition, subtraction, multiplication and division of two dose distributions or a constant. The dose distribution will have the same resolution and coordinates as the first dose distribution. Dose values from the second distribution will be interpolated to the locations of the dose points within the first distribution.

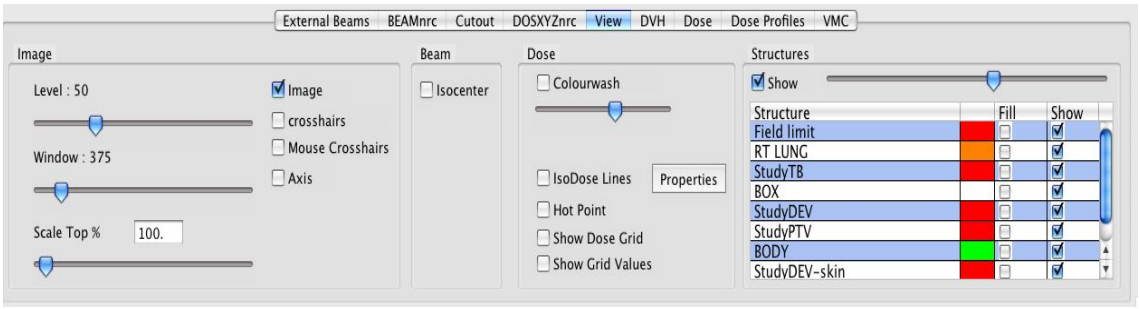


Figure 26: Treatment Planning Window - DVH Tab. The DVH tab contains two listboxes. One which lists the structures to calculate a DVH, and the second which lists the calculated DVHs. The calculated DVHs are separated by their calculated dose distribution (column).

DVH Window The DVH window can display DVH graphs as differential or cumulative. The window also displays DVH statistics such as the pixel volume, max dose, average dose and min dose. An export button will export the selected DVH to a text file. DVH plots are plotted using the number of bins as the number of points and the min and max dose values as the scale on the X axis.

DVH Calculator DVHs are calculated on the axial image resolution. For each axial image, a structure is defined by a closed loop of contour points. Pixels within the closed loop define the volume of the structure. The DVH is constructed by interpolating a dose value at the center of each pixel.

6.3.4 Dose Profiles

The dose profile tab allows the user to define a line with the rulers and calculate a dose profile with the selected dose distribution. Profiles can be normalization to the max value or to a specific value. There is also a distance offset which may be useful for profile comparisons with measured data. The import profile allows the user to copy and past excel data into MMCTP as a new profile. Calculated profiles are displayed within the listbox and can be shown individually or together.

6.3.5 Dose Tab

The dose tab allows the user to create a dose distribution and paint the dose values within the dose grid.

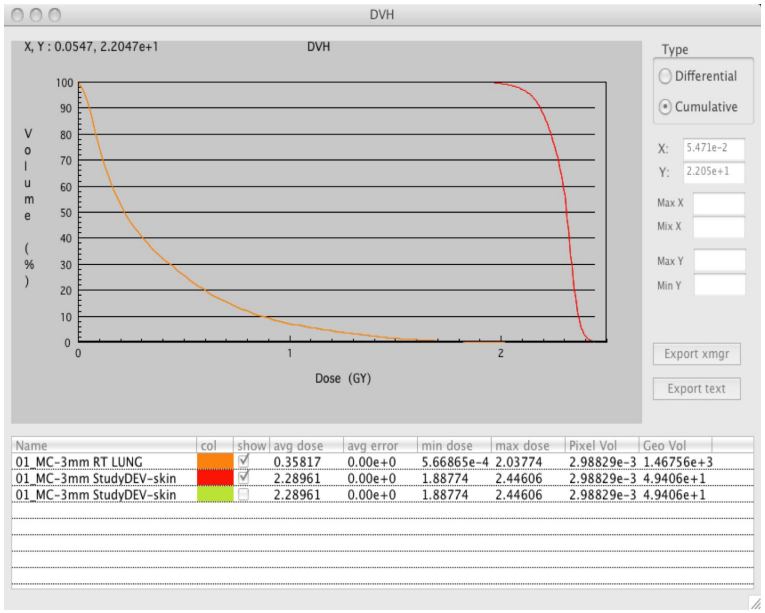


Figure 27: Treatment Planning Window - DVH Window.

7 BEAMnrc within MMCTP

BEAMnrc [4] is a EGS user code developed through the OMEGA project (Ottawa Madison Electron Gamma Algorithm) for the simulation of radiation beams from radiotherapy sources. The code performs the particle transport through the complex geometry of treatment units. Treatment units include linear accelerators, orthovoltage units and Cobalt-60 units. BEAM uses the concept of component modules (CM) to model each component of a treatment unit. The CM modules are stacked perpendicular to the beam axis with a top and bottom surface. The dimensions and materials of each CM module is specified within the BEAM input file. A modeled treatment unit consists of a series of independent CMs sandwiched together. Some examples of CM are: air slabs, jaws, MLC and mirror. Aside from the CMs, the BEAM input file contains transport parameters regarding the energy cutoffs limits for electrons (ECUT) and photons (PCUT), minimum energy required for knock on electrons (AE) and bremsstrahlung photons (AP). In addition, the energy source (monoenergetic, spectrum or phase space file), and the variance reduction options, are included within the input file.

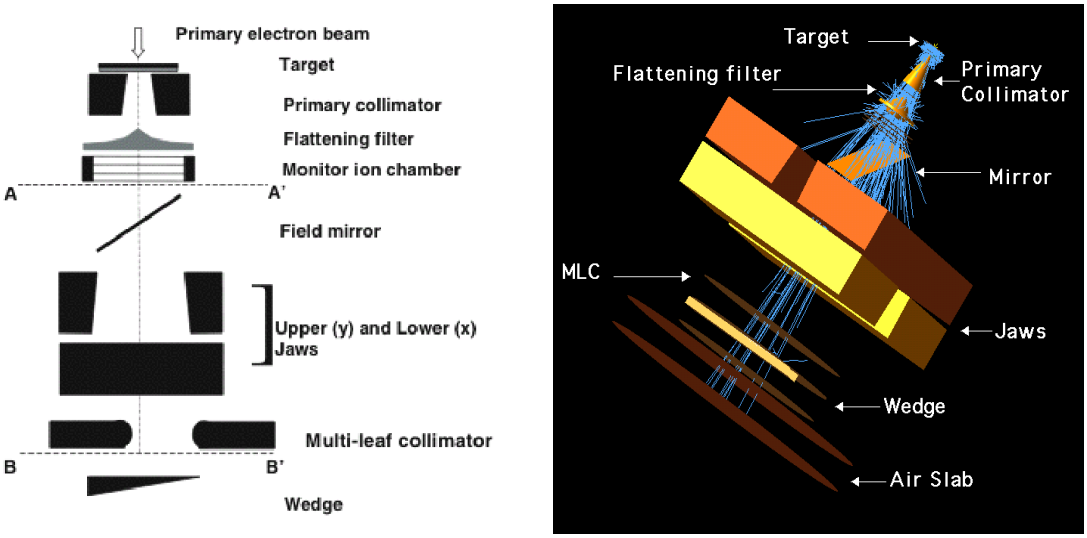
The BEAMnrc output is a phase space file containing information on the physical characteristics of all particles exiting the last CM plane. These characteristics include x, y, z coordinates, direction, particle energy and particle weight. The phase space output can be calculated below any CM. In addition, the phase space can be used as a source for another simulation, or analyzed using user codes to calculate physical quantities. These quantities include: particle energy spectra, fluence, angular distributions and spatial distributions.

7.1 Linac Properties

Virtual linac units are defined within a file named $BEAM.pref$. This file defines and labels the available linacs in terms of radiation type, energy, applicator options MLC type physical wedges, and BEAMnrc variables. There can be multiply energies for each linac unit and each energy will have an associated BEAMnrc input file. This file links the linac names within MMCTP with a predefined BEAMnrc input file. The file format was defined in Figure 8.

7.2 BEAMnrc input files

BEAMnrc input files are all placed within the BEAMnrc folder. These files are used to construct the specific BEAMnrc input file based on the beam geometry properties. For each linac energy, there is a base template file. The template files store specific information related to the linac that does not change from run to run. This includes, initial electron energy distribution, the dimensions and materials of all parts of the machine, the location of output planes, how to track a particle's history, which variance reduction technique to apply, what transport parameters to use, etc. Aside from these, there are four parameters that do change from run to run. Parameters specific to the beam geometry properties include the jaw settings, MLC field, and wedge component. A fourth parameter specifies the initial number of particles for the simulation. MMCTP uses the CM identifiers and linc properties to determine how to edit template input files for simulations. For example, if the CM JAWS is used within the template input file, MMCTP will edit this CM to correspond to the settings of the field openings.



(a) (b)

Figure 28: (a) Schematic drawing of linac components modeled in BEAMnrc Monte Carlo simulations [8], (b) BEAMnrc simulation of modeled CL21EX 18 MV photons (blue) through linac head.

Table 3: BEAMnrc sources coded within MMCTP

BEAMnrc Source #	GUI
0	Yes
1	Yes
3	Yes
19	Yes
21	Yes

There is a limited number of CMs (listed below) which MMCTP can edit. If an addition CM is required, the user will have to modify the source code to include this CM.

7.2.1 Man Inputs - Header

Many of the BEAMnrc main header input options are included within MMCTP. MMCTP will edit the number of initial histories and radius used within DBS. Main input variables, along with the EGSnrc input options can be edited within the Advanced BEAM Properties Window. Source options within MMCTP are listed in table 3. Sources maybe edited by the user within the source GUIs however, MMCTP does not automatically edit source parameters.

7.2.2 CM JAWS

The JAWS is edited by the field X,Y openings. If the template input file includes the identifier JAWS then this CM will be edited. The source code for editing the JAWS cm is found within Class_BEAM.egs_Input_CM_Jaws

Siemens Script within MMCTP is chosen if the linac type is Siemens and the JAW CM is included in the BEAMnrc input file.

Varian Script within MMCTP is shown in figure 29. This script is triggered if the linac type is anything but Siemens and the JAW CM is included in the BEAMnrc input file.

Table 4: BEAMnrc CMs within MMCTP. Auto-scripts are linked to particular linac manufactures and CM inputfile flags. CMs external to this list are not editable within MMCTP

CM	Read	Auto-scripts	GUI
JAWS	Yes	Yes	Yes
DYNJAWS	Yes	Yes	Yes
WEDGE	Yes	Yes	Yes
DYNVMLC	Yes	Yes	Yes
VARMLC	Yes	Yes	Yes
MLC	Yes	No	Yes
TOMOMLC	Yes	Yes	Yes
APPLICAT	Yes	Yes	Yes
BLOCK	Yes	Yes	Yes
SLAB	Yes	No	Yes
FLAT	Yes	No	No

7.2.3 CM DYNJAWS

This CM requires a jaw positioning file for multi-field simulations, otherwise the CM acts as the JAW CM with only one field. Fields are sampled at discrete steps using random numbers and the individual field index value. If the template input file includes the identifier DYNJAWS then this CM is edited within MMCTP. The source code for editing the DYNJAWS cm is found within:
Class_BEAM.egs_Input_CM_DYNJAWS

Dynamic wedge involves one of the jaws sweeping across the field opening to meet the other jaw during irradiation. MMCTP generates this file based on the filed opening, the wedge angle and a gold standard positioning file.

Merged beam sums individual beams into a single simulation with multi-field jaws. MMCTP generates this file based on the field opening of individual beams.

```
x1=-gRTOG.Plan(Plan_Index).Beam(beam_number).Collimator.fields(0).X1
x2=gRTOG.Plan(Plan_Index).Beam(beam_number).Collimator.fields(0).X2
y1=-gRTOG.Plan(Plan_Index).Beam(beam_number).Collimator.fields(0).Y1
y2=gRTOG.Plan(Plan_Index).Beam(beam_number).Collimator.fields(0).Y2

if cm.JAWS.XY_Choice(i-1)="Y" Then
        zytop=cm.JAWS.ZMIN_JAWS(i-1)
        zybot=cm.JAWS.ZMAX_JAWS(i-1)
        cm.JAWS.XFP_JAWS(i-1)=zytop*y2/100
        cm.JAWS.XBP_JAWS(i-1)=zybot*y2/100
        cm.JAWS.XFN_JAWS(i-1)=zytop*y1/100
        cm.JAWS.XBN_JAWS(i-1)=zybot*y1/100

if  cm.JAWS.XY_Choice(i-1)="X" Then
        zxtop=cm.JAWS.ZMIN_JAWS(i-1)
        zxbot=cm.JAWS.ZMAX_JAWS(i-1)
        cm.JAWS.XFP_JAWS(i-1)=zxtop*x2/100
        cm.JAWS.XBP_JAWS(i-1)=zxbot*x2/100
        cm.JAWS.XFN_JAWS(i-1)=zxtop*x1/100
        cm.JAWS.XBN_JAWS(i-1)=zxbot*x1/100
end
```

Figure 29: MMCTP code for generating BEAMnrc JAW positions

7.2.4 CM WEDG

* Note this CM is not part of the standard BEAMnrc distribution!

The WEDG [7] is edited by the static wedge settings within MMCTP. If the template input file includes the identifier "WEDG" then this CM will be edited. The WEDG CM is modified by replace the entire CM string with the appropriate CM string for the specific wedge angle and orientation. The wedge CM files (all angles and orientations) are saved within the configuration folder. MMCTP uses one wedge file to generate the wedge component module of the BEAMnrc input file. An example of a wedge file is shown in Figure 30

If the wedge mode is Static, MMCTP will automatically replace this CM

```
********** start of CM WEDGE with identifier WEDG **********
15.0000,
MLC 30x40cm 15 steel WEDGE
1, 1, 0,
57.6,
58.6, 0.15,
0.7, 0.01, 0, 0, 0.0, ECUT,PCUT etc
PMMA700ICRU
0, 0, 58.75, 63, WDIR,WOR,WPOS(2) general wedge data
0.7, 0.01, 0, 0, interior of CM wedge (assumed AIR)
2, 4, NPROF and NPOINTS for wedge geometry
-9.2, -6.9, 5.1, 9.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, profile points for wedge
-12.65, 0.1, 0.55, 1.4, 1.4, 0.0, 0.0, 0.0, 0.0, 0.0,
12.65, 0.1, 0.55, 1.4, 1.4, 0.0, 0.0, 0.0, 0.0, 0.0,
0.7, 0.01, 0, 0, 0.0, ECUT,PCUT etc for wedge
STEELLESS700ICRU steel
```

Figure 30: BEAM input file, wedge component module for a 15 degree wedge

with the wedge file #ORIENTATION.egsinp. Where # refers to the angle and orientation being a string defined within the beam configurations for each wedge angle. For example, if the beam has a 15 degree RIGHT wedge, then MMCTP will look for a file named 15RIGHT.egsinp. All wedge files are located within the BEAMnrc folder.

7.2.5 CM DYNVMLC

The DYNVMLC is edited by the MLC settings within MMCTP. If the template input file includes the identifier DYNVMLC then this CM will be edited. MLC settings within MMCTP are converted into the DYNVMLC format. The CM supports static, step & shoot and dynamic MLC fields, the MLC mode is defined within MMCTP. MLC settings within MMCTP are converted into the DYNVMLC format. This format include a separate MLC input file for the sequence of MLC positions. Depending on the type of linac, Varian or Siemens, there are two script methods for generating the DYNVMLC CM leaf positions.

Generating DYNVMLC files Within MMCTP, a MLC field is defined at isocenter. A script is required to convert the MLC leaf projections at isocenter to the MLC leaf positions within BEAMnrc. Back-projection is complicated by rounded leaf ends of the MLC. Figure 31 illustrates the geometry of a rounded leaf end.

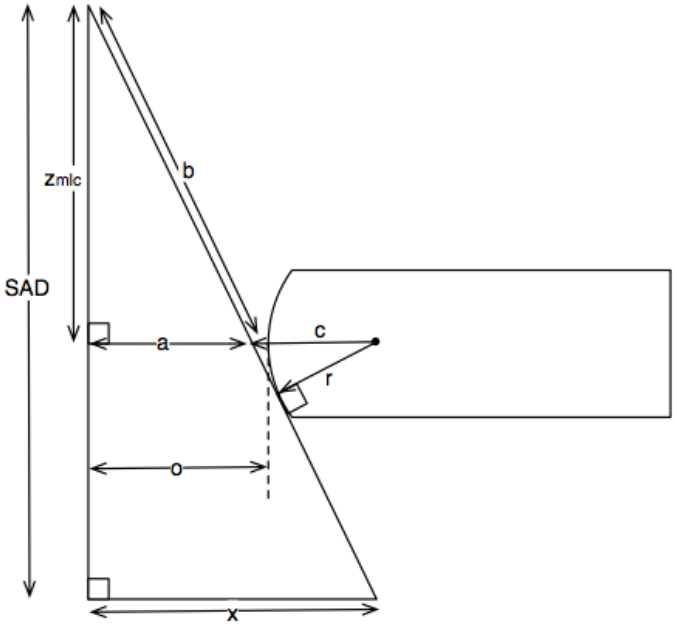


Figure 31: Projection diagram illustrating a ray tangent to the rounded edge of a leaf. [6]

The position of the MLC leaf in the leaf plane is given by o. The position of the leaf in the isocentre plane is considered to be the distance of the open beam from the isocentre and is denoted by x. The radius of curvature of the leaf is given by r (zfocus_ends in BEAMnrc variables) and the distance between the center of curvature of the leaf and the ray which is tangent to the leaf is given by c. The distance the ray travels from the source to the leaf plane is given by b. The distance from the source to the MLC plane is given by $zmlc$ (zmin+zthick/2 in BEAMnrc variables). Finally, the distance between

the source and the isocentre is given by SAD (source to axis distance) which is the nominal isocentre distance in MMCTP.

From this setup one can create the following equations:

$$a + c - r = o \tag{3}$$

$$\frac{z_{mlc}}{b} = \frac{r}{c} \tag{4}$$

$$b = \sqrt{a^2 + z_{mlc}^2} \tag{5}$$

Solving for a yields:

$$a = \frac{o + r}{1 - \frac{r^2}{z_{mlc}^2}} - \sqrt{\left(\frac{o + r}{1 - \frac{r^2}{z_{mlc}^2}}\right)^2 - \left(\frac{o^2 + 2or}{1 - \frac{r^2}{z_{mlc}^2}}\right)} \tag{6}$$

And with the use of similar triangles, we arrive at a solution for x.

$$x = \frac{a \times SAD}{z_{mlc}} \tag{7}$$

Varian Script is defined within the methods of the class BEAM, under the title egs_Input_CM_DYNVMLC_MLC2BEAM_Varian. This script is unique to Varian millennium type MLCs with rounded leaf ends. In the event that the MMCTP beam does not contain a MLC pattern, a default retracted MLC position of 20.3 cm is used for each leaf. A section of the DYNVMLC script code is shown in figure 32

Siemens Script is defined within the methods of the class BEAM, under the title egs_Input_CM_DYNVMLC_MLC2BEAM_Siemens. This script is unique to Siemens type MLCs.

7.2.6 CM SLABS

MMCTP will read-in and re-write this CM. The user can edit SLAB values within the advanced beam properties window.

7.2.7 CM FLAT

MMCTP will read-in and re-write this CM. The user can edit FLAT values within the advanced beam properties window.

7.2.8 CM MLC

MMCTP will read-in and re-write this CM. The user can edit MLC values within the advanced beam properties window.

7.2.9 CM VARMLC

The VARMLC is used to model a focusing MLC with rounded or straight leaf ends. MMCTP will read-in and re-write this CM. VARMLC is edited by the MLC field settings and if the template input file includes the identifier VARMLC, during the creation of a new BEAMnrc input file. The Varian script in DYNVMLC is also used for VARMLC.

7.2.10 CM TOMOMLC

* Note this CM is not part of the standard BEAMnrc distribution!
TOMOMLC is an adaptation of VARMLC for modeling the binary Tomotherapy MLC. TOMOMLC is edited by the MLC field settings and if the template input file includes the identifier TOMOMLC, during the creation of a new BEAMnrc input file.

7.2.11 CM APPLICAT

MMCTP will read-in and re-write this CM. The user can edit APPLICAT values within the advanced beam properties window. If the beam mode is ELECTRON, MMCTP will automatically replace this CM with the applicator file "APP-" + id of the applicator. The applicator file must be located within the BEAMnrc folder. The id of the applicator is defined within the beam configurations under electron applicators. For example, if the applicator id is A10x10, then the file must be labeled APP-A10x10.

7.2.12 CM BLOCK

MMCTP will read-in and re-write this CM. The user can edit BLOCK values within the advanced beam properties window.

7.3 BEAMnrc variables

There is an ongoing process of including additional MMCTP BEAMnrc variables and native BEAMnrc variable as needed. All EGSnrc input variables have been coded within MMCTP. The following is a list of MMCTP BEAMnrc variables

```
#tag Property, Flags = &h0
#tag Note
Used to set queue selection to automatic (true) or manual (false)
#tag EndNote
auto_queue As boolean = true
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Flag set to true is test beam returns an error
#tag EndNote
Beamnrc_error As Boolean
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
beam number index within plan, beam_number starts at 0 index
#tag EndNote
beam_number As Integer
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Flag to auto run calculation
#tag EndNote
calculate As Boolean = false
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
flag set true is addphsp reports "Done" within addphsp output
#tag EndNote
```

```
egs_AddPhsp_Finished As boolean = false
#tag EndProperty

#tag Property, Flags = &h0
egs_auto_shell As Boolean = true
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
number of active jobs
#tag EndNote
egs_BEAMnrc_active_jobs As Integer
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
beamnrc folder within egs home
#tag EndNote
egs_BEAMnrc_dir As String
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
flag set true is calculation has started
#tag EndNote
egs_BEAMnrc_started As boolean
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
value = "CPUTIME per history" within egslog file
#tag EndNote
egs_CPU_time As single = 0
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Desired phsp particle density, used to calculate the number of histories
```

```
#tag EndNote
egs_desired_phsp_particle_density As Integer = 500000
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
name of input file
#tag EndNote
egs_Inputfile As string
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
number of jobs to split simulation
#tag EndNote
egs_jobs As Integer = 8
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
linac index within all configured linacs, BEAM.pref file
#tag EndNote
egs_linac_index As Integer
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Number of histories to run
#tag EndNote
egs_num_histories As int64 = 10000000
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Number of particles per history

calculated by
math=gBEAM.beams(gg.beam_num).egs_phsp_num_particles/StartNohist
```

```
gBEAM.beams(gg.beam_num).egs_particle_per_history=math
#tag EndNote
egs_particle_per_history As single = 1
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Pegs file
#tag EndNote
egs_pegs_file As string
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
If the phsp file is real (false) or a link (true)
#tag EndNote
egs_Phsp_link As boolean
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Variable to read phsp file with BEAMDP,
set true once the phsp file has been read.
#tag EndNote
egs_Phsp_Lookup As boolean = false
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Name of phsp file
#tag EndNote
egs_Phsp_name As string
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Number of particles in phsp file
#tag EndNote
```

```
egs_Phsp_Num_Particles As Integer
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Number of Photons in phsp file
#tag EndNote
egs_Phsp_Num_Photons As Integer
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Variable to search phsp database, false if search has not occured
#tag EndNote
egs_Phsp_Search As Boolean = false
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
size in bytes of phsp file
#tag EndNote
egs_Phsp_Size As int64
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
progress of simulation

-1=Not started
Progress betwen 0:100%
#tag EndNote
egs_progress As single = -1
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
simulation queue
#tag EndNote
```

```
egs_queue As string = "medium"
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
name of shell for simulation
#tag EndNote
egs_Shell As String
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
shell index for simulation
#tag EndNote
egs_Shell_Index As Integer
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
simulation start time
#tag EndNote
egs_Start_Time As string
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
BEAMnrc input file
#tag EndNote
Inputfile As Class_BEAM_Inputfile
#tag EndProperty

#tag Property, Flags = &h0
#tag Note
Number of test histories
#tag EndNote
Num_test_hist As Integer = 15000
#tag EndProperty
```

```
#tag Property, Flags = &h0
#tag Note
Array of split w files, uesed to determine if split files have finished properl
#tag EndNote
w_files(-1) As boolean
#tag EndProperty
```

The following is a list of BEAMnrc header variables included within MMCTP

```
#tag ViewBehavior
#tag ViewProperty
Name="Name"
Visible=true
Group="ID"
InheritedFrom="Object"
#tag EndViewProperty
#tag ViewProperty
Name="Index"
Visible=true
Group="ID"
InitialValue="-2147483648"
InheritedFrom="Object"
#tag EndViewProperty
#tag ViewProperty
Name="Super"
Visible=true
Group="ID"
InheritedFrom="Object"
#tag EndViewProperty
#tag ViewProperty
Name="Left"
Visible=true
Group="Position"
InitialValue="0"
InheritedFrom="Object"
```

```
#tag EndViewProperty
#tag ViewProperty
Name="Top"
Visible=true
Group="Position"
InitialValue="0"
InheritedFrom="Object"
#tag EndViewProperty
#tag ViewProperty
Name="ECUT"
Group="Behavior"
InitialValue="0.7"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="PCUT"
Group="Behavior"
InitialValue="0.01"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="range_rejection"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="ESAVE_Global"
Group="Behavior"
InitialValue="4"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="RNG_1"
Group="Behavior"
InitialValue="97"
Type="Integer"
#tag EndViewProperty
```

```
#tag ViewProperty
Name="RNG_2"
Group="Behavior"
InitialValue="33"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="Max_CPU_hr"
Group="Behavior"
InitialValue="500"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="Incedient_Particles"
Group="Behavior"
InitialValue="1"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="title"
Group="Behavior"
Type="string"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="medium"
Group="Behavior"
Type="string"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="IWATCH"
Group="Behavior"
InitialValue="0"
Type="Integer"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
```

```
Name="IQIN"
Group="Behavior"
InitialValue="0"
Type="integer"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="dose"
Group="Behavior"
InitialValue="0"
Type="integer"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="z_of_front"
Group="Behavior"
Type="string"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="mlc_path"
Group="Behavior"
Type="string"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="IBRSPL"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="IRRLTT"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
```

```
Name="ICM_SPLIT"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NBRSPL"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NSC_PLANES"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="IRESTART"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="ISTORE"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="IO_OPT"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="IDAT"
Group="Behavior"
```

```
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="LATCH_OPTION"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="IZLAST"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="ISOURC"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="DISTZ"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="RBEAM"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="GAMMA"
Group="Behavior"
InitialValue="0"
Type="Integer"
```

```
#tag EndViewProperty
#tag ViewProperty
Name="XINL"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="XINU"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="YINL"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="YINU"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="WINC"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="UINC"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
```

```
Name="VINC"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="MONOEN"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="EIN"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="sigma_src19"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="IFORCE"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NFMIN"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NFMAX"
Group="Behavior"
```

```
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NFCMIN"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NFCMAX"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="INIT_ICM"
Group="Behavior"
InitialValue="1"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="NRCYCL"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="IPARALLEL"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="PARNUM"
Group="Behavior"
InitialValue="0"
Type="Integer"
```

```
#tag EndViewProperty
#tag ViewProperty
Name="ISRCDBS"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="RSRCDBS"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="SSDSRCDBS"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="ZSRCDBS"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="SPCNAM"
Group="Behavior"
Type="string"
EditorType="MultiLineEditor"
#tag EndViewProperty
#tag ViewProperty
Name="USE_REJPLN"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
```

```
Name="Z_REJPLN"
Group="Behavior"
InitialValue="0"
Type="Single"
#tag EndViewProperty
#tag ViewProperty
Name="augmented_rangeR"
Group="Behavior"
InitialValue="0"
Type="Boolean"
#tag EndViewProperty
#tag ViewProperty
Name="IRADDBS"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="auto_mmctp"
Group="Behavior"
InitialValue="true"
Type="boolean"
#tag EndViewProperty
#tag ViewProperty
Name="FS"
Group="Behavior"
InitialValue="0"
Type="Single"
#tag EndViewProperty
#tag ViewProperty
Name="SSD"
Group="Behavior"
InitialValue="0"
Type="Single"
#tag EndViewProperty
#tag ViewProperty
Name="ICMDBS"
Group="Behavior"
```

```
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="ZPLANEDBS"
Group="Behavior"
InitialValue="0"
Type="Integer"
#tag EndViewProperty
#tag ViewProperty
Name="ZRR_DBS"
Group="Behavior"
InitialValue="0"
Type="Single"
#tag EndViewProperty
#tag ViewProperty
Name="RBEAMY"
Group="Behavior"
InitialValue="0"
Type="single"
#tag EndViewProperty
#tag ViewProperty
Name="RMINBM"
Group="Behavior"
InitialValue="0"
Type="Single"
#tag EndViewProperty
#tag ViewProperty
Name="ZSMIN"
Group="Behavior"
InitialValue="0"
Type="Single"
#tag EndViewProperty
#tag ViewProperty
Name="ZSMAX"
Group="Behavior"
InitialValue="0"
Type="Single"
```

```
#tag EndViewProperty
#tag EndViewBehavior
```

7.4 Phsp Database

The phsp database is designed to keep track of all BEAMnrc simulations where the output is a phsp file. The database is stored within the BEAMnrc folder in file called PhaseSpace.rsd. The database stores the following properties; Beam Mode, File Name, Beam Energy, FLEC Opening, Linac Name, Shell, Jaw Opening, MLC Opening, Wedge and Applicator. The database is used to link future simulations with previously completed simulations. MMCTP will automatically attempt to link any new simulation with a previous simulation if a match is found. If a match is not found, MMCTP will run the BEAMnrc simulation and update the database with a new entry.

7.5 BEAMnrc scoring plane

It is important to keep track of the BEAMnrc scoring plane as this value is used to position the phase-space file within the DOSXYZnrc simulation. MMCTP will read the CM number of the scoring plane and determine the Z position of the plane. However, the scoring plane CM must be one of the following CMs: SLAB or BLOCK.

```
zmin=CM.DYNVMLC.ZMIN
zthick=CM.DYNVMLC.zthick
leaf_radius =CM.DYNVMLC.zfocus_ends
z_field   =gRTOG.Plan(Plan_Index).Beam(beam_number).Nominal_Isocenter
gap  = CM.DYNVMLC.Leafgap
inum_leaves=CM.DYNVMLC.numleafs
z_MLC=zmin+zthick/2
abut_gap = gap*z_field/z_MLC/2.0

leaf_a=egs_Input_CM_DYNVMLC_
Leafset(leaf_a,Normal_or_Overtravel_1,z_MLC,leaf_radius,z_field)
leaf_b=egs_Input_CM_DYNVMLC_
Leafset(leaf_b,Normal_or_Overtravel_2,z_MLC,leaf_radius,z_field)

Within egs_Input_CM_DYNVMLC_Leafset
(xleaf as double, mode as integer, z_MLC as double,
radius_leaf as double, z_field as double)

if (Xleaf=0.0) then
    return(0.0)
end
RC=abs(Xleaf/z_field)
A=-4.0/pow(RC,2.0)
B=8.0*z_MLC/RC
C=4.0*(pow(radius_leaf/RC,2.0)+pow(radius_leaf,2.0)-pow(z_MLC,2.0))
Disc=pow(B,2)-4.0*A*C

if(Xleaf<0 and mode=0) then
    res=(B+sqrt(Disc))/(2.0*A)
end
if(Xleaf<0 and mode=1) then
    res=(B-sqrt(Disc))/(2.0*A)
end
if(Xleaf>0 and mode=0) then
   res=(-B-sqrt(Disc))/(2.0*A)
end
if(Xleaf>0 and mode=1) then
   res=(-B+sqrt(Disc))/(2.0*A)
end
return(res)
```

Figure 32: MMCTP code for generating BEAMnrc DYNVMLC positions

7.6 Treatment Planning - BEAMnrc GUI

The BEAMnrc tab encompasses the windows and parameters regarding BEAMnrc simulations. A listbox displays the simulation status for each beam within the selected plan. Simulations are run by clicking on the calculate checkbox. Double clicking on a beam will open the BEAMnrc Window, where the user can edit some basic properties of the simulation.

7.6.1 BEAMnrc Window

This window allows the user to edit parameters specific to the simulation. These include the number of histories, the number of jobs to split the simulation, the PEGS file, queue and the desired number of particles in the phsp file. The desired number of particles can be used to determine the number of initial histories. However, this method to determine the number of histories requires a test run to determine the ratio of particles per history. Test runs also provide estimates on the CPU time per history. Included in this window is an option to run addphsp. Use this option if MMCTP was not able to automatically run addphsp. The addphsp button will send a command to manual add the phsp files regardless of the simulation status. This is can be useful when simulation do not finish properly. However, one should examine why the simulations did not finish on their own prior to using the phsp file for patient recalculation. This window also allows access to the Phase Space Information Window and Advanced BEAM options Window.

7.6.2 Advanced BEAM options

This window allows the user to edit the main input parameters of the BEAM simulation. There is also a button to reload the default template input file. In addition to the main inputs, a few CMs can be modified by selecting them an pressing the edit button. For more information on these values refer to the BEAMnrc user manual [5].

7.6.3 Advanced EGSnrc inputs

This window allows the user to edit the EGSnrc parameters of the BEAM simulation. For more information on these values refer to the BEAMnrc user manual.

7.6.4 Phase Space Information Window

The Phase Space Information Window allows the user to see the phsp properties of the selected beam. There is also an option of linking the current beam

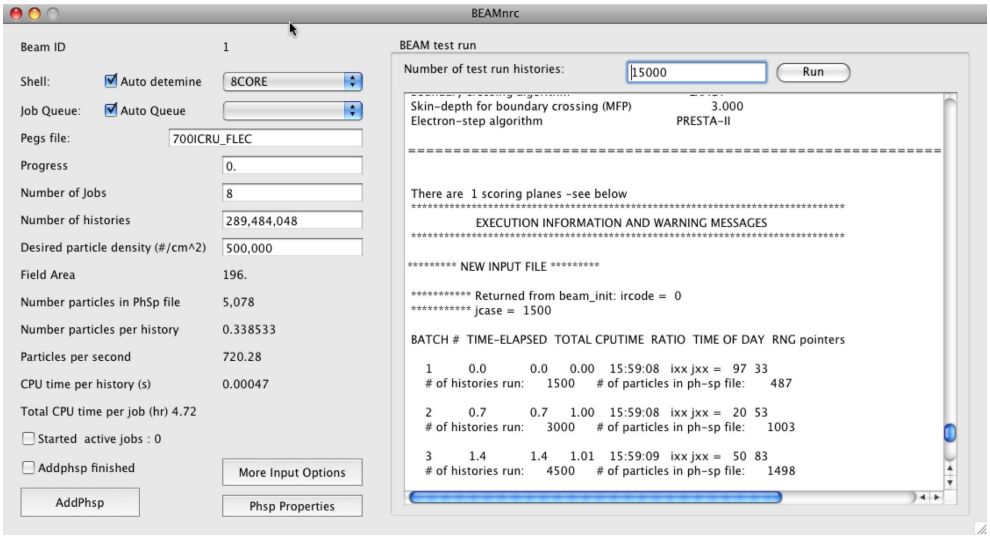


Figure 33: BEAMnrc Window: User can edit the job shell, pegs file, number of jobs, number of histories and desired particle density.

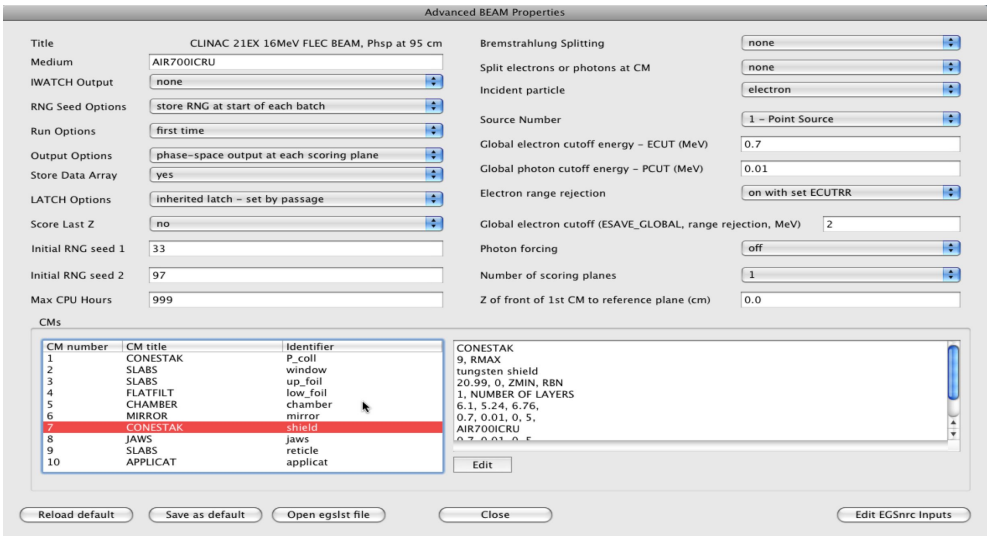
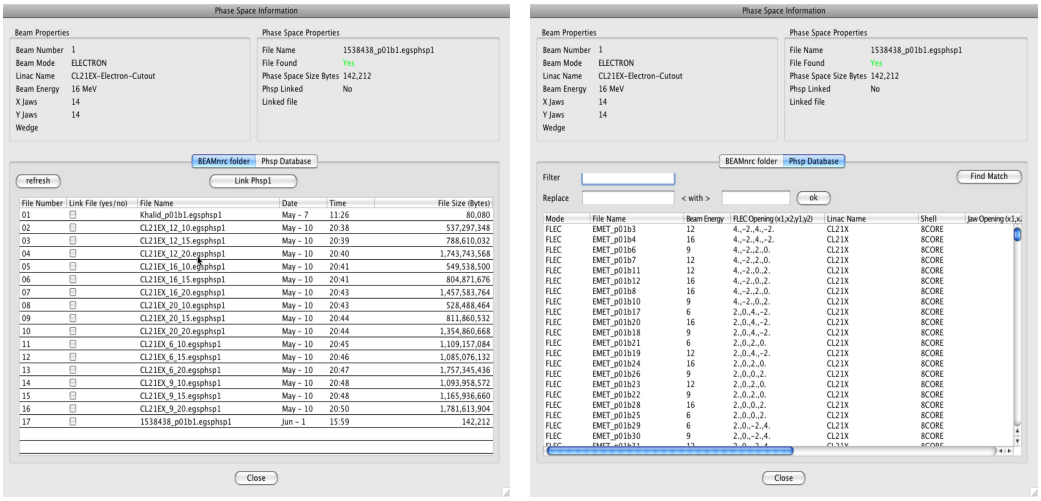


Figure 34: Advanced BEAMnrc Window

with a perviously calculated phsp file. The "Search for phsp files" button will connect to the remote cluster and display a list of all phsp files within the BEAMnrc directory. This action will also update the Phsp properties of the current beam. Selecting a phsp file and pressing on the "Link Phsp1" button will create a soft link within the BEAMnrc directory between the current beam and the selected phsp file. Note that this will overwrite any existing Phsp file which exits for the selected beam.



(a) (b)

Figure 35: Phase Space Information Window: Header information includes properties of the beam, and properties of any existing phase space file. (a) List of all phase space files within the current BEAMnrc folder specific to the login shell. (b) List of all phase space records within the phase space database.

7.7 BEAMnrc Job Submission

MMCTP will submit and refresh the status of BEAMnrc simulations. The user has the option of manually submitting a test run, and running addphsp. Job submission and refresh are controlled by the global MC period settings and by the BEAMnrc control settings.

7.7.1 Auto link with phsp database

If this option is on, MMCTP will search for an existing phsp file which matches the criteria of the current beam properties. If a match is found, MMCTP links the current simulation with the phsp file.

7.7.2 Auto determine shell

If this option is on, MMCTP will determine which shell to submit the current BEAMnrc simulation. This option only applies when there are multiple shells and BEAMnrc linacs configuration on more than one shell. The script searches for an online shell with the greatest number of available job slots.

7.7.3 Run test BEAM

MMCTP will attempt to run a test simulation before submitting a production run. The test run uses a small number of histories 10000 to determine the ratio of particles per history, and CPU time per history. This ratio will be used to determine the required number of histories needed to achieve a desired phsp density.

7.7.4 Run BEAM

Auto run for BEAMnrc is controlled by a timer, a user selected BEAMnrc auto check box within the configurations window and a user selected check box on the beam number in question within the treatment planning window. In this action, the input file is generated, uploaded and submitted to a batch system. Input files generated from MMCTP are named according to the patient ID, plan number and beam number (834901_p01b2.egsinp). This process will replace any previous input files, but it will not kill any ongoing simulation. It is up to the user to terminate an active simulation before resubmitting.

7.7.5 Refresh

Refresh will send a command to check on the simulation status of all beams. This is initially done by reading the $*.lock$ file. If the lock file exists, then the simulation is assumed to be running. If the lock file dose not exist the next step is to look at the log files of each simulation. MMCTP reads the

last line of the log files. If all log files are present, then MMCTP assumes that the simulation has finished and sends the command to add the phsp files together to produce one summed phsp file.

7.7.6 Addphsp

Addphsp is used to add phsp files which belong to the same simulation. MMCTP will automatically submit a call to addphsp once the lock file disappears and all *_w*.egslog files include the text Finished simulation. User can manually call addphsp within the BEAMnrc Window.

7.7.7 Phsp properties

Once the phsp file has been generated, MMCTP will run BEAMDP on the file to readin useful properties of the file. These properties include: file size, number of particles and number of photon particles.

7.8 Default BEAMnrc settings within MMCTP

Within the configurations window, MC settings, there is a section of BEAMnrc default settings. These settings control refresh and run options for BEAMnrc jobs, the removal of w files after a simulation, the auto shell option, the auto link phsp file option, the default and minimum number of jobs and IDAT and IZLAST values. IDAT and IZLAST values will override the values within the template input files.

7.9 Known Issues

Simulation does not finish When a BEAMnrc simulation does not finish properly, the lock file does not disappear and thus MMCTP will assume the simulation is still running. The user will have to manually add the phsp files and search of the final phsp file using the Phase Space Information Window to update the status of the simulation.

Batch Submissions Attention must be made when using batch submissions. If the number of batch jobs exceeds the number of available CPUs, the simulation could finish before all batch jobs begin. This in turn confuses MMCTP as to the number phase space files and status of the lock file.

Table 5: DOSXYZnrc sources within MMCTP. Auto-scripts are linked to a particular source type. Sources external to this list are not compatible within MMCTP

Source	Read	Auto-scripts	GUI
1	Yes	Yes	Yes
2	Yes	Yes	Yes
3	Yes	No	Yes
9	Yes	Yes	Yes
8	Yes	No	No
11	Yes	Yes	Yes

8 DOSXYZnrc within MMCTP

For a complete summary of DOSXYZnrc refer to the user manual [10]. MMCTP will create DOSXYZnrc input files from the McGill RT beam geometry settings, upload these files to the cluster computer, submit jobs to the queueing system and download the resulting 3ddose files. These is also an option for the user to import pre-calculated 3ddose files.

8.1 DOSXYZnrc input files

A template input file "dosxyznrc.egsinp" is housed within the BEAMnrc folder. This file is used to create new DOSXYZnrc input files for CT-based EGSPhant files. A second template input file called "MC_WaterPhantom_Output.egsinp" is used under the condition of a non-CT EGSPhant file. DOSXYZnrc input files are read into memory from either the plan folder or the configurations folder. Once in memory, the user can adjust certain parameters. A new input file is written within the plan folder ahead of each simulation.

With the exception of sources, all of the DOSXYZnrc input options are included within MMCTP. MMCTP will only auto-edit the source parameters of the input file. The remaining parameters are determined by the template input files or are edited manually within the Advanced Options.

```
DOSinp.xiso=gRTOG.Plan(Plan_Index).Beam(beam).Collimator.Fields(0).Isocentre.x
DOSinp.yiso=-gRTOG.Plan(Plan_Index).Beam(beam).Collimator.Fields(0).Isocentre.z
DOSinp.ziso=-gRTOG.Plan(Plan_Index).Beam(beam).Collimator.Fields(0).Isocentre.y
DOSinp.theta=gRTOG.Plan(Plan_Index).Beam(beam).Collimator.Fields(0).Gantry_Angle
DOSinp.phi=gRTOG.Plan(Plan_Index).Beam(beam).Collimator.Fields(0).couch_Angle
DOSinp.phicol=gRTOG.Plan(Plan_Index).Beam(beam).Collimator.Fields(0).Coll_Angle


DOSinp.dsource=gRTOG.Plan(Plan_Index).Beam(beam).Nominal_Isocentre-
gBEAM.Beams(beam).Inputfile.CMs(cm_index).Bottom_z

DOSinp.FILNAM=BEAMnrc_directoy/MC_file_name.egsphsp1
```

Figure 36: MMCTP code for generating source 2 parameters

8.2 EGSPhant Files

MMCTP allows the user to generate muliple EGSPhant files per patient. Each EGSPhant file acts as a class variable for all parameters associated to the EGSPhant file. This includes the input file class for each external beam. Other parameters specific to MMCTP control logic for each DOSXYZnrc calculation are included as supplementary information to the input file. Control logic parameters are saved within the plan folder under the name "DOSXYZ_*.txt" where * is the name of the EGSPhant file.

8.2.1 EGSPhant Materials & Densities

EGSPhant files are created using the CT images and structures. The user has the option of filling in each structure with a material and density, or using an auto fill algorithm.

Contour manual Fill The manual fill option is used to assign specific materials and densities to specific structure volumes. Within the EGSPhant window, right click on the structure name to display a list of possible materials. Assign material names and densities to each structure. Lastly, arrange the fill order such that any overlapping volumes fill larger volumes before smaller volumes.

Clean contour Clean contour sets a specific material and density to all external voxels of the selected contour. This option may be used to remove the CT couch from the MC simulations. This option can be turned on and off.

CT settings auto Fill The auto fill uses the selected CT model to convert the CT HU to a material and density. This option is used when manual fill is turned off. HU are interpolated at the center of each voxel. This is in contrast to averaging the HU over the volume of the voxel.

Manual assignment of voxels Once the EGSPhant voxels have been defined, the user may choose to manual assign all voxels to one material and density. A medium outside the phantom must also be defined. This maybe useful in generating square water phantoms.

CT phantom CT phantoms generated within MMCTP have a specific format. The Y and Z axis within the McGill RT coordinate system are switched to simplify the equations for theta, phi and phicol. Thus the phantom may appear to be upside down in DOSXYZ_show. A CT phantom is generated

Non-CT phantom A non-CT phantom will use the "MC_WaterPhantom_Output.egsinp" as a template input file. Non-CT phantoms do not use the same coordinate systems as the CT phantoms, and the auto-scripts within MMCTP do not update the isocenter, distance or angles. These phantoms are ideal for generating PDDs, output tables or profiles. A PDD is a non-CT phantom defined within the Z axis. The X,Y axis are cross and in-plane directions.

8.3 Treatment Planning - DOSXYZnrc GUI

8.4 DOSXYZnrc tab

This tab displays the DOSXYZnrc simulation status of the selected plan. A listbox sorts egsphant files into columns and beams into rows. The user can access run specific properties for a beam by double clicking on a row. New EGSPhant files are created by right clicking within the lisbox.

The user can submit a DOSXYZnrc run by checking-on the check box for each beam. Note, the DOSXYZnrc auto-run option must be enabled. The

auto-refresh will check the status of all simulations. This is done by checking the individual lock files, and the presence of 3ddose files. If a 3ddose file exists, the simulation is complete and MMCTP will download the file to the corresponding plan folder.

8.4.1 EGSPhant viewer

There is an option to view each EGSPhant file within MMCTP. This feature is useful to ensure the integrity of the EGSPhant file. The user can scroll through the EGSPhant file, reading off the material and density value with the mouse cursor. Loading the entire EGSPhant file into memory takes time and as such there is an option to avoid reading in the voxels. If the voxels are not loaded, the EGSPhant viewer will only display the EGSPhant voxel limits and resolution.

8.4.2 DOSXYZ Properties

The DOSXYZ properties window for a selected beam and egsphant file includes information about the BEAMnrc phase space file, number of histories to run, PEGS file, queue settings, CPU time, number of jobs and the dose normalization values. There is also an option to access more advanced DOSXYZnrc properties.

DOSXYZnrc Test Run The test run will submit a short run to the cluster. This serves to ensure that the run can complete without errors and to determine the CPU time for the desired statistical uncertainty.

8.4.3 3ddose to MMCTP

Once the 3ddose file has been downloaded to the plan folder, the file is ready to be converted into the McGill RT format. This is done by right clicking on the beam in question. MMCTP will reading each 3ddose file, normalize the dose values using the beam weight, number of fractions, MU number, dose calibration value and dose to water value and create a McGill RT dose file.

8.5 3ddose Import

This option was designed for importing 3ddose files from egsphant files which were not created within MMCTP. In general, egsphant files created from

other software packages use a different coordinate system than the one used within MMCTP. This option allows the user to assign each (x,y,z) 3ddose Cartesian axis to a corresponding (x,y,z) MMCTP axis. In addition, the user can apply offsets to each axis. Refer to figure 16 and secton 3.2.2 for a review of the patient coordinate system, and McGill RT dose format.

8.6 DOSXYZnrc Job Submission

MMCTP will submit and refresh the status of DOSXYZnrc simulations. The user has the option of manually submitting a test run. Job submission and refresh are controlled by the global MC period settings and by the DOSXYZnrc control settings

8.6.1 Auto determine shell

If this option is on, MMCTP will determine which shell to submit the current DOSXYZnrc simulation. This option only applies when there are multiple shells and the simulation can run on multiple shells. For example, if the DOSXYZnrc source is 2, then the simulation can only run on the shell where the phsp file is stored. However, if the source number is 9 then the simulation is free to run on all shells there the BEAMnrc linac has been defined. In addition, the script searches for an online shell with the greatest number of available job slots.

8.6.2 Run test DOSXYZnrc

MMCTP will attempt to run a test simulation before submitting a production run. The test run uses a small number of histories to determine the CPU time per history, and the uncertainty per history. These ratios will determine the required number of histories needed to achieve a desired uncertainty and provide an estimate on the simulation time.

8.6.3 Run DOSXYZnrc

Auto run for DOSXYZnrc is controlled by a timer and a user selected DOSXYZnrc auto run check box within the Configurations Window. Lastly the user has to check on the beam number in question within the Treatment Planning Window. In this action, the input file is generated, uploaded and submitted

to a batch system. Input files generated from MMCTP are named according to the patient ID, plan number and beam number and EGSPhant name.

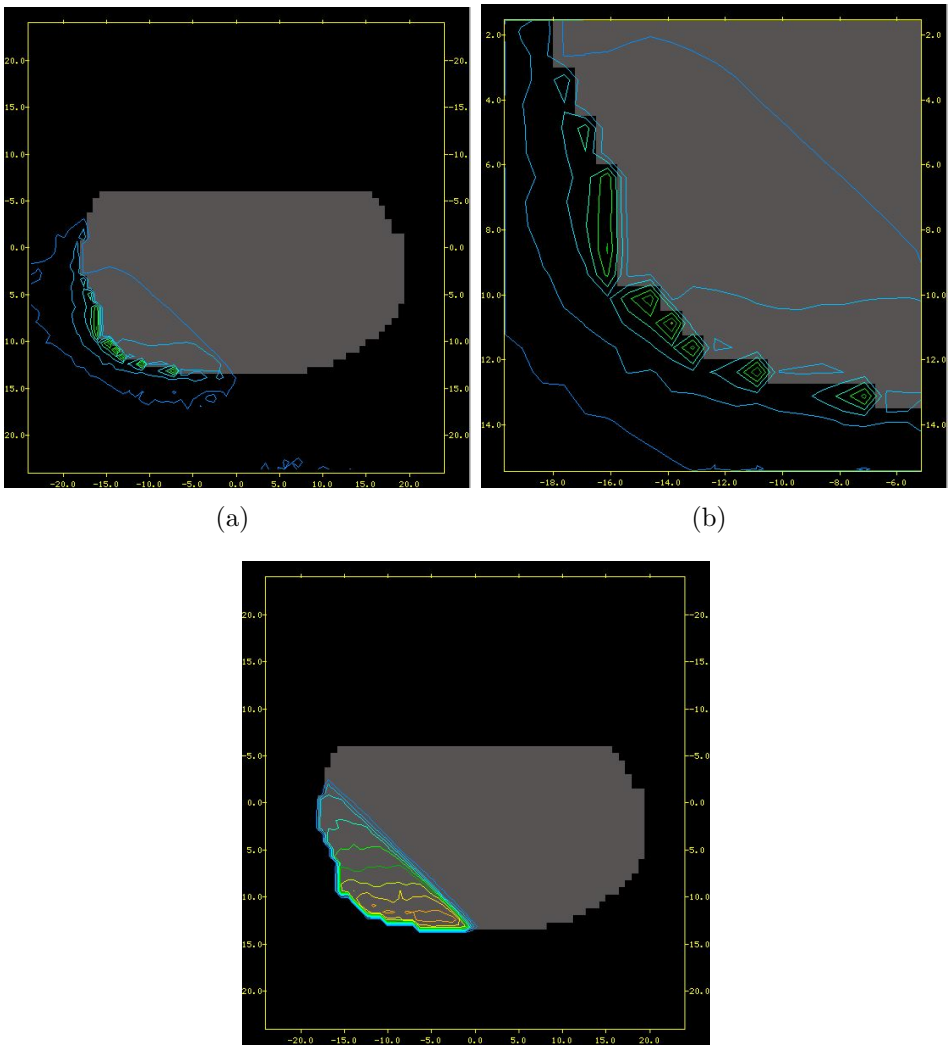
8.6.4 Refresh

Refresh will send a command to check on the simulation status of all running jobs. This is done by reading the $*.lock$ file. If the lock file does not exist, then MMCTP sends a command to check on the status of the $*.3ddose$ file. If the 3ddose file exists, then MMCTP downloads the file to the plan folder.

8.7 Issues

8.7.1 Dose to air

Suppressing the dose to air maybe required to avoid maximum dose values in air, as seen in figure 37.



(a) (b)

(c)

Figure 37: Tangental 6 MV photon beam impinging on homogeneous water patient. Maximum dose values are located within surrounding air voxels from secondary electrons exiting the patient and depositing their energy. This effect can be avoided by suppressing the dose to air.

9 Contour Window

The contour window includes one main canvas for editing within the axial plane, thumbnail image previews, a listbox of structures and display options. Each contour may have multiple segments per slice. Contours are associated with axial images and do not change between plans. The display options allow the user to change the image window and level, the scale size, the contour transparency settings and boolean check boxes to show the image or contours. Changes to the contours, structure names and colours can be saved within the FILE menu.

9.1 Contour editing

The contour window allows the option of editing, creating or deleting contours. For example, bolus may be added as a new contour and included within the patient model. To add or delete contours, right click on the contour listbox. Contour editing features include the following:

- add point
- move point
- delete point
- scale segment
- move segment
- copy segment to superior slice
- copy segment to inferior slice

The user must save all changes by under file menu Save Contours.

References

[1] Andrew Alexander, F Deblois, G Stroian, Khalid Al-Yahya, E Heath, and Jan Seuntjens. Mmctp: a radiotherapy research environment for monte carlo and patient-specific treatment planning. *Phys. Med. Biol.*, 52(13):N297–308, Jul 2007.

[2] Indrin J Chetty, Bruce Curran, Joanna E Cygler, John J DeMarco, Gary Ezzell, Bruce A Faddegon, Iwan Kawrakow, Paul J Keall, Helen Liu, C M Charlie Ma, Dave Rogers, Jan Seuntjens, Daryoush Sheikh-Bagheri, and Jeffrey V Siebers. Report of the aapm task group no. 105: issues associated with clinical implementation of monte carlo-based photon and electron external beam treatment planning. *Med. Phys.*, 34(12):4818–53, Dec 2007.

[3] H Liu, T Mackie, and E McCullough. Modeling photon output caused by backscattered radiation into the monitor chamber from collimator .... *Med. Phys.*, Jan 2000.

[4] Dave Rogers, B A Faddegon, G X Ding, C M Ma, J We, and T R Mackie. Beam: a monte carlo code to simulate radiotherapy treatment units. *Med. Phys.*, 22(5):503–24, May 1995.

[5] Dave Rogers, B R B Walters, and I Kawrakow. Beamnrc users manual. *irs.inms.nrc.ca*, Feb 2007.

[6] Justin Sutherland. Investigating techniques to accurately calibrate non-standard fields. pages 1–120, Apr 2009.

[7] W van der Zee and J Welleweerd. A monte carlo study on internal wedges using beam. *Med. Phys.*, 29(5):876–85, May 2002.

[8] F Verhaegen and Jan Seuntjens. Monte carlo modelling of external radiotherapy photon beams. *Phys. Med. Biol*, Jan 2003.

[9] F Verhaegen, R Symonds-Tayler, H H Liu, and A E Nahum. Backscatter towards the monitor ion chamber in high-energy photon and electron beams: charge integration versus monte carlo simulation. *Phys. Med. Biol.*, 45(11):3159–70, Nov 2000.

[10] B Walters, I Kawrakow, and D.W.O Rogers. Dosxyznrc users manual. *NRC Report PIRS-794*, Jan 2011.

Index