SAND REPORT

SAND2008-3331
Unlimited Release
Printed April 2008

ITS Version 6: The Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes

Revision 1

Brian C. Franke, Ronald P. Kensek, and Thomas W. Laub

Prepared by Sandia National Laboratories Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Officeof Scientific andTechnical Information

P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
NationalTechnical Information Service
5285Port RoyalRd
Springfield,VA22161

Telephone: (800) 553-6847 Facsimile: (703) 605-6900 E-Mail: orders@ntis.fedworld.gov Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2
00-
3331Un
limited
Release
Printed
April
2008

ITSVersion 6:
The Integrated TIGER Series of Coupled Electron/Photon Monte CarloTransport Codes

Revision1

BrianC. Franke, RonaldP. Kensek, and ThomasW. Laub
RadiationTransport Department

Abstract

ITS is a powerful and user-friendly software package permitting state-of-the-art Monte Carlo solution of lineartime-independent coupled electron/photon radiation transport problems, with or without the presence of macroscopic electric and magnetic fields of arbitrary spatial dependence. Our goal has been to simultaneously maximize operational simplicity and physical accuracy. Through a set of preprocessor directives, the user selects one of the many ITS codes. The ease with which the makefile system is applied combines with an input scheme based on order-independent descriptive keywords that makes maximum use of defaults and internal error checking to provide experimentalists and theorists alike with a method for the routine but rigorous solution of sophisticated radiation transport problems. Physical rigor is provided by employing accurate cross sections, sampling distributions, and physical models for describing the production and transport of the electron/photon cascade from 1.0 GeV down to 1.0 keV. The availability of source code permitsthe more sophisticated user to tailor the codes to specific applications and to extend the capabilities of the codes to more complex applications. Version 6, the latest version of ITS, contains (1) improvements to the ITS 5.0 codes, and (2) conversion to Fortran 90. The general user friendliness of the software has been enhancedthroughmemory allocationtoreducetheneedfor userstomodifyandrecompilethe code.

Acknowledgment

There have been many people involved in the development of the ITS codes over the years. We have undoubtedly lost track of the contributions of

some, and even if we had not, there would betoomanytolisthere.ITShasbeen assembledbyborrowingand linking algorithmsfrommany generous sources. We are grateful to those who have shared their methods and grateful also to the many users who have suggested improvements over the years. While recognizing that this is necessarily an abbreviated list, we would like to acknowledge some of the contributions that we deem to have been most significant or most recent.

The ITS codes owe most of their development to John Halbleib. He wrote the TIGER code and was the primary developer of the ITS codes throughout most of their history. The original TIGER code was built on the ETRAN code developed by Steve Seltzer and Martin Berger, both of NIST (the National Institute of Standards andTechnology). Further collaborations with NIST, ledtoimprovementsofthe algorithmsandcross sections usedby ITS.Tom Mehlhorn playedan importantrolein the first integrationof the Integrated TIGER Series.

This release of ITS includes the multigroup capability. The development of MITS (Multigroup ITS) owes much to the contributions of Jim Morel. MITS is also dependent on the CEPXS cross section generating code that was principally developedby Len Lorence.

The parallel capabilityoftheITS codes was implementedbyGregValdez. Much testingand many improvements have been contributedbyWesley Fan. Recent improvementsin the physical modelsin the ITS codes were implementedbyVeronica Klein. Recent improvementsin the regression testing of ITS are due to the coverage analysis work of Lisa Cordova. The Doppler broadening logic was implementedbyTom Quirk. The below1keV logic was implementedby Martin Crawfordbased on suggestions from Ken Adams.

The conversion of the CEPXS code to Fortran 90 was accomplished by Clif Drumm. The conversionoftheXGENandITScodestoFortran90was accomplishedbyMartinCrawford,whohas also contributed numerous features to ITS version 6.

The CAD geometry capability in ITS was originally implemented with the assistance of Steve Warren, now at Kansas State University. The implementation was further refined with the aid of Perry Gray,Fred Gelbard,TimTautges, PaulWolfenbarger, and Matthew Martin. The current engine1 geometry interface with CAD capability was written mostlybyMatthew Martin. The Cholla code for facet geometry was written by John Fowler of Computational Geometry Consulting, Inc.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Contents

Figures

Tables

Intentionally Blank

1
.

I
n
t
r
o
d
u
c
t
i
o
n

t
o

I
T
S

Last Modified Date: 2008/04/17 22:45:40

1 Introduction to ITS

The TIGER series of time-independent coupled electron/photon Monte Carlo transport codes isagroupof multimaterialand multidimensional codes designedtoprovidea state-of-the-artdescription of the production and transport of the electron/photon cascade. The continuous-energy ITS codes arebased primarily on the ETRAN model[1], which combines microscopic photon transport with a macroscopic random walk[2] for electron transport. The multigroup ITS codes are based primarily on the MORSE model[3], with a modification by Sloan[4] to model electron elastic scattering, both of which preserve the angular moments of scattering with discrete scattering angle models. Emphasis is on simplicity of application without sacrificing the rigor or sophistication of the physical model.

1.1 History of the TIGER Series

Table1chronicles the development of the TIGER series, beginning with the EZTRAN[5] and EZTRAN2[6] codes in the early 1970's. These codes were basically user oriented versions of the ETRAN codes. They were severely limited in their application to real physical problems because oftheirrestrictiontoasingle homogeneous material.Overcomingthis limitationwasthe original motivation for the development of the TIGER series.

Table 1. Chronology of TIGER series development

Code	Date	Released	Dimension
EZTRAN	Sep 71	Yes	1-D
EZTRAN2	Oct 73	Yes	2-D/3-Da
TIGER	Mar 74	Yes	1-D
CYLTRAN	Mar 75	Yes	2-D/3-Da
CYLTRANM	Jun 77	No	2-D/3-Da
TIGERP	May 78	Yes	1-D
SPHERE	Jun 78	Yes	1-D
ACCEPT	May 79	Yes	3-D
SPHEM	Jul 79	No	1-D/3-Da
CYLTRANP	Late 81	No	2-D/3-Da
ACCEPTM	Late 81	No	3-D

[a]The first dimensionreferstothe material geometry, whilethe second dimensionreferstothe description of the particle trajectories.

TIGER[7], CYLTRAN[8], and ACCEPT[9] are the base codes of the series and differ primarily in their dimensionality and geometric modeling. TIGER is a one-dimensional multilayer code. CYLTRANemploysafullythree-dimensional descriptionofparticle trajectorieswithinanaxisymmetric cylindrical material geometry and quite naturally finds application in problems involving electronorphotonbeam sources. ACCEPTisageneralthree-dimensional transportcodethat uses the combinatorial-geometry scheme developed at MAGI[10, 11].

The original base codes were primarily designed for transport from a few tens of MeV down to 1.0 and 10.0 keV for electrons and photons, respectively. Furthermore, fluorescence and Auger

1. Introd

processes in the base codes are only allowed for the K-shell of the highest atomic number element inagiven material.For some applicationsitis desirabletohaveamore detailedmodelofthelow energy transport. In the TIGERP[12] and CYLTRANP[13] codes, we added the moreelaborate ionization/relaxation model from the SANDYL code[14] to the TIGER and CYLTRAN codes, and we extended photon transport down to 1.0 keV (all member codes of the ITS system allow transport over the range 1.0 GeV to 1.0 keV).

In CYLTRANM[15], we combined the collisional transport of CYLTRAN with transport in macroscopic electric and magnetic fields of arbitrary spatial dependence using a Runga-Kutta-Fehlberg algorithm[16] to integrate the Lorentz force equations. An important modification of this algorithm[17] made possible the development of the ACCEPTM code[18] , which combines the collisional transport of the ACCEPT code with macroscopic field transport. SPHERE[19]and SPHEM[20] were two special purpose codes that were restricted to multiple concentric spherical shells without and with macroscopic field transport, respectively.

EZTRAN, EZTRAN2, and SPHEM are considered obsolete. Before ITS, that still left us with eight separate code packages to maintain. Five of these – TIGER, CYLTRAN, ACCEPT, TIGERP and SPHERE – had been publiclyreleased and weredisseminated through the Radiation Shielding Information Center at Oak Ridge National Laboratory. CYLTRANM, CYLTRANP and ACCEPTM were not publicly released, but were maintained locally for use throughout Sandia National Laboratories. Maintaining multiple code packages had become quite burdensome for us as well as for users of the codes. As a result, important modifications were not being implemented in a timely fashion. Furthermore, the multiplicity of packages had resulted in uneven development of the various codes such that each code had unique features that had not yet been implemented in the other codes.

In order to remedy this situation we developed ITS (the Integrated TIGER Series), whose full implementation superseded all other versions of the TIGER series codes[21]. The combined program library file was obtained by integrating the eight codes in the first three columns and first threerowsofTable2in suchawayasto minimizetherepetitionof coding thatis common to two or more of these codes. This process led quite naturally to the development of a new code, ACCEPTP. In ACCEPTP, the improved low-energy physics of the SANDYL code was added to the ACCEPT code. Those individual codes appearinginTable1, but notinTable2, wereofa more specialized nature than the others and were no longer supported since their function was duplicated by at least one of the ITS codes. Additional cross-section data and associated logic allowed transportfrom1.0keVto1.0GeV[22]forboth electronsand photons.Anewfree-format, order-independent input procedurebased on descriptive keywords and maximum use of defaults and internal error checking resulted in a very simple and user-friendly input scheme. Integration of the various codes resulted in the availability of additional common options for each code.

Table 2.ITS version5member codes

	Standard Codes	Enhanced Ionization/ Relaxation (PCODES)	Macroscopic Fields (MCODES)	Multigroup (MITS Codes)
TIGER	ITS-TIGER	ITS-TIGERP	N/A	MITS-TIGER
CYLTRAN	ITS-CYLTRAN	ITS-CYLTRANP	ITS-CYLTRANM	N/A
ACCEPT	ITS-ACCEPT	ITS-ACCEPTP	ITS-ACCEPTM	MITS-ACCEPT
CAD	ITS-CAD	ITS-CADP	N/A	MITS-CAD

ITS was further enhanced throughout the 1980’s to improve the physical models, enhance user friendliness, and increase the options available to the user[23]. Beginning in 1994, a set

n

t
o

I
T
S

of multigroup ITS codes (MITS) was developed[24, 25, 26]. The MITS codes inherited the ITS combinatorial-geometry logic, but through alternative physical models, it added a capability for adjoint transport calculations.In1997,webeganthe developmentofacapabilityof trackingparticles on CAD geometries[27]. Throughout the 1990's the original continuous-energy combinatorial-geometry ITS codes were further enhanced with complicated subzoning capabilities, more source options, and flexible biasing schemes. In 2000, ITS, the MITS codes, and the CAD tracking capability were integrated into a single package of codes. In 2002, ITS 5.0 was internally released at Sandia with a verified capability of adjoint particle transport on CAD geometries[28]. The CAD capability was substantially revised in 2004, to improve the efficiency of geometry interrogation queries. Also in 2004, the Cholla capability for using facetted geometry models was implemented. The geometry interface code, engine1, currently allows for ACIS and Cholla geometry. Advanced users may consider linking with their own CAD or other geometry modelers.

Table3showsthe code options availableinthe current versionof ITS.The codesgroupedby rowinTable3willbereferredtoasthe TIGER codes,theCYLTRAN codes,the ACCEPT codes, andtheCADcodes,respectively.Fromlefttoright,thecodesgroupedby columnwillbereferred to as the standardcodes, the PCODES, and the MITS codes, respectively. We acknowledge that some confusion may result from a dual context-dependent use of the term "ITS". In general, we willuse"ITS"tomeanthe complementoftheMITScodes(i.e.,thefirstthree columnsofTable3), and we will use "ITS codes" to refer to the entire series of codes including the MITS codes. The MCODES capabilities have been integrated into the standardcodes and are now enabled by the keywordEBFIELDS.

Table 3.ITS version6member codes

	Standard Codes	Enhanced Ionization/ Relaxation (PCODES)	Multigroup (MITS Codes)
TIGER	ITS-TIGER	ITS-TIGERP	MITS-TIGER
CYLTRAN	ITS-CYLTRAN	ITS-CYLTRANP	N/A
ACCEPT	ITS-ACCEPT	ITS-ACCEPTP	MITS-ACCEPT
CAD	ITS-CAD	ITS-CADP	MITS-CAD

Sincethe initialrelease[29], feedbackfromthe user communityhasbeenofgreat benefittothe developmentoftheITScode system.Asaconsequenceofthis feedback, subsequent versionshave implemented importantimprovementsin physical accuracy, new capabilities, variancereduction, and user friendliness.

2. Overview of the Documentation

Last Modified Date: 2008/04/17 22:45:32

2 Overview of the Documentation

The documentationhasbeen writtenwiththe intentionthatitmaybeusedasareference manual for the expert user and the beginning user alike. Brief overviews are provided on step-by-step execution of the program (in Running ITS), on preprocessor options (in Code Options), and on program keywords (in Summary of ITS Keywords). These may serve as quick references for the expert and as an initial outline for the novice. Each of these are followed by more detailed explanations that in turn may be associated with sections containing further details and/or theory for thecode features.Itis intendedthatall users shouldbeginbyreferringtothe RunningITS section and followreferencesinthe documentationas necessarytogain further understandingof features to be used. Users are encouraged to peruse the entire manual to gain a better understanding of the code package and how various options may be employed. However, some sections apply to restricted subsetsofthecodeoptions availableandmaynotberelevanttotheuser'sspecifictypes of problems.

2.1 Document Sections

The following is a brief description of each of the sections contained in this manual:

- Introduction to ITS: discusses the history of the ITS codes.
- Overview of the Documentation: is this section.

Overview of the ITS Code Package: gives a brief description of the cross section generators and Monte Carlo codes that comprise the ITS code package and a discussion of the code capabilities available.

Installation:gives general guidancefor installingtheITS softwareonaplatform.It discusses some of the modifications that may be necessary in the configure/make system of files.

Running ITS: contains an overview, as well as step-by-step instructions, for executing a calculation with ITS. Some known platform dependencies of the code are listed.

Code Options: contains the preprocessor definitions available for selecting member codes of the Integrated TIGER Series.

Summary of ITS Keywords: contains tables of input keywords available for multigroup forward, multigroup adjoint, and continuous-energy options and the default settings associated with those keywords.

Keywords for ITS: contains an alphabetical listing of the input keywords for all of the ITS codes and descriptions for using each keyword.

Summary of ITS-CAD Keywords: contains a table of keywords available for the parameter file used with CAD calculations and default settings associated with those keywords.

Keywords for ITS-CAD: containsa listingof the parameter keywords for usein the parameter file with the CAD code option and descriptions for using each keyword.

2. Overview of the Documentation

TIGER Geometry: contains the formattingrequirements for inputof 1-D geometry.

CYLTRAN Geometry: contains the formatting requirements for input of 2-D geometry.

ACCEPT Geometry: contains the formatting requirements for input of 3-D combinatorial geometry.

CAD Geometry: contains therequirements for inputof 3-D CAD geometry models.

Output: contains descriptions of the information in each section of an output file.

Suggestions for Efficient Operation: discusses some of the issues involved in using a Monte Carlo code efficiently, such as the proper selection of energy ranges, biasing parameters, and number of particle histories simulated.

PCODES: contains a description of the ITS options for detailed ionization and relaxation modeling.

Electric and Magnetic Fields: contains a description of the field options.

Biasing OptionsandVariance Reduction: contains explanationsofthe biasing settings available in the ITS codes.

Statistics: containsa descriptionof how statistical estimations are performedin ITS.

Automatic Subzoning: contains descriptions of how subzoning is implemented in ITS codes.

Random NumberGenerators: contains descriptionsofthe portable

random number generators implemented.

Adjoint Calculations:providessometheoretical descriptionoftheadjointmodeoftheMITS code option.

Testing of ITS: contains instructions on how to perform installation, commit, and regression tests for ITS.

UnitTesting of ITS: contains descriptions of each of the unit tests for ITS that are contained in the repository and how to perform those tests. This is available as a separate document outside of the manual.

ACIS Library Build for ITS-CAD: contains information on building the ACIS libraries and specific modifications that have been necessary to build the libraries on certain platforms. This is available as a separate document and is only available to Sandia developers.

3. Overview of the ITS Code Package

Last Modified Date: 2008/04/17 22:45:31

3 Overview of the ITS Code Package

ITS consists of three essential code directories:

1 XGEN – the continuous-energy cross-section generation program
2 CEPXS – the multigroup cross-section generation program
3 ITS – the Monte Carlo program

XGEN is used to generate cross sections for the continuous-energy ITS codes. Relatively few physics and modeling decisions are required when these cross sections are generated, but for example, one must choose between the standardcodes and the PCODES. CEPXS is used to generate cross sections for the multigroup ITS codes. Anumber of physics and modeling decisions can be (or must be) made at the time the multigroup cross sections are generated. The heart of ITS is the set of Monte Carlo program files. Here, numerous decisions must be made about the simulation, e.g., problem geometry, physics options, forward vs. adjoint, biasing options, etc.

3.1 New Capabilities Since 5.0

Array allocationistheforemostfeatureaddedtoITSsince version5.0.Therequirementthat ausermodify array-size parametersandrecompilefordifferent calculationshasbeengreatly reduced. Instead,ITSnowreadstheinputfileto dynamically allocatealmostallarraystothe required sizes. While most arrays are dynamically allocated based on the contents of the input file, some array sizes arestill hardwired. Inafew cases, keywords have been made availabletoallowtheusertoincreasethesearraysizeswithoutcode modificationandrecompilation. For example, the CODEZONES-PER-PATH, TRAJECTORY-POINTS, OVERLAP-MAXOUTPUT, and PLOTS-OVAL-RESOLUTION keywords serve this purpose.

Along with automating the memory allocation functions, we have provided the user with keywords that assist in controlling memory usage. The new keywords NO-DEPOSITIONOUTPUT and NO-SZDEPOSITION-OUTPUT can be used to suppress printing deposition in the output file, while the new keywords NO-DEPOSITION and NO-DETAILED-DEPOSITION can be used to limit the tallies and memory allocated for deposition. To limit the memory required for tallying photon flux and escape, line radiation can be tallied with continuum radiation using the new LINE-TALLY-WITH-CONTINUUM and ANNIHILATION-LINETALLY-WITH-CONTINUUM keywords.

Electrons can now be transported in CAD geometries in the continuous-energy codes.

The user can now run the PCODES without uniform shell biasing enabled, and uniform shell biasing is no longer the default behavior of the PCODES. Instead this can be activated with the BIAS-SHELL sub-keywordunder the BIASING keyword.With this change we now recommend that the PCODES be used for almost all calculations.

We have begun to add support for radiation below 1 keV. For now, this is restricted to sourcephotonssampledbelow1keV(i.e., radiationisstillnotproducedbelow1keV)and only absorption is modeled. No scattering of radiation below1 keV is included. See the

3. Overview of the ITS Code Package

BELOW-1KEV keywordfor more details. The new capability for attenuation of source photons below1 keVrequires the Livermore evaluated data library for photons, whichis not distributed with ITS. It is available for download from the IAEA website at http://wwwnds.iaea.org/epdl97/. ITS uses the EPDL97 in ENDF/B-VI format. This file should be located in the XGEN distribution at .../Code/XSdata/epdl97.all.

The efficiency of large combinatorial geometries can be improved using features to read and write the geometry connectivity array. See the READ-CONNECTIVITY and DUMPCONNECTIVITY keywords for more details.

Wehave implementedamoreefficient parallel algorithmandmorerobust parallelerrorhandling. The improved efficiency is most significant for calculations performed on thousands of processors. Dynamic load balancing is now accessed as a keyword, DYNAMIC-MPI, rather than as a preprocessor definition.

The field codes have changed considerably as the MCODES have beenintegrated into the standardcodes. There are now more options for selection of fields without code modification. See the EBFIELDS keywordfor more details. See the following section on changes to inputrequirements also.

The PLOTS capability is now an integral part of the CYLTRAN and ACCEPT codes. PLOTS is invoked only with a keyword, without requiring a unique preprocessor definition. Plottingisnow amenabletousewithspreadsheetsand plottingprogramslikeTecplot R

Material thickness information can be extracted to a separate file while

performing a RAYTRACE calculation. See the AREAL-DENSITY-OUTPUT keywordfor more information.

ITS now performs the proper normalization for volumetric sources in adjoint.

Electron escape and emission data can be output to a PFF formatted file. (The user must supply thepfflibraries.) See the PFF-FILE keywordfor more details.

Deposition data can be written to a separate file in a simple list format with a variety of formatting options. These are options to the FINITE-ELEMENT-FORMATkeyword.

ITS now allows other files to be “included” in an input deck with the INCLUDE-FILE keyword. This allows a user to eliminate duplicate information between input decks. E.g., one file with the combinatorial geometry data used for multiple runs with differing output data requirements.

Adiffutility capable of suppressing differences in numerical data below specified tolerances is now included with ITS. This can simplify the task of performing installation testing of ITS and XGEN on a new platform.

3.2 Changes to Input Requirements Since 5.0

- MCODES was removed as a preprocessor definition and is replaced by the new EBFIELDS keyword. MCODES input decksrequired enabling fields on the geometry line for CYLTRAN and on the material line for ACCEPT. This is no longer allowed, and the EBFIELDS keyword allows the user to define the zones where fields are enabled or disabled using OFF (was value ‘0’), MAGNETIC (was value ‘1’), or BOTH (was value ‘2’).

BIAS-GLOBAL and BIAS-ZONE are obsolete keywords. The BIASING keyword provides equivalent functionality.

DETAIL-IONIZE is an obsolete keyword. This functionality is no longer available.

Last Modified Date: 2008/04/17 22:45:39

4 Installation

As noted in the Code Overview, there are three components in the ITS code package:

1 XGEN – the continuous-energy cross-section generation program
2 CEPXS – the multigroup cross-section generation program
3 ITS – the Monte Carlo program

These code directories should be included in any ITS release.

4.1 Unix, Linux, and Mac Installation

In this section we describe the anticipated *one time only* modifications that may be required to install the ITS package on a new platform. These modifications are made in the configure and Makefile systems. No modifications are anticipated for the codes themselves. Note that the following instructions are written for the ITS code directory, but apply similarly to the XGEN and CEPXS code directories. There should be little difference in the config files for ITS, XGEN, and CEPXS.

WARNING: Before performing any modifications to the files provided, we strongly recommend that an unaltered copy be stored! This will allow for future determinations of code modifications that were necessary.

From within the its/Code directory, execute "./configure". The script will respond with a statement of the form "Configuring for a *-*-* host." Only the last of the three variables (the operating system) is important for configuring. If header and target files exist and are found in the config directory, the configure script will print a statement of the form: 'Created "Makefile" using "config/mh-*" and "config/mt-*"'. If the script prints the statement 'Created "Makefile"' but does not state which header and target files were used, then the necessary files do not exist, the configure failed, and the Makefile will not function.

If the configure failed, the scripts must be modified to use config files customized to the platform. The simplest approach is simply to modify the config/mh-custom and config/mt-custom files to supply the necessary information specific to the platform. The configure command "configure -host=i386-custom" will include the mh-custom and mt-custom files in the Makefile. The "i386" does not affect the Makefile and can be used regardless of the system you are working on. For using the testing and sendn scripts, the nm6CVS file (located in its/Scripts, but also to becopiedto $HOME/bin)mustbealteredtoreplacethetext "PLACEHOLDER"withthesystem response to the command "uname -n".

A more complicated approach is to create header and target files specific to the platform and identifiable by the configure script. As an example, the configure script may identify the host as rs6000-ibm-aix4.3.2.0. This indicates that the operating system has been identified as aix4.3.2.0. The configure script will use the config files config/mh-aix and config/mt-aix to create the

Makefile. On the other hand, if the configurescripts wereto identify the host as i686-unknownexample3.2, then the user would need to create scripts named “config/mh-exa” and “config/mtexa”,or “config/mh-example3.2”and “config/mt-example3.2”,or such.Itwillbe easiesttocreate the new scripts by modifying two of the existing configscripts, so that the required variables and formats are available. (The mh-custom and mt-custom are well commented for this purpose.)

4. Installation

When new config files are created, it is necessary to modify the configure.in file, so that the configurescriptcanfindthenew files.Theuseofan asteriskattheendofthe nameoftheconfig files will indicate whether the name of the operating system must be identified exactly or if only the given leading characters need to be identified. There are several target files (“config/mt-*”) for which the exact name must be identified.

Therearetimeswhenitis desirabletobeabletocompileindifferentwaysonasingle platform. For example, one may wish at times to compile ona solaris machine torun on that same machine, in which case the host and target machines are the same. Or one may wish to (cross-)compile on the solaris machine to run on another machine, in which case the host and target machines are different. In this case, one can create a configure file for the target machine that can be selected as a flag when configuring. One then only needs to specify the target machine in the configure commands (e.g., “configure -target=i386-tflops”). If one is adding a new unique target option, it is necessary to add the option to the configure.in file and to the config.sub file. The changes requiredin the config.sub file maybe identifiedby searching for “tflops”. Specificationofa target machineby namecanbe usefulnotonlyforcross compilingbutalsofor installingthe softwareon an unusual instance of an operating system, while maintaining the target config file for the usual operating system.

There are a few modifications that might be needed to use the CAD capability in ITS with the Cholla libraries. The Cholla library has its own makefile system that is invoked by the ITS makefile system. The name of the compiler (CCC from the mt-* file) and the optimization/debug flag (CCFLAGS

SCRIPT from the Defines.mk file) are passed into the Cholla makefiles. However, if there are compile flags that are required on your system for compiling C++ code, these must be specified in the Cholla makefiles. These modifications must be included in three files: Code/Cholla/Makefile, Code/Cholla/facetbool/makefile, and Code/Cholla/primitives/makefile.

When the Makefile has been successfully created, one must attempt to make an executable. To do this, it is necessary to specify a valid set of preprocessor definitions. This can be done in Defines.mk. Changestothisfilewillnotbeaffectedbythe configure command.TheTemplate.mk file shouldnotbe modified,asit servesasabackupforthe Defines.mk.Valid optionsare listedin this file. Examples of valid options are “RNG = RNG1”, “OPT1 = MITS”, and “OPT2 =”.

An executable can then be produced by executing “gmake”. The ITS makefile system requires a version of make that supports conditionals. The makefile system is currently hardwired to use gmake. If you have an appropriate make utility (e.g., one that handles conditionals), you will need to alias gmake to invoke your make utility. If gmake is not successful, it will be necessary to identify the settings inthe Defines.mk and/or config files that need to be modified. Our general experienceisthattheproblemis(1)mostlikelytobeinthe specificationofpreprocessor definitions in the Defines.mk file or the compilers in the target config file, (2) possibly in other settings in the target config file, and (3) least likely to be in the host config file or Makefile.in.

We then recommend that the installation tests be run to determine that the software is functioning properly on the new platform. To run the tests it is necessary to copy the sendn and nm6CVSfilesfromtheScriptsdirectorytoa$HOME/bindirectory.Itisalso necessarytocopythe Subscripts directory to $HOME/bin/Subscripts. If creating a Makefile requires a command other than simply “./configure” without flags, it may be necessary to modify the $HOME/bin/nm6CVS scripttorecognizethe platform dependency.Aguideforrunningthe installation testsisprovided in the section onTestingof ITS.

When the installation tests have been successfully executed, the softwarehas been successfully installed.

Finally, we recommend that the user consider the advantages of importing the software into a

o
n

CVS version control system[31]. The original files provided should be imported first. Then any modifications required to install the software on the platform can be committed to the repository. This provides a convenient mechanism for storing and tracking future code modifications and extracting changes that have been made to the code over time.

4.2 PC Installation

These installation instructions are for the CG version of ITS usingVisual Studio. The instructions were tested usingVisual Studio 2005 with the Intel Fortran compiler 10.1.011 [IA-32]. These instructions can easily be adapted to XGEN or CEPXS.

Withthecodedirectory installedonyourPC,createanewVisual StudioProject:

File->New->Project
Select Intel(R) Fortran
SelectVisual Studio InstalledTemplate: EmptyProject.
Name: its

•

Location: (base directory)
Uncheck the Create directory for solution.
Click [OK]. Add theproject files into the ITSVisual StudioProject:
For the files in the Code/Source directory:
•
– Click on the ITS project.
– Right-click on Source Files folder in the project tree.
– Select menu item Add->Existing Item.
– Select Source iteminVisual Studio installed templates.
– Browse to the Code/Source directory.
– Select all the FORTRAN files (*.F).
– Click on the [Add] button.
Repeat for the files in the Code/Interface directory.
Repeat for the files in the Code/Modules directory (but OMIT the test-*.F files). Set the type of compilation for the project:
Select menu item Project->its Properties.
Use Configuration drop-list box to select Release, Debug, or All Configurations. Change the compile definition for *.F files to use 132 character lines, free form:
Select menu item Project->its Properties.

4
.

I

n
s
t
a
l
l
a
t
i
o
n

Configuration Properties->Fortran->Language->Source File Format: “Use Free Format”. Set the Data configuration:
Configuration Properties->Fortran->Data->Default Integer KIND:4
Configuration Properties->Fortran->Data->Default Real KIND:8
Configuration Properties->Fortran->Data->Default DoublePrecision KIND:8
Configuration Properties->Fortran->Data->LocalVariable Storage: AllVariablesSAVE
Configuration Properties->Fortran->Data->Initialize stack var...:Yes

Optionally, set run-time checking. The following settings will cause the code to run slower but may help detect coding errors:

Configuration Properties->Fortran->Run-time->GenerateTraceback Information:Yes
Configuration Properties->Fortran->Run-time->Check Array and String Bounds:Yes
Configuration Properties->Fortran->Run-time->Check UninitializedVariables:Yes Add Include Directory:
Configuration Properties->Fortran->General->Additional Include Directories:

.../Code/Source Define the CodeTypePreprocessor definitions:
Select menu item Project->its Properties.
Turn on the preprocessor: Configuration Properties->Fortran->Preprocessor->Preprocess Source File->Yes

- Add the code-specific compiler flags: Configuration Properties->Fortran->Preprocessor Select from the following and enter as comma separated (e.g., “ITS RNG1,ITS CYLTRAN”):
  - Required: ITS RNG1, ITS RNG2, or ITS RNG3
  - Required: ITS TIGER, ITS CYLTRAN, or ITS ACCEPT
  - Optional: ITS MULTIGROUP (not allowed with ITS CYLTRAN)
  - Optional: ITS PCODES (if using a PCODES version of XGEN cross sections)
  - Optional: ITS USE PFF

(requires PFF libraries) Optionally,

change the directory and name for the

executable:

Create the directory, e.g., C:/tmp/its
Change the executable name:

Configuration Properties->Linker->Output File:

C:/tmp/its/its.exe Build your executable:

Build->Build Solution (or Build its)

4. Installation

To test your build:

Copy the cross section file to the executable directory
E.g., copy ../../Tests/RegTests/cross3/taala to C:/tmp/its/fort.11
Copy the input file
E.g., copy ../../Tests/RegTests/Input/itscyl1.inp to C:/tmp/its/
Open a command window.
Change to the C:/tmp/its directory
run “its.exe itscyl1.inp output” from a command prompt.

Optionally,torunusingtheVisual Studioasthe debugger:

Modify Configuration Properties->Debugging->Command Arguments: itscyl1.inp output
Modify Configuration Properties->Debugging->Working Directory: C:/tmp/its Verifing the output:
E.g., diff(... full well-defined path ...)/Tests/RegTests/Output/itscyl1 1.out output

If you want to perform more tests you can match input, cross sections, and output by examining the installation.pm, commit.pm,andregression.pm filesinthe.../Tests/RegTestsdirectory. Expect some differences because:

Some lines have been removed from the stored output files.

Some leading 0's have been added to the stored output files.
There could be some round-offdifferences between different platforms.
We frequently see very small numbers being identically zero on different platforms.

5
.

R
u
n
n
i
n
g

I
T
S

Last Modified Date: 2008/04/17 22:45:46

5 Running ITS

This document contains outlines for running the ITS code. Instructions are provided for running the code by using commands and running the code by using scripts either with or without access to the CVS repository. Known platform dependencies of the code are discussed in the last section.

5.1 Running ITS without Scripts

CROSS SECTIONS: Cross sections must be generated either with XGEN (for the continuous-energy codes) or with CEPXS (for the multigroup codes). The fort.11 file is needed by ITS.

CHECKOUT: Do a "cvs checkout -P its" to acquire a copy of the code on the platform where the repository is located. If necessary, tar the directory and transfer it to the desired platform.

CODE MODIFICATIONS: Code modifications can be made before building an executable.

MAKEFILE SETTINGS: In the directory its/Code, you must alter the Defines.mk settings to specify the necessary definitions for your build of the ITS executable. There are no valid defaults! An unaltered copy shouldremain storedasTemplate.mk.

CONFIGURE: Execute "./configure". If the code has been previously

installed correctly, then the platform and operating system will be identified, and the proper config/mh-* file and con-fig/mt-* file will be included in the Makefile. It may be necessary to specify a target platform when configuring.

MAKE: Execute the Makefile with the “gmake” command to produce the ITS executable, its.x.

INPUT FILE: An ITS input file must be constructed. Examples are available for each code option in its/Tests/RegTests/Input. See the section on Keywords for ITS for additional information.

CAD FILES:If performingaCAD calculation,aprmfile mustbe constructedand either satfiles or facetfiles must be provided to specify the geometry.

EXECUTION: The command “its.x mdat output” executes the ITS code, where mdat is the ITS input file. (The files its.x, fort.11, and mdat are required.) If no input file name is specified, the input file defaults to “its.inp”. If no output file name is specified, the output file defaults to “its.out”. If performing a CAD calculation, the command “its.x prmfile” executes the ITS code. (The files its.x, fort.11, prmfile, mdat, and satfiles or facetfiles are required.)

EVALUATE RESULTS: The output file should be evaluated to determine if the run was successfulandiftheresultsare satisfactory.Foran unsuccessful calculation,anabortcallwillusually be indicated at the end of the output file. It may be necessary to look for errors in the output file (search for “>>>>>”). The abort may have been postponed until all input had been read, and one error may be the cause of additional errors.

5
.

R
u
n
n
i
n
g

I
T
S

5.2 Running ITS without Scripts -Details

CROSS SECTIONS: Cross sections must be generated either with XGEN (for the continuous-energy codes) or with CEPXS (for the multigroup codes). Unless the FILE-NAMES keyword is used in the ITS input, the file must be

named “fort.11”. More detailed discussion of generating cross section files can be found with the documentation for the XGEN and CEPXS codes.

CHECKOUT:Doa“cvs checkout-Pits”toacquireacopyofthemostrecent versionofthecode on the platform where the repository is located. It is recommended that a -P flag be used when checkingoutacopyofthecodeto avoid acquiringemptydirectoriesthat correspondto outdated directorystructure.Avarietyofdateandreleasetagscanbeusedtorequestolder versionsofthe code. See CVS documentation at http://ximbiot.com for additional information. If necessary, tar the directory and transfer it to the desired platform.

CODE MODIFICATIONS: If desired, code modifications can be made before building an executable. If you do not have direct access to the CVS repository, it is strongly recommended that an unaltered version of the code be stored for tracking any intentional or unintentional alterations that you make.

MAKEFILE SETTINGS: In the directory its/Code, you must alter the settings in the Defines.mk file to specify the necessary definitions for your build of the ITS executable. There are no valid defaults! An unaltered copy shouldremain storedinTemplate.mk. Instructions for setting the preprocessor definitions are included in the Defines.mk file and in the Code Options section of this manual. It is important that options be set under the proper variables (e.g., MITS should be selected under OPT1, MPI should be selected under OPT2, etc.). Therearesome options for which it is valid to leave the definition blank, and they are illustratedin Defines.mk (e.g., OPT1 can be MITS, PCODES, or blank).

CONFIGURE: Execute “./configure”. If the code has been previously installed correctly, then the platform and operating system will be identified, and the proper config/mh-* and config/mt
* files will be included in the Makefile. It may be necessary to specify a target platform when configuring. See Section 5.5 for information on known platform dependencies.

MAKE: Execute the Makefile with the “gmake” command to produce the ITS executable, its.x. If the code has been installed on the platform correctly,the correct commands will be used. The ITS makefile system requires a version of make that supports conditionals (it is currently hardwired to use gmake).

INPUT FILE: An ITS input file must be constructed. Examples are available for each code option in its/Tests/RegTests/Input. The keywords relevant to specific code options and their defaults are given in the Summary of ITS Keywords section. Specific instructions for formatting each keywordin the ITS input are given in the Keywords for ITS section.

CAD FILES: If performing a CAD calculation, a prmfile must be

constructed and satfiles or facetfiles must be provided. Instructions for setting the CAD parameters in the prmfile file are provided in the Keywords for ITS-CAD section. One or more satfiles should contain any desired CAD geometry in ACIS SATformat, and facetfiles should contain any desired CAD geometry in CUBIT facet format.

5.

Running ITS

EXECUTION: The command “its.x mdat output” executes the ITS code, where mdat is the ITS input file. (The files its.x, fort.11, and mdat are required to perform a calculation.)

If no input file entered as a parameter to the command, the input file defaults to “its.inp”. If no output file entered asa parameter to the command, the output file defaults to “its.out”.

If performingaCAD calculation, the command “its.x prmfile” executes the ITS code. (The files its.x,fort.11,prmfile,andmdatarerequiredtoperformacalculation,aswellasanygeometryfiles. The names of the mdat file, satfiles, and facetfiles are specified in the prmfile.)

EVALUATE RESULTS: The output file should be evaluated to determine if the run was successfulandiftheresultsare satisfactory.Foran unsuccessful calculation,anabortcallwillusually be indicated at the end of the output file. It may be necessary to look for errors in the output file (search for “>>>>>”). The abort may have been postponed until all input had been read, and one error may be the cause of additional errors. If an error statement is generated, then an anticipated problem has been found in the input and/or cross sections, and diagnostic information should be available. If the code generates an execution error, the user may have introduced a bug viaacode modification.IfabugisfoundinITSthatwasnotintroducedbythe user,pleasenotify the ITS developers providing enough details to reproduce and understand the error. Preferably this would include a description of the error observed, the version of the code being used, diffs showing any code modifications made, and the input and output files demonstrating the bug.

5.

R
u
n
n
i
n
g

I
T
S

5.3 Running ITS with Scripts

CROSS SECTIONS: Cross sections must be generated either with XGEN (for the continuous-energy codes) or with CEPXS (for the multigroup codes).

CHECKOUT: Do a “cvs checkout -P its” to acquire a copy of the code on the platform where the repository is located. If necessary, tar the directory and transfer it to the desired platform.

BACKUP: Making an unaltered backup copy can be very important for tracking code changes when working on a platform that does not have access to the repository.

SCRIPTS: The scripts sendn and nm6CVS that are located in its/Scripts must be copied to $HOME/bin. The scripts in its/Scripts/Subscripts must be copied to $HOME/bin/Subscripts.

CUI FILE: Copy the its/Scripts/its.cui file to your working directory, and edit it for your specific problem. The sections of a cui file are:

1 driver script -the default nm6CVS is usually acceptable.
2 defs -selects code options. See Code Options for more information.
3 prmfile -only required for CAD calculations. See Keywords for ITS-CAD for information.
4 satfile -onlyrequired for CAD calculations with ACIS geometry(repeated for each file).
5 facetfile -only required for CAD calculations with facet geometry (repeated for each file).
6 diffs -this section must be present. Code modification patches may be specified.
7 mdat -contains the ITS input. See Keywords for ITS for more information.

SENDN: Submit the job using the command “sendn its.cui <jobname>”. This script will prompt you for 3 or 4 pieces of information. The information consists of the cross section file tobeused, whetherthecodewillbe compiledinan existingdirectoryorfromacvs checkout,and how to document code modifications.

EXECUTION: If calculations are performed in “interactive” mode (requiring the user to execute the code), the files will be locatedin $HOME/tmp/<jobname>. The code will be executed as “its.x mdat output” for non-CAD calculations and “its.x prmfile” for CAD calculations.

POSTPROC: If calculations are performed in “interactive” mode, the postproc script must be executed in the $HOME/tmp/<jobname> directory to complete the script process.

EVALUATERESULTS:Theoutputfileshouldbe evaluatedto determineiftherunwas successful and if the results are satisfactory. For an unsuccessful calculation, an “ohoh<jobname>.job” file willbereturned. Information willbe includedinthe ohoh file that may indicate the sourceof the error. Additional information may be found in the $HOME/tmp/<jobname> directory. The mlog2 file contains most of the information generated while the scripts were run. If the program compiled successfully but an error was generated during the execution of ITS, an abort call will usually be indicated at the end of the output file. It may be necessary to look for errors in the output file (search for “>>>>>”). The abort may have been postponed until all input had been read, and one error may be the cause of additional errors.

5
.

R
u
n
n
i
n
g

I
T
S

5.4 Running ITS with Scripts -Details

CROSS SECTIONS: The sendn script will request the name of a cross section file. For the continuous-energy codes, the cross section file must be located in $HOME/cross3. For the multi-group codes, the cross section file must be located in $HOME/crossm and must be given a name with the extension “.11”. More detailed documentation on generating cross section files can be found with the XGEN and CEPXS codes.

CHECKOUT:Doa“cvs checkout-Pits”toacquireacopyofthemostrecent versionofthecode on the platform where the repository is located. It is recommended that a -P flag be used when checkingoutacopyofthecodeto avoid acquiringemptydirectoriesthat correspondto outdated directorystructure.Avarietyofdateandreleasetagscanbeusedtorequestolder versionsofthe code. See CVS documentation at http://ximbiot.com for

additional information.

BACKUP:Forworkingon platformsthatdonothavedirect accesstotherepository,itisrecommendedthattwo checkoutcopiesbesetup:onecopytobeusedasaworkingdirectoryinwhich code modifications and builds can be performed, and one unaltered copy that can be compared withto maintainarecordof changesmadetothe working version.The unalteredcopy shouldbe givenanameuniquelyindicatingthecvs version.Forexample,ifthecheckoutwasperformedas (cvs checkout -D ”February1, 2002” its), then the unaltered copy might be named “its01Feb2002”. The name of the directory will be the indication of the version, which is a very important key to repeating a calculation at some later time.

SCRIPTS: The scripts sendn and nm6CVS that are located in its/Scripts must be copied to $HOME/bin. The scripts in its/Scripts/Subscripts must be copied to $HOME/bin/Subscripts. Sendn is a script for launching jobs, nm6CVS is a driver script for performing jobs, and the Subscripts are utilities used by sendn and nm6CVS. Sendn will position the cui file, including the sections of the driver script. The driver script contains the commands necessary to build and execute the program, clean up after itself, and produce relevant result information in a job file. (If necessary, it will also include in the job file information useful in determining the cause of a job failure). $HOME/bin and “.” should be included in your $PATH environment variable.

CUIFILE:The its/Scripts/its.cuifileis availableasan example.Youcancopyittoyourworking directory, and edit it for your specific problem. Other examples are available inTests/Reg-Tests/CUI. The portions of a cui file are:

1 DRIVER SCRIPT: The script nm6CVS is available as a driver script. You may substitute a customized script by including it in the first portion of the cui file.

2 DEFS: Instructions for setting the preprocessor definitions areincludedin the “defs” portion of the its.cui file and in the Code Options section. It is important that options be set under the proper variables (e.g., MITS should be selected under opt1, and MPI should be selected under opt2), because the script will look (i.e., grep) for these specific settings. There should be no spaces in the settings.

There are several options for running the scripts that may be set in the defs portion of the
cui file. Compiler flags may be specified. The user may also request an interactive script. The interactive script should be used on systemsthatrequirejobqueuingorcross compiling.Thefirsthalfofthescriptwillbuildthe executable. The files for running the job will be located in $HOME/tmp/<jobname>. The executable is named “its.x”, the input file is named “mdat”, the parameter file for ITS-CAD is named“prmfile”, and the program output should be directed to a file named “output”.

When the calculationis complete,the user may execute the “postproc” filein the $HOME/tmp/<jobname> directory. This will produce a job file in the directory from which the job was originally submitted.

1 PRMFILE: Instructions for setting the CAD parameters in the “prmfile” portion of the its.cui file are provided in the Keywords for ITS-CAD section. This portion is only required for CAD calculations. If the full path for any satfiles or facetfiles is included in the listing in the prmfile, the satfile and facetfile sections arenot necessary;they areonlyrequired when those files must be located in the same directory as the executable.

2 SATFILE: The “satfile” portion is used to transfer the desired ACIS CAD geometry to the location in the tmp directory where the calculation will be performed. Only two lines are required. The first line should include the desired *.sat file containing the CAD geometry. The second line may contain anything but must contain a return. This portion is only necessary for including ACIS geometry in CAD calculations. The name given to this section (typically “satfile.sat”) should be the name referenced in the prmfile. It is possible to have multiple satfiles as long as each section has a unique name and is referred to appropriately in the prmfile.

3 FACETFILE: The “facetfile” portion is used to transfer the desired Cubit facet CAD geometry to the location in the tmp directory where the calculation will be performed. Only one line is required specifying the path and name of the file containing the facet geometry. This section is only necessary for including facet geometry in CAD calculations. The name given to this section (typically “facetfile.fac”) should be the name referenced in the prmfile. Since one may only include a single zone in each facet file, it is often desirable to have multiple facet files. This is easily accomplished as long as each section has a unique name and is referred to appropriately in the prmfile.

4 DIFFS: Patches to be applied to the code should be inserted in the “diffs” portion of the its.cui file. These patches can be formatted as a cvs diff or as a directory diff. Diffs can be lifted out of job files from previous calculations. Alternatively, one can checkout a copy of the code, make modifications, and then generate a diffs file. On a platform with access to the CVS repository, one can use “cvs diff > diffs” from within the its/Code directory. On a platform without accesstotheCVSrepositoryonecanperformadirectorydiff,butthismust be performed from within the its/Code directory of the unaltered copy of the code, and it must use the -r and -b flags (in that order), such as “diff -r -b . $HOME/itsaltered/Code> diffs”. Then, the diffs file can be used in the its.cui file. Multiple difffiles (diffs from two different executionsofthediffcommand) cannotbe patchedtoa single cvs file.

New files can be added to a build. Within the diffs portion, a line of the following format denotes the start of a new file:

New file: <Directory/Filename> To be included in the compiling and linking of the code, the new file must be referenced in the Makefile.in list of sources. This change in the Makefile.in can also be included in the diffs portion by making the desired change in a copy of Makefile.in and using the cvs diff procedure described above.

7.MDAT:TheITSinputshouldbe includedinthelastportionofthecuifile.Thekeywordsrelevant to specific code options and their defaults are given in the Summary of ITS Keywords

5. Running ITS

section. Specific instructions for formatting each keywordin the ITS input are given in the Keywords for ITS section.

SENDN WITHOUT CVS: Submit the job using the command “sendn its.cui <jobname>”. This scriptprompts you for the following3piecesof information:

1 CROSS SECTIONS: First, it requests the name of the cross section file <cross> to be used. For the continuous-energy codes, the script looks for the file at “$HOME/cross3/<cross>”. For the multigroup codes, the script looks for the file at “$HOME/crossm/<cross>.11”.

2 LOCAL COMPILE: Second, sendn requests the location of a copy of ITS that can be used for making the executable. Thus, you may use a version of the code that you have checked out and modified. You must give a complete pathname for the base ITS directory (e.g., /scratch/temporary/its). No files will be deleted from the make directory as a result of the calculation, but some files may be modified if requested in the diffs section of the cui file. Any new files that have been included in the directory (other than through the diffs section of the cui file) will not appear in the job file, but they may be used if the Makefile.in has been so modified(inwhichcase,thejobfilewillshowthereferencetothenewfileasadifference in the Makefile.in). Attempts to apply diffs in a directory where the diffs have already been applied foraprevious calculation willresultin an error.

3 DIRECTORYDIFF: Next, the script requests a local directory with

which to perform a diff. The diffcommand will be used to compare the two directories and all subdirectories. The job file will not record the version number of either directory, therefore the user may not have enough information in the job file to duplicate a calculation unless the diff directory has a name corresponding to the tag used in the cvs checkout of the code. The directory name appears in the job file.

SENDN WITH CVS: Submit the job using the command “sendn its.cui <jobname>”. This script prompts you for the following 3 pieces of information (and possibly the 4th depending upon yourresponses to the first 3):

1 CROSS SECTIONS: First, it requests the name of the cross section file <cross> to be used. For the continuous-energy codes, the script looks for the file at “$HOME/cross3/<cross>”. For the multigroup codes, the script looks for the file at “$HOME/crossm/<cross>.11”.
2 LOCAL COMPILE:Second,sendnrequeststhe locationofachecked-outcopyofITSthatcan be used for making the executable. Thus, you may use a version of the code that you have checked out and modified. You must give a complete pathname for the base ITS directory (e.g., /scratch/temporary/its).Nofileswillbe deletedfromthemakedirectoryasaresultof the calculation,but some filesmaybe modifiedifrequestedinthediffs sectionofthecui file. Any new files that have been included in the directory (other than through the diffs section of the cui file) will not appear in the job file, but they may be used if the Makefile.in has been so modified(inwhichcase,thejobfilewillshowthereferencetothenewfileasadifference in the Makefile.in). Attempts to apply diffs in a directory where the diffs have already been applied foraprevious calculations willresultin an error.

CVS COMPILE: To request a cvs checkout of the code, you may respond to this request with “none” (or press “Enter”). The cvs checkout will be performed in the $HOME/tmp/<jobname> directory and should not affect any other versions of the code.

5. Running ITS

3. CVS DIFF for LOCAL COMPILE: Enter “none” or simply press “Enter” with no input for this third request. VERSIONforCVS COMPILE:Ifyourresponsetothesecondrequestwas“none”,thescript

requests the version for a cvs checkout. The command will be issued as "cvs checkout <options> its/Code". The response to this request will be used for the <options> and any valid cvs flags may be used that do not contain a slash, "/". Examples of valid syntax for responses are: -D now, -D "March 28, 2001", -D "3 hours ago", -D "2 fortnights ago", -r ITSversion5.0, etc. Another validresponseisto simplypress "Enter" withno input, which will result in the checkout of the most recent version of the code.

4.CVSDIFFforLOCAL COMPILE:Ifyougaveapathnameformakingthe executablebutnot for a directory diff, the script requests the version for a cvs diff. The syntax for responses to this request are the same as for specifying the cvs version for a checkout. The directory in which the executable is made will be compared with the repository using the cvs diff command. This version (that will appear in the job file) and the results of the cvs diff(that willalsoappearinthejobfile)canbeusedtoreproducea calculation.

CVS UPDATE/DIFF for CVS COMPILE: If you did not give a pathname for making the executable, the scriptrequestsa version fora cvs update and diff. In this case, the script will checkoutaversionofthecodeusingtheoptionsinthethirdresponse,applythe" diffs"from the cui file, attempt to update to the version of the code specified in this response, and then compare the resulting code to the repository using the cvs diffcommand with the options specified in this response. The version requested here (that will appear in the job file) and the results of the cvs diff (that will also appear in the job file) can be used to reproduce a calculation.

For either of these, if no option is specified for the cvs diff, "-D now" will be used. The date and time of the submission of the calculation, which are recorded in the job file, and the results of the cvs diff can be used to reproduce the calculation at a later date.

EXECUTION: On platforms where the interactive script is used because cross compiling is required, it may be necessary to move the files to perform the calculation. However, it will be necessary to move the files back to their original location after execution.

If executionfailsorifaftertheruntheoutputfileisfoundto containerrors,itmaybe desirable to use the tmp directory as a working directory to debug the code or calculation. However, after the calculation has been performed successfully, it is likely that one must resubmit the job with correctedinputsothatthejobfilewill accuratelyreflectthecode modificationsandtheinputused to perform the calculation.

POSTPROC: When the interactive script is used, postproc must be executed to produce a job file. The job file will be placed in the directory from which the job was originally launched, just as with the non-interactive script.

EVALUATE RESULTS: Errors may occur at a number of stages in the calculation. The stage at which the error occurred is usually indicated near the start of the ohoh file. Some common causes of errors are:

1 Configure -If the platform has not been used before, the necessary config files may not be present.
2 Make -Ifthedefshavenotbeenproperly specifiedorTemplate.mkhasbeen modified,the definitions may not have beenproperly setby the scripts. TheTemplate.mk must contain certainwordstobereplacedbythe scripts.

5
.

R
u
n
n
i
n
g

I
T
S

3. Execution -If an error statement is generated (search in the output for ">>>>>"), then an anticipated problem has been found in the input and/or cross sections, and diagnostic information should be available. Such an error might also result if the Template.mk was modified, and an executable with the wrong code options is being used. If the code generatesan executionerror,the usermayhaveintroducedabugviaacode modification.Ifabug is foundin ITS that was not introducedby the user, please notify the ITS developersproviding enough details to reproduce and understand the error. Preferably this would include a description of the error observed, the version of the code being used, diffs showing any code modifications made, and the input and output files demonstrating the bug.

5.5 Platform Dependencies

The following is a list of platforms on which ITS has been successfully built and tested. Following the name of the platform is the system type and operating system.

- Red Storm (Catamount)

  The driver script is functional for building the executable on a compile node in interactive
  mode.
  Jobs are submitted to the queue using the qsub command.
  If not using the scripts, the configurecommand must specify Red Storm as the host platform:

configure -host=i386-redstorm

ACIS12 is available for CAD calculations.

Thunderbird(Linux)

- The driver script is functional for building the executable on a compile node in interactive mode. Jobs are submitted to the queue using the qsub command. If not using the scripts, the configure command must specify Thunderbird as the host plat

f

o

r

m

:

c

o

n

fi

g

u

r

e

-

h

o

s

t

=

i

3

8

6

-

t

b
i
r
d

A
C
I
S
1
2

i
s

a
v
a
i
l
a
b
l
e

f
o
r

C
A
D

c
a
l

c
u
l
a
t
i
o
n
s
.

- Purple or Up (AIX)

  The driver script is functional for building the executable on a compile node in interactive
  mode.
  Jobs are submitted to the queue using the psub command.

  psub -ln <nodes> -g <processors> -tM <max time> its.com If not using the scripts, the configure command must specify Purple as the host platform: configure -host=i386-purple ACIS15 is available for CAD calculations. The ACISR15 libraries were provided by Spatial (rather than built compiling the source code on purple).

- QA,QB(HP Alpha,Tru64 OSF1V5)

  The driver script is functional for building the executable in interactive mode. ACIS12 libraries areavailable for CAD. The ITS C++ CAD interface code should be compiled with the

5
.

R
u
n
n
i
n
g

I
T
S

debug flag. Jobs are submitted to the queuing system using the bsub command. The ACIS library path must be set in the config/mt-osf5 file for running on QSC or QT.

- Crater (AMD Athlon, Linux) The driver scriptis functionalinall modes.WithMPI,the scriptissetto use2processors.

The mpirun command should be used to specify the number of processors for parallel calculations.

Pegasus (64-bit Intel, Linux) The driver scriptis functionalinall modes.WithMPI,the scriptissetto use2processors.

The mpirun command should be used to specify the number of processors for parallel calculations.

PC

- See the Installation section for guidance on installing ITS on a PC.

6
.

C
o
d
e

O
p
t
i
o
n
s

Last Modified Date: 2008/04/17 22:45:44

6 Code Options

Numerous code options can be selected for compiling the ITS codes. These options have been implemented as preprocessor definitions. It is necessary to choose between the multigroup (MITS) codes and the continuous-energy codes. (The selection between forwardand adjoint mode in the MITS codes is made as an input option.) Other options include choosing between the 1-, 2-, and 3-dimensional geometry representations and choosing

a random number generator.

The code options must be specified in either the CUI file (if using the scripts to execute the code) or in the Defines.mk file (if building an executable). Refer to the section on Running ITS for more details on applying definitions. Abbreviated versions of these definitions are used in the CUI and Defines.mk.

6.1 Preprocessor Definitions

ITS MULTIGROUP (torequest the multigroup codes; omitted for continuous energy codes)

Select the code as one of:

ITS TIGER (1-D)
ITS CYLTRAN (2-D cylindrical geometry; 3-D transport)
ITS ACCEPT (3-D)

Select the random number generator as one of:
ITS RNG1 (generator in 3.0, only available in serial)
ITS RNG2 (RANMAR)
ITS RNG3 (MersenneTwister, only availablein development version)

To select to run on a parallel platform:
ITS USE MPI

Other available options are:
ITS PCODES (more ionization and relaxation for continuous energy codes)
ITS USE PFF (pfformatted output)
ITS CAD (for linking with Cholla or ACIS CAD geometry)
ITS USE CHOLLA (for facet geometry)
ITS USE ACIS12 (the path for ACIS libraries is specified in the config/mt-* file)

6.2 Definition Requirements

ITS RNG1, **ITS RNG2**, or **ITS RNG3** must be selected as the random number generator.
ITS TIGER, **ITS CYLTRAN**, **ITS ACCEPT**, or **ITS CAD** must be selected as the code.
ITS CAD mustbe accompaniedby **ITS ACCEPT** (automatic in the makefile).
ITS CYLTRAN is not currently functional for **ITS MULTIGROUP**.
ITS RNG1 cannot be used with **ITS MPI**.
ITS USE PFF cannot be used with **ITS MULTIGROUP**.
ITS CAD must be used with **ITS USE ACIS12** or **ITS USE CHOLLA**.

7.

Summary of ITS Keywords

Last Modified Date: 2008/04/17 22:45:41

7 Summary of ITS Keywords

This section contains listings of keywords relevant to the ITS (continuous-energy) codes, the MITS codesin forwardmode, and the MITS codesin adjoint modeinTables4,5, and6,respectively. Only those keywords applicable to the code and mode are listed in each table. Many keywords in forwardmode do not apply to adjoint mode and vice versa. The keywords are listed approximately in order of importance. In addition, for each keywordthe default code behavior is listed.The default behaviorwillbeemployedbythecodeifthekeywordisnotfoundintheinput deck.

The secondarykeywords used for biasing are listedinTable7. These are secondary keywords for the BIASING keyword. The sub-keyword, limitations on the code and mode with which the sub-keyword can be used, and the default code behavior are listed in the table. In addition, the global and zone-dependent features of the sub-keywordare listed.

More detailed descriptions of the syntax, sub-keywords, and use of these keywords are contained in the Keywords for ITS section. In some cases, these keywords or their defaults depend uponthecodeoption(preprocessor definition)beyondthe choiceofMITSorITS.Alistingofthe available preprocessor definitions is contained in the section on Code Options.

Table 4.ITS keywords and default settings

KEYWORD	DEFAULT
**** GEOMETRY ****	
GEOMETRY	required
**** SOURCE ****	
ELECTRONS or PHOTONS	electron source
ENERGY or SPECTRUM	1.0 MeV monoenergetic
POSITION	point source at origin (TIGER and ACCEPT) on axis at minimum-z (CYLTRAN)
DIRECTION	monodirectional source in positive-z direction
CUTOFFS	electrons: 5% of maximum;photons: 0.01 MeV
**** OUTPUT OPTIONS ****	
ELECTRON-EMISSION	off
ELECTRON-ESCAPE, PHOTON-ESCAPE	off
ELECTRON-FLUX, PHOTON-FLUX	off

PULSE-HEIGHT	off
ESCAPE-SURFACES	all escape surfaces
PLOTS (CYLTRAN and ACCEPT Only)	off
**** COMMONLYUSED OPTIONS ****	
TITLE	blank title
HISTORIES or HISTORIES-PER-BATCH	1000 histories
BATCHES	20 batches
BIASING	no biasing parameters are activated
EBFIELDS	no fields

7. Summary of ITS Keywords

Table4(continued).

KEYWORD	DEFAULT
**** RARELYUSED OPTIONS ****	
DYNAMIC-MPI (MPI Only)	processor load balancing
TASKS (MPI Only)	number of processors available
RESTARTand/or DUMP	no restart and no restart dump
PRINT-ALL or NO-INTERMEDIATE-OUTPUT or NO-DEPOSITION-OUTPUT or NO-OVERLAP-OUTPUT or OVERLAP-OUTPUT-MAX or NO-SZDEPOSITION-OUTPUT or NO-GEOMETRY-TABLE	final batch in output; intermediate in fort.12 all results data written to output
NO-DEPOSITION or NO-DETAILED-DEPOSITION	detailed dose and charge deposition
RANDOM-NUMBER	0(converted to 5^{19})
NEW-DATA-SET	1run
INCLUDE-FILE	all data in single input file
FILE-NAMES	default names (fort.3, fort.11, etc.)
FINITE-ELEMENT-FORMAT(ACCEPT Only)	no fort.3 output
PFF-FORMAT(ITS PFF Only)	no pffoutput
REFLECTION-ZONE (ACCEPT Only)	no reflection zone
DEPOSITION-UNITS (ACCEPT Only)	dose in MeV per source particle charge deposition in electrons per source particle
ECHO	on
BELOW-1KEV	off

LINE-TALLY-WITH-CONTINUUM or ANNIHILATION-LINE-...	line radiation is tallied separately for photon escape and flux
DUMP-CONNECTIVITY (ACCEPT Only)	off
READ-CONNECTIVITY (ACCEPT Only)	off
CODEZONES-PER-PATH (ACCEPT Only)	20
TRAJECTORY-POINTS (CYLTRAN and ACCEPT Only)	500
PLOTS-OVAL-RESOLUTION(ACCEPT Only)	360
**** DEVELOPMENT USE ONLY ****	
CUTOFF-PHOTONS-ESCAPE	energy of cutoffphotons is deposited locally
DOPPLER	No Doppler broadening
NO-COHERENT	photon coherent scattering is simulated
NO-INCOH-BINDING	binding effects in incoherent scattering are included
NO-KICKING	terminal processing includes kicking
NO-KNOCKONS	secondary knock-on electrons
NO-STRAGGLING	energy-loss straggling
RESTART-HISTORY	no restart
SIMPLE-BREMS	more accurate bremsstrahlung distributions

7. Summary of ITS Keywords

Table 5.MITS forwardkeywords and default settings

KEYWORD	DEFAULT
**** GEOMETRY ****	
GEOMETRY	required
**** SOURCE ****	
ELECTRONS or PHOTONS	electron source
ENERGY or SPECTRUM	mono-group source in the highest-energy group

POSITION	point source at origin
DIRECTION	monodirectional source in positive-z direction
CUTOFFS	bottom of the lowest-energy group for each species
**** OUTPUT OPTIONS ****	
ELECTRON-ESCAPE	off
ELECTRON-FLUX	off
PHOTON-ESCAPE	off
PHOTON-FLUX	off
ESCAPE-SURFACES	all escape surfaces
PLOTS (CYLTRAN and ACCEPT Only)	off
**** COMMONLYUSED OPTIONS ****	
TITLE	blank title
HISTORIES or HISTORIES-PER-BATCH	1000 histories
BATCHES	20 batches
BIASING	no biasing parameters are activated

7. Summary of ITS Keywords

Table5(continued).

KEYWORD	DEFAULT
**** RARELYUSED OPTIONS ****	
TASKS (MPI Only)	number of processors available
DYNAMIC-MPI (MPI Only)	processor load balancing
RESTARTand/or DUMP	no restart and no restart dump
PRINT-ALL or NO-INTERMEDIATE-OUTPUT or NO-DEPOSITION-OUTPUT or NO-OVERLAP-OUTPUT or OVERLAP-OUTPUT-MAX or NO-SZDEPOSITION-OUTPUT or NO-GEOMETRY-TABLE	final batch in output; intermediate in fort.12 all results data written to output

NO-DEPOSITION or NO-DETAILED-DEPOSITION	detailed dose and charge deposition
RANDOM-NUMBER	0(converted to 5_{19})
NEW-DATA-SET	1run
MICRO	deposition calculated via flux-folding
INCLUDE-FILE	all data in single input file
FILE-NAMES	default names (fort.3, fort.11, etc.)
FINITE-ELEMENT-FORMAT(ACCEPT Only)	no fort.3 output
REFLECTION-ZONE (ACCEPT Only)	no reflection zone
DEPOSITION-UNITS (ACCEPT Only)	dose in MeV per source particle charge deposition in electrons per source particle
ECHO	on
LINE-TALLY-WITH-CONTINUUM or ANNIHILATION-LINE-...	line radiation is tallied separately for photon escape and flux
DUMP-CONNECTIVITY (ACCEPT Only)	off
READ-CONNECTIVITY (ACCEPT Only)	off
CODEZONES-PER-PATH (ACCEPT Only)	20
PLOTS-OVAL-RESOLUTION(ACCEPT Only)	360
**** DEVELOPMENT USE ONLY ****	
CUTOFF-PHOTONS-ESCAPE	energy of cutoffphotons is deposited locally
RESTART-HISTORY	no restart

7. Summary of ITS Keywords

Table 6.MITS adjoint keywords and default settings

KEYWORD	DEFAULT
ADJOINT	forward
**** GEOMETRY ****	
GEOMETRY	required
**** DETECTOR ****	
DETECTOR-RESPONSE	required
**** SOURCE OUTPUT OPTIONS ****	
SOURCE-SURFACES	all escape surfaces
SPECTRUM	only a flat forwardspectrum is used
ELECTRON-SURFACE-SOURCE	off
ELECTRON-VOLUME-SOURCE	off
PHOTON-SURFACE-SOURCE	off

PHOTON-VOLUME-SOURCE	off
**** COMMONLYUSED OPTIONS ****	
TITLE	blank title
HISTORIES or HISTORIES-PER-BATCH	1000 histories
BATCHES	20 batches
BIASING	no biasing parameters are activated
PLOTS (CYLTRAN and ACCEPT Only)	off
**** RARELYUSED OPTIONS ****	
TASKS (MPI Only)	number of processors available
DYNAMIC-MPI (MPI Only)	processor load balancing
CUTOFFS	top of the highest-energy group for each species
RESTARTand/or DUMP	no restart and no restart dump
PRINT-ALL or NO-INTERMEDIATE-OUTPUT or NO-OVERLAP-OUTPUT or OVERLAP-OUTPUT-MAX or NO-GEOMETRY-TABLE	final batch in output; intermediate in fort.12 all results data written to output
RANDOM-NUMBER	0(converted to 5^{19})
NEW-DATA-SET	1run
INCLUDE-FILE	all data in single input file
FILE-NAMES	default names (fort.11, fort.12, etc.)
REFLECTION-ZONE (ACCEPT Only)	no reflection zone
AREAL-DENSITY-OUTPUT(ACCEPT Only)	no areal density file
ECHO	on
CODEZONES-PER-PATH (ACCEPT Only)	20
PLOTS-OVAL-RESOLUTION(ACCEPT Only)	360
**** DEVELOPMENT USE ONLY ****	
RESTART-HISTORY	no restart

7. Summary of ITS Keywords

Table 7. Biasing sub-keywords, default settings, and properties

KEYWORD	DEFAULT	Global/Local
BIAS-SHELL (PCODES Only)	natural line radiation	global
COLLISION-FORCING	natural photon interactions	no global settings; specified by zone
ELECTRON-RR (ForwardOnly)	natural number of photon-produced secondary electrons are followed	global Russian Roulette probability; activated by zone
NEXT-EVENT-ESCAPE	off, unless using PHOTON-ESCAPE or PHOTON-SURFACE-SOURCE	global
PHOTRAN (ForwardOnly)	secondary electrons are tracked	no secondary electrons set globally; exceptions by zone
SCALE-BREMS (ITS Only)	natural bremsstrahlung production	global scaling factor; activated by zone (activates SCALE-IMPACT)
SCALE-EP (MITS Only)	natural electron-to-photon production	global scaling factor; activated by zone
SCALE-IMPACT (ITS Only)	20% of brems scaling if used; otherwise, natural impact ionization	global scaling factor; activated in zones with SCALE-BREMS
SCALE-PE (MITS Only)	natural photon-to-electron production	global scaling factor; activated by zone
TRAP-ELECTRONS (ForwardOnly)	no trapping above cutoff energy	both global and local; the more stringent applies
	**** DEVELOPMENT USE ONLY ****	
ELECTRAN (ForwardOnly)	secondary and scattered photons are tracked	no secondary/scattered photons set globally; exceptions by zone
NO-BANK (MITS Only)	secondary particles are banked	global

8. Keywords for ITS

Last Modified Date: 2008/04/17 22:45:24

8 Keywords for ITS

The input keywords must be specified in either the CUI file (if using the scripts to execute the code) or in the input file (if executing the code manually). Refer to the documentation on Running ITS for more details on specifying the input file in the CUI file or otherwise.

8.1 Input Notation

The keywords that are appropriate to use depend upon the code options that have been selected in building the executable (and whether the MITS code is being run in forward or adjoint mode). An overview of the preprocessor definitions is available in the Code Options section. An overview of the keywords that apply to MITS forward, MITS adjoint, and the forwardcontinuousenergy ITS codes is available in the Summary of ITS Keywords section. In this section, following each keywordareanyrestrictions on the code options or mode. The designation “ITSOnly”refers tothe continuous-energycodes(i.e.,notMITS).The designation “ForwardOnly”referstobothethe MITS and ITS codes, unless otherwise noted. All other designations refer to the preprocessor definitions used in building the executable.

Most primary keywords are order-independent. The two exceptions to this are the NEWDATA-SET keywordand the SUBZONE-ONLYusage of the GEOMETRYkeyword.

Most keywords must be used once and not repeated in an input file. Exceptions are BIASING, ECHO, and NEW-DATA-SET. Most sub-keywords should be used only once per use of their primary keyword. Some exceptions arethe sub-keywords of ESCAPE-SURFACES, SOURCESURFACES, and GEOMETRY.

Parameters are associated with the preceding keywordappearing on the same line. If parametersare omitted, default valuesmayapply. Consideration shouldbegiventothefactthatin some situationsthis defaultvalueis invalidandwilltriggeranerror.Valuesexpectedonlines following a keyword are not optional, unless otherwise stated.

Comments may be inserted in the input deck. Anything appearing to the right of an asterisk anywhere in the input deck will be treated as a comment and ignored by the code.

Input is not case sensitive, with only one significant exception. File names entered with the FILE-NAMES keywordwill be used exactly as provided in the input deck. Character input provided with the TITLE and DEPOSITION-UNITS keywords will appear in the output file exactly as provided, but this input does not affect the performance of the code.

8.2 Keywords

1. **ADJOINT** *(MITS Only*)

Syntax: ADJOINT

Default: Forward

This keywordtriggers adjoint transport. It can be used at any point in the input deck. Forward and adjointruns canbe mixed, but for adjoint calculations the adjoint specification must be made for each new-data-set.

8. Keywords for ITS

Adjoint moderequires3mandatory keywords: GEOMETRY,DETECTOR-RESPONSE, anda surface or volume source. The quantity of interest must be specified using the DETECTORRESPONSE keyword with one of the secondary keywords CHARGE, DOSE, ESCAPE or KERMA. The forward source(s) must be described using one or more of the source keywords with prefix PHOTON-or ELECTRON-and suffix SURFACE-SOURCE or VOLUMESOURCE. If a surface-source is requested, the user may further select surfaces using the SOURCE-SURFACES primary keyword. For ACCEPT, the default is to use, in a combined tally, all surfaces through which a particle enters the escape zone. For TIGER, the default is that both surfaces are specified.

2. **ANNIHILATION-LINE-TALLY-WITH-CONTINUUM** *(Forward Only*)

Syntax: ANNIHILATION-LINE-TALLY-WITH-CONTINUUM Default: Annihilation line radiation is tallied separate from the continuuum for photon escape and flux.

This keywordtriggers annihilation line radiation to be tallied in the

corresponding energy bin of the continuum radiation for photon escape and flux.

3. **AREAL-DENSITY-OUTPUT** *(Adjoint Raytrace Only*)

Syntax: AREAL-DENSITY-OUTPUT

Default: No areal density data file

This keyword triggers the creation of a file containing areal density data for the rays in a raytrace calculation.Itcanbeusedatanypointintheinputdeck.Thiskeywordrequiresadjoint mode and that a raytrace calculation (see the DIRECTION-SPACE secondary keyword of the PHOTON-SURFACE-SOURCE keyword)is being performed. Bydefault the file name is “fort.4” but the file name can be changed using the FILE-NAMES keyword.

The format of the areal density data file is as follows. The first record contains the text “location:” followed by the x, y, and z coordinates of the point where the rays originate. Each record after the first contains the angle number, the polar angle (degrees) measured fromthe positivezaxis,the azimuthalangle measuredfromthe positivexaxis,andthe total areal density (g/cm^2)ofall material encounteredbytheray until escapefromthe geometry.

4. **BATCHES**

Syntax: BATCHES [parameter(1)]

Example: BATCHES 100

Default: 20 batches

Number of batches of primary particles to be run. [parameter(1)] batches are performed in order to obtain estimates of statistical uncertainties. Each batch contains an equal number of source particles selected either by the HISTORIES keyword or by the HISTORIESPER-BATCH keyword. Accuracy of the estimates degrades substantially for fewer than 10 batches. Although increasing the numberof batches improves this accuracy, it also increases the overhead(run time).Werecommend thatat least20 batchesbe used.

5. **BELOW-1KEV** *(PCODES Only*)

8. Keywords for ITS

Syntax: BELOW-1KEV PHOTONS [parameter(1)]

Example: BELOW-1KEV PHOTONS 1.0E-3

Default:Nocross sectionsbelow1keV.NoLLNLcross sectionsused.

WARNING: Thisisa limited capability! Read someof the limitations below. This keywordindicates that photon energies below1 keV may be included in the calcula
tions. [parameter(1)] is the transition energy (in MeV) from the XGEN cross-section data to

the LLNL EPDL97 cross-section data set. To use this feature, cross sections must be generated in XGEN using the BELOW-1KEV keyword. The new capability for attenuationof source photons below1 keVrequires the Livermore

evaluated data library for photons, which is not distributed with ITS. It is available for download from the IAEA website at http://www-nds.iaea.org/epdl97/. ITS uses the EPDL97 in ENDF/B-VI format. This file should be located in the XGEN distribution at .../Code/XSdata/epdl97.all.

Only photon absorption is modeled using the LLNL data. Thus, absorption is the only
photon physics that will be modeled below the transition energy.

Only source particles will be generated below1 keV (i.e., there is no scattering to or
photonproduction below1keV).

Cross sections below1keV containhigh levelsof uncertainty. Usersare encouragedto
consider how these cross section uncertainties may be amplified to even greater uncertainty in their Monte Carlo results.

6. **BIASING**

Syntax: BIASING Default: No biasing parameters are activated. Photons, electrons, and positrons are followed globally (if data is included in the cross sections). Electron trapping is performed at the electron cutoff energy.

Selectively turns on the input-zone-dependent bias parameters specified (Photon Collision
Forcing, Photran, Russian Roulette, Scale Electron-to-Photon Interactions, Scale Photon-to-
Electron Interactions, and/or ElectronTrapping). This primary

keywordmayberepeated,
but some secondary keywords should not be repeated.

Restrictions: ELECTRON-RR is disallowed in adjoint. The user cannot simultaneously spec
ify SCALE-EP and SCALE-PE in the same input zone.
Note that SCALE-EP and SCALE-PE always refer to scaling the forward cross section, even
in adjoint. Thus, when using SCALE-PE in adjoint, an electron-adjuncton would be more
likely to turn intoa photon-adjuncton.

(a) **BIAS-SHELL** *(PCODES Only*)

Syntax: BIAS-SHELL

Example: BIAS-SHELL
Default: Samples line radiation based on the actual probability.

8. Keywords for ITS

This keywordactivates a biasing option that samples line radiation uniformly among all possible lines. Photon weights are adjusted based on the actual probability of sampling the given line radiation.
Using this biasing option can make PCODES moreefficient for the calculation of flux or escape of line radiation. However,this option can make PCODES less efficient for the calculation of quantities like energy

deposition, where the importance of line radiation is likely to be proportional to the actual probability.

(b) **COLLISION-FORCING** Syntax:

COLLISION-FORCING

[parameter(1)]

Example: COLLISION-FORCING5

14-68

0.3 0.2 0.1 0.5 0.1 15 Default:

Photon cross sections determine interaction probabilities. This keywordspecifies the photon forced interaction probabilities. This keywordmust be followed by two sets of [parameter(1)] numbers. The first set contains the input

zones for which forced interaction probabilities are to be specified. The second set specifies the forced probabilities for the corresponding input zones. This keywordmay be repeated, but prior settings may be overwritten. For each zone

the last setting in the input deck will be used.

(c) **ELECTRAN** *(Forwar*d *Only*) Syntax:

ELECTRAN Example: ELECTRAN 1, 4-6, 8Default: Secondary and scattered photons arefollowed. If photon cross sections are not provided for MITS, then this keywordis unnecessary. Thiskeywordindicatesthatelectron-produced secondaryphotonsarenottobe tracked and scattered photons arenot to be tracked. (Coherent scattering of photons is allowed; the NO-COHERENT keywordcan be used to deactivate this physics.) Additional parameters are optional and specify exceptions. Zones in which secondary photons are to be tracked are listed beginning on the following line. Adash indicates that all zones between two numbers are included. Beginning the

list with a dash includesall zonesfrom1tothe indicated zone. Endingthe list witha dash includesall zones from the last number given to the last zone number. Using ELECTRANwithaphotonsourceprovidesthe equivalentofafirst-collisionelectron source. Using ELECTRAN with an electron source provides electron-only transport.

The ELECTRAN secondary keyword can be repeated.

(d) **ELECTRON-RR**

(Forward Only)

Syntax:

ELECTRON-RR [parameter(1)] [keyword]

Example:

ELECTRON-RR 0.1 CUSTOM-RR 14-689

Default: Russian Roulette is not used. Natural photon-to-electron cross sections are used.

8. Keywords for

r

I
T
S

[parameter(1)] is the Russian Roulette survival probability used in determining the number of photon produced secondary electrons followed. If [parameter(1)] is omitted or 0.0, Russian Roulette will be used such that the natural number of electrons (the number produced if SCALE-BREMS or SCALE-EP had not been used) would be followed, if Russian Roulette and SCALE-BREMS or SCALE-EP were used throughout the problem. The keyword **CUSTOM-RR** indicates that customized Russian Roulette logic has been included by the user in function FLRRK. The additional parameters set the zones for which Russian Roulette is to be turned on. The list of input zones beginning on the following line specify the regions of the problem where Russian Rouletteistobe used. The fractionofphotonproduced secondary electrons to be followed is the inverse of the scaled bremsstrahlung production. The ELECTRON-RR secondarykeywordmayberepeated,but [parameter(1)]willonly be set to the value in the last occurrence of the keyword.

(e) **NEXT-EVENT-ESCAPE**

Syntax: NEXT-EVENT-ESCAPE [keyword] Example: NEXT-EVENT-ESCAPE OFF Default: Feature is off, unless photon-escape or photon-surface-source is specified. With this keyword, a more efficient calculation of photon escape can be made. For differential escape scoring, this featureis automatically activated. The use of this keyword with the **OFF** sub-keywordcan be used to deactivate next-event-escape

logic when differential photon escape is requested.

(f) **NO-BANK** *(MITS Only*)

Syntax: NO-BANK Default: Secondary particles are banked and relative weights are kept at unity. With this secondary keyword, secondary particles arenot banked. Rather,their weights are changed to account for multiplicity and absorption.WARNING: Using this feature is strongly discouraged.

(g) **PHOTRAN** *(Forward Only*) Syntax:

PHOTRAN Example: PHOTRAN 1, 4-6, 8Default: Secondary electrons are followed. If electron cross sections are not provided for MITS, then this keywordis unnecessary. The keywordPHOTRAN indicates that photon-produced secondary electrons are not to be tracked. Additional parameters are optional and specify exceptions. The zones in which secondary electrons are to be tracked are listed beginning on the following line. Adash indicates that all zones between two numbers are included. Beginning the list withadash includesall zonesfrom1tothe indicated zone. Endingthelistwithadash includes all zones from the last number given to the last zone number. Using PHOTRAN with a electron source is

allowed, since it only excludes tracking of secondary electrons. Using PHOTRAN with a photon source can provide photon-only transport, if no exception zones are specified. The PHOTRAN secondary keywordmay be repeated.

8. Keywords for ITS

(h) **SCALE-BREMS** *(ITS Only*) Syntax: SCALE-BREMS [parameter(1)] [parameter(2)] Example: SCALE-BREMS 500.02 4-6,9 Default: Natural bremsstrahlung cross sections are used. [parameter(1)] is a scale factor used to modify bremsstrahlung production so as to increase the photon population without increasing the number of primary histories. For example, if [parameter(1)] is set equal to two, then there will be twice as much bremsstrahlung photon production. The ELECTRON-RR secondary keyword can be used to control the number of secondary electrons generated by this increased population of photons. [parameter(2)]

is the index of the material, according to the order in which the materials are read from the cross section file, on which the impact ionization scaling is based if SCALE-IMPACT is not used. The default is material number 1. The additional parameters beginning on thefollowing line specify those zones in which bremsstrahlung biasing is activated.

The SCALE-BREMS secondary keywordmay be repeated, but [parameter(1)] and [parameter(2)] will only be set to the value in the last occurrence of the keyword.

(i) **SCALE-EP** *(MITS Only*) Syntax: SCALE-EP [parameter(1)] Example: SCALE-EP 2.0 4,5,6, 8-9 Default: Natural electron-to-photon cross sections are used. This keywordspecifiesthe factorby whichthe electron-to-photoncross sectionsareto be scaled and in which input zones the scaled cross sections are to be used. [parameter(1)] is the scaling factor. The additional parameters beginning on the following line are the input zones in which the scaled cross sections are applied.

The SCALE-EP secondary keyword may be repeated, but [parameter(1)] will only be set to the value in the last occurrence of the keyword.

(j) **SCALE-IMPACT** (*ITS Only*) Syntax: SCALE-IMPACT [parameter(1)] Example: SCALE-IMPACT 20.0 Default: Natural probability of electron impact ionization, except that if the SCALE-BREMS keywordhas been used, then the scale factors for electron impact ionization will be based on the bremsstrahlung scaling (such that for an electron slowing from the maximum energy to the global electron cutoff energy for every five bremsstrahlung interactions there will be one electron impact ionization event in the material specifiedby [parameter(2)]of SCALE-BREMS). [parameter(1)] is used to scale electron impact ionization so as to increase the photon population (line radiation) without increasing the number of primary histories. The cross sections are scaled such that an electron slowing from the maximum energy to the global electron cutoff energy will, on the average, undergo a number of ionization

events equal to [parameter(1)] in each material. The values by which the cross sections may be scaled in every material are written to output. The cross sections will never

s

f
o
r

I
T
S

be scaled down (such that fewer ionization events are simulated than with the natural

cross sections).
WARNING: Impact ionization scaling is only activated in zones in which SCALE-BREMS
is activated.

(k) **SCALE-PE** *(MITS Only*)
Syntax: SCALE-PE [parameter(1)]
Example: SCALE-PE 0.5 3-7 Default: Natural photon-to-electron cross sections are used.
This keywordspecifiesthe factorby whichthe photon-to-electroncross sectionsareto be scaled and in which input zones the scaled cross sections are to be used.
[parameter(1)] is the scaling factor.
The additional parameters are the input zones in which the scaled

cross sections are
applied.

The SCALE-PE secondary keyword may be repeated, but [parameter(1)] will only be set to the value in the last occurrence of the keyword.

(l) **TRAP-ELECTRONS** *(Forward Only*)

Syntax: TRAP-ELECTRONS [parameter(1)] [parameter(2)]

ITS Example: TRAP-ELECTRONS 0.25
1-5

0.25 0.25 0.25 0.35 0.35

MITS Example: TRAP-ELECTRONS425
14-68
40 35 40 35 35

Default: The trap-electron energy is the electron cutoff energy.

ForITS,theglobalelectrontrappingcutoffspecifiedby[parameter(1)]isenergyinMeV. For MITS, the global electron trapping cutoffis the lower energy bound of the group specified by [parameter(1)]. This keyword must be followed by two sets of [parameter(2)] numbers. The first set contains the input zones for which electron trapping energies are to be specified. The second set specifies the energy below which trapping is tested for the corresponding input zones. In ITS the electron trapping energy is specified directly in MeV. In MITS the corresponding energy group is specified. (The lower energy group bound is used in forward, and the upper energy group bound is used in adjoint.) The TRAP-ELECTRONS secondary keywordmay be repeated, but [parameter(1)] and [parameter(2)] will only be set to the value in the last occurrence of the keyword. Electrontrapping is either ineffective or not implemented correctly yet for adjoint calculations. Electron trapping does not function with Cholla facet geometry. The code will always assume that an electron is not trapped if it is in a part that contains facetted geometry using the Cholla geometry library. The keyword **CUSTOM-TRAP** indicates that customized trapping logic has been includedby the userin subroutineSAVE.

7. **CODEZONES-PER-PATH** *(ACCEP*T *Only*)
8
.

K
e
y

w
o
r
d
s

f
o
r

I
T
S

Syntax: CODEZONES-PER-PATH [parameter(1)]

Example: CODEZONES-PER-PATH 50

Default: 20.

With this keyword, the user can increase the number of zones that a particle may pass through.Achangeisrequired only when specifiedby ITS output.

8. **CUTOFF-PHOTONS-ESCAPE** *(Forward Only*)

Syntax: CUTOFF-PHOTONS-ESCAPE

Defaults: Energy of photons below the cutoff energy is deposited locally.

This keywordspecifies that photons falling below the cutoff energy are assumed to escape from the problem. A diagnostic in the output states the average energy of photons per history that is assumed to have escaped from the problem.

Use of the ELECTRAN biasing featurecauses all electron-produced photons and all scattered photons (except for coherent scattering; see the NO-COHERENT keyword) to be considered below the cutoff energy.

9. **CUTOFFS**

Syntax: CUTOFFS

[parameter(1)] -

[parameter(6)]

Example: CUTOFFS

0.01 0.12 * For ITS

0.5 0.2

Example: CUTOFFS484950230 * For MITS

13

45 45

134

45 45 40

Defaults: In ITS the global electron cutoff energy equals 5% of the maximum source energy, and the global photon cutoff energy equals 0.01 MeV. In MITS forwardmode, the cutoff group is the last group for each species. In MITS adjoint mode, the cutoff group is the first group for each species.

This keyword specifies the global cutoff energy for electrons [parameter(1)] and photons [parameter(2)] (and only in MITS, [parameter(3)] is reserved for another particle type). In MITS forwardmode, the cutoff energy is the lower energy bound of the specified group. In MITS adjoint mode, the cutoff energy is the upper energy bound of the specified group.

This keyword can also be used to specify local cutoffgroups. The numbers of input zones for which local cutoff groups are to be specified are given for electrons [parameter(3)] (or in MITS, electrons [parameter(4)], photons [parameter(5)], and a reserved data position). For each of these parameters that are non-zero, two sets of data must follow. The first set contains the input zones for which local cutoff groups are to be specified. The second set

specifies the local cutoff energy for the corresponding input zone. The more stringentofthe
local and global cutoffwillbe usedby the code. InMITSthe indicesrefertothelocalgroup
numbersastheyweregeneratedbyCEPXSbefore they werereversed(for adjoint calculations)inthe Monte Carlo.In forwardmode, electrons which slow down below the lowest energy in the cutoffgroup areno longer transported and their energy and charge are locally deposited. Photons which downscatter below the cutoff group or which are absorbed (only in the default cutoff group) have their energy locally deposited and transport terminated. In adjoint mode, particles which speed up beyond the highest energy in the cutoffgroup will no longer be transported.

WARNING: The CUTOFFS keywordshould be used with caution in adjoint mode.

10. **DEPOSITION-UNITS** *(ACCEP*T *Forwar*d *Only*)

    Syntax: DEPOSITION-UNITS [keyword] [keyword] [parameter(1)] Example: DEPOSITION-UNITS MASS SCALE 620.5 'MeV-cm2/g-ph' 'el-cm2/g-ph' Default: Dose in units of MeV per source particle and charge deposition in units of electrons per source particle.

    This keywordmodifies the default unitsof charge and energy deposition outputs for the ACCEPT codes. Three secondary keywords may be used with DEPOSITION-UNITS: **MASS**, **VOLUME**, and**SCALE**. MASS and VOLUME are mutually exclusive. If the MASS keyword is used, deposition values for each input zone will be divided by the mass of material in the zone in units of grams. If the VOLUME keywordis used, deposition values for each input zone will be divided by the volume of the zone in units of cm^3. If the SCALE keywordis used (either separately or in addition to the MASS or VOLUME keyword), deposition values foreachinputzonewillbe multipliedbythescalingfactor [parameter(1)].Thetwo character strings on the line following the keyword are the new units for energy and charge deposition that will be used only for labels in the output file. The character strings can be up to 15 characters long and should be enclosed in single quotation marks.

    WARNING: Accurate MASS or VOLUME scaling depend on the accuracy of the volume data used. Internal calculation of zone volumes is not available for all subzoned bodies or body combinations. Refer to the GEOMETRYkeywordfor further information.

11. **DETECTOR-RESPONSE** *(Adjoin*t *Only*)

Syntax: DETECTOR-RESPONSE

Default: No default. User must specify a detector response.

This keyword is the means by which the user specifies what single quantity of interest (known as the detector response in the forwardmode) is desired for the adjoint calculation.

(a) **CHARGE** Syntax: CHARGE Default: There is no default quantity of interest in the adjoint calculation.

This keywordspecifies charge deposition for the quantity of interest to be determined in an adjoint calculation. The following MATERIAL sub-keywordmust be present.

8. Keywords for ITS

i. **MATERIA**L

Syntax: MATERIAL [parameter(1)]

Example: MATERIAL5

Default: No default, this sub-keywordmust be present.

This specifies the material in which charge deposition is calculated.

ii. **LOCATIO**N

Syntax: LOCATION

Default: Point charge deposition calculated at the origin.

Seethe POSITION keywordfor secondary keywords. Normalization logic is only included for POINT and VOLUME distributions.

(b) **DOSE** Syntax: DOSE Default: There is no default quantity of interest in the adjoint calculation.

This keywordspecifies energy deposition for the quantity of interest to be determined in an adjoint calculation. The following MATERIAL sub-keywordmust be present.

i. **MATERIA**L

Syntax: MATERIAL [parameter(1)]

Example: MATERIAL5

Default: No default, this sub-keywordmust be present.

This specifies the material in which energy deposition is calculated.

ii. **LOCATIO**N

Syntax: LOCATION

Default: Point dose deposition calculated at the origin.

Seethe POSITION keywordfor secondary keywords. Normalization logic is only included for POINT and VOLUME distributions.

(c) **ESCAPE** Syntax: ESCAPE [keyword] Example: ESCAPE PHOTONS Default: Thereis no default quantity of interest in the adjoint calculation. There is no default source particle type. This keyword specifies particle escape (or leakage) for the quantity of interest to be determined in an adjoint calculation. The tertiary keyword must be present and be either **ELECTRONS** or **PHOTONS** for electron-escape or photon-escape,respectively.

The user can further specify the type of escaping quantity through the following sub-keywords:

i. **GROUP** Syntax: GROUP [parameter(1)] Example: GROUP8 Default: The particle escape is integrated over all energy groups.

This sub-keywordspecifies the group index of the quantity of interest. The order of the group structure is that produced by CEPXS before the inversion which occurs in the Monte Carlo in adjoint mode.

8. Keywords for ITS

ii. **LOCATIO**N
Syntax: LOCATION
Default: No default surface for particle escape.

See the POSITION keyword for secondary keywords. Only the SURFACE keywords are functional. The sub-keyword SURFACE specifies the surface through which forwardescape detector-response will be calculated.

iii. **BINT** Syntax: BINT [parameter(1)] [parameter(2)]
Example: BINT 30.0 45.0 Default: The particle escape is integrated over lab angles from 0-90 degrees.

The reference direction is LOCAL-NORMAL to the specified escape surface. The escape distribution is between angles given by [parameter(1)] and [parameter(2)] with defaultsof0and90degrees,respectively. The escape direction distribution bin is based on a cosine-law to yield particle current.

(d) **KERMA** Syntax: KERMA
Default: There is no default quantity of interest in the adjoint calculation. This keywordspecifies KERMA (Kinetic Energy Released in MAterial) for the quantity of interest to be determined in an

adjoint calculation. Operationally, the kerma dose is calculated from the photon flux. Photon-generated electrons are assumed locally deposited with a small correction for escaping bremsstrahlung. The following MATERIAL sub-keywordmust be present.

i. **MATERIAL** Syntax: MATERIAL [parameter(1)] Example: MATERIAL5 Default: No default, this sub-keywordmust be present.

This specifies the material in which KERMA is calculated.

ii. **LOCATIO**N
Syntax: LOCATION
Default: Point KERMA calculated at the origin.

Seethe POSITION keywordfor secondary keywords. Normalization logic is only included for POINT and VOLUME distributions.

12. **DIRECTION** *(Forwar*d *Only*)

Syntax: DIRECTION [parameter(1)] [parameter(2)]
Example: DIRECTION 90.0 90.0
Default: Reference direction is positive-z direction.

Thiskeywordisusedtodefinethesourcereferencedirectionandthe distributionofparticles

inanglerelativetothereferencedirection (either isotropicor cosine-law). [parameter(1)] is the spherical polar angle θ, in degrees, and [parameter(2)] *(CYLTRA*N *o*r *ACCEP*T *Only*) is the azimuthal angle φ that define the reference direction.

Themeaningofsource parametersspecifiedwiththe DIRECTIONand POSITIONkeywords is illustrated in Figure 1.

e
y
w
o
r
d
s

f
o
r

I
T
S

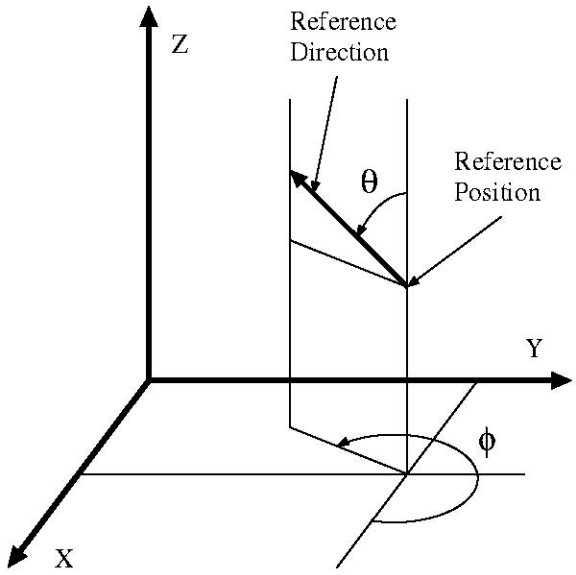


Figure 1. Source position and reference direction

(a) **ISOTROPIC** Syntax: ISOTROPIC [parameter(1)] [parameter(2)] Example: ISOTROPIC 10.0 45.0 Default: Monodirectionalin thereference direction. Defines the distribution of source particles as isotropic between angles given by [pa
rameter(1)] and [parameter(2)], withrespect to thereference direction. The default valuesfor [parameter(1)]and [parameter(2)]are0and90degrees,respectively.

(b) **COSINE-LAW** Syntax: COSINE-LAW[parameter(1)] [parameter(2)] Example: COSINE-LAW Default: Monodirectionalin thereference direction. Defines the distribution of source particles as proportional to the cosine of the angle

with respect to the reference direction. The source is distributed between angles given by [parameter(1)]and [parameter(2)]with defaultsof0and90degrees,respectively.

13. **DOPPLER** *(ITS Only*)

Syntax: DOPPLER

Default: Incoherent photon scattering will exclude Doppler broadening.

This keyword causes incoherent photon scattering to be simulated with both binding effects and Doppler broadening. The use of this keywordis restricted to problems containing only single element materials. The effects of Doppler broadening are most profound at low energies; at photon energies above 2 MeV the keyword shows no significant effect when compared to the default incoherent photon scattering settings.

14. **DUMP**

Syntax: DUMP

Default: no dump

If the DUMP keywordis present, a dump file will be written after each batch to “fort.10”. When a dump file is written, any existing file with the specified name will be overwritten. If the dump file is to be used for a subsequent restart (see primary keywordRESTART), it must be saved. See the FILE-NAMES keywordfor specifying alternative names.

In serial,adumpfilewillbe writtenaftereach batch.In parallelwith static load-balancing,a dump file will be written after each cycle. In parallel with dynamic load-balancing enabled (see primary keyword DYNAMIC-MPI), a dump file will be written after each batch has been accumulated, but the frequency is controlled with the intermediate output setting on theTASKSkeyword.Adumpfileisalways writtenattheendofa calculation.

The code will always attempt to write a dump file before aborting a calculation, regardless of whethera dump file has beenrequested.

15. **DUMP-CONNECTIVITY**

Syntax: DUMP-CONNECTIVITY

Default: no connectivity dump

If the DUMP-CONNECTIVITY keywordispresent,a connectivity dump file willbe written after each batch to “fort.9”. When a dump file is written, any existing file with the specified namewillbe overwritten.Ifthedumpfileistobeusedfora subsequentrestart(see primary keyword READ-CONNECTIVITY), it must be saved. See the FILE-NAMES keyword for specifying alternative names.

In serial, a connectivity dump file will be written after each batch. In parallel with static load-balancing, a connectivity dump file will be written after each cycle, if the master process is performing batches of Monte Carlo work. In parallel with dynamic load-balancing, a connectivity dump file cannot be written. Since each process accumulates connectivity information in isolation and that information is not gathered to the master process, only the connectivity data learned by the master process can be written to the dump file.

16. **DYNAMIC-MPI** *(MP*I *Only*)

Syntax: DYNAMIC-MPI

Example: DYNAMIC-MPI

Default: Static load balancing is used.

Thiskeywordappliesonlyforparallelprocessing.Withdynamicload balancing,the master process does not perform Monte Carlo calculations; instead, it reads input, coordinates the Monte Carlo work performed on other processes, and writes output. Some capabilities, such as RAYTRACE, DUMP-CONNECTIVITY, and particle trajectory PLOTS, are not allowed with dynamic load balancing.

17. **EBFIELDS** (*ITS*, *CYLTRA*N *o*r *ACCEPT*, *an*d *non-PCODE*S *Only*)

K
e
y
w
o
r
d
s

f
o
r

I
T
S

Syntax: EBFIELDS [keyword(1)] [parameter(1)] [INPUT-MODEL]
[parameter(2)] [parameter(3)] [keyword(2)] [parameter(4)]

Example1: EBFIELDS 1-TESLA * constant magnetic fieldof1Tesla alongz axis
MAGNETIC * activate magnetic field 1-7 *in input
zones1through7

Example 2: EBFIELDS CURRENT-EB * uniform axial electric field BOTH *
activate magnetic and electric fields 2,5 *in input zones2and5

Example3: EBFIELDS CUSTOM-B * custom magnetic field only,3(Z,B) pairs
INPUT-MODEL

```
0.0000    .5    * at 0. cm, field strength = 0.5 Tesla
20.000    .6    * at 20 cm, field strength = 0.6 Tesla
40.000    .7    * at 40 cm, field strength = 0.7 Tesla
  MAGNETIC                  * activate magnetic field
    1-7         * in input zones 1through 7
    OFF         * turn offfield
      6         * in zone 6
```

Example 4: EBFIELDS B-3D-MESH * magnetic field on uniform mesh

30 10 100 *30 binsinx,10iny, 100inz

-1.0,-1.0,-1.0 1.0,1.0,1.0 * min and max x,y,z values

1.24E-01,-1.24E-01,5.20E+00 * values of the magnetic field vectors

... *(3 values on eachof31x11x 101 lines) MAGNETIC * activate

magnetic field only 1-* in all input zones
Example5: EBFIELDS USER6 * user defined/coded field description Default: Fieldsoffinall zones; constant magnetic fieldof1Teslaalongthe z-axis;no electric field.

This keywordoffers the user choices for specifying electric and magnetic fields effects and

allows the user to set which input zones shall have these effects computed.
Fields are no longer activated in each zone through the GEOMETRY input on the zone-
specification or material-specification, for CYLTRAN and ACCEPT,respectively.

The value of [keyword(1)] determines the field model for the electric/magnetic fields. It can
be one of **1-TESLA**, **CURRENT-EB**, **CUSTOM-B**, **B-3D-MESH**, or **USER**. **1-TESLA** is the
default.

8
.

K
e
y
w
o
r
d
s

f
o
r

I
T
S

The value of [keyword(2)] can be one of **MAGNETIC**, **BOTH**, or **OFF**. The effect of these is to activate the magnetic fields, activate

both electric and magnetic fields, or deactivate fields, respectively,inthe zones specifiedbythelist [parameter(4)] beginningonthe following line.

For **1-TESLA**, there is no electric field and a constant magnetic field of 1 Tesla along the z-axis. Using 'BOTH' for input zones still only enablesa magnetic field.

For **CURRENT-EB**,thereisa uniform axial electric fieldof1MV/cmandan externalmagnetic field as if it had been generated by an infinitely long, uniform-current-density beam with radius of 0.15 cm and total current (which would be along the z-axis) of 150 kAmps.

For **CUSTOM-B**, there is no electric field and a magnetic field specified in the following manner. **INPUT-MODEL** must appear on a subsequent line, followed by Z,B pairs. On each line [parameter(2)] and [parameter(3)] contain distance along the z-axis and the total magnetic field(inTesla)at that z-value,respectively. Thezvalues mustbe strictly increasing. For particles outside the range of z values, the magnetic field closest to that position will be used. For particles within the z-grid, an interpolation based on linear field lines is used (i.e., values of radii such that the product of the enclosed area times the total magnetic field is constant). The azimuthal magnetic field is zero, and the radial and axial components are determined such that the mathematical divergence of the magnetic field is zero. The user should NOT specifya valueof zero for the total magnetic field while using CUSTOM-B.

For **B-3D-MESH**,the valuesofthe magnetic fieldarespecifiedonaregular, uniform,rectangular mesh.With this option, the next line contains3integers, whichrepresent the number of uniform bins in the x-, y-andz-directions, respectively (to be referred to as maxI, maxJ andmaxK,respectively).Thenextline contains6valuescorrespondingtotheminimumand maximum extents of the mesh in the Cartesian coordinate axes (i.e., minimum-x, minimum-y, minimum-z, maximum-x, maximum-y, and maximum-z). This should encompass all space in which the field may be turned on. Following this are (maxI+1)*(maxJ+1)*(maxK+1) lines. On each line, the first three values are the x-, y-and z-components of the magnetic field (inTesla). Any other values on the line are ignored. The order of the values should advance from minimum to maximum Cartesian coodinates, advancing first in x, then iny, and then in z. This is illustrated by the following do-loop structure, which might produce therequired inputby evaluatinga magnetic field asa functionB:

```
do K=0,maxK
    do J=0,maxJ
        do I=0,maxI
```

```
                x = xmin+I*(xmax-xmin)/maxI
                y= ymin+J*(xmax -xmin)/maxJ
                z = zmin+K*(zmax -zmin)/maxK
                write(...) Bx(x,y,z),By(x,y,z),Bz(x,y,z)
            end do
        end do end do
```

Thus, the first maxI+1 values are along the minimum-y and minimum-z edge of the rectangular mesh. The first (maxI+1)*(maxJ+1) values are on the minimum-z face of the mesh.

8. Keywords for ITS

For **USER**, the value of [parameter(1)] must correspond to the user modifying subroutine BFLD to specify their own field description. The value for [parameter(1)] must be larger than5, as values1 -5arereserved. **INPUT-MODEL** and the associated real numbers, and also OFF, MAGNETIC, and BOTH are available for user-defined application.

18. **ECHO**

Syntax: ECHO [parameter(1)]
Example: ECHO0
Default: echoing is active

If “ECHO 0” is inserted in the input stream, subsequent input will not be echoed to the output.If“ECHO1”is inserted(onalinebyitself)intheinputstream,all subsequentinput will be echoed.

19. **ELECTRONS** *(Forwar*d *Only*)

Syntax: ELECTRONS
Default: electron source if no PHOTONS primary keywordis used.

This keyworddefines the source particles to be electrons rather than photons.

20. **ELECTRON-EMISSION** *(ITS ACCEP*T *Forwar*d *Only*)

Syntax:

E
L
E
C
T
R
O
N
-
E
M

ISSION

[parameter(1)]SURFACES

[parame

t
e
r
(
2
)
]
S
U
R
F
A
C
E

[
p
a
r
a
m
e
t
e
r
(
3
)
]
B
O
D
Y

[
p
a

r
a
m
e
t
e
r
(
4
)
]
[
p
a
r
a
m
e
t
e
r
(
5
)
]
[
p
a
r
a
m
e
t
e
r
(
6

)
]
[
k
e
y
w
o
r
d
]
[
p
a
r
a
m
e
t
e
r
(
7
)
]
E
x
a
m
p
l
e
:
E
L
E
C

TRON-EMISSION3SURFACES2SURFACE1BODY43

3
S
U
R
F
A
C
E
2
B
O
D
Y
5
2
1
A
Z
8
D
e
f
a
u
lt
:
E
l
e
c
t
r
o
n
e
m
i

s
s
i
o
n
i
s
n
o
t
t
a
ll
i
e
d
.

This keywordsignals that electron emission is to be tallied. If [parameter(1)] is positive, all electrons entering the zone number specified will be terminated. If [parameter(1)] is negative, all electrons entering the zone number specified (by the absolute value of the parameter) will continue to be transported in the zone.

WARNING: InCG, electron emission tallieswillbemadeiftheypassedthrougha specified surface to exit the zone they were in prior to entering the electron-emission zone. In many cases the two adjacent zones will share a surface, so that the surface that is passed through to exit the prior zone will be the same as the surface passed through to enter the electron-emission zone, but this is not always the case so the user is encouraged to give this point careful consideration. In CAD, tallies are made on the surfaces of the electron emission zone as particles enter the zone.

8
.

K
e
y
w
o

r
d
s

f
o
r

I
T
S

The number of surfaces on which tallies are desired is specified with [parameter(2)]. This must be followed immediately with2 [parameter(2)] lines. The user must specify the surface indices of the body numbers for which tallies are desired using [parameter(3)] and [parameter(4)],respectively.Figure2onpage77 illustrateshowmanyofthebodytypesusedin ACCEPT have their surfaces numbered. The rectangular parallelepiped (RPP) is numbered the same as the BOX where **A**1 points in the positive x-direction, **A**2 points in the positive y-direction, and **A**3 points in the positive z-direction. The right circular cylinder (RCC) is numbered the same as the TRC.

The subsurfacing (equidistant subdivision) of each surface must be specified using [parameter(5)] and [parameter(6)] to request the number of divisions in the two dimensions. The meaning of these "U" and "V" parameters depend upon the surface being subsurfaced and are discussed below. The output will refer only to a single subsurface number, correspondingtoincrementingthe [parameter(6)] value first.For example,the6th subsurfaceina"33" subsurfacing correspondstothe2nd U-divisionandthe3rdV-division. TheUandVcoordinates are always incremented by increasing in their assigned coordinate axes. For example, on an RPPU andV always increase from the minimum Cartesian dimension to the maximum Cartesian dimension regardlessof the surface. Ona BOXU andV always increase with respect to moving along the **A**1, **A**2, and **A**3 vectors away from the **V**position.

With a CAD geometry emission zone, the**BODY** keywordand the subsurfacing divisions are still required but they will not used. With a CAD geometry part, there is no subsurfacing capability. Each faceisa single surface tally. If spatialresolutionis desired, the CAD geometry should be modified to cut the faces into smaller patches. For ACIS geometry, surface numbers are assigned in the same order as when they are viewed in Cubit. For Cholla geometry,eachfileis consideredtocontainasingle surface.Ineithercase, surfacesarenumbered based on the order they are encountered in the prmfile.

Currently, combining ACIS and Cholla formats into a single emission zone is not allowed.

For binning in energy and angle, the same secondary keywords apply as for ELECTRONESCAPE (NBINE, NBINT, NBINP). Directions for electron emission tallies arerelative to the local surface normal. That is, zero degrees theta is the normal into the emission zone. Theta is always defined as into the emission zone (no matter how bodies may be used to construct the zone or how bodies may border the zone), so there should never be a theta greater than 90 degrees. Zero degrees phi is defined differently depending upon the surface. The 90 degree phi coordinate is determined based on a right-hand rule of zero theta and zero phi.

In some caseszerodegreesphimaybea user-definedreferencedirection.Tospecifysucha reference direction, the input line with the two subsurfacing integers may contain the keyword **AZ**, followed by any body number except an RPP. The zero reference vector is the component of theVvector of the **AZ** body that is perpendicular to theHvector of the sub-surfaced body. If that definition of the reference vector is null, the code attempts to use the +i, +j, and +k vector for the zero azimuth vector, in that order. If no **AZ** body is specified, the+i vectorwillbeusedasthebody vector.Thiskeywordmaybeappliedtoall surfacesof aTRC,RCC,andTOR.Fortheplanar surfacesofaTRCorRCC,thereferencedirectionalso defines the zero direction for any angular division specified with the NBINP sub-keyword.

With a CAD geometry emission zone, the zero azimuthal reference direction is the positive x-coordinate axis. If the surface normal coincides with the x-coordinate axis, then the zero azimuth is the positive y-coordinate axis. If present, the **AZ** keywordwill have no effect on the tallies.

8.

Keywords for ITS

This feature is currently available for the surfaces of the following bodies:

(a) RPP – The U-V surfaces are determined as follows: for constant-x planes, they are the yand z divisions; for constant-y planes, they are the z and x divisions; for constant-z planes,theyarethexandydivisions,respectively. Zerodegreesphiis definedbythe Ucoordinate axis.

(b) BOX – The U-V surfaces are determined as follows: for constant-**A**1 planes, they are the **A**2 and **A**3 divisions; for constant-**A**2 divisions, they are the **A**3 and **A**1 divisions; for constant-**A**3, they are the **A**1 and **A**2 divisions,respectively.Zerodegreesphiis defined by theU coordinate axis.

(c) WED – The U-V surfaces are determined in the same manner as for the BOX, except there is no surface4and surface2is uses the vector **A**1 - **A**2 forUdivisions. Therectangular divisionsof surfaces5and6 are not modified for the triangular surfacesof the wedge, so only half of the divisions may be tallied. Zero degrees phi is defined by the Ucoordinate axis.

(d) RCC– Ontheplanar surfaces1and2,theU-V divisionsareinangle abouttheaxisof the cylinder and in radius, respectively. On the cylindrical surface, the U-V divisions are in angle about the axis of the cylinder and in the axial coordinate (or height) of the cylinder, respectively. The zero phi direction on the planar surfaces is determined by the zero angle about the axis, which is determined by the **AZ** logic. The zero phi direction on the cylindrical surface is in the direction of the **H**vector of the body.

(e) TRC – The U-V divisions are determined in the same manner as for the RCC, except that the zero phi direction on the conical surface is in the direction of the point of the cone.Notethatthedirectiontothepointofthe conedependsupontherespectiveradii of the bases of the TRC, rather than on the **H**vector of the body.

(f) SPH – The U-V divisions are azimuthal about the laboratory z axis with respective to the postive-x axis and in polar angle with respect to the positive-z axis. The zero phi direction is tangential to the surface of the sphere in the direction of the positive-z pole of the body.

(g) TOR – The U-V divisions are the poloidal-phi and toroidal-phi directions. This is analogoustothe divisionofthe cylindrical surfaceoftheRCC,wherethetorusisacylinder with its axis wrapped in a circle. Thus, the RCC azimuthal angle about the axis is the TOR poloidal angle, and the RCC axial coordinate is theTOR toroidal angle. TheU divisions begin at the outer-most radius of the torus and are incremented such that the 90 degree poloidal angle is the **H**vector. TheVdivisions beginatalocation specifiedby the **AZ** logic and are incremented such that the 90 degree toroidal angle is determined by the **H**vector crossed with the **AZ** vector. The zero phi direction is tangential to the surface of the torus, such that it is in the same direction as the **H**vector of the torus on the outside and in the opposite direction of the **H**vector of the torus on the inside.

21. **ELECTRON-ESCAPE** *(Forwar*d *Only*)

Syntax: ELECTRON-ESCAPE

Default: Differential electron escape is not tallied.

This keyword signals that differential electron escape is to be tallied. The following are secondary keywords associated with this primary keywordthat describe the bin structure used in tallying electron escape.

Keywords for ITS

(a) **NBINE**

Syntax: NBINE [parameter(1)] [keyword]
ITS Example: NBINE5USER

0
.8 0.6
0.4 0.2
0.0
MITS
Example
:
NBINE5
USER
10 20 30
40 50
Default:
In ITS,
10 bins
of equal
width
are
used. In
MITS,
the bin
structure

corresponds to the electron group structure.

[parameter(1)] is the number of energy bins.

In ITS, choices for [keyword] are:

i. **LOG** – Logarithmic grid spacing, with some parameter x such that $E_{cutoff} = x^{[parameter(1)]} E_0$, where E_{cutoff} is the cutoff energy and $E0$ is the source energy, and the grid values are defined as $E_{i+1} = xE_i$.

ii. **USER** – User defined energy grid. The code will then read lower bound energies (MeV) for the number of energy bins specified by [parameter(1)] in descending order. The maximum lower bound must be less than the maximum source energy.

For either of these tertiary keywords, the user must ensure that the energy is less than
or equal to the global electron cutoff(if less than, the grid will be truncated).
In MITS, this keywordallows the user to “collapse” the default bin structure (which is
one bin for every group in the electron group structure) into fewer bins.

i. **USER** – The USER secondary keyword **must** be present, and this line must be followed by [parameter(1)] numbers (integers) which specify group indices. The lower-bound energies of the specified groups form the lower-bound energies of the escape bins. The indices should appear in strictly increasing order.

(b) **NBINT**

Syntax: NBINT [parameter(1)] [keyword]
Example: NBINT5USER

10.0 30.0 90.0 135.0 180.0
Default: 18 bins of 10 degrees each up to 180 degrees.

This keywordallows the user to define a polar bin structure for recording electron es

cape. [parameter(1)] specifies the number of polar bins.
If the **USER** secondary keywordis present, then this line must be followed by [param
eter(1)] numbers which specify angle bins in ascending order up to 180 degrees.

If the **DIRECTION-SPACE** *(CYLTRA*N *or ACCEP*T *Only*) keyword is used, [parame
ter(1)] polar bins are internally generated and azimuthal bins are generated such that

angular bins are approximately equal in size. See the warning under sub-keyword
NBINP.

If no secondary keyword is present, then [parameter(1)] equal width bins will be defined.

(c) **NBINP** *(CYLTRA*N *or ACCEP*T *Only*)

Syntax: NBINP [parameter(1)] [USER]
Example: NBINP5USER

8
.

K
e
y
w
o
r
d
s

f
o
r

I
T
S

10.0 30.0 140.0 235.0 360.0
Default:1binof 360 degrees.

Thiskeywordallowsthe userto definean azimuthalbinstructureforrecording electron escape. [parameter(1)] specifies the number of azimuthal bins. If the **USER** secondary keywordis present, then this line must be followed by [parameter(1)] numbers which specify angle bins in ascending order up to 360 degrees. If the USER keywordis not present, then [parameter(1)] equal width bins will be defined. **Warning:**NBINP cannot be used with the NBINT/DIRECTION-SPACE option.

22. **ELECTRON-FLUX** *(Forwar*d *Only*)

Syntax: ELECTRON-FLUX [parameter(1)] [parameter(2)]
Example: ELECTRON-FLUX35

Default: No electron flux tallied.

This keywordsignals that electron flux is to be tallied in all subzones for input zones [parameter(1)] through [parameter(2)]. The automatic subzoning features of the ITS codes are discussed in more detail in the Subzoning section. If either parameter is omitted or 0, flux will be calculated in all zones. The same secondary keywords apply as for ELECTRONESCAPE (NBINE, NBINT, NBINP).

Calculation of electron flux in zones where macroscopic fields have been specified is not allowed. The user must ensure that the zone dependent electron cutoffenergies (see keyword CUTOFFS) for zones [parameter(1)] through [parameter(2)] are all equal.

23. **ELECTRON-SURFACE-SOURCE** *(Adjoint Only*)

Syntax: ELECTRON-SURFACE-SOURCE

Default: No forwardelectron surface sources in adjoint mode.

This keyword signals that electron escape is to be tallied. The following keywords are secondary keywords associated with this primary keywordthat describe the bin structure used in tallying electron escape. Directions are in the LOCAL-NORMAL frame unless the DELTA0-AVE sub-keywordis used.

(a) **NBINE**

Syntax:
NBINE
[parameter(1)]
USER
Example:
NBINE5
USER
111

21
31
41
Default: The bin structure corresponds to the electron group structure.

This keywordallows the user to “collapse” the default bin structure (which is one bin for every group in the electron group structure) into fewer bins. [parameter(1)] specifies the number of energy bins. The **USER** secondary keywordmust be present, and this line must be followed by [parameter(1)] numbers (integers) which specify group indices(beforeinversion, for adjoint). In adjoint mode, the upper-bound energies of the specified groups form the upper-bound energies of the escape bins. The indices should appear in strictly increasing order.

8. Keywords for ITS

(b) **NBINT**

Syntax: NBINT [parameter(1)] [keyword] [keyword] Example: NBINT6USER COSINE-LAW

5. 10.5 22.25 45 70 90

Default: Nine cosine-law sources within one of nine equal polar-angle bins with azimuthal symmetry about the z-axis.

This keyword allows the user to specify the number, angular-extent, and type of angular distribution of forward sources for an adjoint calculation with electron surface sources. [parameter(1)] specifies the number of source polar-angle bins. Without either the USER or DIRECTION-SPACE keyword, this will generate [parameter(1)] equal solid-angle polar-angle bins. When the **USER** keyword is used, the following line must contain [parameter(1)] values which arethe upper bounds of the polar-angle bins. When the **DIRECTION-SPACE** *(CYLTRA*N *an*d *ACCEP*T *Only*) keywordis used , [parameter(1)] polar bins are internally generated and azimuthal bins are generated such that angular bins are approximately equal in size. DIRECTION-SPACE can only be used with the DELTA0-AVE setting.

The final keyworddescribes the source angular distribution within these bins. It should be one of the following (COSINE-LAWis the default):

i. **ISOTROPIC** The forward-source is uniform in angle within the specified angular bin, e.g. a thin surface of a radioactive material.

ii. **COSINE-LAW** The forward source has a cosine-law distribution with respect to the surface normal, e.g., this corresponds to the “isotropic-flux” sources of cosmic particles.

iii. **DELTA0-AVE** The forward source is the average, within the specified angular bin, of all plane-wave sources with normals within the specified angular bin. For each plane wave, the source is a delta function normal to the plane, $\delta(\Omega\text{-}1)$.

(c) **NBINP** *(CYLTRA*N *or ACCEP*T *Only*)

Syntax: NBINP [parameter(1)] [USER]

Example: NBINP5USER

10.0 30.0 140.0 235.0 360.0

Default:1binof 360 degrees.

Thiskeywordallowstheusertodefinean azimuthalbinstructureforrecordingangular distribution of

forwardsources for an adjoint calculation with electron surface sources. [parameter(1)] specifies the number of azimuthal bins. If the **USER** secondary keyword is present, then this line must be followed by [parameter(1)] numbers which specify angle bins in ascending order up to 360 degrees. If the USER keywordis not present, then [parameter(1)] equal width bins will be defined. NBINP cannot be used with the NBINT/DIRECTION-SPACE option.

24. **ELECTRON-VOLUME-SOURCE** *(Adjoint Only*)

Syntax: ELECTRON-VOLUME-SOURCE [parameter(1)] [parameter(2)]

Example: ELECTRON-VOLUME-SOURCE57

Default: No volume source.

8. Keywords for ITS

This keywordsignals that a forward source (i.e., the flux of adjunctons) is to be tallied in all subzones for input zones [parameter(1)] through [parameter(2)]. If either parameter is omitted or 0, the source will be calculated in all zones. The same secondary keywords apply as for ELECTRON-SURFACE-SOURCE (NBINE, NBINT, NBINP), but the angular distribution must always be ISOTROPIC.

25. **ENERGY** *(Forward Only*)

Syntax: ENERGY [parameter(1)] [keyword] [parameter(2)]

Example: ENERGY 4.3

Example: ENERGY GROUP1 (MITS Only)

Default: For ITS, 1.0 MeV monoenergetic source. For MITS,a mono-group sourcein

the highest energy group.

InITS,the sourceis specifiedasamono-energetic sourcewith energy [parameter(1)]inMeV. InMITS,the sourcemaybe specifiedasa mono-energetic sourceorasa mono-group source. For an electron source, the user may specify a monoenergetic source with the energy specified by [parameter(1)] in MeV. For any particle species, if the keywordGROUP appears on the line, then the associated [parameter(2)] specifies the single group index which will be used for all the source particles. The specific energy of the individual histories will be sampled uniformly over the width of the group. If the user desires a mono-energetic photon source, the cross sections generated by CEPXS must include the appropriate single-energy (or “source-line”) group.

26. **ESCAPE-SURFACES** *(Forward Only*)

Syntax: ESCAPE-SURFACES [parameter(1)]

Example: ESCAPE-SURFACES3

Default: All surfaces of the escape zone (tallied as a single surface) for ACCEPT.

Both surfaces (ZMIN and ZMAX) for TIGER. All three surfaces (ZMAX, ZMIN, and RMAX) for CYLTRAN.

This specifies the number of escape surfaces for which the integral particle escape or the requested particle escape willbe displayed. This keywordmustbe followedby[parameter(1)] separate lines of the secondary keywordSURFACE.

Any specification for CAD will be ignored. All escaping particles are scored in a single body/surface tally.

(a) **SURFACE** Syntax: SURFACE [keyword] (non-ACCEPT codes) or SURFACE [parameter(1)] BODY [parameter(2)] (ACCEPT codes) Example: SURFACE ZMAX (non-ACCEPT codes) or SURFACE3BODY2

(ACCEPT codes)

This sub-keywordspecifies the surface index through which the escaping particles will be calculated.

8. Keywords for ITS

For TIGER and CYLTRAN, surface **ZMIN** refers to the minimum-z surface and surface **ZMAX** is the maximum-z surface. For CYLTRAN, surface **RMAX** is the lateral escape surface at maximum radius. For ACCEPT, it is necessary to specify both the surface index and the **BODY** number. Figure2 on page77illustrates how many of the body types used in ACCEPT have their surfaces numbered. In the case of the arbitrary polyhedron (ARB), the user explicitly specifies the order in the input. The right circular cylinder is numbered the same as the truncated right-circular cone (TRC). The rectangular parallelepiped (RPP) is numbered the same as the BOX where A1 points in the positive x-direction, A2 points in the positive y-direction, and A3 points in the positive z-direction.

The user is cautioned to make sure that the desired surface has a unique description for the way the user has specified the geometry, otherwise the result may be invalid. For example,ifa zoneis definedastheunionoftwo bodiesandthosetwo bodieshave coincident surfaces, a particle may exit the zone surface through either one of the two body surfaces.

27. **FILE-NAMES**

Syntax: FILE-NAMES

Default: Default names (based on Fortran unit names) are used.

This keywordallows the user to specify names of files to be opened for

input and output. File names must be enclosed in single quotation marks.

(a) **AREAL-DENSITY-FIL**E
Syntax: AREAL-DENSITY-FILE
[keyword]
Example: AREAL-DENSITY-FILE
'areal.dat'
Default: Areal density data is written to "fort.4".

This keywordspecifies that areal density data will be written to the file named [keyword].
(b) **CONNECTIVITY-DUMP-FILE** Syntax: CONNECTIVITY-DUMP-FILE [keyword] Example: CONNECTIVITY-DUMP-FILE 'connect.dat' Default: Connectivitydata neededfora connectivityreadis writtento "fort.9".

This keyword specifies that connectivity data will be written to the file named [keyword].

(c) **CONNECTIVITY-READ-FIL**E

Syntax: CONNECTIVITY-READ-FILE

[keyword]
Example: CONNECTIVITY-READ-FILE

8
.

K
e
y
w
o
r
d
s

f
o
r

I
T
S

'connect.dat' Default:
Connectivity data is read from "fort.13".
This keyword specifies that connectivity data will be read from the file named [keyword].

(

d

)

D

U

M

P

-

F

I

L

E

S

y

n

t

a

x

:

D

U

M

P

-

F

I

L

E

[

k

e

y

w

o

r

d
]

E
x
a
m
p
l
e
:

D
U
M
P
-
F
I
L
E

,

D
u
m
p
’

Default:Dumpdata neededforarestartis writtento “fort.10”. This keywordspecifies that dump data will be written to the file named [keyword].

(e)

I
N
T
E
R
M

E

D

I

A

T

E

-

F

I

L

E

S

y

n

t

a

x

:

I

N

T

E

R

M

E

D

I

A

T

E

-

F

I

L

E

[

k

e
y
w
o
r
d
]

E
x
a
m
p
l
e
:

I
N
T
E
R
M
E
D
I
A
T
E
-
F
I
L
E

,

I
n
t
e

r
O
u
t
’

D
e
f
a
u
l
t
:

I
n
t
e
r
m
e
d
i
a
t
e

o
u
t
p
u
t

i
s

w
r

i
t
t
e
n

t
o

“
f
o
r
t
.
1
2
”
.

This keywordspecifies that intermediate output will be written to the file named [keyword].

(f) **FINITE-**

E
L
E
M
E
N
T
-
F
I
L
E

(
A
C
C

E
P
T

F
o
r
w
a
r
d

O
n
l
y
)

S
y
n
t
a
x
:

F
I
N
I
T
E
-
E
L
E
M
E
N
T

-FILE [keyword]

Example:

FINITE-ELEMENT-

FILE,torus.dat'

Default:

Finite element

d
a
t
a

i
s

w
r
i
t
t
e
n

t
o

“
f
o
r
t
.
3
”
.

This keywordspecifies that finite-element output will be written to the file named [keyword].

(g) **PFF-**

F
I
L
E

(
/

T
S

P
F
F

O
n
l
y
)

S
y
n
t
a
x
:

P
F
F
-
F
I
L
E

[
k
e
y
w
o
r
d
]

E

x
a
m
p
l
e
:

P
F
F
-
F
I
L
E

,

e
m
i
s
s
i
o
n
.
p
f
f
,

D
e
f
a
u
l
t
:

E
l
e
c
t
r
o
n

e
m
i
s
s
i
o
n
,

e
l
e
c
t
r
o
n

e
s
c
a
p
e
,

o
r

p
h

o
t
o
n

e
s
c
a
p
e

d
i
s
t
r
i
b
u
t
i
o
n
s

a
r
e

w
r
i
t
t
e
n

t
o

“
d
i
s
t
r
i
b
.
p
f
f
”
.

This keywordspecifies that electron emission distribution output will be written to the file named [keyword].

(h) **PLOT-FILE** *(CYLTRA*N *o*r *ACCEP*T *Only*)

Syntax: PLOT-FILE [keyword]

Example: PLOT-FILE

8. Keywords for ITS

’weasel.
dat’ Default: Plot
data is written to
“WEASEL.OUT”.
This
keywordspecifies
that plots output
will be written to
the file named
[keyword].

(i) **RESTART-FIL**E
Syntax: RESTART-FILE
[keyword]
Example: RESTART-FILE
’Restart’

Default: Restart data is read from “fort.14”. This keywordspecifies that restart data will be read from the file named [keyword].

(j) **XSECTION-FIL**E
Syntax: XSECTION-FILE
[keyword]
Example: XSECTION-FILE
’XSfile’

Default:Cross sectionsarereadfrom “fort.11”. This keywordspecifies that cross sections will be read from the file named [keyword].

28. **FINITE-ELEMENT-FORMAT** *(ACCEP*T *Forwar*d *Only*)

Syntax: FINITE-
ELEMENT-
FORMAT[CHARGE]
[DOSE] [NO-SIGMA]
[LIST] [LISTZONE]
Example: FINITE-
ELEMENT-
FORMATCHARGE
Default: no finite-element
file

This keyword signals the creationofaTecplot R

[32] finite-element-format file written to “fort.3”. The file contains charge deposition (secondary keywordCHARGE) and/or energy deposition (secondary keywordDOSE) for each subzone. If neither secondary keywordis used, both energy and charge deposition will be written.

The NO-SIGMA secondary keyword will suppress the statistical uncertainty information
from being printed to the finite-element file. The LIST secondary keywordwill suppress all node and element information. This option will simply list the dose and/or charge deposition data. Unlike the default behavior, the list capability will include the dose for all zones, whether they were subzoned or not. Counting each zonethatisnot subzonedasasingle subzone,theline numberinthe finite-elementfile will correspond to the subzone number for the geometry.

The LIST-ZONE secondary keywordhas the same effect as the LIST secondary keyword, but
it also includes zone and subzone numbers in the listing of deposition data. Withoutthe LIST keyword, eachITS zone thatis subzonedis writtenasa separateTecplot zone. Each subzoneofanITS zonerepresentsan elementintheTecplot zone.The energyand charge deposition data arewritten as volume-averaged values associated with each element. SinceTecplot expects nodal values, each element node has energy and/or charge deposition valuesbutonlythefirstnodeofeach elementrepresentsthe correct volume-averagedvalue for that element. This can cause results viewed in Tecplot to appear different than they should.Arobustmethodforviewingdataistousea conversionprogramcalled “its2tec”so that each node is repeated for each

element.

Anotherrobust methodforviewingdataistouseEnsight R

[33].

Ensight will display elementaldatabutwillnotreadanTecplot file.Aconversionprogram called “its2exo”ispart of the ITS distribution in theTools directory. This program will convert the data from the Tecplot format to an Exodus format.

29. **GEOMETRY** This keyword signals the beginning of the geometry information. The choice among the usages depends on which of the member codes of ITS has been selected: TIGER, CYLTRAN, ACCEPT,orCAD.Keywordusage informationandadetailed discussionofthe GEOMETRY
keywordinput and syntax requirements is contained in the section specific to the member code chosen. The ACCEPT Geometry section is also relevant to the use of CAD geometry.

30. **HISTORIES**

    Syntax: HISTORIES [parameter(1)]

    Example: HISTORIES 100000

    Default: 1000 histories

    This specifies the total number of primary particle histories to be followed. [parameter(1)] cannotbegreater than 2,147,483,647 (i.e.,2^{31}-1).To simulate more histories, the HISTORIESPER-BATCH keywordshould be used. HISTORIES and HISTORIES-PER-BATCH are mutually exclusive keywords.

31. **HISTORIES-PER-BATCH**

    Syntax: HISTORIES-PER-BATCH [parameter(1)]

    Example: HISTORIES-PER-BATCH 10000

    Default: 1000 total histories

    This specifies the number of primary particle histories to be followed per batch. [parameter(1)] cannotbegreater than 2,147,483,647 (i.e.,2^{31}-1).To simulate more histories, the number of batches should be increased. HISTORIES and HISTORIES-PER-BATCH are mutually exclusive keywords.

32. **INCLUDE-FILE**

Syntax: INCLUDE-FILE

Example: INCLUDE-FILE

GeometryInp

Default: All input is read from a single file.

Thisallowstheuserto includeinput parametersfromotherfiles.Withthiskeyword,auser can eliminate redundant data between separate input decks. For example, the geometry data can be in one file, and that file can be included in multiple input decks.

8. Keywords for ITS

33. **LINE-TALLY-WITH-CONTINUUM** *(Forward Only*)

Syntax: LINE-TALLY-WITH-CONTINUUM

Default: Line radiation is tallied separate from the continuuum for photon escape and flux.

This keyword triggers line radiation to be tallied in the corresponding energy bin of the continuum radiation for photon escape and flux. This does not affect annihilation line radiation tallies, which are treated separately with the ANNIHILATION-LINE-TALLY-WITHCONTINUUM keyword.

34. **MAGNETIC-TRAP-LIMITS** *(CYLTRAN or ACCEPT Only*)

Syntax: MAGNETIC-TRAP-LIMITS [parameter(1)] [parameter(2)]

Example: MAGNETIC-TRAP-LIMITS 10 1E+5

Default: Limits are3and 1000000,respectively.

Withthiskeyword,the usercanchangethe counter limitsfor determiningthataparticleis trappedinamagneticfieldinavoid.Twotypesoftrapsaredetected:amagneticmirrortrap, in which a particle may bounce back and forth within the trap, and a particle that is circling within the magnetic field with no velocity parallel to the magnetic field lines. Both of these conditions are detected in the code by counting occurences of incidents that may indicate the condition, with count limits set by parameter(1) and parameter(2), respectively. In both casesthereisatrade-off:alow countlimitmaybemorecomputationallyefficient(byspending less effort tracking trapped particles) but may be less accurate (by killing particles that are not trapped). The first counter measures the number oftimes that the particle changes direction with respect to the magnetic field. The second counter measures the number of timesthat successivetimesteps movethe particleinadirectionthatis strictly perpendicular to the magnetic field lines.

Sincethe transportofaparticle betweenreflectionsinamagneticmirrormaybequite expensive, a relatively low number is recommended for the first counter. Since there are typically 1000timestepsforaparticleto completeacircleinamagneticfield,arelativelyhigh number is recommended for the second counter. Neither limit may be set lower than 2.

35. **MICRO** *(MITS Forward Only*)

Syntax: MICRO Default: Energyandcharge depositionis determinedbyfoldingthefluxintotheappropriate cross section (e.g., restricted stopping power for electron energy deposition), with an additional microscopic contribution for particles that fall below cutoff.

With this keyword, energy and charge deposition areentirely microscopic. This has been observed to be more efficient for calculating charge deposition, but less efficient for calculating energy deposition.

36. **NEW-DATA-SET**

Syntax: NEW-DATA-SET

8. Key

Example: NEW-DATA-SET

Default: one run.

This keywordsignifies that the data set for a particular Monte Carlo run has been read and thatthedatasetforanewMonteCarlorun follows.ItspurposeistopermitmultipleMonte Carlo runs within a single code execution. **Its usage is an exception to the rule that the primary keywords are order independent.**

The input data must be given for each problem, and the input data sets for the different problems must be separated from one another by a line containing this keyword. Different cross section files can be used for different data sets, if they are specified using the FILENAMES/XSECTION-FILE keyword.

For CAD calculations, the prmfile and CAD geometry files will not be reread for each run. Therefore, all of the problems must use the same CAD calculation parameters and the same CAD geometry.

37. **NO-COHERENT** *(ITS Only*)

Syntax: NO-COHERENT

Default: Coherent photon scattering will be included in the calculation.

This keyword is intended for development purposes only. This keyword deactivates the simulation of coherent photon scattering.

38. **NO-DEPOSITION** *(Forward Only*)

Syntax: NO-DEPOSITION [CHARGE] [DOSE]

Example: NO-DEPOSITION CHARGE

Default: Energy and charge deposition are tallied for every subzone.

Thiskeywordsuppresses tallyingof energyandcharge deposition.Usingonlythe CHARGE secondary keyword suppresses tallying of charge deposition. Using only the DOSE secondary keyword suppresses tallying of energy deposition. Omitting both secondary keywords or using both will suppress tallying of both energy and charge deposition.

This keyword will reduce memory requirements, because memory is not allocated for the tallies that are suppressedbythis keyword. Since the deposition is not tallied, it cannot be obtained usinga coderestart.

39. **NO-DEPOSITION-OUTPUT** *(Forwar*d *Only*)

Syntax: NO-DEPOSITION-OUTPUT
Default:Energyandcharge
depositionoutputis writtento
“fort.12”andunit6output.

This keywordsuppresses energy and charge deposition output. This may prove useful when the FINITE-ELEMENT-FORMATkeywordis used with numerous subzones. The output can be obtained later by performing a code restart without this keyword.

40. **NO-DETAILED-DEPOSITION** (*Forwar*d *Only*)

8.

Keywords for ITS

Syntax: NO-DETAILED-DEPOSITION [CHARGE] [DOSE]
Example: NO-DETAILED-DEPOSITION CHARGE
Default: Detailed energy and charge deposition are tallied for every subzone.

This keywordsuppresses tallying of energy and charge deposition details. These “details” indicate whether deposition was due to primary or secondary particles for ITS, and whether the deposition was due to particles at the cutoff energy or higher for MITS. Using only the CHARGE secondary keywordsuppresses detailed tallying of charge deposition. Using only the DOSE secondary keywordsuppresses detailed tallying of energy deposition. Omitting both secondary keywords or using both will suppress detailed tallying of both energy and charge deposition.

This keyword will reduce memory requirements, because memory is not allocated for the tallies that are suppressedbythis keyword. Since the deposition is not tallied, it cannot be obtained usinga coderestart.

41. **NO-GEOMETRY-TABLE**

Syntax: NO-GEOMETRY-TABLE

Default: Geometry-dependent input settings are written to output.

This keywordsuppresses the printing of geometry-dependent input settings.

42. **NO-INCOH-BINDING** *(ITS Only*)

Syntax: NO-INCOH-BINDING

Default: Incoherent photon scattering will include binding effects.

This keywordis intended for development purposes only. This keyword causes incoherent photon scattering to be simulated in the Klein-Nishina or free-electron approximation.

43. **NO-INTERMEDIATE-OUTPUT**

Syntax: NO-INTERMEDIATE-OUTPUT

Default: Intermediate output is written to “fort.12”.

This keywordspecifies that output will be written only upon completion of the calculation.

44. **NO-KICKING** *(ITS Only*)

Syntax: NO-KICKING Defaults:
Terminal processing of electrons
and positrons includes kicking,
except when EBFIELDS is enabled.

This keyword is intended for development purposes only. The “kicking” of electrons and positronsisan approximationthat movesthe particleto accountfor transportat lower energies. This is based on the remaining practical range of the particle. Using this keywordwill cause the particle energy and charge to be locally deposited. When EBFIELDS is enabled, thereis no kicking, and using this keywordisredundant.

8. Keywords for ITS

45. **NO-KNOCKONS** *(ITS Only)*

Syntax: NO-KNOCKONS

Defaults: Secondary knock-on electrons are produced.

This keywordis intendedfor development purposes only. Production of secondary knock-on electrons is disabled, however primary electron energy loss and energy-loss straggling are not affected. See the NO-STRAGGLING keyword.

46. **NO-OVERLAP-OUTPUT** *(CAD Only)*

Syntax: NO-OVERLAP-OUTPUT

Default: Overlaps in CAD geometry are accumulated and reported.

This keyword suppresses accumulation and reporting of overlaps in CAD geometry. For large parallel calculations, this may improve the parallel efficiency when static load balancing is being used.

47. **NO-STRAGGLING** *(ITS Only*)

Syntax: NO-STRAGGLING

Defaults: Energy-loss straggling is applied to electrons.

This keywordis intended for development purposes only. Energy-loss stragglingis disabled for all electron transport.

48. **NO-SZDEPOSITION-OUTPUT** *(Forward Only*)

Syntax: NO-SZDEPOSITION-OUTPUT
Default:Energyandcharge depositionoutputis writtento “fort.12”andunit6output for all zones and subzones.

This keywordsuppresses energy and charge deposition output for all subzones. This may prove useful when the FINITE-ELEMENT-FORMATkeywordis used with numerous sub-zones. Unlike the NO-DEPOSITION-OUTPUT keyword, this keywordwill allow deposition quantities for zones to be printed out. Ifthe NO-DEPOSITION-OUTPUT keywordis used, this keywordhas no effect.

49. **OVERLAP-OUTPUT-MAX** *(CAD Only*)

Syntax: OVERLAP-OUTPUT-MAX 20
Default: A maximum of 10 overlaps in CAD geometry are accumulated and reported.

Thiskeywordsetsthelimitonthe accumulationandreportingofoverlapsinCADgeometry. For large parallel calculations, large values will decrease the parallel efficiency when static load balancing is being used.

Setting this to zero disables the accumulation and reporting of overlaps in CAD geometry (see NO-OVERLAP-OUTPUT).

o
r

I
T
S

50. **PFF-FORMAT** *(ITS PFF Only*)

Syntax: PFF-FORMAT[SIGMA]

Example: PFF-FORMATSIGMA

Default: no pfffile

This keyword signals the creation of a pff-formatted file written to "distrib.pff". The file contains one of: (1) electron emission distributions for each subsurface, (2) electron escape distributions for each surface, or (3) photon escape distributions for each surface. Only one ofthesethreewillbegiveninthepfffile,withprioritygivenintheorder listedaboveamong those distributions requested in the input deck. The optional secondary keyword **SIGMA** will include statistical uncertainty informationin thepfffile.

51. **PHOTONS** *(Forward Only*)

Syntax: PHOTONS [parameter(1)]

Example: PHOTONS1

Default: electron source

This keyworddefines the source particles to be photons rather than electrons.
For ITS only, if [parameter(1)] is0 or omitted, then unscattered photons will be excluded
from photon flux and escape scores. Otherwise, unscattered photons will be included in
photon flux and escape scores.

1 **PHOTON-ESCAPE** *(Forward Only*) See ELECTRON-ESCAPE
2 **PHOTON-FLUX** *(Forward Only*) See ELECTRON-FLUX
3 **PHOTON-SURFACE-SOURCE** *(Adjoint Only*)

See ELECTRON-SURFACE-SOURCE, with the following additions.

For the PHOTON-SURFACE-SOURCE keyword,the DIRECTION-SPACE secondary

keywordhas the following optional secondary keywords. Syntax: . . . DIRECTION-SPACE [keyword][keyword][keyword] [parameter(1)] Example: NBINT6DIRECTION-SPACE DELTA0-AVE RAYTRACE RAYPRINT

2

5
0

9
0
.
1

7
5
.
5

2
7
0

D
e
f
a
u
l
t
:

F

u
l
l

t
r
a
n
s
p
o
r
t

r
a
t
h
e
r

t
h
a
n

a

r
a
y
-
t
r
a
c
e

calculation.

Although not strictly a secondary keywordof **DIRECTION-SPACE**, the **DELTA0-AVE** secondary keywordisrequired whenever **DIRECTION-SPACE** is present. The secondary keyword **RAYTRACE** causes an uncollided kerma calculation to be performed. This calculation performs a ray-tracing activity for every angle bin produced by the direction space procedure. The photon surface source is attenuated along the centroids of each angular bin using the total photon interaction

cross section. The resulting photon source is folded with the photon kerma cross section for the specified detector material. The kerma is presented in the usual direction space output format. The RAYTRACE keyword is not allowed with the **USER** secondary keywordof **NBINT**. No other output isproduced unless the additional secondary keyword **RAYPRINT** is used.

The **RAYPRINT** secondary keywordallows the user to print out the ray segment data for any or all of the rays generated with the RAYTRACE keyword. Ray segment data contains alltheinformation neededto performa1-D transport calculationalongthepathofaray. **RAYPRINT** with no parameters prints out ray segment data for all rays. If [parameter(1)] is present, it is the number of rays for which the user is requestingray segment data and requires [parameter(1)] pairs of angles beginning on the next line of input. The angle pairs correspondtothetheta(polarangle,0to180degrees)andphi (azimuthalangle,0to360degrees) of each angular bin for which the user wants ray segment data printed. The specified angles do not have to be exactly the angular bin centroids, ITS will find the angular bin that the given pair falls in and print the ray segment data for that centroid. The **RAYPRINT** secondarykeywordrestrictswhichraydatais writtentothearealdensityfileifanyrestrictions are specified (see the **AREAL-DENSITY-OUTPUT** keyword).

55. **PHOTON-VOLUME-SOURCE** *(Adjoin*t *Only*)

See ELECTRON-VOLUME-SOURCE

56. **PLOTS-OVAL-RESOLUTION** *(ACCEP*T *Only*)

Syntax: PLOTS-OVAL-RESOLUTION parameter(1)

Example: PLOTS-OVAL-RESOLUTION 36

Default: 360

Withthiskeyword,theusercanchangethenumberofpointsusedtorepresent curvededges of bodiesinplots. Increasingthisvaluemayalsoincreasethe maximum numberof trajectory points plotted (see the TRAJECTORY-POINTS keyword).

57. **PLOTS** *(CYLTRA*N *or ACCEP*T *Only*)

CYLT

RAN

Syntax:

PLOTS

[parameter(1)] [parameter(2)] [parameter(3)] [parameter(4)]
CYLTRAN
Example: PLOTS0305
ACCEPT
Syntax: PLOTS [parameter(1)] [parameter(2)] ... [parameter(7)]
ACCEPT
Example: PLOTS3050590180 0505180 90 050500
Default: No plots are generated.

8. K

e
y
w
o
r
d
s

f
o
r

I
T
S

This keyword causes geometry and/or particle track data to be written to a file in a form that can be plotted. The plot data can be customized with the sub-keywords. Particle track data can only be written from the master process, so trajectories will not be plotted if the DYNAMIC-MPI keywordis used.

For **CYLTRAN** this keywordis used to plot the geometry for any of the CYLTRAN codes. With fields, the electron and positron trajectories are plotted in zones where macroscopic fieldsaredefined.Useofthiskeywordwillproducea ρ z plot of that portion of the problem cylinder bounded by [parameter(1)] through [parameter(4)], which define the minimum ρ, the maximum ρ, the minimum z, and the maximum z,respectively, in cm. If the parameters are left blank, the entire problem cylinder will be plotted.

For **ACCEPT** use of this keyword will produce [parameter(1)] parallel projections of the body specification as given under the GEOMETRY keyword. [parameter(2)] through [parameter(7)] are repeated [parameter(1)] times on separate lines. [parameter(6)] and [parameter(7)] specify the spherical polar angles φ and θ, respectively, in degrees that define the direction from which the geometry is to be viewed. [parameter(2)] through [parameter(5)] specify the minimum x, the maximum x, the minimum y, and the maximum y,respectively, incmoftheplottedprojection.Ifthe PLOTS keywordis used,all parametersarerequired; there are no defaults. With fields, the electron and positron trajectories are plotted in zones where macroscopic fields are defined only on the final projection.

(a) **ORBITS** *(ITS CYLTRAN or ACCEPT Only*)

Syntax: ORBITS [parameter(1)]
Example: ORBITS 10
Default: Electron and positron trajectories associated with every 5th source par

ticle of the first batch will be plotted.

Electron and positron trajectories associated with source particles of the

first batch that are multiples of [parameter(1)] are to be plotted in those zones for which macroscopic fields have been defined.Ablank for [parameter(1)] will cause all electron and positron trajectories of the first batch to be plotted. This keyword can only be used when the EBFIELDS primary keywordis also used to activate fields.

(b)

P
A
G
E
-
H
E
A
D
E
R
(
C
Y
L
T
R
A
N

o
r

A
C
C
E
P
T

O
n
l
y

)

Syntax: PAGE-HEADER [keyword]

Example:

P
A
G
E
-
H
E
A
D
E
R

variables=”XROT”,”YROT”
Default: The page header is a blank line.

(c) **BREAK-**
I
N
D
I
C
A
T
O
R

(
C
Y
L
T
R
A
N

o
r

A

C

C

E

P

T

O

n

l

y

)

S

y

n

t

a

x

:

B

R

E

A

K

-

I

N

D

I

C

A

T

O

R

[

k

e
y
w
o
r
d
]

E
x
a
m
p
l
e
:

B
R
E
A
K
-
I
N
D
I
C
A
T
O
R

zone
Default: Thebreak indicatorisa blank line.

(d) **PLOT-3DAXIS** *(CYLTRA*N *or ACCEP*T *Only*)

.

K
e
y
w
o
r
d
s

f
o
r

I
T
S

Syntax: PLOT-3DAXIS Example: PLOT-3DAXIS Default: The coordinate axes are not indicated on the plots.

This keywordwill cause the 3D coordinate axes to be drawn on the plots.

58. **POSITION** *(Forward Only*) Syntax: POSITION

Default: The source is located at the origin.

Thiskeyworddefinesthepositionofthesource.Itmustbe

followedbyoneofthefollowing

sub-keywords to further describe its spatial distribution:

(a) **POINT**

Syntax: POINT [parameter(1)] (TIGER codes)
or POINT [parameter(1)] [parameter(2)] [parameter(3)] (non-TIGER codes)
Example: POINT 2.0 (TIGER codes)

or POINT 0.0 0.0 2.0 (non-TIGER codes) Default:The sourceisapoint sourceatthe origin.

This sub-keywordspecifies that the source is a point. For TIGER, the single location is the z coordinate of the point. For the non-TIGER codes, the three parameters specify thex,yandz coordinates,respectively.
The default directional distribution is mono-directional in the positive z-direction, but the reference direction and source distribution may

be specified by the DIRECTION keyword.

(b) **LINE** *(CYLTRA*N *or ACCEP*T *Only*)

Syntax: LINE Example: LINE
3.2 8.1 5.5
9.1 3.4 3.0 Default: There is no default LINE distribution.

This keywordspecifies the source as a line. The command line following the keyword contains the x, y, and z coordinates of one end of the line. The next command line containsthex,y,andz coordinatesofthe otherendofthe line. The sourceis sampled uniformly along the line segment.
The default directional distribution is mono-directional in the positive z-direction, but the reference direction and source distribution may be specified by the DIRECTION keyword.

(c) **DISK** *(CYLTRA*N *or ACCEP*T *Only*) Syntax: DISK [parameter(1)]
[parameter(2)] [parameter(3)]
Example: DISK 0.0 0.0 2.0
Default: There is no default DISK spatial distribution.

8
.

K
e
y
w
o
r
d
s

f
o
r

I
T
S

This keywordspecifiesthe sourceasa disk. Thethree parameters specifythex,y,and

z coordinates of the location of the center of the disk.
The default angular distribution is mono-directional in the

positive-z direction. The ori
entation is normal to the source direction (i.e., in the x-y axis). Both the source direction
and orientation are controlled by the DIRECTION keyword.

This sub-keywordshouldbe followedby the following tertiary keyword:

i. **RADIU**S

Syntax: RADIUS [parameter(1)] [keyword]
Example: RADIUS 3.5 RADIAL-BIASING
Default: Zero radius point source.

[parameter(1)] specifies the radius of the disk source. This parameter may be followed by the keyword **RADIAL-BIASING** that will cause source particles to be sampled uniformly in radius (rather than uniformly in area).

(d) **ANNULUS** *(CYLTRA*N *or ACCEP*T *Only*) Syntax: ANNULUS

[parameter(1)] [parameter(2)] [parameter(3)] [keyword][pa rameter(4)]

Example 1: ANNULUS 0.0 0.0 2.0

4.0 2.0 RADIAL-BIASING

Example2: ANNULUS 0.0 0.0 2.0 PROFILE4

0.0 0.5 0.8 1.0
2.1 2.4 2.9 3.1 Default: There is no default ANNULUS spatial distribution.

This keywordspecifies the source as an annulus. The first three parameters specify the

x,y, andz coordinatesof the locationof the centerof the annulus. The default angular distribution is mono-directional in the positive-z direction. The ori
entationis normaltothe sourcedirection (i.e.,inthex-y axis). Boththe sourcedirection
and orientation are controlled by the DIRECTION keyword.

If the optional PROFILE keywordis not present, then the following line must contain
two parameters that specify the outer and inner radii of the annulus. These parameters
may be followed by the keyword **RADIAL-BIASING** that will

cause source particles
to be sampled uniformly in radius (rather than uniformly in area).

If the **PROFILE** keyword is present, the user may specify a radial distribution to be sampled from. [parameter(4)] following the PROFILE keywordspecifies the number of radial bins. Starting on the following line, [parameter(4)] values specify the cumulative probability array. This is followed by [parameter(4)] values specifying the corresponding radial array.

(e) **RECTANGLE** *(CYLTRA*N *or ACCEP*T *Only*) Syntax:

RECTANGLE [NORMAL] [REVERSE] [A1GRID parameter(1)] [A2GRID parameter(2)]

Example: RECTANGLE

1.0 1.0 0.0
1.0 2.0 0.0
3.0 2.0 0.0

8. Keywords for ITS

Example: RECTANGLE NORMAL REVERSE A1GRID2A2GRID2
1.0 1.0 0.0
1.0 2.0 0.0
3.0 2.0 0.0
0.6 0.2 0.2 0.0 Default: There is no default

RECTANGLE spatial distribution. This keywordspecifies the source asarectangle. The three lines following the keyword must containthex,y,andzcoordinatesof vectorsV1,V2,andV3 specifying3cornersof therectangle. (V1-V2) mustbe orthogonalto (V3-V2).The defaultreferencedirectionis in the positive-Z direction. An alternative reference direction and angular distribution may be specified with the DIRECTION keyword. If the **NORMAL** sub-keyword is used, thereference direction willbe definedby (V1-V2)X(V3-V2). The NORMAL subkeywordwill override the reference direction given by the DIRECTION keyword, but any angular distribution will still be valid. If the NORMAL sub-keyword has been used, the **REVERSE** sub-keywordmay be used to reverse the reference direction to be definedby (V3-V2)X(V1-V2). An arbitraryrectangular sourcegridmaybespecifiedbyfurther specifyingthe number of grid divisions along each direction of the rectangular source and the source strength within each grid cell. Aparameter following the **A1GRID** sub-keywordspecifies the numberof divisionsin the (V1-V2) direction.Aparameter following the **A2GRID** sub-keyword specifies the number of divisions in the (V3-V2) direction. Then, following the3 vectors specifying the corners of therectangle, the user mustprovide [parameter(1)]x[parameter(2)] values specifying the relative source strength of each grid cell. Cell assignments are made by first incrementing in the (V1-V2) A1GRID direction and secondarily in the (V3-V2) A2GRID direction. Regardless of source strengths input per cell, results are normalized to a single source particle over the entire source. The code uses uniform random sampling of source positions within each rectangular cell. If [parameter(1)] and [parameter(2)] are both unity or omitted, then there is only one grid cell, the relative strength must be 1.0, and the user should not attempt to specify the source strength within the cell.

(f) **SURFACE** Syntax: SURFACE [keyword] (non-ACCEPT) or SURFACE [parameter(1-3)] [keyword] (CYLTRAN) or SURFACE [parameter(1)] BODY [parameter(2)] [keyword] (ACCEPT)

Example: SURFACE ZMAX (non-ACCEPT)
or SURFACE 1.0 2.5 5.0 OUTWARD (CYLTRAN)
or SURFACE 3BODY 2 (ACCEPT)

Default: There is no default for the surface source option.

This sub-keywordspecifies the surface from which source particles will be started.
For TIGER and CYLTRAN, surface **ZMIN** refers to the minimum-z surface and surface

ZMAX is the maximum-z surface. For CYLTRAN, surface **RMAX** is the lateral escape
surface at maximum radius.
For CYLTRAN, an arbitrary cylindrical surface can be specified. [parameter(1)] is the

lowerZ coordinate. [parameter(2)] is the upperZ coordinate. [parameter(3)] is the ra
dius of the cylindrical surface. By default the reference direction is inwardnormal, but

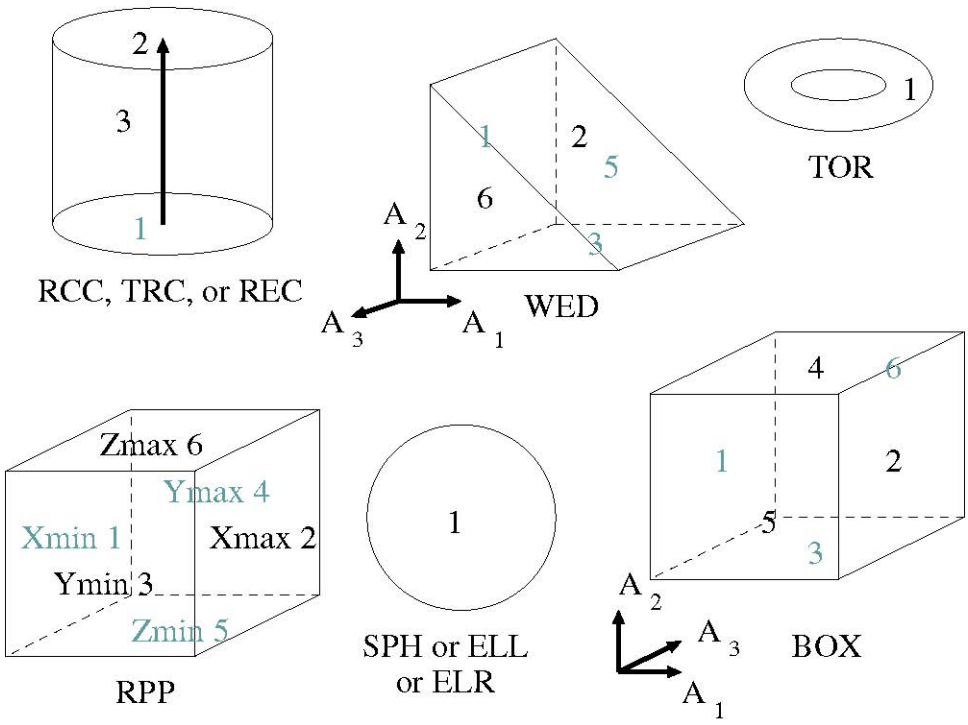


Figure 2. Surface indices for ACCEPTbodies

thereferencedirectionmaybe definedas outwardnormalbyusingthekeyword **OUT-**
WARD. Thereference direction cannotbe changed with the DIRECTION keyword. For ACCEPT, it is necessary to specify both the **SURFACE** index number and the associated **BODY** number. The body referred to need not be part of the actual zone description of the geometry. Figure2 on page 77 illustrates how many of the body types used in ACCEPT have their surfaces numbered. In the case of the arbitrary polyhedron (ARB), the user explicitly specifies the order in the input. The right circular cylinder is numbered the same as the truncatedright-circular cone (TRC). The rectangular parallelepiped (RPP) is numbered the same as the BOX where A1 points in the positive x-direction,A2pointsinthe positivey-direction,andA3pointsinthe positivez-direction. By default the reference direction is inwardnormal, but the reference direction may be defined as outwardnormalby using the keyword **OUTWARD**. Thereference direction cannot be changed with the DIRECTION keyword.

(g) **UNIFORM-ISOTROPIC-FLUX** *(CYLTRA*N *or ACCEP*T *Only*) Syntax: UNIFORM-ISOTROPIC-FLUX [parameter(1)] Example: UNIFORM-ISOTROPIC-FLUX [parameter(1)] Default: There is no default UNIFORM-ISOTROPIC-FLUX source.

Thiskeywordallowsforthe simulationofauniformisotropic radiationfield.Thereference direction is inwardnormal, and the distribution is cosine-law on the surface. The reference direction cannot be changed with the DIRECTION keyword.

ForCYLTRAN,no parameteris necessary;acylindersufficientto surroundtheproblem is automatically used.

8. Keywords for ITS

For ACCEPT, [parameter(1)] specifies the index of the body over which the source will be sampled. **WARNING:**Thereis no diagnostic to ensurethat anywhereoutside of this body is the escape zone.

(h) **VOLUME** Syntax: VOLUME [parameter(1)] [SHELL] [parameter(2)] [ZONE] [parame ter(3)]

Example: VOLUME 3

Example: VOLUME 3SHELL 1.5 (ACCEPT codes)

Example: VOLUME 2ZONE 5 (ACCEPT codes)

Default: There is no default for the volume source option.

For TIGER, [parameter(1)] is the input-zone index.

For ACCEPT, [parameter(1)] is a body index. For a SPH or RCC body, the keyword **SHELL** may be used with [parameter(2)] specifying the inner radius of the spherical or cylindrical shell. For

ACCEPT, the body referred to need not be part of the actual zone description of the geometry. The keyword **ZONE** may be used with [parameter(3)] to specify an input zone that will be used to further specify the extent of the source. All source particles will be inside both the specified body and the specified zone.

59. **PRINT-ALL**

Syntax: PRINT-ALL Default: Only the
cumulative results for the final batch
will be written to the output file.

Thisprimary keyword causesthe cumulative resultsfrom each batchtobe writtentothe output file. In parallel calculations, other control keywords may prevent every batch from being written to output.

60. **PULSE-HEIGHT** *(ITS Only*)

Syntax: PULSE-HEIGHT [parameter(1)] [parameter(2)]

Example: PULSE-HEIGHT47

Default: No spectrum of absorbed energy will be calculated.

This keyword causes the spectrum of absorbed energy to be calculated for input zones [parameter(1)] through [parameter(2)]. These parameters correspond to the order of the input zones as those zones weredefined. If the parameters areleft blank, the spectrum of absorbed energy will be calculated for the entiregeometry. Certain biasing schemes, such as those activated by the sub-keywords SCALE-BREMS and SCALE-IMPACT, are inconsistent with this calculation; PULSE-HEIGHT will cause them to be deactivated (a message so informing the user is written to the output file). The following secondary keyword describes the energy bin structure used in tallying the spectrum of absorbed energy.

(a) **NBINE** Syntax: NBINE [parameter(1)] [keyword]

f
o
r

I
T
S

Example:
NBINE6USER
1.99999 1.0 0.5 0.25
0.00001 0.0 Default:
ten bins of equal width
plus total absorption
and escape (i.e., 12
bins total).

If [keyword] is not specified then [parameter(1)] is the number of desired equal-width bins plus 2, to account for both total absorption and escape. The highest energy bin is only total absorption if the source is mono-energetic. If [keyword] is USER, [parameter (1)] is the number of bin energies to be read. The only choice for [keyword] is:

i. **USER -**User defined energy grid. The code will thenread the lower boundsof the energy bins (MeV) in descending order as in the above example. The maximum lower bound must be less than the maximum source energy. In the above example, the first lower bound and the last two lower bounds were chosen to ensure that total absorption (full source particle energy absorbed in the selected region) and total escape (no energy absorbed in selected region for a given source particle), respectively, would be accounted for.

Note that the primary keywordalone, with no other parameters or keywords, will result in the calculation of the spectrum of absorbed energy for the entire geometry using the default bin structure.

61. **RANDOM-NUMBER**

S

y

n

t
a
x
:

R
A
N
D
O
M
-
N
U
M
B
E
R

[
p
a
r
a
m
e
t
e
r
(
1
)
]

E
x
a

m
p
l
e
:

R
A
N
D
O
M
-
N
U
M
B
E
R

4
2
6
5
6
4
1
5
4
2

D
e
f
a
u
l

t
:
0
(
c
o
n
v
e
r
t
e
d

t
o
5
1
9
)

[parameter(1)]isthe initialrandomnumberseedfortheMonteCarlorun.Thiskeywordcan

be used to start a run with the final random number from an earlier run.
For RNG1, this keywordcan also be used in debugging to isolate the offending primary his
tory. Fora similar purpose, the more sophisticated user can use this keywordin conjunction
with a print of the initial random number seed of a source particle, IRSAV.

For RNG2 and RNG3, the state of the RNG is more than a single seed, but this keyword can
beusedtospecifyabatch.SeetheRESTART-HISTORYkeywordforamethodof debugging

with those generators.

See the Random Number Generators section for further discussion of issues concerning ran
dom number routines.

62. **READ-CONNECTIVITY**

Syntax: READ-CONNECTIVITY

Default: Geometry connectivity informationisreadfroma file.

8. Keywords for ITS

Geometry connectivity informationisreadfromthe “fort.13”file.Thiscanbeafilegenerated by ITS using the DUMP-CONNECTIVITY keyword, or it can be a file generated by some other means. Incorrect connectivity information will not cause the code to give inaccurate results, but it may decrease the efficiency of the calculation.

The connectivity information used by ITS is based on each body entry in a zone description in the input geometry. Even if a single body is used numerous times in describing a zone or in describing multiple zones, the code maintains connectivity information for each usage of the body. The connectivity information indicates which zones a particle may enter upon

leaving the body definition. Thus, the connectivity file must contain at least one entry for each body-entry in the geometry description.

The format of the connectivity file is as follows: The first line must contain a single number indicating the number of connectivity values to follow. This must be at least two times the number of body entries in the zone definitions. The following data is the connectivity array and forms a kind of linked list. The (2n − 1)th value is a zone to which the corresponding *n*th body entry is connected. The (2*n*)th value is the location in the array of the next zone connection for the *n*th body entry. If there are N body entries, the first 2N values must be devoted to the initial data of those body entries. If a body does not connect to any zone, a value of zero should be used. If the connectivity data does not continue at another point in the array, a value of zero should be used to indicate that. The connectivitydata beyond the first 2N values can appear in any order, but careful attention should be given to how the data is linked together.

63. **REFLECTION-ZONE** *(ACCEP*T *an*d *ACCEPT*P *Only*)

Syntax: REFLECTION-ZONE [parameter(1)]

Example: REFLECTION-ZONE 20

Default: No reflection zone.

Particles undergo specularreflectionatthe boundariesofthe selected zone. Source particles cannot be initiated in the reflection zone, and no particles can enter the reflection zone. This keywordcannot be used with EBFIELDS or with a CAD zone.

64. **RESTART**

Syntax: RESTART

Default:Anew calculationis performed insteadofarestart.

The problem is restarted at the batch number of the data in the “fort.14” file. Adump file must have been written, saved, and named “fort.14” (see FILE-NAMES keywordfor alternative names). The batch size on the restart run must be the same as those on the dump file to permit accurate computation of statistical uncertainties. If specified otherwise, the batch size will be set equal to the batch size used to generate the dump file. The number of batches specified in the restart input file should

be the desired number of additional batches.

“Infill” batches represent gaps in the simulation that created the restart dump. If a calculation finishes successfully, there will be no infill batches in the dump file. If the dump file contains infill batches, those will be run in addition to the number of batches requested. No output of results will be written until all infill batches have been completed.

8. Keywords for ITS

65. **RESTART-HISTORY** *(RNG2 and RNG3 Only*)

Syntax: RESTART-HISTORY

Default:Anew calculationis performed insteadofarestart.

This keywordspecifies that a restart will be executed using the state of the random number generator specified in the file “rngstate.dump”. This file is written when the program exits due to an error in execution. It contains the state of the random number generator at the start of the particle history in which the error occurred. Only the offending history will be executed. This allows the user to repeat the single history in which an error occurs.

In parallel, the restart must be performed with static load balancing on one processor, since the RNG state is not passed to subtasks.

66. **SIMPLE-BREMS** *(ITS Only*)

Syntax: SIMPLE-BREMS

Default: more accurate bremsstrahlung distributions.

This keyword is intended for development purposes only. This keyword specifies that bremsstrahlungangular distributionissampledfromasimpleapproximationtothe Bethe-Heitler formula insteadof the combinationof formulasby Koch and Motz.

67. **SOURCE-SURFACES** *(MITS Adjoin*t *Only*)

Syntax: SOURCE-SURFACES [parameter(1)]

Example: SOURCE-SURFACES3

Default: All surfaces of the escape zone (tallied as a single surface) for ACCEPT.

Both surfaces (ZMIN and ZMAX) for TIGER.

This specifies the number of surfaces on which either ELECTRON-SURFACE-SOURCEs or PHOTON-SURFACE-SOURCEs will be tallied. These specified surfaces should correspond to adjuncton-escape surfaces. This keyword must be followed by [parameter(1)] separate lines of the secondary keywordSURFACE.

Any specification for CAD will be ignored. All escaping particles are scored in a single body/surface tally.

(a) **SURFACE** Syntax: SURFACE [keyword] (non-ACCEPT codes) or SURFACE [parameter(1)] BODY [parameter(2)] (ACCEPT codes) Example: SURFACE ZMAX (non-ACCEPT codes) or SURFACE3BODY2 (ACCEPT codes) This sub-keywordspecifies the surface index through which the

escaping particles will be calculated. For TIGER and CYLTRAN, surface **ZMIN** refers to the minimum-z surface and surface **ZMAX** is the maximum-z surface. For CYLTRAN, surface **RMAX** is the lateral escape surface at maximum radius. For ACCEPT, it is necessary to specify both the surface index and the **BODY** number. Figure2 on page77illustrates how many of the body types used in ACCEPT have their surfaces numbered. In the case of the arbitrary polyhedron (ARB), the user explicitly specifies the order in the input. The right circular cylinder is numbered the same as the truncated right-circular cone (TRC). The rectangular parallelepiped (RPP) is numbered the same as the BOX where **A**1 points in the positive x-direction, **A**2 points in the positive y-direction, and **A**3 points in the positive z-direction.

68. **SPECTRUM**

Syntax: SPECTRUM [parameter(1)] [keyword] [keyword] [keyword]

Example: SPECTRUM5

1.00 0.80 0.76 0.53 0.00

5.0 4.0 3.0 2.5 2.0

Example: SPECTRUM5NUMBER-PER-BIN

0.40 0.08 0.46 1.06

5.0 4.0 3.0 2.5 2.0

Example: SPECTRUM5NUMBER-PER-BIN-PER-MEV

0.20 0.04 0.46 1.06

5.0 4.0 3.0 2.5 2.0 Default: mono-energetic source (ITS) or mono-group source (MITS)

In forwardmode, thiskeywordspecifies that the energy distributionisa spectrum. [parameter(1)] is the number of energy grid points describing the spectrum (or one more than the number of energy bins in the spectrum). The spectrum follows this keyword, decreasing monotonically to 0.0. The corresponding spectrum energy grid is given on the next line. If the **NUMBER-PER-BIN** or **NUMBER-PER-BIN-PER-MEV** keywords appear, the code will convert the distribution to a cumulative distribution internally. The examples given above result in identical spectra.

In any of these formats, the spectrum does not have to be normalized (e.g., the cumulative distribution of the spectrum does not have to begin with 1.0). If the spectrum is not normalized, the code will produce a warning, normalize the spectrum, and proceed with the calculation. The distribution is normalized based only on the spectrum data provided. If a portion of the spectrum falls below the cutoff energy, source particles sampled from below the cutoffwill not be tracked (but information about the fraction of such rejected particles will appear in the output).

The spectrum may contain line sources and ranges of the spectrum with zero probability, such as in the following example:

Example: SPECTRUM4

1.0 0.5 0.5 0.0

1.3325 1.3325 1.1732 1.1732

y
w
o
r
d
s

f
o
r

I
T
S

The user may bias the energy sampling of source particles. If the **BIASED** keywordappears on the same line as the SPECTRUM keyword, the first spectrum read is the spectrum sampled from, and the second spectrum is the true spectrum of the source particles. Bothspectra mustbeonthe same energygrid.Inthe following example,the 1.3325MeVlinewillbe sampled3times more oftenby the code than the 1.1732 MeV line (but the particle weights will be adjusted to account for the fact that the true source has equal probability of each line).

Example: SPECTRUM4BIASED

1.0 0.25 0.25 0.0

1.0 0.5 0.5 0.0

1.3325 1.3325 1.1732 1.1732

In adjoint mode, the **SPECTRUM** keywordallows the detector response to be calculated by folding with multiple forward source spectra during a single calculation.

Syntax: SPECTRUM [parameter(1)] ADJOINT-SPECTRA [keyword] [keyword] Example: SPECTRUM2ADJOINT-SPECTRA PHOTON

5

2

.

3

1

4
1
.
5
6

1.00 0.80 0.76 0.53 0.00
14.2 13.1 10.1 8.6 5.4
1
.
0
0

0
.
5
0

0
.
2
5

0
.
0

1
4
.
0

1
2
.
0

9
.
0

5
.
0

Example: SPECTRUM1ADJOINT-SPECTRA
PHOTON NUMBER-PER-BIN 5
0.462 0.0924 0.5313 1.2243
14.2 13.1 10.1 8.6 5.4

Default:The detectorresponseisonly calculatedbyfoldingwithaflatforwardspectrum.

In adjoint mode, the secondary keyword **ADJOINT-SPECTRA** must be included on the same line as the SPECTRUM keyword, as well as a keywordspecifying the type of source particle as one of **PHOTON** or **ELECTRON**. [parameter(1)] specifies the number of forward spectra. [parameter(1)] lines must follow, each containing two parameters: the first parameter specifies the number of energy grid values (the number of bins plus one) in the corresponding spectrum, and the second parameter specifies the magnitude (or source strength) of the spectrum. Then,2 [parameter(1)] lists must follow. For each spectrum, the first list is the distribution, and the second list is the corresponding energy bin grid. The energies need not correspond to the energy divisions on the cross section set, however source energies cannot include energies for which cross sections are not available. The keywords **NUMBER-PER-BIN** or **NUMBER-PER-BIN-PER-MEV** can be added to the keyword line to specify those formats for the spectra, otherwise the format is assumed to be a cumulative distribution of the number of particles per bin.

8
.

K
e
y
w
o
r
d

s

f
o
r

I
T
S

WARNING: The results are always multiplied by the source strength factor. It is also possible to specify an unnormalized spectrum. While it is possible to apply a source strength factor to an unnormalized spectrum, this is not likely to be a desired feature. If no source strength factor is specified, none will be applied (that is, the factor will be set to 1.0).

The default is scoring with a flat forward spectrum and is included in the output if this keyword is not used. If this keyword is used to specify forward spectra, output will be generated based on folding with the forwardspectra. In either case results will be given in the energy bin structure specified with the **NBINE** sub-keywordof the SURFACE-SOURCE selected. However, in the case of the default, results will correspond with folding against a flat spectrum of unit strength for each energy span reported. That is, the unit strength is applied to each energy span, not to the entire energy span of the problem. If this keyword is used, results will correspond to the response due to source particles in the energy span reported.

69. **TASKS***(MP*I *Only*)

Syntax:TASKS [parameter(1)] [parameter(2)] [parameter(3)] Example:TASKS50 100 1.1 Default: Number of processors available is used. Intermediary output after every batch. No termination due to stray processes.

This keyword applies only for parallel processing. [parameter(1)] specifies the number of processors to which batches can be distributed. If [parameter(1)] is negative, the master process will only distribute work and process results, but static load-balancing will still be used (unless the DYNAMIC-MPI keywordis used). [parameter(2)] specifies the number of batches between intermediary outputs. [parameter(3)] is a factor multiplying the running average of the batch time such that if any batch time

exceeds this time, the batch is considered to be in an infinite loop, and if all other processors have completed their calculations, the run will be terminated.

Iftherunis terminatedforan assumed infiniteloop,the random numberseedforthat batch is output so that, subsequently, that batch alone can be run by using that random number seed as the parameter for the RANDOM-NUMBER keyword. If the code itself detects an error condition that is not an infinite loop and calls ABORTX, the number of random numbers to the beginning ofthe offending history is output and the “rngstate.dump” file is written. The usercan subsequentlyrunonlytheoffendingbatch(withthe RANDOM-NUMBERkeyword) or the offending history (with the RESTART-HISTORYkeyword).

70. **TITLE**

Syntax:

TITLE [titledata]

Example:

TITLE

Adjoint

Dose

calculatio

n

i
n

A
l

b
o
x

i
n

S
a
t
e
l
l
i
t
e

G
P
S
-
4

D
e
f
a
u
l
t
:

n
o

t
i
t
l
e

This keywordsignals that the next line of input contains [title data], which is a title of up to 80 columns that will be written to the output file and will be used as the title on any plots that are generated.

8
.

K
e
y
w
o
r
d
s

f
o
r

I
T
S

71. **TRAJECTORY-POINTS** *(CYLTRA*N *or ACCEP*T *Only*)

Syntax: TRAJECTORY-POINTS parameter(1)
Example: TRAJECTORY-POINTS 1000
Default: 500

Withthiskeyword,the usercanchangethe maximum numberofpointsplottedper particle track. Achange is required

only when specified by ITS output. For CYLTRAN, this value will automatically be at least 4. For ACCEPT, this value will automatically be at least one more than the resolution for plotting curved edges (see the PLOTS-OVAL-RESOLUTION keyword).

9. Summary of ITS-CAD Keywords

Last Modified Date: 2008/04/17 22:45:30

9 Summary of ITS-CAD Keywords

This section contains a listing of keywords relevant to the prmfile used with the CAD codes. There are no defaults. The ITS Simulation section must always be included. The other sections may always be included, and they will only be read if the geometry engine has been included in the executable. More detailed descriptions of the syntax and use of these keywords are contained in the Keywords for ITS-CAD section.

Table 8. ITS-CAD keywords and default settings

KEYWORD	FUNCTION
**** ITS SIMULATION ****	
ITS BEGIN	Marks start of section
INPUT FILE	Specifies the ITS input file
OUTPUT FILE	Names the ITS output file
MODE	Usage of CAD and CG geometry
KD TREE PARAMS	Parameters for efficiency grid
ITS END	Marks end of section
**** ACIS GEOMETRY ****	
ACIS BEGIN	Marks start of section
ACIS END	Marks end of section
**** CHOLLA GEOMETRY ****	
CHOLLA BEGIN	Marks start of section
CHOLLA END	Marks end of section

10. Keywords for ITS-CAD

Last Modified Date: 2008/04/17 22:45:29

10 Keywords for ITS-CAD

This section contains the keywords for the prmfile (parameter file). This input file is required when using the CAD preprocessor definition to compile. The ITS-CAD codes do not necessarily imply that CAD geometries are being used, since the MODE keyword can be used to specify a CG ONLYcalculation. Refer to the section on Running ITS for moreinformation on setting up the prmfile section of a CUI file or using a prmfile otherwise.

There are multiple sections of input in the prmfile, with each section isolated by BEGIN and END keywords. The ITS simulation section is within the ITS BEGIN and ITS END keywords. It contains the names of the ITS input and output files and the simulation parameters. All keywords within the ITS section are required. Other sections specify the simulation geometry. If the executable includesa geometry engine (e.g.,ACIS), thena section mustbe specifiedfor that engine (e.g., ACIS BEGIN and ACIS END) even if that geometry section is empty.

All keywords should be entirely uppercase. For keywords with sub-keywords or parameters, the parameter or sub-keywordmust appear on the same line as the keyword. Underscores do not denote spaces and must be included in the keywords.

Commentsare allowedina parameterfileandareprecededwiththe#symbol. Anythingto the right of a comment symbol is ignored. Extra characters after valid input is an error and will cause an error message and an abort. Blank lines are ignored. Text between sections of input is ignored. A missing * END card is an error and will cause an error message and an abort. A repeated or missing keywordis an error and will cause an error message and an abort. There are no default values. Missing data is an error will cause an error message and an abort.

The geometry specified in the prmfile may be assigned to zone numbers between 0 and 1 million. The zone numbers do not need to be sequential, and the numbering may contain gaps. However, the assigned zone numbers willbe condensed for useby ITS. Thus,if7 unique zone numbers are used in the prmfile (e.g., 10 51 1201350), then ITS will consider these to be zones 1through7, numberedas(4751236). Thatis,inthis example,thebody assignedas0willbe referred to as zone 1, the body assigned as 51 will be referred to as zone 7.

1 **ITS BEGIN**
2 **INPUT FILE**

Syntax:

I
T
S

B
E
G
I
N

This keywordis required.

Syntax: INPUT FILE [keyword]

Example: INPUT FILE its.inp

Default: none.

This keywordspecifies the name of the ITS input file as [keyword].

3. **OUTPUT FILE**

Syntax: OUTPUT FILE [keyword]

10. Keywords for ITS-CAD

Example: OUTPUT FILE its.out

Default: none.

This keywordspecifies the name of the ITS output file as [keyword].

4. **MODE**

Syntax: MODE [keyword]

Example: MODE HYBRID

Default: none.

This keywordspecifies the transport mode as one of: **CAD ONLY** ,**HYBRID**,**MIRROR CG**, or **CG ONLY**.

5. **KD TREE PARAMS**

Syntax: KD TREEPARAMS

[parameter(1)] [parameter(2)]
[parameter(3)] [parameter(4)]
[parameter(5)] [parameter(6)]
[parameter(7)] Example:KD
TREEPARAMS9919922-10 Default: none.

We recommend using the parameter settings shown in the example, but users may wish to
study the effect of other parameter settings on the efficiency (and accuracy) of their calcula
tions.

parameter(1) sets the maximum depth of any branch of the kd-tree.
parameter(2) sets the maximum desired number of faces per leaf of the kd-tree. The code

will attempt to refine the kd-tree until either this constraint is satisfied or the maximum tree
depth has been reached.
parameter(3) sets the tree depth at which the cost-function heuristic changes from a solid-

angle heuristic to an octree (midpoint). This is only active when the user has specified a

solid-angle heuristic with [parameter(6)]=-1.
parameter(4) sets the axis flag, which determines upon which Cartesian axis to place the
nextsplit-
plane.Avalueof0or2startswithasimplerotationthroughthe3axes.Anyother value starts with the longest axis of the cell which will be split. Additionally, the behavior is
different for values greater than 1. For such values, the split-plane location will be accepted
onlyifit “usefully” dividestheaxisinsuchawaythatthereisatleastoneobject completely
to the right and another object completely to the left of the split-plane. If such a “useful”
split-plane is not found, such a split-plane will be searched for on the next (by rotation)
axis. If all three axes do not result in a “useful” split-plane, the cell will not be divided any
further. For parameter(4) values less than or equal to one, a split-plane will always occur on

the chosen axis (i.e., at the “default” location, if an optimized one cannot be determined -see
parameter(7)).

parameter(5) sets the split-plane flag, which is only used when the cost function is something
other than an octree (see parameter(6)). For the value of 2, the split plane is placed at a
midpoint between two nearest (but greater than 1.E-6 cm) bounding-box boundaries. For

10. Keywords for ITS-CAD

any other value, the split-plane is placed on one of the bounding-box boundaries. At this
time, only the valueof2is functional. parameter(6) sets the cost function, which determines the heuristic for placing the split-plane.Avalueof0isan octree, whichsimply meansthe split-planeis placedatthe midpoint ofalengthofacell.Avalueof-1isthe solid-angle heuristicof Havran.Anyothervalue,W, isana¨ıve heuristic which minimizes the function (W*Nsp+abs(Nleft -Nright)), whereNleft is the number of objects completely to the left of the plane, Nright is the number completely to the right, and Nsp is the number straddling the split-plane.

parameter(7) sets the default split-plane location, which is only used for parameter(4) values lessthan2.A zerovalueplacesitonthe midpoint (similartoanoctree),while1placesiton the maximum boundary. The valueof0isrecommended.

6. **ITS END**

Syntax: ITS END

This keywordis required.

7. **ACIS BEGIN**

Syntax: ACIS BEGIN

This keyword is required if the ITS executable is built with an ACIS geometry engine included. Spatial Corporation marketsthe3DACIS Modelerasacommercially availableproduct [34]. This version of ITS supports linking with the 3D ACIS Modeler version R12, R15, or R16,

which must be purchased directly from Spatial Corporation.

Between the ACIS BEGIN and END statements, the user may supply the ACIS geometry. Each line should contain the name of a file and the zone number to which the geometry belongs. For example: acispart.sat2 Each ACIS file may contain information for one zone or multiple zones. If the satfile contains more than one ACIS body and the user specifies a positive zone number, then the code will sequentially number each ACIS body asaunique zone number beginning with the specified zone number. If the user specifies a negative zone number, the code will use the absolute value and assign every ACIS body in the file to that single zone. It is also possible to assign information in multiple files to a single zone, such as: acispart1.sat3 acispart2.sat3 It is also possible to combine multiple geometry descriptions into a single zone description, such as describing some zone faces with Cholla facets and other zone faces with ACIS surfaces.

8. **ACIS END**

Syntax: ACIS END

This keyword is required if the ITS executable is built with an ACIS geometry engine included.

10. Keywords for ITS-CAD

9. **CHOLLA BEGIN**

Syntax: CHOLLA BEGIN

This keyword is required if the ITS executable is built with a Cholla geometry engine included. The Cholla engine allows for the use of facet geometry in the format written from Cubit.

Between the Cholla BEGIN and END statements, the user may supply the Cholla geometry. Each line should contain the name of a file and the zone number to which the geometry belongs. For example: /home/geometry/chollapart.fac3 Each Chollafilemay contain informationforonlyone zone,buta zonemaybe describedby information in more than one file, such as: chollapart1.fac3 chollapart2.fac3 It is also possible to combine multiple geometry descriptions into a single zone description, such as describing some zone faces with Cholla facets and other zone faces with ACIS surfaces.

10. **CHOLLA END**

Syntax: CHOLLA END

This keyword is required if the ITS executable is built with a Cholla geometry engine included.

11. TIGER Geometry

Last Modified Date: 2008/04/17 22:45:37

11 TIGER Geometry

The geometry of the TIGER codes is the simplest of the ITS member codes. It is strictly one dimensional.Aparticle trajectoryis describedonlyin termsofthezcoordinateof positionandthe z direction cosine. Nevertheless, this is often all that is necessary, and, because the TIGER codes are the fastest and simplest to use, they should always be considered. They are especially useful in obtaining accurate answers to questions involving very basic transport phenomena.

11.1 Problem Geometry

Beginning at z =0.0,layersarestackedalongthepositivezaxis accordingtotheorderinwhich they are read in as described under Geometry Syntax. For each layer the user must define: (a) the material index, (b) the number of subzones into which the layer is to bedivided for purposes of scoring charge deposition, energy deposition, and particle flux (see the discussion of Automatic Subzoning), and (c) the thickness of the layer. The material indices are defined by the order in which the materials are specified in executing the cross-section generating code.

Interior voids must not be defined. Voids within the geometry have no effect upon one-dimensional transport. Omission of voids allows for increased efficiency in the calculation and is a requirement for TIGER geometry descriptions.

11.2 Conventions for Escaping Particles

In addition to quantities internal to the problem geometry such as charge deposition, energy deposition and particle flux, radiation that escapes may also be scored. Because geometry is defined as infinite slabs with only a finite z-dimension, particle escape is classified as one of:

1 Radiation that escapes from the maximum-z boundary of the problem.
2 Radiation that escapes from the minimum-z boundary of the problem.

In forwardmode, these definitions may be applied by the user with the ESCAPE-SURFACES keyword to specify where escaping particles are to be tallied. In adjoint mode, the SOURCESURFACES keywordspecifies where escaping adjuncton particles are to be tallied. The default in either mode is to perform tallies on both surfaces.

11.3 Geometry Syntax

Syntax: GEOMETRY[parameter(1)]

[parameter(2)] [parameter(3)] [parameter(4)]

Example: GEOMETRY3

310.1

1
1
0

1
2
.
0

2
5
0
.
1
5

1
1
.

T
I
G
E

R

G
e
o
m
e
t
r
y

Default: no default

[parameter(1)] is the number of input layers. Immediately after the keywordline there must follow a series of [parameter(1)] lines, one for each layer, containing [parameter(2)] through [parameter(4)]. These specify the material, the number of subzones, and the layer thickness in cm.

12.
CY
LT
RA
N
Ge
om
etry

Last Modified Date: 2008/04/17 22:45:36

12 CYLTRAN Geometry

Therearea varietyof experimentalproblemsin whichthe symmetryrequirementsoftheCYLTRANcodesaresatisfiedtoagoodapproximation.Thisis especiallytrueinthose experimentsfor which the radiation source is itself a cylindrical beam, as in the case of many pulsed and steady-state electron accelerators.Theonly essentialrequirement, however,isthatthe material geometry, as specified by the input zones, be cylindrically symmetric. The trajectories themselves are fully three dimensional. In a code modification, the more sophisticated user may wish to define a nonaxisymmetric sourceor,inthe caseofCYLTRANM,anon-axisymmetric field configuration,along with whatever azimuthal tallies he desires. Note that the logic is already included for scoring azimuthally-dependent escape distributions (see keywords ELECTRON-ESCAPE and PHOTONESCAPE) and azimuthally-dependent charge deposition, energy deposition, and particle fluxes (see keywords ELECTRON-FLUX and PHOTON-FLUX, the remainder of this section, and the discussion of Automatic Subzoning).

12.1 Problem Geometry

The material geometry for the CYLTRAN codes consists of a right circular

cylinder of finite length, the axis of which coincides with the z axis of the Cartesian system that describes the particle trajectories. The location of this cylinder, hereafter referred to as the problem cylinder, along thezaxisis completelyarbitrary.Theentirevolumewithintheproblemcylindermustbespecified in termsof materialorvoidinput zones,eachof whichis boundedbytwoandonlytwo cylinders coaxialwiththezaxisandtwoandonlytwoplanes perpendiculartothezaxis.

The material configuration is then conveniently described by the half section of the problem cylinder obtained by passing a plane through its axis. An example of such a half section is shown in Figure 3. The horizontal base line is the axis of the problem cylinder, and the other horizontal lines are labeled by the radii of the corresponding cylindrical boundaries. The vertical lines are labeled by the z coordinate of the corresponding plane boundaries. The solid lines are actual material boundaries; the broken lines are not. The dashed lines are employed either to complete the perimeteroftheproblemcylinderhalf sectionortobreakmorecomplex zonesofagiven material (e.g.,thosehavingL-shapedhalf sectionsinFigure3)intothesimplerinput zonesrequiredbythe code(i.e.,zoneswhosehalf sectionsarerectangles).Thedottedlines describesubzoningofagiven input zone for purposes of obtaining charge deposition, energy deposition, and flux profiles.

Each zoneinFigure3is boundedbysolidand/or dashedlinesand containsa materialindex (circled).Azeroindex definesavoid zone; otherwise, the material indices aredefinedbythe order in which the materials are specified in executing the cross-section generating code. Each of these input zones requires a single input cardfor its description. The dotted lines illustrate subzoning of a particular input zone into equal axial and/or radial increments. Azimuthal subzoning is also possibleas describedundertheGeometrySyntax section.Italso followsfromthat discussionthat separateinputcards describingthese subzonesarenotrequired.This descriptionis accomplished internally by the code using the subzoning parameters specified on the input carddescribing the input zone. This feature allows the user to obtain three-dimensional charge deposition, energy deposition, and fluxprofiles withina given input zone witha single input card.

The following input cards describe the problem geometry illustrated in Figure 3.

```
        GEOMETRY6
12.
CY
LT
RA
N
Ge
om
etry
```

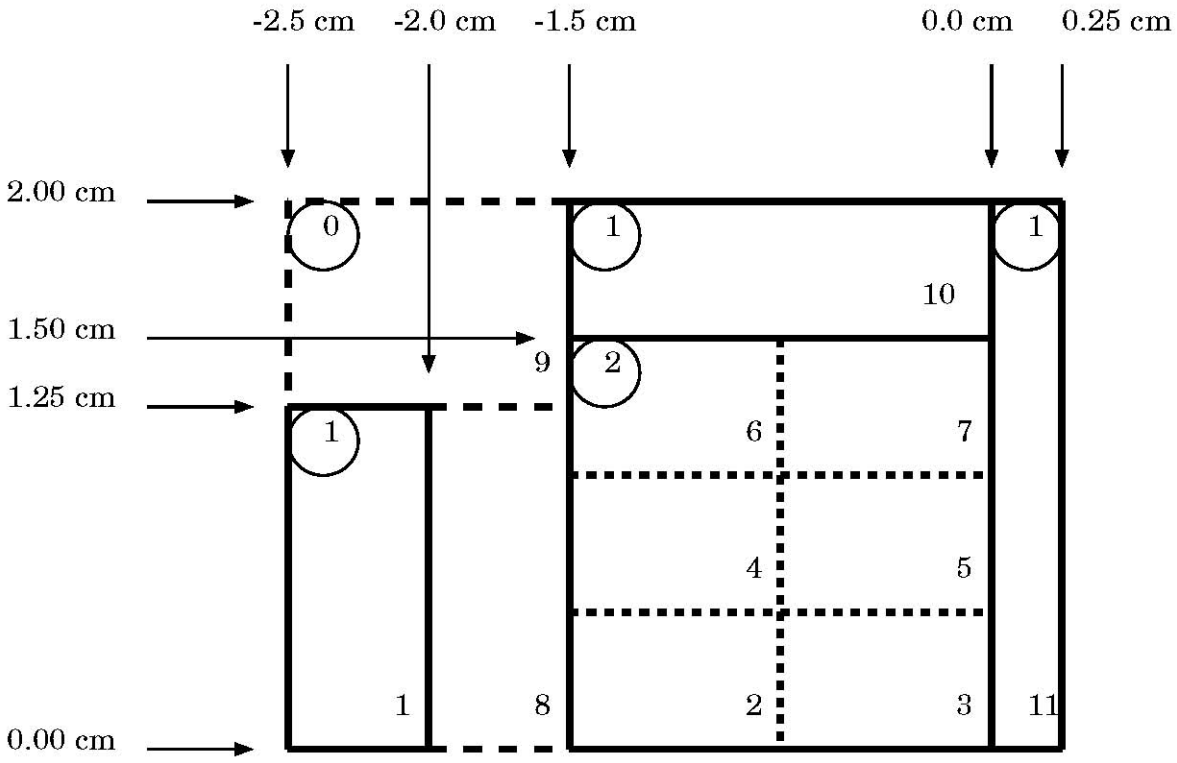


Figure 3. Example of the half section of a problem cylinder

-2.50 -2.00 0.00 1.251
-1.50 0.00 0.00 1.502132
-2.00 -1.50 0.00 1.250
-2.50 -1.50 1.25 2.000
-1.50 0.00 1.50 2.001

0.00 0.25 0.00 2.001

The numbers in the lower right hand corners of the subzones demonstrate how the code internally numbers these subzones. Subzone numbers are immediately assigned as each geometry cardis read. Therefore, subzones are numbered before the next cardis read. In the example, the second card, which describes the second input zone, generates6 subzones (subzones 2-7). The next input zone, therefore, has a subzone number of 8. It is important that the user understand this numbering scheme in order to properly interpret spatially-dependent outputs.

12.2 Conventions for Escaping Particles

In addition to quantitiesinternal to the problem cylinder, such as charge deposition, energy deposition, and particle flux, radiation that escapes from the problem cylinder may also be scored. Because the geometry is defined in

cylindrical coordinates of thickness z and radius r, particle escape is classified according as one of:

1. Radiation that escapes from the maximum-z boundary of the problem cylinder.

12.
CY
LT
RA
N
Ge
om
etry

1 Radiation that escapes from the minimum-z boundary of the problem cylinder.
2 Radiation that escapes from the maximum-r curved lateral boundary of the problem cylinder.

In forwardmode, these definitions can be applied by the user with the ESCAPE-SURFACES keyword to specify where escaping particles are to be tallied. In adjoint mode, the SOURCESURFACES keywordspecifies where escaping adjuncton particles are to be tallied. The default in either mode is to perform tallies at all three surfaces.

12.3 Geometry Syntax

Syntax: GEOMETRY[parameter(1)]

[parameter(2)] [parameter(3)] ... [parameter(9)]

Example: GEOMETRY3

-0.5 0.20 0.0 10.531150

0.2 12.45 0.0 10.510001

-0.5 12.45 10.5 12.052

Default: no default [parameter(1)] is the number of input zones. Immediately after the keywordline there must follow a series of [parameter(1)] lines, one for each input zone, containing [parameters(2)] through [parameter(9)]. These parameters specify the minimum z boundary, the maximum z boundary, the minimum ρ boundary, the maximum ρ boundary, the material, the number of φ subzones, the number of ρ subzones,andthe numberofz subzones.All boundariesare given in cm. When the parameters trailing the material index are left blank, no subzoning is imposed.

1
3
.

Last Modified Date: 2008/04/17 22:45:34

13 ACCEPT Geometry

The ACCEPT codes provide experimenters and theorists with a method for the routine solution of coupled electron/photon transport through three-dimensional multimaterial geometries described by the combinatorial method. In the combinatorial scheme, the problem input zones arebuiltupoutofprimitive bodies.Thisisin contrasttomore traditional schemesthat definethe zonesin termsof bounding surfaces.We find the combinatorial methodof specifying input zones in terms of solid bodies to be simpler, more intuitive, and less ambiguous than specification in terms of boundary surfaces. The combinatorial scheme also learns as the calculation progresses; at any particular time it makes use of information obtained from past experience in order to improve the efficiency of its search procedures used in particle tracking. This same learning ability precludes the requirement, typical of many other geometry schemes, for inputting a substantial amount of tracking information.

With the ACCEPT codes the user employs the combinatorial-geometry method in order to describe the three-dimensional material configuration of the problem. This task is accomplished in five distinct steps:

1 Define the location and orientation of each solid geometrical body required for specifying the input zones.
2 Specify the input zones as combinations of these bodies.
3 Specify zones to be subzoned and subzoning schemes, if necessary.
4 Specify the volumes of the subzones, if necessary.
5 Specify the material in each input zone.

The following sections discuss these five steps. Each section is divided into a general discussionof the geometry concepts anda more specific discussionof the datarequiredby the code.

The data is inserted in free format form with spaces or commas as delimiters. Syntax is discussed in Sec. 13.7 and simple examples can be found in its/Tests/RegTests/Input.

Due to limitations in the accuracy of the geometry tracking algorithm used in ITS, we discourage using geometry with important features smaller than 1E-6 cm. In some cases, bodies with such small dimensions will trigger input errors.

13.1 Specification of Input Bodies

13.1.1 Body Definition

The combinatorial-geometry method requires a library of geometrical body types from which the user may choose in order to describe his problem configuration. The information required to specify each body type in a three-dimensional Cartesian system is as follows:

1. Rectangular Parallelepiped (RPP) – Specify the minimum and maximum values of the x,y andzcoordinates that boundarectangular parallelepiped whose six sides areperpendicular to the coordinate axes.

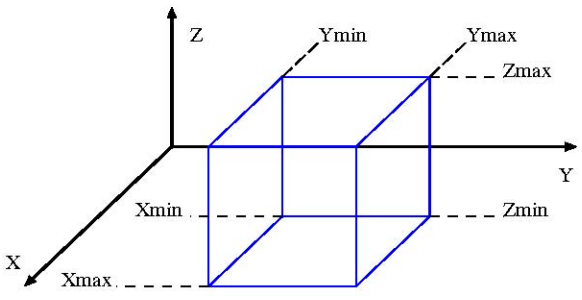


Figure 4. Rectangular Parallelepiped(RPP)

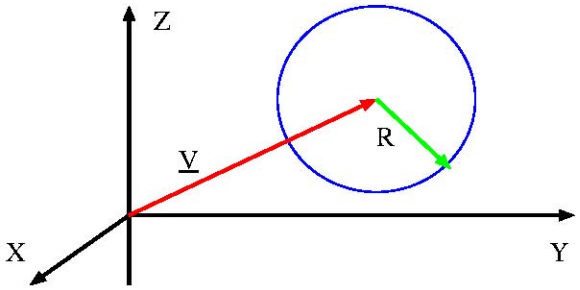


Figure 5. Sphere (SPH)

13. ACCEPT Geometry

1 Sphere(SPH) – Specify the components of the radius vector **V**to the center of the sphereand the radiusRof the sphere.

2 Right Circular Cylinder (RCC) – Specify the components of a radius vector **V**to the center of one base, the components of a vector **H**from the center of that base to the center of the other base, and the radiusRof the cylinder.

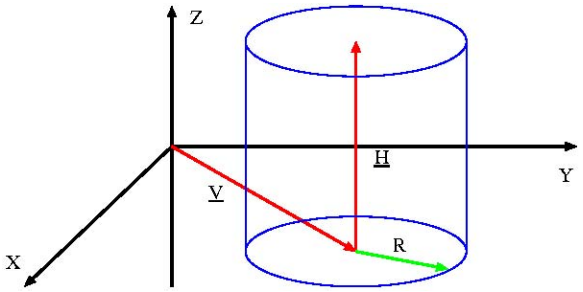


Figure 6. Right Circular Cylinder (RCC)

4. Right Elliptical Cylinder (REC) – Specify the components of a radius vector **V**to the center of oneofthe elliptical bases,the componentsofa vector **H**from the center of that base to the center of the other base, and the components of two vectors **R**1 and **R**2 that define the major and minor axes, respectively, of the bases.

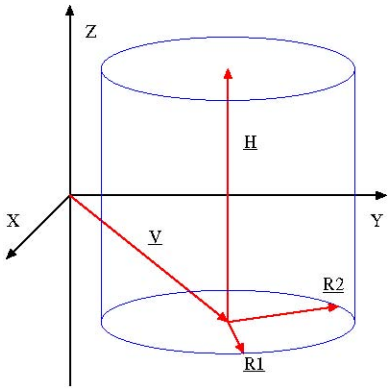


Figure 7. Right Elliptical Cylinder (REC)

1 Truncated Right-Angle Cone (TRC) – Specify the components of a radius vector **V** to the centerof one base, the componentsofa vector **H**from the center of that base to the center of the other base, and the radiiR1 andR2 of the first and second bases, respectively.
2 Ellipsoid (ELL) – Specify the components of the radius vectors **V**1 and **V**2 to the foci of the prolate ellipsoid and the length of the major axis R. This ellipsoid must be prolate – to

1
3
.
A

CCEPTGeometry

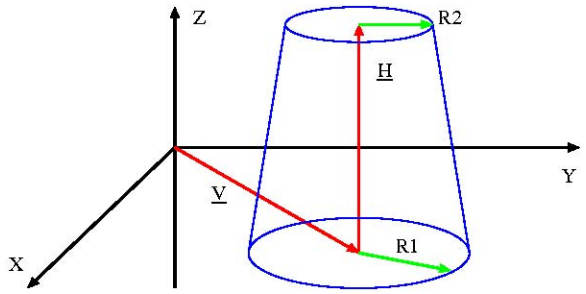


Figure 8. Truncated Right-Angle Cone (TRC)

specify an oblate ellipsoid use the ELR (see next primitive) instead of the ELL. The foci of a prolate ellipsoidareonthemajoraxis,theaxisaboutwhichtheellipseisrotatedtoformthe body. The center of the body lies halfway between the foci. The square of the length of the major axis equals the sum of the square of the length of the minor axis plus the square of the distance between the foci.

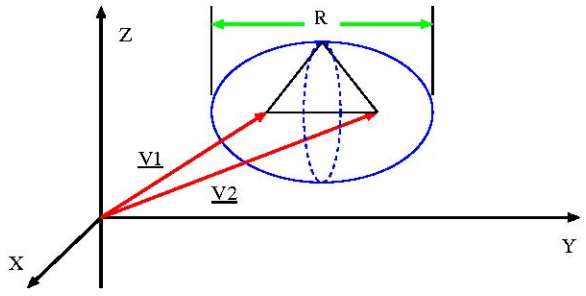


Figure 9. Ellipsoid (ELL)

1 Ellipsoidof Revolution (ELR) – Specify the componentsofa radius vector **V**to the center of the ellipsoid, the componentsofa vector **H**from the center of the ellipsoid to the apex along the axis of revolution, and the semi-axis length **R**in a direction perpendicular to the axis of revolution. Notice this ellipsoid maybe eitherprolate(**H** > R) or oblate(**H** < R).

2 Wedge (WED) – Specify the components of a radius vector **V**to one of the corners and the components of three mutually perpendicular vectors **a**1,**a**2,and**a**3 starting at that corner and defining the wedge such that **a**1 and **a**2 are the two legs of the right triangle of the wedge. **Warning:** If the wedge is to be used as a subzoning body for CAD geometry the **a**1, **a**2, and **a**3 vectors must forma right-handed system.

3 Box (BOX) – Specify the components of a radius vector **V** to one of the corners and the components of three mutually perpendicular vectors **a**1, **a**2, and **a**3 starting at that corner and defining a rectangular parallelepiped of arbitrary orientation. **Warning:** If thebox is

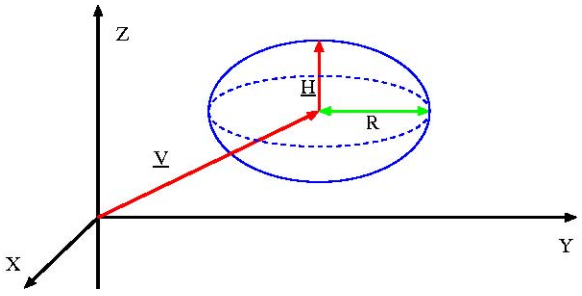


Figure 10. Ellipsoid of Revolution (ELR)

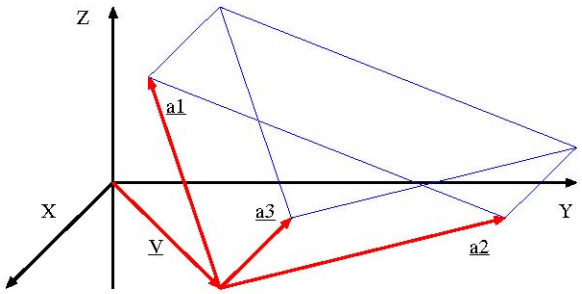


Figure 11. Right Angle Wedge (WED)

13. ACCEPTGeometry

to be used as a subzoning body for CAD geometry the **a**1, **a**2, and **a**3 vectors must form a right-handed system.

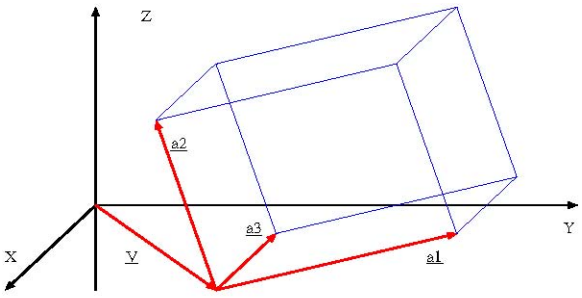


Figure 12. Box (BOX)

1 ArbitraryPolyhedron(ARB)–Specifythe componentsofk(k =4,5,6,7,or8)radius vectors, **V**1 through **V***k*, to the corners of an arbitrary non-reentrant polyhedron of up to six sides, andspecifythe indicesofthe cornersofeachfaceby meansofaseriesoffour-digit numbers between “1230” and “8765” (enter zero for the fourth index of a three-cornered face). The

digits must appear in either clockwise or counterclockwise order.
2 Torus (TOR) – The vector **V** specifies the coordinates of the centroid of the torus, the unit vector **H**specifiestheaxisofrevolution,themajor radiusRspecifiesthe distancefromthe centroidofthetorustothe centerofthe ellipsetoberotated,the radiusRH specifies the axis of the ellipse parallel to the **H**vector, and the radiusRp specifies the other axis of the ellipse. For now, only circular (RH =R*p*)tori are allowed.

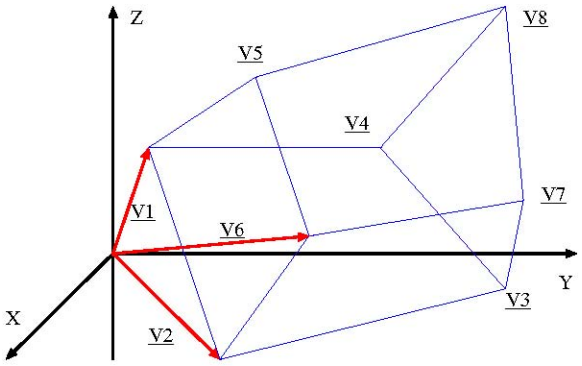


Figure 13. Arbitrary Polyhedron (ARB)

13.1.2 Body Data

The body data begin immediately after the line containing the GEOMETRY keyword. The method of describing each of the body types is discussed in the body definitions section and

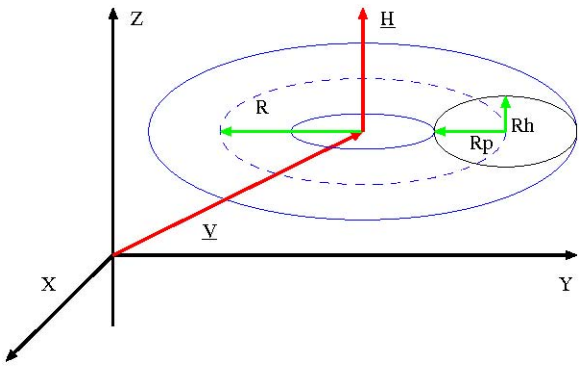


Figure 14. Torus (TOR)

illustratedinTable9. The descriptionof each newbody must begina new lineof input,andthe first parameter on that line must be the appropriate three character code for the body type. Table 9lists the additional input parameters required (no defaults) for each body type in their proper sequence. The user is free to distribute these parameters over as many lines as he pleases. Aline with the keywordEND signals that the description of all of the problem bodies is complete.

13.2 Specification of Input Zones

13.2.1 Input Zone Definition

Having defined the necessary geometrical bodies, the user must then resolve the entire problem geometry into input zones satisfying the following criteria:

1 An input zone may consist only of either a single homogeneous material or a void.
2 Every point of the problem geometry must lie within one and only one input zone.
3 The final input zone mustbea void zone surrounding therestof theproblem geometry that is entered through a non-reentrant surface; any particle entering this zone is treated as an escape particle.

Input zones are specified as appropriate combinations of the previously defined bodies. Such combinationsmaybeas simpleasjusta singlebody,ortheymay consistof complex intersections, unions and differences of various bodies. We illustrate the principles of input zone specification with the following examples where, for simplicity, we omit the escape zone. Each example involves only two zones,AandB, definedby the cross hatchingin Figure 15.

InFigure15a, zoneAconsistsofasphere,body#1,thatis tangentto zoneB, which consistsof a right circular cylinder, body #2. Input zone specification is simply

A= +1,

B=+2. That is,

inputzoneAconsistsofallspatialpointsthatliewithinbody#1, and similarly for zone

B. In Figure 15b, the sphere is inserted into a hole that has been cut in the cylinder so that

13. ACCEPTGeometry

Body Type	Real Data Defining Particular Body	
BOX	Vx Vy Vz A1x A1y A2x A2y A2z A3x A3y	A1z A3z
RPP	Xmin Xmax Ymin Ymax Zmin	Zmax
SPH	Vx Vy Vz R	
RCC	Vx Vy Vz Hx Hy R	Hz
REC	Vx Vy Vz Hx Hy R1x R1y R1z R2x R2y	Hz R2z
ELL	V1x V1y V1z V2x V2y R	V2z
TRC	Vx Vy Vz Hx Hy R1 R2	Hz
WED	Vx Vy Vz A1x A1y A2x A2y A2z A3x A3y	A1z A3z
ARB	V1x V1y V1z V2x V2y V3x V3y V3z V4x V4y V5x V5y V5z V6x V6y V7x V7y V7z V8x V8y Face Descriptions (see note below)	V2z V4z V6z V8z
TOR	Vx Vy Vz Hx Hy R RH Rp	Hz
ELR	Vx Vy Vz Hx Hy R	Hz
END	No Data	

Table 9. Data required to describe each body type

Note: The final lineof the arbitrary polyhedron input containsa four-digit number
for each of the six faces. Thirty data values are required for this body type; if there are fewer than eight corners and six faces, zero values must be entered.

A=+1,

B=+2-1. Thus, **input zoneBconsists of all spatial points that lie within body #2 AND not within body #1**. Input zoneBis specified as the difference between two bodies.

In Figure 15c, bodies #1 and #2 consist of the same homogeneous material (or void), but they are imbedded within a second right circular cylinder, body #3, of another material. The specification is

A=+1OR+2,

B=+3-1-2. Thus, **input zoneAconsists of all spatial points that lie within EITHER body #1 OR body #2**. Thisis an exampleof input zone specification asa unionof bodies.

In Figure 15d, the intersection of body #1 and body #2 consists of a single homogeneous material; therestof the space within body#3is filled with another material. The specificationis A=+1+2,

13. ACCEPTGeometry

B=+3-1OR+3-2.

Thus, **input zoneAconsistsof all spatial points that lie within body#1 AND within body#2**.

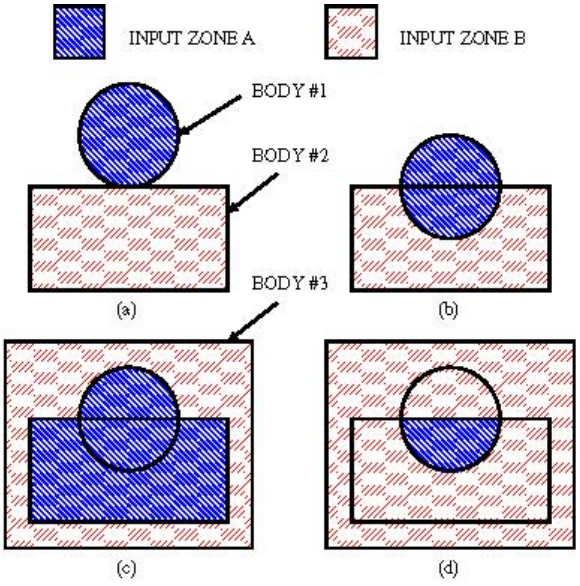


Figure 15. Illustration of various methods of combining bodies for specification of input zones

Note that:

1 TheORoperatorreferstoall followingbody numbers untilthe nextOR operatorisreached or a new input zone is initiated.
2 The AND operator is implied before every body number that is not preceded by an explicit OR operator, except that the firstOR operatorofa unionis an implied EITHER.

Though Figs. 15a and 15b are useful for demonstrating how input zones are constructed, they are not good examples of transport geometries because they are reentrant. By reentrant we mean that there are some paths by which escaping particles can reenter those geometries. They can be made non-reentrant by enclosing them completely in a non-reentrant body such as a sphere and letting the escape zone be the region outside the sphere.

13.2.2 Input Zone Data

Geometrical specification of the input zones begins immediately after the line containing the END parameter for the body data. The method of describing the input zones in terms of the input bodies is discussed in Sec. 13.2.1. Body numbers are determined by the order in which the bodies arereadin.The descriptionofeachnewinputzonemustbeginonanewlineofinput,andthefirst parameter on that line must be a character string beginning with the letter Z. Anything following thisZ,thatisnot separatedbya delimiter,isignored.Itisour conventionto followtheZwith the zone number to improve readability for the user, but the code will number zones in the order theyarereadregardlessofthe

numbering aftertheZ.TheZparameteris followedbya stringof
1
3
.
A
C
C
E
P
T
G
e
o
m
e
tr
y

parameters that specifies the input zone following the form of the right hand sides of theequations
of Sec. 13.2.1. For example, input lines describing the two input zones in Figure 15d are:

Z001 +1 +2
Z002 +3 -1 OR +3 -2

The user is free to distribute the parameters necessary for describing an input zone over as many lines as he pleases. Aline with the keywordEND signals that the description of all of the problem input zones is complete.

13.3 Subzone Specification

13.3.1 Subzone Definition

InVersion 2.0 we began implementing automatic subzoning features into the ACCEPT codes. (See the section on Automatic Subzoning for further details.) In addition to reductions in memoryrequirementsandruntime,this powerfuloption eliminatestheburdensometaskof otherwise generating an input-zone description for each individual subzone. The ACCEPT codes now feature the full three-dimensional subzoning of input zones consisting of a single body of type RCC, RPP,BOX, SPH, WED, TRC, andTOR and subzoning for some multi-body input zones. Automatic subzoning is available for CAD zones based on RPP subzoning of the CAD bounding box. All of the available subzoning schemes can be used as non-conformal subzone overlays for obtaining simple profiles within complicated CG or CAD zones. Each subzoning scheme divides the sub-zone entity into equal intervals in three different dimensions based on three integers supplied by the user.

Single Body Subzoning The simplest type of subzoning involves a zone composed of a single body. The following is a description of how the three subzoning integers are used for these types:

1 RPP – The three integers correspond to subzoning along the three

Cartesian directions, x, y, and z, respectively. Here, the body-based coordinate directions are the same as those of the laboratory system. Distances are measured along the axes from the point (Xmin, Ymin, Zmin) defined in Figure 4.

2 BOX – The three integers correspond to subzoning along the three Cartesian directions, $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, respectively, as defined in Figure 12. Distances are measured from the point defined by the radius vector **V**.

3 WED – The three integers correspond to subzoning along the three Cartesian directions, $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$,respectively, as defined in Figure 11. This subzoning is similar to the BOX, except that subzones are cut by the sloped surface of the wedge. Distances are measured from the point defined by the radius vector **V**.

4 RCC – The three integers correspond to subzoning azimuthally about the cylinder axis, in distance from the axis (radially), and in distance along the cylinder axis (axially) from the center of the base defined by the radius vector **V**in Figure 6, respectively.

5 TRC – The three integers correspond to the same subzoning definitions used for the RCC: azimuthal, radial, and axial.

6 SPH – The three integers correspond to subzoning azimuthally about the laboratory z axis as measured with respect to the positive x axis, in polar angle as measured with respect to the laboratory z axis, and in distance from the center of the sphere (radially), respectively.

13. ACCEPTGeometry

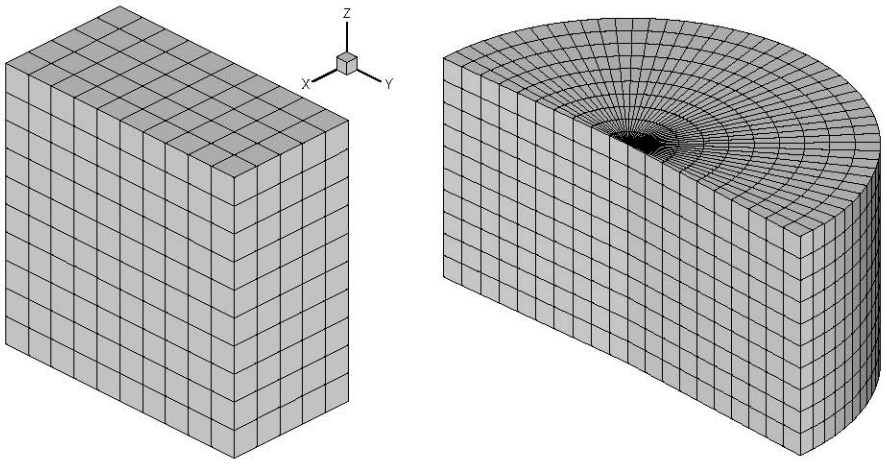
7. TOR – The three integers correspond to subzoning in the poloidal angle, the cross section radius of the torus, and the toroidal angle (angle about the axis of revolution). This is analogous to the subzoning of an RCC, where the torus is a cylinder with its axis wrapped in a circle. Thus, the RCC azimuthal angleis theTOR poloidal angle, the RCC radial coordinate is the cross section radius, and the RCC axial coordinate is the toroidal angle. Currently, subzoningisrestrictedtocircular tori.

   The poloidal angle is the angle between the cross section radius vector **r** (currently, **r** = **R**H = $\mathbf{R}_p$)and the major radius vector**R**inFigure14.The senseofrotationforthe poloidalangleis such thata radius vector **r** in the direction of the unit vector **H**isa 90-degree poloidal angle. The poloidal

subzonesarecreatedby dividingthe 360-degree angle-spacebythe numberof poloidal subzones requested.

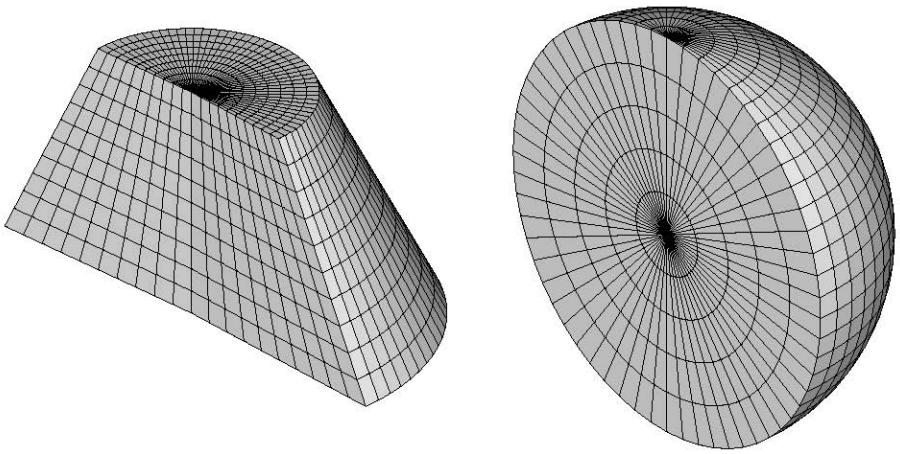
The subzoning of the cross section radius is currently limited to circular shells since the torus mustbecircular.Theradial subzonesarecreatedbydividingthecross sectionradiusequally into the number of cross section subzones requested.

The toroidal angle about the axis of revolution proceeds as the calculation of any azimuthal angle about an axis. In this case, the azimuthal angle is determined by the dot product of the perpendicular component of the radial vector **R**with the reference vector. The toroidal subzonesarecreatedby dividingthe 360-degree angle-spacebythe numberoftoroidalsubzones requested.

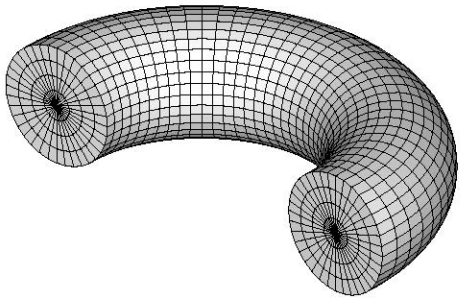


RPP subzoning RCC subzoning

13. ACCEPT Geometry



TRC subzoning SPH subzoning



TOR subzoning

Multi-Body Subzoning Therearecurrently8multi-body subzone entities availablein ITS. These subzone entities can be divided into2types: closed shells and cylindrical-like shells. The closed shellsarethe SPH-SPHandTOR-TOR.Bothoftheserequirethatthetwobodiesbe concentric.The SPH-SPH subzoningisshowninFigure16.TheTOR-TORmustbecomposedoftorithatsharean axis of rotation, as shown in Figure 17. Subzoning is based on the same coordinate systems as the single body subzoning. The number of “radial” subzones specify the number of subzone layers between the inner and outer boundary of the shell.

The other 6 multi-body subzone entities all use cylindrical-like body-based coordinates for subzoning: azimuthal, radial, and axial. The azimuthal and axial boundaries have their usual meaning. The radial subzone boundaries are equally spaced between the inner and outer zone boundaries for all axial

coordinates. This is best illustrated in the curved subzone boundaries of the SPH-RCC. For the SPH-RCC, the center of the sphere must lie on the axis of the RCC. In all other cases the bodies defining the inner and outer radii must be coaxial with each other. For the TRC-TRC, the wide parts of the frusta need not be on the same side.

13. ACCEPTGeometry

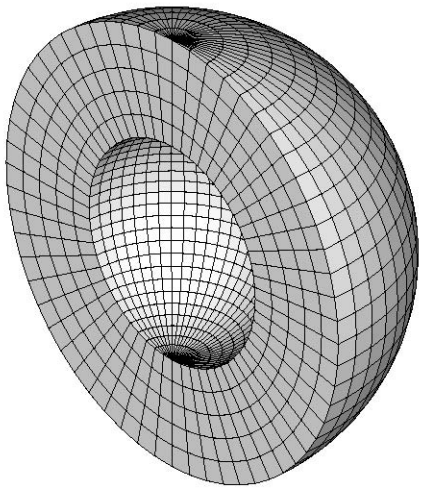


Figure 16. SPH-SPH subzoning

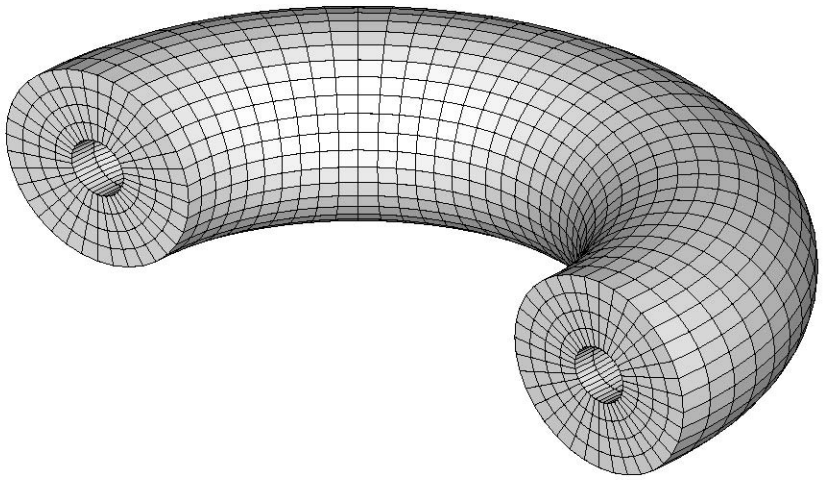
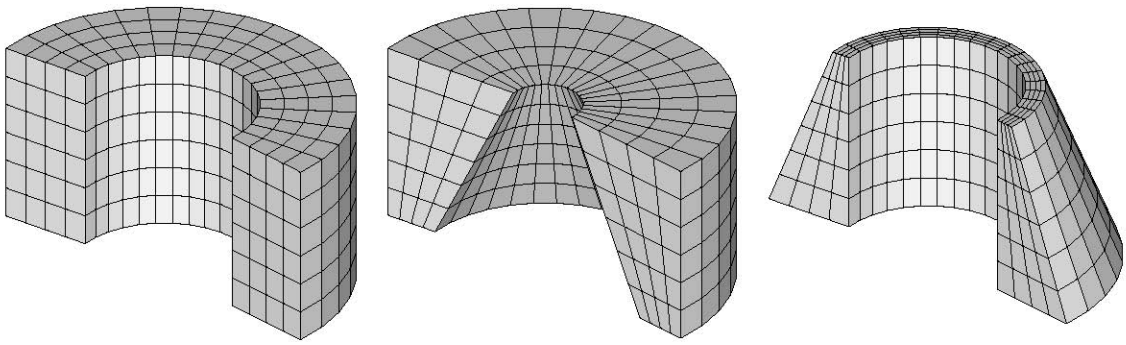
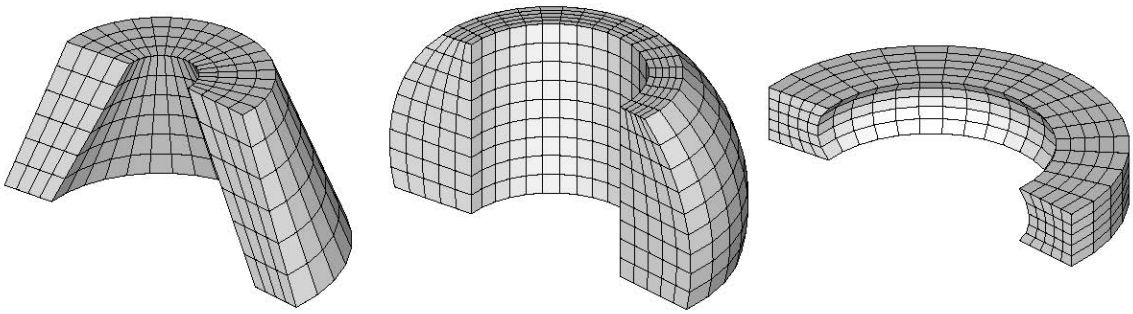


Figure 17. TOR-TOR subzoning

13. ACCEPTGeometry



RCC-RCC subzoning RCC-TRC subzoning TRC-RCC subzoning



TRC-TRC subzoning SPH-RCC subzoning RCC-TOR subzoning

Mostof these zonesaredefined entirelyasthe subtractionof onebodyfrom another. However, twoofthese zonesrequireathirdbodyto completethe definitionoftheinput zone.The definition ofthe SPH-RCCrequiresthe unionwith anotherRCC,butthe dimensionsoftheRCCare dictated by the SPH and RCC desired. The second RCC must provide an extension of the two planes perpendiculartotheaxisofthe subtractedRCCandmustbelargeenoughin radiusto encompass all of the subzoned region. The planes of the RCCs need not be symmetric about the center of the sphere. The RCC-TOR will typicallyrequire the subtractionof another cylinder with radius equal to the torus radius of revolution and other parameters identical to the first RCC. (This is done to eliminate the hole in the center of the torus.) If employing one of these entities as an explicit subzone overlay, then only the two bodies need to be specified.

Automatic CAD Subzoning Amethod for subzoning CAD zones is provided. The user specifies the zone number that should be subzoned and the desired number of divisions in the x-, y-, and z-coordinates. ITS uses the bounding box of the CAD zone to create an RPP. This RPP serves as a subzone overlay for tallying purposes.

Whilethisisa simpleand automated subzoning technique,theresulting overlaymaynotbe an efficient or reasonable subzoning scheme for the CAD zone. In some cases, a user may be able to manually specify a more efficient (nearly conformal) subzoning scheme for the CAD zone.

e
o
m
e
tr
y

Subzone Overlays Subzone overlays (alternativelyreferredtoas non-conformal subzoning) can be specified either implicitly or explicitly. The alternative is conformal subzoning. Conformal subzoning requires that the zone be defined by the same bodies as the subzoning scheme (with two exceptions stated in the description of multi-body subzoning that require an additional body to complete the zone description). Non-conformal subzoning requires that the zone be contained entirelywithinthe subzoneoverlay.Ifthisisnotthecase,errorscanresultdueto invalid subzone indexes calculated during the transport process.

Implicit subzoneoverlayscanbeusedby incorporatingthe subzoneentityasthefirstbody(or bodies) in the zone description. In this case, the overlay can be extracted from the zone description.

Explicit subzone overlays use different combinatorial descriptions for the input zone and the subzoning entity, and therefore use different geometrical descriptions for the transport process and for the subzone tallying process. In these terms, CAD subzoning is always explicit, but can be automatedby having ITS convert the CAD bounding box into an RPP overlay.

13.3.2 Subzone Data

The automatic subzoning capability is invoked in the following way. In the subzoning section of the GEOMETRYdata, the keyword **SUBZONE** must appear followed by a parameter that specifies the number of the zone to be subzoned. If the CG zone description begins with the body number(s) that define the subzoning scheme (as either a conformal subzoning or an implicit non-conformal subzone overlay) or if one wishes to automatically subzone a CAD zone based on the RPP of the bounding box, then nothing else needs to appear on this line. To impose an explicit subzone overlay upon either a CG zone or a CAD zone, the keyword **OVERLAY** must be followedbyacombinatorial descriptionofthe overlay scheme.Only1or2bodiesareneededforthe OVERLAYdescription. The multi-body subzoning descriptions should consist of one body minus another body.

The line immediately following must contain three integers that define the number of equal-increment subzones into which the zone is to be divided along the three coordinate directions. The three orthogonal directions corresponding to these three integers are defined in Sec. 13.3.1.

Additional information is required for multi-body subzoning and for azimuthal subzoning. For multi-body subzoning, there is a set of tertiary keywords defining various input zones to be subzoned that consist of the difference between two bodies. In this case, one of the following keywords must follow the three subzoning integers: **RCC-RCC**, **RCC-TRC**, **TRC-TRC**,

TRC-RCC, **SPH-SPH**, **TOR-TOR**, **SPH-RCC**, and **RCC-TOR**.

To perform azimuthal subzoning on any cylindrical, conical, spherical, or toroidal subzone entity, a reference direction is required to define the zero azimuth. The reference direction for a SPH or SPH-SPH is defined as the positive x-axis. A reference direction may be supplied for other subzoning entities (RCC, TRC,TOR, RCC-RCC, RCC-TRC, TRC-TRC, TRC-RCC,TOR-TOR, SPH-RCC,and RCC-TOR)byreferenceto anotherbody.To specifysuchareferencedirection,the input line with the three subzoning integers should contain the keyword **AZ**, followed by any body numberexceptanRPP.Thezeroreference vectoristhe componentoftheVvectorofthe **AZ** body that is perpendicular to theHvector of the subzoned body. If no **AZ** body is specified, the +i vector will be used as the body vector. If the definition of the reference vector is null, the code attempts to use the +i, +j, and +k vector for the zero azimuth vector, in that order.

Asanexample,toimposeanexplicit non-conformal subzoneoverlayoninputzoneAinFigure 15c (zone number 1) based on the surrounding RCC body (body number 3), one would specify:

```
SUBZONE1OVERLAY+3 110 10
```

13. ACCEPT Geometry

Aline with the keywordEND signals that the description of the problem subzoning is complete.

13.4 Volume Specification

13.4.1 Volume Definition

The specification of volumes is optional. The code will attempt to normalize flux results by volume.The DEPOSITION-UNITSkeywordalso allowsthe usertorequestthat energyandcharge depositionresultsbe normalizedbyVOLUMEorbyMASS(densitytimesvolume).Thecodewill default to using a volume of 1.0. The user should consider whether accurate volumes are needed for the normalization of the results that are of interest.

The volumes of the problem subzones may be specified through one of several automated or manual methods that are available. If automatic subzoning has not been requested for any input zone, the input zones are the same as the problem subzones. The volume of the escape zone is never specified. If the user wishes to supply the volumes for a run in which he has requested subzoning, he must ensure that the volumes are specified in the proper sequence.

Ageneral scheme for the precise calculation of the volumes of zones defined by the combinatorial method is not possible. The user may select an option via [parameter(1)] associated with the GEOMETRYkeyword. The default value of0 will cause the code to set all volumes to 1.0 cm^3. Avalueof1allowsthe usertoreadinthe volumesas described below.Ifthe geometryissuch that a satisfactory method exists for calculating the volumes internally, the user may set [parameter(1)]equalto2anduseacode modificationto insertthe necessarylogicattheproperplacein Subroutine VOLACC.

A value of3 triggers the code to automatically calculate subzone volumes. The volumes of input zones that are not subzoned are set to 1.0 cm^3.Volumes can be obtained for some zones. In CAD, the CAD-VOLUMES keywordwill make the code calculate the volumes of all zones. In CG, ageneral volume calculation algorithm is not available. The volumes of zones for which conformal subzoningis available canbe calculatedbyrequesting only1subzone(1 intervalin eachof the 3 dimensions). However, the user needs to be aware that these volumes are only available for the simple single body or multi-body combinations for which subzoning is available. Attempts to calculate volumes for more complicated zones may result in non-conformal subzoning (see the discussion of Automatic Subzoning for more detail) and inaccurate volumes.

Avalueof4for the first parameter on the GEOMETRYkeywordhas the sameeffect as option 3, except that the code attempts to determine whether subzones are inside or outside of a zone (and adjust volumes accordingly) and the user may then overwrite subzone volumes for selected zones as described below. The method for determining whether subzones are outsideofa zoneis based on simply checking the mid-point of subzone against the zone definition. This method will be adequate if subzone boundaries conform to the boundaries of the zone in some manner. For subzone schemes that do not conform to zone boundaries, the user must supply the volumes for the intersection of the subzones with the zone.

If the value of [parameter(1)] is negative, the logic for setting subzone volumes will proceed based on the absolute value of the parameter, but printing of the volumes to the output file will be suppressed.

13.4.2 Volume Data

If [parameter(1)] associated with the GEOMETRYkeywordis equal to 1, the array containing the volume data for the problem subzones is inserted immediately after the line containing the END parameter for the data specifying the subzoning schemes. The input zones are numbered according to the order in which they are read. If an input zone is not to be subzoned, then it becomes a single subzone. Therefore, if automatic subzoning has not been requested for any input zone, the subzones are identical with the input zones (except that the escape zone is not included among the subzones). If an input zone is to be subzoned, the subzones are numbered by incrementing the body-based coordinates in an order corresponding to the inverse of the order of the three integers specifying the subzoning (see previous subsection). For example, if an input zone consisting of an RPP is to be subzoned, the subzones are generated by first incrementing thez coordinate, then they, and finally the x. The volume array must contain an entry for each problem subzone (no defaults), excluding only the escape zone.

If [parameter(1)] associated with the GEOMETRY keyword is equal to 4, then the code first calculatesall subzone volumes analytically.Forany subzonewitha midpoint outsideofthe zone, the volume is set to zero. These volumes can be overridden for subzones of user-specified zones. These appear immediatelyafter the line containing the END parameter which terminates the input zone descriptions. The first line should give the index of the desired zone. The following lines should include the subzone index (a dummy value that is ignored) and the subzone volume (in cm^3). Any additional parameters will be ignored. There should be one line for each subzone, even for subzones that lie entirely outside of the original zone (that have zero volume). This format is repeated for each zone. The user can specify as many zones (that are subzoned, of course) as desired. This sectionis terminated witha line containing the keyword“END”.

If [parameter(1)]isnotequalto1 or4, these volumeinputdataareomitted,andno “END” statement is required.

13.5 Material Specification

13.5.1 Material Definition

Amaterial index is assigned to each input zone. Azero index defines a void zone; otherwise, the material indices are defined by the order in which the

materials are specified in executing the cross-section generating code. The method of inputting the material indices is described in the Geometry Syntax section.

13.5.2 Material Data

Materials are referred to only by index number. This index corresponds to the order the material cross sections were generated by the XGEN or CEPXS, which is the same as the order of the material requests in the input deck for those codes.

13.6 Conventions for Escaping Particles

Any particle entering the escape zone is assumed to have escaped. In CG, the user is responsible for constructing a geometry such that this is true. In CAD, the escape zone is automatically determined.

In forwardmode, the ESCAPE-SURFACES keywordmay be used to indicate the surfaces on which escape particle tallies are desired. In adjoint mode, the SOURCE-SURFACES keywordmay

be used to indicate surfaces on which escape adjuncton particles are to be tallied as sources contributing to the response. The default in either mode is to perform total escape tally (equivalent to a single surface surrounding the geometry) for all particles entering the escape zone. If using thesekeywords,theuserisresponsibleforensuringthatthe surfaceslistedunderthesekeywords area completedescriptionof escape surfacesin forwardmodeandarean accuraterepresentation of the surface-source quantity desired in adjoint mode.

13.7 Geometry Syntax

Syntax: GEOMETRY[parameter(1-2)] [SUBZONE-ONLY] [VOID] [CAD-VOLUMES]

(Input Body Descriptions)

END

(Input Zone Descriptions)

END

(Subzoning Descriptions)

END

[parameter(3)]

Example: GEOMETRY01

RCC
0.0 0.0
0.0 0.0
0.0 1.0
0.25

RCC
0.0 0.0
0.0 0.0
0.0 1.0
0.5

S
P
H

0
.
0

0
.
0

0
.
5

2
.
0

S

P
H

0
.
0

0
.
0

0
.
5

3
.
0

E
N
D

Z
1

+
1

Z
2

+
2

–
1

Z
3

+
3

–
2

Z
4

+
4

–
3

Z
5

–
4

E
N
D

SUBZONE1

110 10

SUBZONE2

11010 RCC-RCCAZ4

13. ACCEPTGeometry

END

3

1

2

1

0

Default: no default This keywordsignals the beginning of geometry input for the ACCEPT codes. The absolute value of [parameter(1)] determines the option for setting the subzone volumes in cm^3:

(a)0(default) causesall volumestobeset internallyto1.0.

(b)1 causesthecodetoread volumesfromtheinputstream.

(c) 2 requires that the user provide the necessary logic for computing the volumes at the appropriate place in subroutine VOLACC.

(d)3 causes volumesofall subzonestobe calculated, whilethe volumesofinput zonesthat are not subzoned are set to 1.0 cm^3.

(f)4hasthe sameeffectas3,but subzone volumesfor user specified zonesmaybe overwritten.

If [parameter(1)] is negative, printing of volumes to the output file is suppressed. Trackingdebugis turnedofforon accordingto whether [parameter(2)]issetequalto0(default) or not,respectively. **WARNING:**The tracking debug feature may produce substantial amountsof information.Werecommendthatasinglehistorybe simulatedto assessthesize of the resulting output file produced. The **SUBZONE-ONLY** sub-keyword causes ACCEPT to process input through the geome try, write a FINITE-ELEMENT-FORMATfile, and then quit. No further processing of input is performed and no particle transport is performed. This feature is used to produce a refined subzone structure without energy or charge deposition results. The resulting file can be used when mapping the results of a previous calculation performed on a coarse subzone structure onto this refined subzone structure. There is a tool to perform this mapping called MAPPER. **The SUBZONE-ONLYfeature is an exception to the rule that keywords are order independent.** When the SUBZONE-ONLYkeywordis used, no input will be read that appears beyond the data of the GEOMETRYkeyword.

The **VOID** sub-keyword causes all zone materials assignments to be set to zero. The input for zone materials is still required; however, allmaterial assignments will be disregarded. This feature is useful if performing a stochastic volume calculation.

The **CAD-VOLUMES** sub-keyword causes the code to calculate volumes of all CAD zones. Warnings will be included in the output for any part for which the volume cannot be calculated.

What follows the keyword line is the list of primitive bodies used to construct the combinatorial geometry followed by an END line, the list of input zones constructed from the
1
3
.
A
C
C
E
P
T
G
e
o
m
e
tr
y

primitive bodies followedbyanENDline,thelistof subzoning specifications followedbyan
ENDline,andpossiblyalistofzonevolumesdependinguponthesettingof
[parameter(1)]. Immediately after this information, there must follow a series of lines, one for each input zone, with each containing [parameter(3)]. These parameters specify the material for each input zone.
1
4
.

C
A
D

G
e
o
m
e
t
r
y

Last Modified Date: 2008/04/17 22:45:35

14 CAD Geometry

This document discusses the special requirements of the GEOMETRY keyword and special considerations when constructing or adapting a CAD

model to be used in ITS. The Cholla library providesa facet geometry capability with ITS for interrogating CAD models. Links are also available for coupling ITS with the ACIS modeler. Spatial Corporation markets the 3D ACIS Modeler as a commercially available product [34]. This version of ITS supports linking with the 3D ACIS Modeler versions R12, R15, and R16, which must be purchased directly from Spatial Corporation.

14.1 GEOMETRYKeyword

CAD geometries may have differentrequirements than combinatorial geometry (CG) depending on the MODE selected in the prmfile. The alternative modes are CAD ONLY, HYBRID, MIRROR CG, and CG ONLY. For the MIRROR CG, and CG ONLYmodes, the GEOMETRYkeyword requirements are the same as the ACCEPT requirements when the code is not compiled with the CAD option. However, the GEOMETRYkeywordhas different requirements for the CAD ONLY and HYBRID modes.

There are several terms that must be defined to describe CAD ONLYand HYBRID geometry requirements:

The "**escape zone**"inCAD geometryisnot strictly defined.Aparticle leavingaCAD geometry zone and not directed toward another CAD zone is in the escape zone. In HYBRID mode, the escape zone must be defined in CG and must surround all CAD zones.

The "**undefined void**"inCAD geometryisanyregionthatisnot otherwise definedintheCAD description. This is the void space between CAD zones, which is not explicitly defined.

The "**defined void**" in HYBRID mode is the CG equivalent of the CAD undefined void. It is the second-to-last zone specified and shoulddefine all CG space that is not otherwise defined. (It is possible to have other CG and/or CAD zones that are explicitly defined as being voids.) All of the space defined by CAD zones must be enclosed in this defined void.

14.1.1 CAD ONLY

In the CAD ONLYmode, combinatorial geometry features can be used to describe the source distribution. That is, the POSITION of the source (or in adjoint mode, the LOCATION of the DETECTOR-RESPONSE)mayreferenceabodyinthefirstsectionofthe GEOMETRYinput.Body information can also be used to specify subzone overlays. Other body and zone information may be present in the input file, but all CG zone definitions will be disregarded.

The material assignments section of the GEOMETRYkeyword must be used to assign materialstoCAD zones. There mustbea material assignedto eachCAD zoneplus2additional void assignments at the end of the list for the undefined void and the escape zone.

Sourcesmay existinthe escape zone. However,ifa source particle initiatedinthe escape zone does not intersect the problem, a warning will be printed to the output file stating that the source particle wasrejected.Asimple methodfor avoiding numerous warning messages(duetoasource with an

angular distribution which only causes some particles to intersect the geometry) is to use the HYBRID feature to specify only a defined void and an escape zone such that the source is not in the escape zone.

14. CAD Geometry

14.1.2 HYBRID

Both CG and CAD geometry descriptions are used. The last two zones in the CG description must be a defined void and an escape zone. The defined void corresponds to CAD's undefined void and must be specified as the escape zone minus all other space defined in CG. The escape zone must have boundaries such that a particle entering cannot possibly reenter the problem geometry.

If a particle is known to be in a CG zone, the CAD geometry will not be interrogated, and vice versa. Therefore, overlapping geometry descriptions may yield inaccurate results. Specification of non-overlapping geometriesisa userresponsibility;no checkis performedintheITScode,and no errors will be generated.

Material indices in the GEOMETRYkeywordshould first assign materials to each of the CAD zones, then assign materials to each of the CG zones. Since a defined void and escape zone are includedintheCG description,no additional material assignmentsarerequired.

14.1.3 MIRROR CG

This mode is intended for development purposes and is not recommended to users.

The CAD and CG models should describe exactly the same geometry. Any discrepancies in particle distance-to-boundary and particle zone location inquiries during transport will be printed to the output file. The transport will proceed using the CG results in the event of a discrepancy.

14.1.4 CG ONLY

Code compiled with the CAD option will execute as if the CAD option had not been used, except that the input and output files will be opened with names specified in the prmfile.

14.2 Conventions for Escaping Particles

Any particle entering the escape zone is assumed to have escaped. In HYBRID and CG ONLY modes,the userisresponsiblefor definingaCGescape zonesuchthatthisistrue.InCAD ONLY mode, the escape zone is automatically defined.

The ESCAPE-SURFACES and SOURCE-SURFACES keywords are not functional for CAD geometry. All particles escaping the geometry are tallied intoa single surface.

14.3 CAD Models

CAD models must be “healed” to be used in ITS. Gaps between CAD surfaces will result in lost particlesinthe transportprocess. Each timea particleis losta diagnosticis writtentothe output file. Large numbers of lost particles can result in inaccurate results and may indicate an inadequately healed geometry.

14.4 Geometry Syntax

The format of the geometry keywordfor CAD is the same as for the ACCEPT codes. However, there are different requirements regarding what information must be present and differences in how the information is used.

BodiesmustbespecifiedifthecodeisnotrunninginCAD ONLYmode.InCAD ONLYmode, CG bodies may be used to specify the spatial distribution of a source.

1
4
.

C
A
D

G
e
o
m
e
t
r
y

Zones are required if the code is not running in CAD ONLY mode. If zones are present in CAD ONLYmode, they are ignored. For mirroring, the zones

must be identical to the CAD description in both geometry and zone numbering.

Although bodies and zones are optional in CAD ONLY mode, the END statements must be included even if the sections are otherwise empty.

In HYBRID mode, CG zones may be superimposed on the CAD geometry. The escape zone must be specified as the last zone. The escape zone should surround the CAD and CG geometry. A“defined” void (corresponding to a CAD “undefined” void) must be specified as the second to last zone. This must encompass all space definedby CAD geometry.

Materials must be specified for zones in all modes. In CAD ONLYmode, materials should be assigned to all CAD zones, and two additional void assignments should be made for the undefined void and the escape zone. For HYBRID calculations, material assignments should be made for all CAD zones followed by material assignments for all CG zones. Since the undefined void and escape zone are part of the CG description, no extra assignments are required.

15. Suggestions for Efficient Operation

Last Modified Date: 2008/04/17 22:45:47

15 Suggestions for Efficient Operation

The generaloperational limitationsonthe member codesofITSare definedbythe scopeofthe keyword input. An attempt has been made to implement automatic memory allocation for the arrays in ITS that are most likely to change from one calculation to another. Some arrays still have hard-wired dimensions. If while processing the keyword input, an array dimension required by a particular problem exceeds the default allocations, the execution aborts with a message to that effect being written to the output file. These dimensions can be found and modified in the its/Code/Modules/params M.F file.

WARNING: Care must be taken in reducing a hard-wired allocation to zero since this may result in the upper bounds of the dimensions of certain arrays being set to zero; this will result in a fatal error since the lower bounds of the dimensions of most arrays is one.

Perhaps more important is the fact that the choice of certain input parameters can markedly affect the efficiency of the calculation; that is, the user’s ability to obtain statistically meaningful output in a reasonable amount of time:

1 Obviously, the number of histories should be kept as small as possible. All member codes provide the user with estimates of the statistical uncertainties of the output data. Assuming that these uncertainties vary like the square root of the number of histories, these estimates thenserveasaguidetotheultimatechoiceofthenumberof

histories.Theusermustdecide what level of statistical accuracy is acceptable for his or her particular application.

2 To achieve good parallel efficiency, the number of BATCHES should be an integer multiple of the number of processors. The workload is distributed in batches. Having fewer batches than processors will result in some processors remaining idle. The first parameter of the TASKS keyword can be used to specify the number of processors desired or the code can determine the number of processors available.

3 In our experience dynamic load balancing, DYNAMIC-MPI, is only preferable over static load balancinginthree situations:(1)the masterprocessis locatedonaprocessorthat should notbe performingalarge amountof computationbutispreferableforI/O operations,(2)the processors on the system are expected to perform at different speeds so the calculation must beload balanced dynamically,or(3)thecomputersystemisvery unstablesuchthatdynamic error handling is more efficient in making progress toward completing the calculation. In the first case, the user should consider using a negative value for the first parameter of the TASKS keyword.

4 ThenumberofBATCHESshouldbeatleast20andshouldnotbe excessive.Foracalculation runningonhundredsofprocessorsformorethanaday,itisnot excessivetohave thousands of batches. However, there is communication, processing, and output overhead associated with beginning and ending batches and having too many can affect the efficiency.

5 Electroncutoffs shouldbeaslargeas possible.For example,ifthe sourceis monoenergetic,a global electroncutoffequalto5or10percentofthe source energymaybe adequate (depending upon the quantity of interest). Because the logarithmic energy grid used in the electron

15. Suggestions for Efficient Operation

transport technique becomes much finer at low energies, following electron histories down to low energies becomes very time consuming. On the other hand, running time is not very sensitive to the value of the photon cutoff energy because low energy photons have a high probability of being absorbed after only a few interactions.

6. Similarly, electron trapping energies should be as large as possible. For example, consider the simulation of photoemission by low-energy photon sources. Because accurate simulation of boundary crossings is important, electron cutoffs must be low. On the other hand, if bremsstrahlung production is not important, as is likely in this case, electron trapping energies may be as high as the maximum source energy.

7.Therequestedenergy,angle,andspatialresolutionsshouldbenohigherthan necessary.Demanding excessiveresolutiononlymakesitmoredifficult – i.e.,costly–toobtain statistically meaningful output.

8. Processing large sets of tally data can affect the efficiency of the code.

This can impact the efficiency of the calculation by stressing the memory limits of the system, increasing time for processing results, increasing the time for parallel communications, or increasing the time for writing output. Users should consider whether all of the requested tallies are desired. For problems with large numbers of zones or subzones or with highly differential tallies, users should consider using the following keywords to manage I/O and memory usage: NO-INTERMEDIATE-OUTPUT, NO-DEPOSITION-OUTPUT, NO-SZDEPOSITIONOUTPUT,NO-DEPOSITION, NO-DETAILED-DEPOSITION, LINE-TALLY-WITH-CONTINUUM, or ANNIHILATION-LINE-TALLY-WITH-CONTINUUM.

Finally, the judicious use of a number of other variance reduction options can markedly increase the efficiency of certain calculations. Specific examples of these are discussed under the keyword BIASING. Some of these are discussed in even more detail in the section on Biasing. Users are warned, however, that the reckless and indiscriminate use of biasing procedures can leadtomisleadingresults.Anyuseofbiasingschemesshouldbecarefully consideredandscrutinized.

1
6
.

O
u
t
p
u
t

F
i
l
e
s

Last Modified Date: 2008/04/17 22:45:42

16 Output Files

This section describes in detail the information present in output files. The output file can be divided into3types of information: pre-processing information (anything preliminary to the Monte Carlo calculation), processing information (generated while the Monte Carlo calculation is performed), and results. There may not be any information generated while the Monte Carlo calculation is being performed. The output file is organized into sections generally designated by new carriage control pages (“1” in the first column). Each

section of the output file is presented separately here. The meaning of each statement or value (and its units) that may appear in the output file is discussed. Many sections discussed here may not appear in a given output file, as the output depends upon both the code options and output options requested.

16.1 Pre-Processing Information

16.1.1 Output Header

Next, the title “Program ITS” is given, followed by the version number and date of the code release. The authors of the software are listed, and author contact information is provided.

The next section states the preprocessor directives used to obtain the executable with which the calculation was performed.

Anumerics interrogation section is included. This is mostly for testing purposes, to determine whethertestfailuresmaybeduetodifferent numericalrepresentationsonanewplatformorusing a new compiler.

16.1.2 CAD Parameters *(CA*D *Only*)

If the CAD preprocessor directive is used, the flow logic integer requested for the calculation is stated with a table for determining the meaning of the flow logic integer. Also, the number of CAD zones that have beenreadis stated.

The prmfile is echoed.

16.1.3 Reading Input

Unless the “ECHO 0” keywordis used, input is echoed to the output as it is read. This providesthe userwith informationaboutwhenandwhyanerror occursduringthereadingofinput. However, for large input files, it may be desirable not to echo the input to the output file, so as to minimize the size of the output file. The entire input file is included in a job file, so this section may be redundant.

Minimal processing is performed while the input is being read, but some processing information may be reported in this section as it is performed. An example of this is (for **ACCEPT**) information on subzone volumes and how they were obtained. For **CAD** calculations, negative volumes indicate that the volume of the subzone intersected with the CAD zone and the volume of the CG subzone were in agreement. This means that the subzone lies entirely within the zone, and therefore CG-based electron trapping logic may be activated in that subzone. The negative values are merely informative, and absolute values are used for all result normalizations.

u
t
p
u
t

F
i
l
e
s

Error and warning messages may be included in this section. These messages are generally preceded by ">>>>>" and include the words "ERROR" or "WARNING".

16.1.4 Reading Cross Section Data

For ITS (not **MITS**), this section includes:

The title of the XGEN calculation that produced the cross section file.

The number of cross section sets available in the cross section file is printed. This is the number of materials for which data was generated using XGEN.

- For each cross section set in the file the following parameters (used by XGEN to produce the cross sections) are stated:
  - The density of the material in g/cm^3.
  - The "detour" of the material. This is the ratio of the practical range to continuous-slowing-down-approximation range for an electron at the maximum energy of the set.
  - The "I(BL)" factor is Seltzer's empirical modification to the Blunck-Leisegang formulation for sampling from a truncated collisional energy-loss straggling distribution for electrons.
  - For each element in the material:
    - Z is the atomic number of the element.
    - A is the mean atomic weight of the element.
    - W is the weight fraction of the element in the material.
  - ITRM is the level of data contained in the cross section file. If less than 5, ITS cannot be run.
  - ISGN is the cross section model: 1=Mott Electron, 2=Mott Positron, 3=Screened Rutherford Electron, 4=Screened Rutherford Positron. This should always be 1, unless the XGEN code has been modified.
  - ISUB is the number of electron substeps taken per step in the condensed history algorithm. If the number of substeps per step is allowed to vary across the energy grid, then ISUB applies only to the first energy span. The XGEN default is for ISUB to remain constant across the energy grid, but see INDEX/JSUB under the DATAPREP DATA for more information. The XGEN keyword SUBSTEP can be used to control the ISUB value.

– INAL is the option used in XGEN for calculating eta in the Mott elastic cross section.

– ICYC is the option used in XGEN for generating the electron energy grid. ICYC=1 means that a logarithmic scheme has been used (see the next parameter), and this should always be the case unless the XGEN code has been modified.

– NCYC is the parameter determining the spacing of the electron energy grid. Successive energies are related by $E_{i+1} = 2^{-1/NCYC} E_i$. The XGEN keywordSTEP can be used to control the NCYC value.

– NMAX is the number of electron energy grid values. The XGEN keywordELECTRON-GRID-LENGTH can be used to control the NMAX value.

– EMAX is the maximum energy in MeV of the electron energy grid. The XGEN keyword ENERGY can be used to control the EMAX value.

16. Output Files

– EMIN is the minimum energy in MeV of the electron energy grid.

– RMAX is the maximum electron range in g/cm^2 in the material.

– MMAX is the number of angular bins in the multiple scattering distribution. The XGEN keywordELECTRON-ANGLE-BINS can be used to control the MMAX value, as well as the distribution of angular bins.

– The value of INDEX is stated. This is a three digit number. If the first digit IDST is 1, then the cumulative multiple scattering distributions are to be read into ITS from the cross section set. If the second digit IAVE is 2, then the average cosines for the multiple scattering distributions are to be read. If IAVE equals 1, then ITS cannot be run. If the thirddigit JSUB is 2, then the number of substeps per step for the condensed history algorithm is allowed to vary across the energy grid, in which case the number of substeps is read from the cross section set.

Alist of the data sets read from the cross section file for each material.

•

For **MITS**, this section includes:

The first line of the title of the CEPXS calculation.

- The number of materials for which data is available in the cross section set. This is the
- number of unique material compositions for which data was generated using CEPXS.

The number of unique materials (both composition and density) labeled as material-densities. This is greater than or equal to the number of unique material compositions. (Some data is identical for materials with the same composition but different densities, and less memory is required by taking advantage of this.)

The number of energy groups for all species.

The particle species coupling scheme used to generate the cross sections.

If electrons are included in the cross sections, the electron energy group structure in MeV.

If photons are included in the cross sections, the photon energy group structure in MeV, and information about fluorescence lines, if any are included in the photon cross sections. Fluorescence information includes the element and shell-transition generating the line.

For each material:

- The density of the material in g/cm^3.
- The "detour" of the material. This is the ratio of the practical range to continuous-slowing-down-approximation range for an electron at the maximum energy of the set.
- For each element in the material:
  - Zis the atomic number of the element.
  - Ais the mean atomic weight of the element.
  - Wis the weight fraction of the element in the material.

16. Output File

e
s

The scheme used to generate Fokker-Planck scattering-angles. The scattering angles may be energy-and material-dependent to mimic ITS substeps, or they may be constant. Either way, the maximum scattering angle will be written to the output (as the cosine of the angle).

The treatment of positrons. Positrons may be tracked, not tracked, or treated as electrons. In the last two cases, annihilation will occur at the pair interaction site.

If photons are included in the cross sections, the group into which annihilation radiation is produced, and whether thatgroupisa linegroup.

16.1.5 Processing Input

This section will only contain information if errors or warnings are generated while processing input and cross section data. Checks are performed for inconsistencies and other errors in input keywords,cross section data, energy ranges, etc. These messagesare generallyprecededby “>>>>>” and include the words “ERROR” or “WARNING”.Awarning is generally generated instead of an error if an inconsistency has been resolved by the code in such a way as to allow the calculation to proceed. The user should always check for such warnings to determineifthe calculation performed was the desired calculation.

16.1.6 Storage Requirements vs. Allocations

This section compares the size of array dimensions required with the allocation for those arrays. The user may be able to decrease some array allocations to cope with memory constraints. Generally, an error will have been generated before this point if an array has insufficient allocation. Hard-wired array dimension parameters are specified in its/Code/Modules/params M.F and may be over-ridden in the calc params routine.

16.1.7 Geometry-Dependent Input

Geometry-dependent biasing settings and material assignments are listed for each zone.

For **TIGER** and **CYLTRAN**, the spatial extent of each zone is then listed.

Since this section can be large if the geometry consists of many zones, the NO-GEOMETRYTABLE keywordcan be used to suppress this section of the output.

16.1.8 Source Information

The energy, spatial, and directional distribution of the source is stated. In adjoint, this is the distribution of the adjuncton source (as specified by the DETECTOR-RESPONSE).

16.1.9 Output Options

The number of batches and histories per batch are stated.

Optional output requests are stated here. For each differential quantity requested, the binning structure is stated. For example, if ELECTRON-FLUX is requested, then the energy, polar-angle, and azimuthal-angle binning structure for the electron flux is stated. Keywords that will trigger binning information to be printed here are: ELECTRON-or PHOTON-in combination with -SURFACE-SOURCE, -VOLUME-SOURCE, -ESCAPE, or -FLUX, ELECTRON-EMISSION, and PULSE-HEIGHT.

16. Output Files

16.1.10 Physical Options

The options used for modeling physics are listed. This includes physics for which input keywords provide switches or cross section scaling, and physics that may or may not be included, such as fluorescence lines and positron annihilation.

16.2 Monte Carlo Output

16.2.1 Parallel Processing *(MPI Only)*

The following information is printed before the Monte Carlo calculation begins:

Whether the load distribution is static or dynamic. This is directly related to use of the **DYNAMIC-MPI** keyword.

Number of processes. This is the number of processors determined to be available.

Master option. If equal to 1, then the master process performs Monte Carlo batch calculations. If equal to 0, then the master does not perform Monte Carlo calculations. The latter is always the case for dynamic load balancing.

Numberof tasks(requested). Thisis the numberofprocessors that the userrequested.

Number of tasks (adjusted). If the number of processors requested was greater than the number of processors available, then the number actually used is stated.

Intermediate print. This is the frequency (in number of batches) with which the intermediate output will be written.

Allowed time factor. This number (if greater than zero) is the factor by which batch times are allowed to deviate from the average batch time. That is, if a batch requires more time than the average batch time multiplied by this factor and all other processes are completed, then the run is terminated.

The following information is printed during the Monte Carlo calculation:

For static load balancing calculations at the start of each cycle, the number of tasks performing Monte Carlo calculations and the initial random number seed.

The subtask number and the random number seed assigned to initiate a batch. This is printed for the master task (subtask 0) and for batches corresponding to the intermediate output print frequency.

The task number, the number of random numbers used, and the batch time. This is printed when the master finishesa batch or whena batch, corresponding to the intermediate output print frequency, returns its results to the master.

16.2.2 Monte Carlo Errors

Errors that occur during the Monte Carlo calculation will appear before the startof theresults. This information does not have a section header. The information will appear to be at the end of the physical options section, or for **MPI** it will appear to be part of the parallel processing section. These messages generally are preceded by “>>>>>”, include the word“ERROR”, and list information that may be useful for understanding the error.

16. Output Files

16.3 Results

Except in the diagnostics tables containing accounting information, every output quantity is followed by a one-or two-digit integer that is an estimate of

the one-sigma statistical uncertainty of that quantity expressed as a percentage of the quantity. Details of the method used to obtain these statistical data are given in the Statistics section.

Almost all quantities in the results section are carried over during a dump and restart. Exceptions to this are random number information that is specific to the latest batch or cycle and some timing data that is specific to the latest batch or current run.

16.3.1 Diagnostics

Timing information:

The percent of the problem completed. In the output file, this states that the problem is 100 percent complete, unless the PRINT-ALL keywordis used. This can be a useful marker to determinetheprogressofacalculation,eitherinthe intermediateoutputfileorintheoutput file if the PRINT-ALL keywordis used.

The number of batches remaining in the calculation is printed.

The average time per batch is printed.

•

Random number seed and usage information:

The initial random number seed of this cycle. In serial, a cycle is the entire run. For MPI with static load balancing, a cycle is one set of batches performed in parallel. For example, to perform 20 batches on 5 processors requires 4 cycles of 5 batches each. For MPI with dynamic load balancing, a cycle has no meaning, and the random number seed reported is the seed for the next batch.

The initial random number seed of this batch. In serial, this batch is the last batch calculation performed. For MPI, this batch is the last for which a set of results is received by the master task.

The initial random number seed for the next batch. This is the seed that would be used

•

if the calculation continued to perform another batch. This seed will be used if a restart is performed. This seed may be used with the RANDOM-NUMBER keyword to initiate a calculation using an independent series of random numbers.

The number of random numbers generated in this batch.

Cumulated number of random numbers generated. This is the total number of random numbers used in the current run. This counter is reinitialized when the NEW-DATA-SET keywordis used, butit willbe maintained througharestart.

Average source energy in MeV. In adjoint, this refers to the adjuncton source. This includes tallies for the energy of source particles that are determined to have energies below the cutoff energy that are not tracked.

For **ACCEPT** sources that include zone rejection, several diagnostics are provided on the efficiency of the source location sampling.

6. Output Files

For calculations that include electric or magnetic fields, a list of tallies precedes the standard diagnostics tables. Thesearemostly self-explanatory.An electrontrappedina magneticmirrorin a void will never terminate “naturally”, so when such a condition is identified the electron is terminated, anda counterindicates the numberof times this occurred. The “cornerproblem” arises insome caseswhenaparticle’scurved trajectoryinafieldwouldgoaroundacornerandapproach a surface tangentially. An algorithm for coping with the corner problem has been implemented, and the counter records the number of times the more computationally expensive algorithm is invoked.

For **CYLTRAN** the number of times that a particle is rejected due to geometry problems is reported.

For ITS **CYLTRAN** and **ACCEPT** the number of times that “kicking” is terminated due to geometry errorsisreported. Kickingis the terminationprocedure for electronsin the continuous-energy codes. It consists of moving the electron a random fraction of its residual practical range in rectilinear motion.

For ITS (not **MITS**), there are two diagnostics tables. The history table reports the number of events simulated:

PRIM – primary histories simulated.
SEC –secondary electron histories simulated. Thisisthe sumof KNOCK,PE,PAIR,COM, and AUGER. An electron history is not simulated unless the electron is produced above the cutoffenergy and is not immediately trapped.

KNOCK – knock-on electrons simulated.

- PE– photo-electrons simulated.
  PAIR –pair-production electrons and positrons simulated.
  COM – Compton-produced electrons simulated.
  AUGER – Auger-produced electrons simulated.
  BREM – bremsstrahlung photons simulated.

RAD – unscaled bremsstrahlung events sampled to account for radiation energy-loss straggling of electrons.

XRAY – fluorescence photons simulated.

REJ. LAND – numberof times that samplingof electron energy-loss straggling wasrejected, either to preserve the mean energy loss or because the energy loss for the step would have been greater than the initial electron energy.

REJ. PEAL – number of times that sampling of the photo-electron emission angle yielded a scattering cosine with an absolute value greater than one. (The scattering cosine was set to 0.99999.)

PRIM STEPS – condensed history steps simulated for primary electrons.

SEC STEPS – condensed history steps simulated for secondary electrons.

16. Output Files

NBLK – number of times that sampling of electron energy-loss straggling would have resulted in electron energy gain. (Energy loss was set to zero.)

INCOH. SCAT – incoherent (Compton) photon scattering events.

COH. SCAT – coherent photon scattering events.

The second ITS table reports the number and energy of secondaries produced. This includes particles produced below cutoffthat are not tracked, except in the case of knock-on electrons. The ENERGY is the mean energy in MeV of this type of secondary produced per primary history. The AVE ENERGY is the average energy in MeV of this type of secondary produced per interaction that produced this type of secondary. The NUMBER/PRIMARYis the estimated mean number of secondaries of this type that would be produced per primary history in an unbiased calculation (i.e., without cross section scaling). The NUMBER GENERATED is the number of production events simulated by the code.

FIRST KNOCK – knock-on electrons produced by primary electrons.

TOTAL KNOCK– knock-on electronsproducedby all electrons.

PHOTO-ELECTRON – photoelectric interactions producing electrons.

PAIR –pair production interactions producing electrons and positrons.

COMPTON – incoherent Compton scattering events imparting energy to electrons.

AUGER – Auger electronsproducedby either photon or electron interactions.

FIRST BREMSSTRAHLUNG – bremsstrahlung photons produced by primary electrons.

TOTAL BREMSSTRAHLUNG– bremsstrahlung photons produced by all electrons.

X-RAY(P-IONIZATION) – fluorescence x-rays produced by photon interactions. For standardcodes (not **PCODES**), this applies only to K-shell fluorescence x-rays.

X-RAY(E-IONIZATION) – fluorescence x-rays produced by electron interactions. For standardcodes (not **PCODES**), this applies only to K-shell fluorescence x-rays.

ANNIHILATION QUANTA – photonsproducedby positron annihilation.

For **MITS**, the history diagnostic table reports the number of events simulated. All tallies disregardnon-unity particle weights that may arise due to biasing.

Primary histories – primary particle histories initiated. (In adjoint, these are adjuncton source particles and may includea mixof particle species.)

Unscattered primary photons – photons that do not interact from initiation to escape.

Electron escape scores – all electrons entering the escape zone through any surface.

Photon escape scores – all photons entering the escape zone through any surface.

Cutoffscores – all particles slowing down (in adjoint, speeding up) pasta cutoff energy.

16. Output Files

- Boltzmann and FP interactions – any type of scattering or

absorption interaction with the medium. The breakdown of these interactions follows with descriptions indented.

– FP interactions – electron Fokker-Planck angular scattering interactions.

– Electron elastic interactions – electron angular scattering interactions without correlated energy loss.

– Photon coherent scattering – photon angular scattering interactions without energy loss.

– Electron inelastic interactions (non-absorption) – an electron interaction from which a particle emerges, other than Fokker-Planck or elastic scattering. As the transport is performed, it is possible to have an electron interaction produce a photon but not an electron, since the particles emerging from an interaction are sampled without preserving the identity of the original particle.

– Photon inelastic interactions (non-absorption) –a photon interactionfrom whichaparticle emerges, other than coherent scattering.

– Electron inelastic interactions (absorption) – an electron interaction from which no particles emerge.

– Photon inelastic interactions (absorption) – a photon interaction from which no particles emerge.

CPSL too small for scor/cpsl – In adjoint only, if an isotropic surface-source is desired, this tally records how many times an escaping adjuncton had an angular cosine too small to allow for the necessary angular weighting of the score. Currently, the angular cosine limit is

$$1 - 10^{-13}$$

.

Non-physical upscatters (in adjoint, non-physical downscatters) – If cross sections aregenerated using differing energy group structures for different particle species, then it is possible to have a cross section for producing particles in one group (A) from another group (B) that extends lower in energy. If a particle has an energy in group B below the lower energy bound of group A, and the sampling of an interaction produces a particle in group A, then the particleproduced must havea higher energy than the interacting particle thatproduced it. This is non-physical. Such interactions are allowed, but they are tallied so that the user may know how many such interactions occurina simulation.

Exponent argument less than -88 in forcing logic – In **CYLTRAN** and **ACCEPT** only, if collision forcing is used for photons, the code may attempt to use an exponential argument less than -88. The value -88 is used instead to avoid underflow errors on some platforms. This parameter (defined as C88 in the Modules/params M.F file) can be adjusted.

Exponent argument less than -88 in sampling distance to collision – In **TIGER**, the problem described in the previous bullet can arise whether collision forcing is used or not.

Exponent argument less than or equal to -88 in next-event logic, score was not tallied. If the NEXT-EVENT-ESCAPE keywordis used(orifthelogicis usedduetoa PHOTON-ESCAPE or PHOTON-SURFACE-SOURCE request), then scores are not tallied for which the probability of

escape is small (and may cause an underflow error on some platforms). Note that because the tally is proportional to the probability of escape which is small, this is unlikely to affect results. The C88 parameter can be modified to determine the effect. Using a value

greater than -88 (C88 less than 88) may result in a speed up, since ray-tracing is halted for any escape path when the probability falls below e^{-C88}.

- Exponent argument greater than -88 in next-event logic, score was tallied. This tally is the complement of the previous tally. If next-event logic is used for photon escape, then the sum of these two tallies should equal the number of photon escape scores (unless therehave been lost particles during the next-event logic).

The number of source particles and total energy in MeV of source particles rejected for being below the cutoff energy. (In adjoint, this means adjuncton source particles.) Because energy spectra are allowed to extend below global cutoff energies, this tally allows the user to assess the significance of source particles that were not tracked and whether the cutoff energy must be lowered to achieve an accurate simulation.

For the **PCODES**, a table of the number of electron impact ionizations is given for each shell, followed by a similar table for photoelectric ionizations.

16.3.2 Integral Electron Emission (*ITS ACCEPT Only*, *ELECTRON-EMISSION keyword*)

For each surface on which electron emission is calculated, the number and energy of electrons is reported. This is the average number per source particle and the average cumulative electron energy per source particle.Tallies are onlyrecorded forelectrons with energies above the electron cutoff energy.

For each surface, a row in the table is labeled by the surface index and body index supplied by the user. The last row of the table reports tallies for any electrons emitted into the cavity through a surface not specified by the user. Any tallies in this row may be an indication of an error in the problem specification.

16.3.3 Integral Escape *(Forwar*d ***Only***)

Fora photon source, the number and energy escape fractions for unscattered primary photons is printed.

For ITS (not **MITS**),the number escape fraction(thatis, numberper source particle)isgivenfor electron generated secondary electrons (E-SEC), photon-generated secondary electrons (P-SEC), and annihilation photons. The number escape fraction is given for x-rays generated in each material. For the non-**PCODES**,x-rays areonly generated for the K-shell. This information is given for each surface requested or all surfaces by default (see the ESCAPE-SURFACES keywordfor more information).

The next section contains tables of the number and energy escape fractions for each particle species. For all codes, information is given for electrons and photons. For **MITS**, information is also given for positrons. For ITS, positrons are counted as electrons here. This information is given for each surface requested or all surfaces by default (see the ESCAPE-SURFACES keyword for more information).

For ITS, the following values are stated:

Energy escape fraction below cutoff. These electrons and positrons are not included in any other output tallies.

Net charge escape fraction below cutoff. These electrons and positrons are not included in any other output tallies.

16. Output Files

- Net charge escape fraction above cutoff. This quantity may be different than the number escape fractiongivenforelectronsintheabovetable,

becausethisquantityassignsanegative weight to escaping positrons.

The energy of cutoffphotons assumed to have escaped is stated. This quantity should be zero, unless the CUTOFF-PHOTONS-ESCAPE keywordhas been used.

16.3.4 Boundary Currents *(TIGE*R *Only*)

The electron boundary currents at each material interface are reported. For bothtransmission (particles going in the positive-z direction) and reflection (particles going in the negative-z direction), the number currents and energy currents are listed with the percent statistical uncertainty. The currents will include electrons below the cutoff energy, unless the NO-KICKING keywordis used.

Positrons arescored with electrons in this tally,in the same way and also with positive weights.

16.3.5 Energy and Charge Deposition *(Forwar*d *Only*)

Anew section is initiated for every 50 subzones, and header information is repeated, unless one of the NO-DEPOSITION-OUTPUT or NO-SZDEPOSITION-OUTPUT keywords areused. Results are normalized fora single source particle. The total energy and charge deposition and three subdivisions of each areprovided, unless the NO-DEPOSITION or NO-DETAILED-DEPOSITION keywordsareused.ForITS,the subdivisionsarebasedonthephysicsresultinginthe deposition. For MITS, the subdivisions are based on the mechanics of the code. For ITS (not **MITS**), the four columns are PRIM, E-SEC, P-SEC, andTOTAL.

PRIM is deposition directly by the primary particle. For photon sources the PRIM energy deposition should be zero, since only electrons deposit energy.

E-SEC is the redistribution of energy or charge due to the secondaries of an electron (or positron),i.e., knock-on electronsorAuger electronsproducedby electron(orpositron)ionization. In ITS, the primary electron energy-loss associated with production of knock-on electrons is not directly correlated with the production of knock-on electrons. All primary energy-loss is recorded under PRIM. The E-SEC tally records a negative energy deposition fortheproductionofa knock-on electronandrecords positive energy depositionforallsubsequent energy-loss interactions by the secondary electron. Therefore, since this is a redistribution tally, it is not uncommon to find negative energy deposition tallies for E-SEC.

P-SEC is the redistribution of energy and charge due to the secondaries of a photon, e.g., a Compton electron. Note that this includes the energy of a photo-electron produced below the cutoff energy and therefore not tracked. As with the E-SEC tally, it is not uncommon to find negative energy deposition tallies for P-SEC, since thisisaredistribution tally.

TOTAL is all deposition in the subzone. This should equal the sum of the other three.

For **MITS**, the four columns are MICRO, TRACK, FLCUT, andTOTAL.

MICRO is deposition due to microscopic deposition tallies. This should only be non-zero if the MICRO keywordis used.

16. Output Files

TRACK is deposition due to flux-fold tallies. If using the MICRO keyword, this is only due to continuous-slowing-down of electrons.

FLCUT is deposition by particles falling below the cutoffenergy or electrons being trapped.

TOTAL is all deposition in the subzone. This should equal the sum of the other three.

For **TIGER**,the energy deposition section appears beforethe charge deposition section. Energy deposition is given in MeV $\times$ cm^2/g. Charge deposition is in electrons $\times$ cm^2/g. The columns for both sections report the subzone number, the material number for the subzone, the minimum-z edge of the subzone in cm, the maximum-z edge of the subzone in cm, the minimum-z edge of the subzone in g/cm^2, the maximum-z edge of the subzone in g/cm^2, and the four deposition values (with percent uncertainties). At the end of each table is a row of totals across all subzones in the problem.

For **CYLTRAN**, the energy deposition section appears before the charge deposition section. Energy deposition is given in units of MeV, and charge deposition is given in units of electron charge (i.e., one electron deposits a charge of 1.0, and one positron deposits a charge of -1.0). The columns for the energy deposition table report the subzone number, the material number for the subzone, the mass of the subzone in grams, the volume of the subzone in cm^3, and the four deposition values (with percent uncertainties). The columns for the charge deposition table report the subzone number, the material number for the subzone, the minimum-z edge in cm, the maximum-z edge in cm, the inner radius edge in cm, the outer radius edge in cm, the lower azimuthal boundary in degrees, the upper azimuthal boundary in degrees, and the four deposition values (with percent uncertainties). At the

end of each table is a row of totals across all subzones in the problem.

For **ACCEPT**, energy and charge deposition information are contained in the same table. Unless the DEPOSITION-UNITS keywordis used, energy deposition is given in units of MeV, and charge depositionisgiveninunitsofelectroncharge(i.e.,oneelectrondepositsachargeof1.0,and one positron deposits a charge of -1.0). The first column contains the subzone number. If subzoning is not activated, the zone number is the same as the subzone number. If subzoning is activated in the problem, the start of each zone is marked by a line stating the “INPUT ZONE NUMBER”, and for the zones that are subzoned, an additional line states the total deposition quantities for the zone and the following line notes how the subzone numbers are incremented within the zone. The second column states the material number for the subzone. The remaining columns report the four deposition values (with percent uncertainties) for energy deposition and then for charge deposition.Attheendofthetableisarowoftotalsacrossall subzonesintheproblem.Thetotals arenot normalizedbythe DEPOSITION-UNITSscaling factors,sotheyarealwaysinunitsofMeV for energy deposition and electron charge for charge deposition.

The energy conservation fraction is the sum of all energy accounted for by escape and deposition divided by the average source energy.

The charge conservation fraction for an electron source is the sum of all charge accounted for by escape and deposition in units of electron charge. The charge conservation fraction for a photon source is one minus the sum of all charge accounted for by escape and deposition. For either source, this statistical quantity should converge to one.

16.3.6 Particle Flux *(Forwar*d ***Only***)

All flux estimates are obtained via a track-length tally and are normalized to one source particle. Results are normalized by the volume of subzone, the energy interval (for energy differential

tables), and the angular interval (for angle differential tables). The volume used for this normalization may not be the actual volume of the subzone, depending upon the geometry volume option requested. Unless a negative GEOMETRY flag is used, the volumes of all subzones are printed in the Reading Input section of the output (see section 16.1.3), and these volumes are used for the normalization. WARNING: In**ACCEPT**, automatic subzone volume calculation may not accurately describe the intersection of subzones with the zone when nonconformal subzone overlays are used. In **CYLTRAN** and **TIGER**, subzone volumes are always accurate because of the simplified geometries.

In **TIGER**, the energy spectrum flux is stated in units of #/MeV, the energy spectrum and angular distribution flux is stated in units of #/MeV/sr, and totalangular distribution flux is stated in units of #/sr. In **CYLTRAN** and **ACCEPT** (if the subzone volumes are accurate) the energy spectrum flux is stated in units of #/cm^2/MeV, the energy spectrum and angular distribution flux is stated in units of #/cm^2/MeV/sr, and total angular distribution flux is stated in units of #/cm^2/sr.

Electron *(ELECTRON-FLU*X *keyword*) For ITS (not **MITS**), the “electron left at flux cutoff energy” is listed before the energy spectrum of the electron flux. This is given as sets of two rows of numbers. The first row contains the subzone numbers. The second row (in columns aligned with the subzone numbers) contains the number of electrons per source particle left at the flux cutoffenergyineach subzoneandthe associatedpercent uncertainty. Eachrow containsuptoten subzone quantities.

Atable of “energy spectrum of electron flux” appears for each subzone of the zones requested with the ELECTRON-FLUX keyword. For each energy interval requested, the electron flux and percent uncertainty are stated.

Atable of “energy spectrum and angular distribution of electron flux” is given for each sub-zone of the zones requested. For **ACCEPT** and **CYLTRAN**, the azimuthal interval is stated in the header for each table. For **TIGER**,the azimuthal intervalisalwaysimpliedtobe0to360degrees, since it is a one-dimensional code that cannot resolve the azimuthal direction. Column headers state the polar (theta) interval in degrees for the corresponding header. (If the direction-space option is used, the azimuthal and polar information appear in the opposite locations.) The energy interval associatedwiththefluxvaluesisstatedatthestartofeachrow.Thefluxvalueandperc ent uncertainty is then stated for the energy interval given at the start of the row, the polar interval stated above the column, and the azimuthal interval stated in the table header.

At the bottom of each energy-angular flux table, the total angular distribution of flux is stated for the entire energy interval for which flux was tallied. This total energy interval is stated at the startoftherow.Again,theangular intervalscorrespondtothepolar intervalinthecolumnheader and the azimuthal interval in the table header.

Photon *(PHOTON-FLU*X *keyword*) A table of “energy spectrum of photon

flux" appears for each subzoneofthe zonesrequestedwiththePHOTON-FLUXkeyword.Foreach energy interval requested, the photon flux and percent uncertainty are stated. The flux from continuum radiation and line radiation are listed separately. The continuum radiation appears in the same format as the electronflux values.Atthe startofeachrow containingline radiation information,the energy of the line is stated. Annihilation radiation (if applicable) appears first. Then, line radiation fluxes are given for each possible transition in each material. The transition producing the line radiation is stated with the line energy at the start of each row. (Only K-shell transitions are simulated in the standardITS codes. The **PCODES** allow more detailed line radiation. For line radiation fluxes

in **MITS** calculations, line radiation groups must be present in the cross section set, and therefore must be requested in the CEPXS input.)

Atableof"energyspectrumandangular distributionofphotonflux"isgivenforeach subzone ofthe zonesrequested.The continuumandline radiationarestated separately.The table formatis like the electron energy-angular flux, with the addition of line radiation as in the energy spectrum photon flux.

At the bottom of each energy-angular flux table, the total angular distribution of flux is stated for the entire energy interval for which flux was tallied. The total includes both continuum and line radiation.

16.3.7 Spectrum of Absorbed Energy (*ITS Only*, *PULSE-HEIGHT keyword*)

Theenergy intervalsforrecordingtalliesarelistedatthestartofeachrowofdata.Thenumber of tallies and percent uncertainties in the estimate follow. The results are in units of number of tallies per MeV, and are normalized to a single history (i.e., a single source particle).

It is common practice to have the first energy interval and last energy

interval be very small. The first energy interval, spanning a range from the source energy to an energy slightly smaller thanthe source energy,records talliesforthe total absorptionof source energy withinthe detector. The last energy interval, spanning a range from slightly larger than zero energy to zero, records tallies for no source energy absorption within the detector.

The spectrum of absorbed energy is a pseudo-pulse height distribution. It differs from a true pulseheight distributioninthatITSisa ClassIelectron transportcode, meaningthatitdoesnot correlate electron energy loss with the energy imparted to knock-onelectrons. (Energy loss due to bremsstrahlung production is sampled with correlation to secondary production.) Electron energy loss is sampled based on straggling distributions to determine the energy lost by an electron. The energy lost is deposited in the medium. The production of knock-on electrons is sampled separately. Energy initiatinga knock-on electronisremovedfromthe medium.Inagiven history, it is possible to have more energy removed from a subzone than is deposited, but statistically the energy deposition is accurate as the number of histories is increased. Pulse height distributions are tallies on a per-history basis. It is possible to have physically unrealistic negative absorbed energy tallies. Adiagnostic following the absorbed energy table states the number of counts that were rejected due to negative energy deposition.

16.3.8 Electron Emission (*IT*S *ACCEP*T *Only*, *ELECTRON-EMISSIO*N *keyword*)

The first line of each header states that the table reports the energy spectrum and angular emission distributionofelectrons.Thesecondlinestateswhichsurfaceofwhichbodytheelectron emission has been tallied for, and if applicable also reports the subsurface index for that surface. Results are normalized for a single source particle and are given in units of #/MeV-sr (number of electrons emitted divided by both the energy interval and solid angle interval). On the fifth line, the azimuthal interval is stated (or polar interval, if direction-space binning is used).

Column headers state the polar (theta) interval in degrees for the corresponding header. (If the direction-space option is used, the azimuthal and polar information appear in the opposite locations.) The energy interval associated with the electron emission values is stated at the start of each row. The emission value and percent uncertainty is then stated for the energy interval given atthestartoftherow,thepolar intervalstatedabovethecolumn,andthe azimuthal intervalstated in the table header.

t

F
i
l
e
s

16.3.9 Escape Spectra *(Forwar*d ***Only***)

All escape estimates are obtainedby tallying particles entering the escape zone. For **ACCEPT**, the accuracy of the escape tallies depends upon an adequately defined escape zone (e.g., the geometry shouldbe non-reentrant). Escaperesultsare normalizedtoone source particle,the energy interval (for energy differential tables), and the angular interval (for angle differential tables).

Separate tables of escape information are given for each surface requested. For **ACCEPT**, the defaultistoreporttheescape information“throughall surfaces”inanintegratedtable.For **TIGER** and **CYLTRAN**,the defaultistoreport escape information separatelyforeachofthe2or3possible escape surfaces. See the ESCAPE-SURFACES keywordfor more information.

The energy spectrum escape is stated in units of #/MeV, the energy spectrum and angular distribution escape is stated in units of #/MeV/sr, and total angular distribution flux is stated in units of #/sr.

Electron *(ELECTRON-ESCAP*E ***keyword***) For each surface requested with the ESCAPE-SURFACES keyword, two tables of data may be given depending upon the resolution requested with the ELECTRON-ESCAPE keyword.

A table of “energy spectrum of escaping electrons” is written first. For each energy interval requested, the electron escape and percent uncertainty are stated.

A table of “energy spectrum and angular escape distribution of escaping electrons” is presented next (if angular binning was requested). For **ACCEPT** and **CYLTRAN**, the azimuthal interval is stated in the header for each table. For **TIGER**, the azimuthal interval is always implied tobe0to360degrees, sinceitisa one-dimensionalcodethat cannotresolvetheazimuthaldirection. Column headers state the polar (theta) interval in degrees for the corresponding header. (If the direction-space option is used, the azimuthal and polar information appear in the opposite locations.) The energy interval associated with the escape values is stated at the start of each row. The escape value and percent uncertainty is then stated for the energy interval given at the start of the row, the polar interval stated above the column, and the azimuthal interval stated in the table header.

At the bottom of each energy-angular escape table, the total angular distribution of escape is stated for the entire energy interval for which escape

was tallied. (This total energy interval is statedatthe startoftherow.) Again,the angular intervals correspondtothepolarintervalinthe column header and the azimuthal interval in the table header.

Escaping positrons are included in the electron escape tallies.

Photon *(PHOTON-ESCAP*E ***keyword***) For each surfacerequested with the ESCAPE-SURFACES keyword, two tables of data may be given depending upon the resolution requested with the PHOTON-ESCAPE keyword.

Atable of “energy spectrum of escaping photons” is written first. For each energy interval requested, the photon escape and percent uncertainty arestated. The escape of continuum radiation and line radiation arelisted separately. The continuum radiation appears in the same format as the electron escape values. At the start of each row containing line radiation information, the energy of the line is stated. Annihilation radiation (if applicable) appears first. Then, line radiation escape is given for each possible transition in each material. The transition producing the line radiation is stated with the line energy at the start of each row. (Only K-shell transitions are simulated in the standardITS codes. The **PCODES** allow more detailed line radiation. For line radiation escape in **MITS** calculations, line radiation groups must be present in the cross section set, and therefore must be requested in the CEPXS input.)

1
6
.

O
u
t
p
u
t

F
i
l
e
s

Atable of “energy spectrum and angular escape distribution of escaping photons” is presented next (if angular binning was requested). The continuum and line radiation are stated separately. The table format is like the electron energy-angular escape, with the addition of line radiation as in the energy spectrum of escaping photons.

At the bottom of each energy-angular escape table, the total angular distribution of escape is stated for the entire energy interval for which escape was tallied. The total includes both continuum and line radiation.

16.3.10 Sources and Responses *(Adjoin*t *Only*)

In adjoint, a separate output section exists for each source energy spectrum. If no forward source spectrum is specified, results are based on the default flat energy spectrum (i.e., a unit strength flat energy spectrum is applied to each energy range for which results are reported). An output headeris written for each setof output, anda new setof outputis initiatedif more than8 columns are required. Each adjoint output header first describes the detector:

Detectorresponsetype(DOSE,KERMA,CHARGE,ESCAPE ELECTRONSorESCAPEPHO-TONS)

Units of the response values. For TIGER or for ACCEPT surface-sources, the units are (MeV/g)/(source-particle/cm2) for dose or kerma, (electrons/g)/(source-particle/cm2) for charge, or number/source-particle for escape. For ACCEPT surface sources with an escape detector, the units are (number/cm2)/(source-particle/cm2). For ACCEPT volume-sources, the units are (MeV/g)/(source-particle/cm3) for dose or kerma, (electrons/g)/(source-particle/cm3) for charge, or (number/cm2)/(source-particle/cm3) for escape.

For escape, the energy range of particles detected.

For escape, the angular extent of particles detected.

The location of the detector. (Surface for escape; point or volume for deposition.)

For deposition, the material for the detector.

Next, the sources are described in a header section that applies to all detector-response values that follow:

The location of the source. For ACCEPT, the default is that surface sources are “through all surfaces” of the escape zone.

The angular distribution of the source (isotropic, cosine-law, or delta0-ave).

The angular extent of the source in degrees. For energy intervals only, the source extends over the full angular range. For energy and angle intervals, separate headers may be provided for sources extending over only a portion of the azimuthal range (normally) or polar range (for direction-space).

Ifresultsare normalizedtoone source particle,thatis stated. Otherwise,the sourcestrength is stated.

Finally, sources are described by row and column labels that apply only to detector-response valuesin the correspondingrows and columns:

Row labels state the source energy ranges in MeV for each detector response value.

Column labels state the polar range (normally) or azimuthal range (for direction-space) of the sources in degrees.

16.3.11 CAD diagnostics *(CAD*, *no*t *C*G *ONL*Y *mode*)

Source particle location:

Totalcalls–The numberoftimesasource particle zone numberwasrequested(shouldequal the number of histories).

Percent right (mirroring only) – The percentage of agreement between CAD and CG.

Percent wrong (mirroring only) – The percentage of disagreement between CAD and CG.

Percent on boundary – The percentage of source particles for which the zone number could not be determined because the particle was located on a boundary.

Percent unknown – The percentage of source particles for which the zone number could not be determined.

Percent bad – The percentageof source particles for which CAD experienceda failure while trying to determine the particle location.

- Percent rejected – The percentage of source particles rejected due to failure to the determine zone number.

Distance to boundary:

Total calls–The numberof timesthe distancetoa boundary wasrequested.

Percent right (mirroring only) – The percentage of agreement between CAD and CG.

Percent wrong (mirroring only) – The percentage of disagreement between CAD and CG.

Percent no-hits –Thepercentageof attemptsthat failedto locatedaboundaryinthedirection requested.

Percent rejected – The percentage of calls that resulted in a particle being rejected due to failure to find a valid distance to boundary. This is not the same as a percentage of source particlesrejected, since someof theserejected particlesmaybe secondary particlesandmay havea varietyof weightsdueto biasing.

The sidestepping functionality is currently disabled due to concerns about consistency with the geometry tracking logic. The following diagnostics remain as an indication of how efficient the algorithm might be.

- 1sidestep attempted– The number of times that, because a boundary could not be located or the source particle location could not be

determined, the particle was pushed 1×10^{-7} cm perpendicular to the direction of flight, and the distance to a boundary was attempted to be determined.

16. Output Files

2sidesteps attempted– The number of times that, because the first sidestep failed, the particle was pushed 1×10^{-7} cm perpendicular to the direction of flight and perpendicular to the direction of the first sidestep from the location of the particle before the first sidestep was attempted. The difference between this value and the “1 sidestep attempted” value is the number of times that the first sidestep succeeded and transport continued with the particle on a slightly altered path.

3 sidesteps attempted – The number of times that, because the second sidestep failed, the particle was pushed 1×10^{-7} cm perpendicular to the direction of flight (in the opposite direction of the first sidestep).

4 sidesteps attempted – The number of times that, because the third sidestep failed, the particle was pushed 1×10^{-7} cm perpendicular to the direction of flight (in the opposite direction of the second sidestep).

Sidestepsfailed –Thenumberoftimesthata

- distancetoboundarycouldnotbe determined eitheralongthe original particlepathoralong4differentoffset particlepaths.

Boundary crossing location:

Total calls– The number of times that the zone number of a particle location was requested aftera particle crosseda boundary.

Percent right (mirroring only) – The percentage of agreement between CAD and CG.

Percent wrong (mirroring only) – The percentage of disagreement between CAD and CG.

Percent on boundary – The percentage of particles for which the zone number could not be determined because the particle was located ona boundary.

Percent unknown – The percentage of particles for which the zone number could not be determined.

Percent bad – The percentage of particles for which CAD experienced a failure while trying to determine the particle location.

Percent rejected – The percentage of particles rejected due to failure to determine the zone number.

16.3.12 Timing Data

All times are reported in seconds. In serial calculations, the timing uses the function cpu time if it is available. In parallel calculations, the timing uses wall-clock time. Astatement appears in the output indicating whether a cpu timer or wall-clock timer was used.

BatchtimesrecordonlythetimeduringwhichtheMonteCarlo calculationisbeing performed and exclude all I/O, pre-processing, and post-processing. In a RESTART calculation, only the cumulative Monte Carlo time, average batch time, and average history time include the times of the previous run(s).

- Startup time (s) – For **CAD** calculations, this is the time required to process the CAD geometry and input.

16. Output Files

Allocation sizing time (s) – This is the time to read input and cross sections to determine necessary array sizes and to allocate arrays.

Input processing time (s) – This is the time to read input and cross sections and perform preprocessing before initiating the Monte Carlo calculation.

Batch initialization time (s) – This is the time required to initialize variables before each Monte Carlo batch calculation. In a parallel calculation, this time is only computed on the master process.

Output processing time (s) – This is the time required for post-processing and writing output, including intermediate output. In a parallel calculation, this time is only computed on the master process.

Monte Carlo cycle time (s) – This is only meaningful if using **MPI** with static load-balancing. This is the sum across all cycles of the longest batch time for each cycle.

Longest Monte Carlo time (s) – This is only meaningful if using **MPI**. This is the longest batch time across all batches.

Elapsed executiontime(s) –ThisisthetotalruntimeforITS.This includesI/O,preprocessing, Monte Carlo, and post-processing within the current run.

Cumulative Monte Carlo time (s) – This is the sum of all batch times.

Averaged Monte Carlo batch time (s) –This is the average time required to perform a batch calculation for all batch calculations performed.

Average Monte Carlo history time (s) – This is the sum of all batch times divided by the number of Monte Carlo histories simulated.

Last Modified Date: 2008/04/17 22:45:43

17 PCODES

The SANDYL[14] code is a three-dimensional multimaterial code. Its construction was oriented toward relatively low-energy photon sources and the understanding of internal electromagnetic phenomena in complex geometries. In particular, it includes a detailed modeling of atomic shell ionization and relaxation phenomena for electron and photon energies down to 1.0 keV. The early codes of the TIGER series, on the other hand, were developed primarily for relativistic electron beam applications, where atomic shell effects usually play only a minor role and are, consequently, treated in a more cursory fashion. Nevertheless, these potential low-energy limitationswerepurely incidental,andtherewasnoreasonwhythemore complete descriptionof atomic shell effects available in the SANDYL code could not be included in the codes of the TIGER series. Those codes in ITS that include the more detailed ionization/relaxation physics from the SANDYL code are referred to as the PCODES. The PCODES contain the logic necessary for describing ionizationandrelaxationofallK,L1,L2,L3,M(average),andN(average) shells having binding energies greater than 1.0 keV for elements with atomic numbers Z=1 to Z=100.[14][12]

Once a photoionization or electron impact ionization event has occurred, several different relaxation cascades are possible. A large quantity of atomic relaxation data is required for the stochastic description of these cascades. These data, together with the photoionization probabilities, are tabulated in Ref. [12], which also includes a discussion of the cross sections for electron impact ionization and details of the implementation of these processes in the PCODES.

The standardcodesonly includea descriptionofthe electron impact ionizationoftheKshell of the highest-atomic-number element in a given material. Similarly, following either this impact ionization or a photoelectric event, these codes only model relaxation processes (production of Auger electrons and fluorescent photons) from this same shell.

Last Modified Date: 2008/04/17 22:45:33

18 Electric and Magnetic Fields

In many instances the value of strictly collisional transport models is questionable because the actual experiments involve macroscopic electric and magnetic fields whose effects upon radiation transport not only cannot be neglected but may even be moreimportant than the collisional effects. Inordertoaddressthis situation,wehave developedamodelthat combines sophisticatedcoupled electron/photon collisional transport with transport in externally applied macroscopic electric and magnetic fields of arbitrary spatial dependence.

The model allows magnetic fields in both material and void regions. Of course, magnetic fields alone will only deflect electrons without changing their energy. The procedure for combining collisional energy loss and deflection with magnetic deflection has been described elsewhere in detail.[15][35][36] Briefly stated, the rectilinear random-walk substeps of the field-free model[1][2] are replaced by numerically-integrated segments of field trajectories in vacua whose integrated areal densities are equal to those of the substeps. Sampled collisional deflections are superimposed upon the electron direction at the end of each of these vacuum-trajectory segments. The numerical integration scheme determines those locations along the segment that correspond to energy deposition and secondary production (knock-on electrons, bremsstrahlung photons and relaxation particles from electron impact ionization), as well as the intersections of the trajectory segments with material boundaries. Magnetic fields shouldbe ignoredinregions where transport is collision dominated, because the combined simulation is quite expensive in such cases, though the results are the same as for collisions alone.

Electric fields (or combined electric and magnetic fields) are allowed only in void regions. This restriction has been imposed because no sufficiently general scheme has been derived to accountfortheeffectsof changesinthe electron energyproducedbythe macroscopic electric field upon the energy-dependent multiple-interaction collisionalprocesses withina given substep. For those applications where electric fields are present within material media (e.g., potential buildup in dielectrics and sustaining fields in gas lasers), special algorithms[37] that depend upon the ratio of the electric potential gradient to the electronic stopping power must be introduced to handle thisdifficulty. Evenwiththerestrictionof electric fieldstovoidregions,the modelis applicableto a wide variety of problems – for example, problems involving accelerating diodes.

The method for accurately integrating the vacuum equations of motion in order to obtain the vacuum-trajectory segments is the essential feature of the model, whether the fields are present in material or in void regions. In voids the integration is interrupted whenever the trajectory intersects a material

boundary or a problem-escape boundary. In material regions the integration is alsoandmorefrequently,interrupted whenevertheareal density traversedcorrespondstoalocation whereenergy is to be deposited or secondary production occurs, or equals the areal density of the appropriate substep. In the latter case, collisional scattering and energy loss are accounted for and a new trajectory segment is initiated. Afourth-to fifth-order Runge-Kutta-Fehlbergroutine with automatic step-size control (RKF)[16], substantially modified to include boundary-crossing logicandother constraints,isemployedtointegratetheequationsofmotionin vacua.Thereasons for this choice are discussed at length in Ref. [36].

The current algorithm includes a major improvement over the method described in Ref. [16]. The basic RKF integrator was designed to integrate over some specific interval of the independent variable. However, model applications invariably require interruption of the integration at the roots of any one of several possible constraint functions that are functions of the dependent

variables. The most common example is the root corresponding to the intersection of an electron trajectory with a zone boundary. Other examples are the roots corresponding to the locations for energy deposition and secondary production. In the pre-ITS versions of the field codes we were forced to use relatively crude approximate solutions at these roots, which limited the overall accuracy of the model predictions to something substantially less than the inherent accuracy of the integrator.[16] The algorithm now includes an extended RKF procedure[17] that permits interruption of the integration at any one of a number of constraint functions of both the dependent and independent variables with an accuracy that is comparable to the inherent accuracy of the integrator. The moresophisticated user is free to add his own constraint functions for interrupting the integration. The constraint functions must be defined in subroutine CSTR.

Last Modified Date: 2008/04/17 22:45:28

19 Biasing Options andVariance Reduction

From a practical, if not theoretically rigorous, point of view, biasing in Monte Carlo can generally be described as the distortion of the natural analog to achieve variance reduction in certain desired output quantities. Variance reduction refers to the attainment of lower statisticaluncertainty for the same amount of run time or, equivalently, the attainment of the same statistical uncertainty in a lesser amount of run time. Except where absolutely necessary, biasing should be used sparingly. In any case, it should be used with great care. Reckless use of biasing (overbiasing or underbiasing) can lead to results that are erratic and/or easily subject to misinterpretation. Nevertheless, there are biasing options in ITS that are easilyaccessed via input keywords and that have proven very useful in specific applications.

Zone-dependent particle cutoff energies can be specified with the CUTOFFS keyword. The BIASING keywordmay be used to simultaneously

specify global biasing parameters, the zones in which global biasing parameters are to be applied, and zone-dependent biasing parameters.

19.1 Zone-Dependent CutoffEnergies

The user has the option of varying the cutoffenergy for each species from zone to zone so long as the zone-dependent cutoffs are greater than or equal to the global cutoff energy for the particle species. The option is activated via the appropriate parameter associated with the CUTOFFS keyword.

For electrons in the continuous-energy codes, when the energy of the electron in a given zone falls below the cutoff for that zone, a check is first made to see if it is trapped in the sense described under sub-keywordTRAP-ELECTRONS. If so, the history is terminated via on-the-spot depositionofchargeand energy. Otherwise, exceptforwhenamagneticor electric fieldisusedor the NO-KICKING keywordis used, a final calculation of non-local energy and charge deposition is made based on the residual range of the electron. When a magnetic or electric field is used, a relatively low electron cutoff energy should be used because the history is always terminated via on-the-spot deposition of the charge and remaining energy of the electron. It is important to rememberthatthereisnoproductionof secondary particlesby electrons belowthe zone-dependent cutoff, nor is there any contribution to electron flux and electron escape by such electrons.

For photons,and electronsinthemultigroupcode,whenthe energyofaparticlefallsbelowthe cutofffor the zone the particle is in, the history is terminated via on-the-spot deposition of charge and energy. Because photons can travel relatively large distances before interacting and photon transportis generally inexpensive compared to electron transport, this cutoff energy should be used with caution.

The zone-dependent cutoffoption has proven useful in problems that involve the generation of bremsstrahlungin oneregion and deposition causedby thatbremsstrahlungin anotherregion.A relativelyhighcutoffmaybeusedinthe converterzone(s)since low-energyelectronsarerelatively inefficient for producing bremsstrahlung. On the other hand, in the zone(s) where deposition is dominatedbybremsstrahlung transport,the usermaybe interestedinthe detailsofthe deposition from the low-energy bremsstrahlung-produced secondaries (e.g., interface effects, in which case he may not want to kill those electrons with Russian Roulette; see Sec. 19.3), or he may not want electron transport in those zones at all (bulk deposition; see Sec. 19.5).

19.2 Forced Photon Collisions

Anoptionis availableforforcingaselected fractionofphotonsenteringagivenzoneorleaving a collision site within a zone to interact in that zone. The option is activated via the COLLISIONFORCING

sub-keyword. The option is useful for forcing photons to interact in certain regions where their natural interaction probability is so small as to make it difficult to obtain statistically significant results. The values chosen for the forcing fractions must be greater than zero and less than one. These are specified by the appropriate parameter for the given zone as described under the COLLISION-FORCING sub-keyword. Care must be taken not to overbias. Aforcing fraction of 1.0 will prevent any photons from escaping from the given zone and will prevent them from making contributions elsewhere in the results of the calculation (e.g., prevent them from contributing to the escape fractions).

19.3 Russian Roulette

When SCALE-BREMS or SCALE-EP options are used to increase the secondary photon population, it may be desirable to reduce the number of secondary electrons generated from the interaction of this artificially high bremsstrahlung population. The ELECTRON-RR option allows the user to activate Russian Roulette in specified zones of the geometry. The default survival probability (when the feature has been activated) will return the secondary electron population to the naturally occurring number.

Although this procedure is very efficient for predicting external bremsstrahlung, it can lead to statistically poor results for the profiles of energy deposition, charge deposition, and electron flux in regions of the problem where these profiles are determined by the transport of and secondary electron production by the bremsstrahlung radiation. In such cases, the survival probability should be adjusted to ensure that a sufficiently large fraction of the secondary electrons are followed.

19.4 Next-Event Estimator for Photon Escape

Fora geometrythatishighly absorbentto secondaryphotons generatedinthe transportprocess, scoring as leakage photons only those secondary photons that actually escape the geometry while being tracked can be quite inefficient. (In the continuous-energy code, the contribution to total leakage from uncollided source photons is not scored because this contribution can usually be calculated analytically and might otherwise dominate the total leakage to the extent that the scattered contribution cannot be determined. However, a flag on the PHOTONS keyword can be used to change this default.) This is remedied by using the next-event estimator for photon leakage.With this estimator,ascoreis obtained each timeaphoton emergesfromacollision. The score is simply the emergent photon weight times the probability of escape without further interaction.

This option is automatically activated as a method of variance reduction for differential leakage when the PHOTON-ESCAPE keywordis used. Otherwise, the option is not used for the default prediction of integral leakage unless explicitly activated via the sub-keywordNEXT-EVENTESCAPE.

19.5 Photon OnlyTransport

In some calculations, electron transport is only important in a small portion of the geometry. In this case, it is more efficient to simply turn off electron production throughout much of the

geometry rather than using a high electron cutoff energy. The PHOTRAN sub-keywordprovides this functionality. Zones in which electron transport is desired can be specified.

19.6 Scaling of Bremsstrahlung Production *(IT*S *Only*)

The user may artificially increase the bremsstrahlung production to improve the statistical accuracy of bremsstrahlung output without increasing the number of primary electron histories, which would be much more time consuming. The option is activated via the SCALE-BREMS sub-keyword. The cross sections are scaledbya factor specifiedby the user.

Simultaneous scaling of the cross section for electron impact ionization probabilityis also desirable (see Sec. 19.8) and is performed by default when SCALE-BREMS is used. The material selected as the basis for scaling the impact ionization (the second parameter associated with the SCALE-BREMS sub-keyword) should be that material which one would expect to dominate the bremsstrahlung production.

This option is used primarily for the prediction of external bremsstrahlung production (e.g., prediction of the environment of an x-ray source). Consequently, a Russian Roulette procedure may be desirable to reduce the number of secondary electrons generated. See Sec. 19.3 for details on the Russian Roulette feature.

19.7 Scaling of Electron-to-Photon Interactions *(MIT*S *Only*)

Different types of interactions that have the same effect are not distinguished in the MITS codes. Therefore, it is not possible to simply scale the bremsstrahlung production of photons or the electron impact ionization production, as is done in the continuous-energy codes. Instead, the SCALE-EP feature allows the scaling of all photon-producing electron interaction cross sections. The cross sections of all materials are scaled by a factor specified by the user, but the scaled cross sections are only used in regions specified by the user.

19.8 Scaling the Probability for Electron Impact Ionization *(IT*S *Only*)

An option similar to that described in Sec. 19.6 permits the user to artificially increase characteristic x-ray production by scaling the cross section for electron impact ionization. This option is activated via the SCALE-IMPACT sub-keyword. The cross section for electron impact ionization of each material

is scaled so that an electron slowing down in that material from the maximum source energy to the global electron cutoff energy will, on the average, generate a number of ionization events equal to the value of the parameter associated with this sub-keyword. Aseparate scaling factor is calculated for each material. Impact ionization scale factors are rounded to the nearest integer and are never allowed to be less than one. Impact ionization scaling is implemented in the same zones in which bremsstrahlung scaling is implemented.

If SCALE-BREMS is used and SCALE-IMPACT is not, electron impact ionization is scaled. The factor for this scaling is determined such that impact ionization events between the maximum source energy and the global electron cutoff energy equal 20% of the bremsstrahlung events over the same energy rangein the material givenby the second parameter on the SCALE-BREMS keyword. This number of ionization events is used to determine scale factors for each material.

19.9 Scaling of Photon-to-Electron Interactions *(MIT*S *Only*)

The user may artificially increase the production of secondary electrons to improve the statistical accuracy of dose or charge deposition in a region of the problem rather than increasing the

numberof histories,whichmaybemoretime consumingifrelativelyfewphotons naturallyinteract in the region. This option is activated via the SCALE-PE sub-keyword. The cross sections of all materials are scaledbya factor specifiedbythe user, but the scaled cross sections are only used in regions specified by the user.

19.10 Trapped Electrons*(Forwar*d *Only*)

In certain problems where only electrons that cross certain boundaries are important, the option activated by the TRAP-ELECTRONS sub-keywordmay be employed to reduce run time significantly. The parameter associated with this sub-keywordis the global electron trapping energy. In addition, zone-dependent electron trapping energies may be specified. Internally, an array of zone-dependent electron trapping energies is obtained, each element of which is the greater of the global trapping energy, the zone-dependent trapping energy, or the zone-dependent cutoffenergy for that particular zone (see Sec. 19.1). The option becomes effective when an electron is trapped, that is, does not have enough energy to escape from a subzone. When an electron with energy less than the zone-dependent trapping energy is trapped, its history is immediately terminated via local (on-the-spot) deposition of its charge and remaining energy. This option is commonly used when one is primarily interested in the accurate transport of those electrons escaping from all or some portion of the problem geometry. Great care should be taken in

employing this option where production of secondaries (e.g., bremsstrahlung) may be important because there is no secondary production by electrons whose histories are terminated in this fashion. It is important to note that the contribution to leakage from the subzone of any untrapped electrons with energies above the zone-dependent cutoff is much more rigorous than that of untrapped electrons with energies below the zone-dependent cutoffbecause a much cruder form of transport is employed for the latter. The decision as whether an electron is trapped or untrapped is based on subzone boundaries in the TIGER codes, axial and radial subzone boundaries in the CYLTRAN codes, and subzone (in subzoned regions) or code-zone (regions separated by an OR operator in the input-zone definitions and input zones defined without the OR operator; see the ACCEPT Geometry section) boundaries in the ACCEPT codes.

Last Modified Date: 2006/10/06 19:52:56

20 Statistics

A significant advantage of the ITS system is the computation of statistical uncertainties for virtually all output quantities. Under the default option, the total number of histories of primary particles are run in 20 equal batches. The output routine is called at the end of each batch. Immediatelybeforeeachwrite statement,acallismadetoSubroutineSTATS.Thisroutine(a)recalls the statistical variables from the previous batch corresponding to the output quantities about to be written, (b) computes the estimate of the statistical standard error (in percent) based on the number of batches that have been run, and (c) saves the statistical data from the current batch so thatit willbe available for the next batch. Unless the keywordPRINT-ALLis used, only the final results based on the total number of completed batches are printed out. The user may specify a number of batches other than 20 by using the keywordBATCHES as described in the Keywords for ITS section.

Under normal operation virtually every Monte Carlo output quantity is followed by a one-or two-digit integer from0through 99 (estimates even greater than 99 are shown as 99) that is the best estimate of the statistical standarderror expressed as a percent of that output quantity:

$$(S.E.)_N = \frac{100\sqrt{\langle x^2\rangle_N - \langle x_N\rangle^2}}{|\langle x_N\rangle|\sqrt{N-1}},$$

where

$$\langle x_N\rangle = \frac{1}{N}\sum_{i=1}^{N} X_i,$$

$$\sum_{i=1}^{N}$$

and

$$\frac{1}{N}\sum_{i=1}^{N} \langle x_N^2 \rangle = X_i^2 ,$$

The X_i's are the values of the quantity obtained from each batch, and N is the total number of completed batches. Should the more sophisticated user wish to add additional tallies to any of the Monte Carlo member codes, he will find subroutine STATS to be a useful utility. STATS provides estimated statistical uncertainties for all output quantities. It has three formal parameters. The first is a temporary array containing the current batch values for the quantities for which statistical estimates are desired when the routine is called. The routine returns the cumulative batch averages in this same array for printing. The second parameter is an array that will return the statistical estimates for printing. The thirdparameter is the number of batch values to be processed with this call (the length of the first two parameter arrays). An additional advantage to using the STATS utility is thatresults willbe includedin the arrays employedin the RESTARTfunctionality.

21. Automatic Subzoning

Last Modified Date: 2006/10/06 19:52:57

21 Automatic Subzoning

Automatic subzoning refers to that feature of the ITS codes whereby the user may direct a particular member code to internally divide a given input zone into subzones for the purpose of obtaining the spatial variation of energy and charge deposition, electron flux, and photon flux within the given input zone. This powerful feature has the potential for substantially reducing (a) user time for generating the input file, (b) machine memory requirements, and (c) machine CPU time.Without this feature, the user would have to describe each subzone asa separate input zone.Forexample,a 10x10x10subzoningofarectangular parallelepipedwiththeACCEPTcodes wouldrequireatleast1000linesofinput without automatic subzoning.

Becausethe subzonesare defined in terms of equal increments of the intrinsic coordinates of the input zone, thereis no need for explicit storageofthe boundariesof subzones,andverylittle memoryisrequiredto locatethe subzone containing an arbitrary point within the input zone. Finally, because the internal subzone boundaries are not material discontinuities, they can be ignored by the CPU-intensive tracking logic.

Automatic subzoning and related coding has grown with the development of the ITS system, and is still not as complete as we would like. In the TIGER codes, it was already virtually complete inVersion 1.0. In that version of the CYLTRAN codes, there was some capability for pseudo-subzoning a solid annulus input zone in the radial and axial directions. However, the only reduction was in the sense of Item (a) above. Once these “subzones” were generated internally, they were treated like any other input zone. Their boundary information was permanently stored, and particles weretracked through themjust like any other input zone. InVersion 1.0 there was no subzoning of any kind in the ACCEPT codes. InVersion 2.0, we implemented full automatic subzoning in the sense of Items (a), (b), and (c) above in the ACCEPT codes for input zones consisting of a single RCC or RPP body. In the latter we allowed three-dimensional subzoning, and in the former we allowed radial and axial subzoning. No additional automatic subzoning was implemented inVersion 2.0 of CYLTRAN. InVersion 3.0, we substantiallyextended this feature. We implemented full automatic subzoning in the CYLTRAN codes, extending it to three dimensions with the addition of azimuthal subzoning. Moreover, the ACCEPT codes featured the full three-dimensional subzoning of input zones consisting of a single body of type RCC, RPP, BOX, orSPH. SinceVersion3.0, automatic subzoninghasbeen addedforinput zonesofasinglebody oftypeTRC,WED,orTOR.Also,anewtypeof subzoninghasbeenaddedbasedon multi-body input zones. These are generally shells that consist of one body subtracted from the middle of another body. These entities are the RCC-RCC, RCC-TRC, TRC-TRC, TRC-RCC, SPH-SPH,TORTOR, SPH-RCC, and RCC-TOR.

There are, however, some aspects of the coding of ITS that do not yet take full advantage of or are not completely consistent with the philosophyof automatic subzoning.We discuss them here in terms of three basic questions that are frequently asked as a particle trajectory evolves within the problem geometry:

(1) Point location: In what input zone (or subzone) does a given point of the trajectory lie?

(2)Tracking: Fora particleatagiven pointina given input zone(or subzone) witha given direction, how far will it go before reaching the boundary of the given input zone (or subzone) if it continues moving in the given direction?

(3)Trapping test (electrons only): Foran electronatagivenpointinagiveninput zone(or subzone), what is the minimum distance to the surface of that input zone (or subzone)?

21.
Aut
om
atic

Su
bzo
nin
g

In principle we would like the code to respond to such queries on a subzone rather than an input zone basis. In practice, it is felt that the overhead for doing the former may be so excessive astonegatetheabove mentioned advantagesof automatic subzoning.Wehavetakena middleof theroadapproachbasedonourjudgmentofwhatisthebestoverallchoiceandwhatis feasibleat this time. SinceVersion 3.0, subzone-based electron trapping (see the section on Biasing) has been extendedtothe ACCEPTcodesandto azimuthal subzones(in additiontothepreviously available axial and radial subzones) in the CYLTRAN codes.

In general, the features of the current ITS codes that use zones rather than subzones have to do with the actual scoring of certain spatially-dependent output quantities: energy deposition, electron flux, and photon flux. All of the scoring is subzone based, but there are alternative variations to what we currently use whose variance-reduction consequences have not been fully explored.

Energy deposition and scoring of volume-averaged electron flux are coupled. Both quantities are scored at the same point, as in Item (1) above, which is randomly selected along an electron random-walk substep, or partial substep if the substep encounters a material discontinuity. Because there is one score per substep, the use of automatic subzoning will not lead to variance reduction by increasing the number of scores per electron. Rather, the variance reduction from automatic subzoning is achieved because the time required to track an electron is reduced when subzone boundaries can be ignored.

There is, however, an alternative method of scoring these quantities, not yet implemented in ITS, that could potentially lead to further variance reduction in applications where the random walk substeps are greater than the subzone dimensions. We refer to this method as track-length apportioning. The segments of the substep within each subzone must first be determined. These segments times the electron weight are then scored as volume-averaged fluxes in the appropriate subzones. Similarly, the segments, as fractions of the total substep, are used to apportion the substep energy deposition among the subzones. Multiple scoring of these quantities per substep may leadto significant variancereduction. The caveatto this approachis thatitrequires whatis equivalent to tracking, as in Item (2) above, among the subzones. It was the avoidance of tracking among subzone boundaries that wasprimarilyresponsible for the variancereduction achievedby automatic subzoning. Nevertheless, the more sophisticated user may use code modifications to implement this method. This is rather easily done for the TIGER codes (see discussion of photon flux below), but becomes progressively more difficult for the CYLTRAN and ACCEPT codes.

In the CYLTRAN and ACCEPT codes, scoring of volume-averaged photon flux is done in a fashion similar to that of electron flux. Here, however, instead

of obtaining one score per sub-step, there is one score per free-flight photon trajectory segment between collisions or input-zone boundaries. Again, scoring is at a point, as in Item (1) above, that is randomly sampled along the free-flight segment. Automatic subzoning does not increase the number of scores, and variance reductions again derive from the avoidance of tracking among the subzone boundaries. However, when the dimensions of the subzones are much smaller than the average free-flight photon trajectory segments, there could actually be a variance increase relative to the variance that would have been obtained had each subzone been defined explicitly as an input zone, since the number of scores is greatly reduced in the former case. Whether there is an overall variance increase or reduction is, of course, very problem dependent.

Because photon mean free paths tend to be much larger than electron random-walk substeps, and because one-dimensional tracking is relatively simple, it was decided to apportion the free-flight photon trajectory segments among the subzones in the TIGER codes when scoring photon flux. Thus, significant additional variancereduction, overand above thatfrom automatic subzoning alone, may accrue to the estimation of photon fluxes with the TIGER codes, unless subzone boundaries tend to be much larger than photon mean free paths. Using code modifications, this

logic may easily be extended to the calculation of electron flux in the TIGER codes.

21.1 Non-Conformal Subzone Overlays

At times, subzoning is desired for a zone that is quite complicated. For this purpose, a more general method is provided based on overlaying the zone with the subzoning of a simple entity. An input zone with arbitrary shape (that may include unions or CAD geometry zones) can be subzonedbyfirstcompletelyenclosingitinanysimplebodyoroneoftheshells availableforsubzoning (RCC-RCC, etc.). The details of activating subzone overlays are described in the ACCEPT Geometry section.

Non-conformal subzoningaffectsthe implementationofthe electron trapping logic.With non-conformal subzones it is not sufficient to determine that an electron is trapped within a subzone; it is further necessary to determine whether the electron is trapped within the zone, if it is possible that the subzone lies on the zone boundary.

Therearecurrently three methods for imposing subzone overlays. The first is to explicitly state the overlay to be used for subzoning the zone using the

OVERLAY sub-keyword. The second method, which applies only to CG zones, isto intersect the first code zone with the simple body (orbodies)insuchawaythatthefirstbody(ortwo
bodies,forashell)inthedescriptionofthe input zone specifies the entity to be used for subzoning. The additional intersection should result in a logically equivalent description of the input zone, since the original input zone description is required to be completely enclosed by that with which it is intersected (but this remains the user's responsibility). The subzone structure is now based on the first body (or first two bodies, for a shell). In both of the first two methods, it is the user's responsibility to choose a shape that makes sense; i.e., the subzone boundaries should describe a reasonable profile through the zone of interest. The thirdand most automated method, which applies only to CAD zones, is to base the subzoning on the RPP that is the axis-aligned bounding box of the zone.

The specified subzoning structure may describe some subzones that are completely outside of the original input zone. It is desirable not to include output for such subzones, and an attempt is made to exclude them from both the output file and the finite-element-format output written to the fort.3 file.

For CG, two checks can be made to determine whether to exclude a subzone from output. These checks are made if the "GEOMETRY 4" option is used. It is performed by comparing a point inside the subzone with the zone. (This point is chosen as the simple average of the eight vertices of the subzone. The vertices are subzone "corner" points that are used if finite-element formatted output is requested.) If the point is not within the zone, then the subzone is flagged for exclusion from the finite-element output. (For example, if the original zone is a quarter bend of a torus and the subzone structure is based on subzoning the full torus with the number of angular subzones around the circumference of the bend divisible by four, then only the subzones corresponding to the quarter bend of the torus will be included in the output.) Thus, if the full subzoning is not conformal, but all individual subzones are entirely inside or entirely outside of the zone, then this will be sufficient to accurately calculate subzone volumes.

The secondCG check canprovideamorerigorous determinationof whether subzones coincide with the zone. For nonconformal subzoning to use accurate volume information, the volumes (of the portion of subzones within the zone) must be supplied. Volumes can be input via the "GEOMETRY1" or "GEOMETRY4" option. (These volumes can be calculated via a Monte Carlo calculation. Using a void geometry and the UNIFORM-ISOTROPIC-FLUX source, the flux of zones and subzones will be proportional to the volume.) The preliminary flag from the first check will be overwritten if the subzone has a non-zero volume. This covers the case where part of the

subzone (containing the interior point) lies outside the input zone. It should not be possible to find the interior point inside the zone for a subzone with non-zero volume overlapping the zone. The code flags this condition as an error.

For CAD, one of two methods can be used for determining whether to exclude a subzone from output. For the “GEOMETRY 3” option, the volume is calculated for the intersection of each subzone with the zone. For the “GEOMETRY1” or “GEOMETRY4” option, the volumes of subzones can be input directly. (It may be useful to use the former option once and subsequently use the latter option to input the calculated volumes directly to avoid therepeated computational expensive of the volume intersections.) In either case, the output will be suppressed for subzones with zero volume.

22.
Random
Number
Generators

Last Modified Date: 2008/04/17 22:45:45

22 Random Number Generators

Monte Carlo calculationsrelyonthe useof pseudo-random numberstosimulatethe stochastic nature of physical processes. Therefore, one must be careful to correctly implement the available randomnumber generator(RNG).Weprovidetheuserachoiceof3RNGs.Inthis document,we first discuss the properties of the RNGs provided and then share the issues of concern we have encountered in the implementation and use of RNGs.

22.1 Portable Random Number Generators

RNGshavebeenmade availableascompile optionsinITStorelievethe userof concerns about the integrity and implementation of the intrinsic RNG on his system. All 3 RNGs are double precision and as implemented, never generate zeros or ones.

RNG1,animplementationofanRNGintheMCNP code[38],hasacyclelengthof2^{46} and allows easy access to its seed. However, RNG1 is not implemented as a parallel RNG. RNG2 and RNG3will functioninboth serialand parallel.Topermitacommon interfacefortheRNGs,RNG1 is used to seed each batch performed with RNG2 or RNG3. Regardless of which RNG is selected in the code, the user will be seeding and receiving seed information from RNG1. Because the states of RNG2 and RNG3 are quite large, they will be written to a file when required for debugging. Only the RESTART-HISTORYfeature requires access to these states.

RNG2is an implementationof the RANMAR RNG[39, 40]. It hasa state spaceof2^{4656} [41], consistingof numerous cycles with an average cycle lengthof

approximately 2^{100} [40]. Each batch of RNG2 is seeded with a combination of 97 random numbers from RNG1.

RNG3isan implementationofthe MersenneTwister RNG[42].Ithasacyclelengthof 2^{19937} -1. Each batchinarunwithRNG3is seededwithacombinationof624 random numbersfrom RNG1.

22.2 Range

The random numbers that are generated should be uniformly distributed between 0.0 and 1.0 exclusively. There are certain places in the Monte Carlo software where random numbers that are identically 0.0 or 1.0 are unacceptable,resultingin fatal execution errors.

22.3 Access to the Seed

The state of an RNG consists of those variables that are used to determine the next random number in the sequence. The state of RNG1 consists of a single variable. Access to this variable is required to assure independent random number sequences in separate runs, to have a restart capability, and to facilitate debugging. If an error condition is detected while the Monte Carlo is in progress, for RNG1 the starting random-number seed for the current source particle (variable RIR-SAVin randat.h) is printed within the terminating subroutine, ABORTX. For RNG2 and RNG3, the starting random-number seed for the current batch and the number of random-numbers generated in the batch beforethe current history (variables RIRA and CNRN(2) in randat.h) areprinted, and the initial RNG state for the current source particle is printed to a file.

One symptom of incorrect implementation of an RNG may be obtaining “0” percent statistical uncertainties for all quantities; this may happenif the same seedis being usedto start each batch so that the results from each batch are identical.

22.
Random
Number
Generators

22.4 Reproducibility

ARNG is designed to produce the same sequence of random numbers for the same starting seed.

22.5 Cycle Length

Computer generated random numbershavea finitecyclelength(the numberof random numbersgeneratedbeforethecyclerepeatsitself).Oneshould neverrunsomanysourceparticlesina givenrun so as to exceed the RNG cycle length.Asingle source particle will likely use many random numbers. The number of random numbers used in a calculation is presented in the output.

22.6 Speed

Timings indicate that for our implementations RNG2 is generally faster than RNG3 and that both RNG2 and RNG3 are faster than RNG1. However, these timings vary depending on the platform and compiler.

23
.
A
dj
oi
nt
C
al
cu
lat
io
ns

Last Modified Date: 2006/10/06 19:52:34

23 Adjoint Calculations

The adjoint calculation mode is a complement to the forwardcalculation mode. In some situations where forwardcalculations are inefficient, adjoint calculations offer an efficient alternative. In general, if one is interested in a variety of responses with distributions in space, angle, and energy(e.g.,doseandcharge deposition distributions)duetoa limited numberof radiation sources (e.g., a monoenergetic electron beam), then it will be more efficient to use the forward method. However,ifoneisinterestedina limited numberofresponses(e.g.,doseatapoint)duetoalarge number of radiation sources with distributions in space, angle, and energy (e.g., sources from various directions), then it may be more efficient to use the adjoint method.

From linearalgebra,givenan innerproduct(,),an operator T , and two functions S and *R*, the following equality holds:

$$(\mathrm{T}\,S, R)=(S, \mathrm{T}\,\dagger R),$$

where T † is the adjoint operator of T . For our purposes in the MITS code, we can think of the T as the forwardmode of the computer program, and we can think of T † as the adjoint mode of the computer program. Running MITS in both forwardand adjoint modes to calculate a single response R due toa single source S should yield identical results (within statistics) since the code is solving the same problem using the same cross sections.

The forward computer program T “operates” on a single forward source *S*, which must be described in space, energy, and angle. For Monte Carlo, we can think of the program T as tracking particles, and the particle source is specified by *S*. The result of *T*S is the particle flux. If the user wants a value of dose, for example, that is the value of the inner product, then the flux must be folded with the appropriate response R to obtain that value. For dose, the

response is the restricted stopping power for the region of interest. For a single source, the particle flux *T*S can be folded with many different responses to obtain, for example, doses in various materials throughoutthe geometry, doseprofiles withina single material, charge depositionprofiles, escaping particle distributions as a function of energy and/or angle, etc. As a convenience to the user, our codes will compute the innerproduct fora libraryof commonresponses.

Similarly, the adjoint computer program T † operates on a single response *R*, so the user must specify what single type of response (e.g., dose at a point) is desired. For Monte Carlo, we can think of the program T † as tracking "particles" called adjunctons, and the "adjuncton source" corresponds to the response *R*. The result of T †R is a particle importance map, that gives the importance of particles (as a function of species type, space, energy, and angle) to the specified response. Theinnerproduct can nowbe evaluated for many different typesof forwardsources *S*. As a convenience to the user, our codes will compute the inner product for a library of common sources.

In forwardmode, the distribution (in space, energy, and angle) and the species type of a single source must be specified using the POSITION, ENERGY or SPECTRUM, DIRECTION, and ELECTRON or PHOTON keywords. Then, a variety of responses may be selected. Energy and charge depositionarecalculatedthroughouttheproblemby default. The usermayrequest escape distributions (using the ELECTRON-ESCAPE, PHOTON-ESCAPE, and ESCAPE-SURFACES keywords), electron surface emission distributions (using the ELECTRON-EMISSION keyword), and flux distributions (using the ELECTRON-FLUX and PHOTON-FLUX keywords).

23. Adjoint Calculations

In adjoint mode, a single response must be specified with the DETECTOR-RESPONSE keyword. The user must specify the type of response as DOSE, CHARGE, KERMA, or ESCAPE and must describe some properties (such as spatial extent) of the detector. This specifies the space, energy, and angle distribution of an adjuncton source. Then, a variety of sources may be selected. The user must request at least one source, since there are no defaults. The user may specify a surface source of each species (ELECTRON-SURFACE-SOURCE and PHOTON-SURFACE-SOURCE) anda volume sourceof each species (ELECTRON-VOLUME-SOURCE and PHOTON-VOLUMESOURCE).

In addition to the choice of spatial distributions of sources in adjoint mode,

the user has the option of “binning” in energy and angle. This binning represents dividing the energy and angle domains into separate, independent sources of energy and angle extents given by the bin. Thus, if one divides a surface source into 9 equal polar-angle bins, then the output will contain the requested response for a source with a distribution from 0 to 10 degrees, a source from 10 to 20 degrees, etc. Such data may be post-processed assigning weighted distributions in angle and energyto determinetheresponsetoawide varietyof combinationsof energyand angular source distributions. On the other hand, it may be more accurate (due to the approximation of binning the sources before folding with the actual distributions) and the output will be greatly condensed if the user is able to specify an angular distribution of interest (such as COSINE-LAW), the energy spectra of interest (using the SPECTRUM keyword), and allow the code to fold the energy and angular distributions. The choice between folding with source distributions during the calculation or in post-processing is left to the user.

2
4
.
T
e
s
t
i
n
g
o
f

I
T
S

Last Modified Date: 2008/04/17 22:45:48

24 Testing of ITS

This section discusses the tests available for code maintenance. This includes installation tests, committests,andregression tests.Thefirst subsection discusseshowthetestscriptcanbeusedto run these tests. The second subsection discusses how to perform and evaluate installation testing.

24.1 Testing Scripts

The “tests.pl”scriptintheTests/RegTestsdirectoryisusedto executethe installation, commit, andregression tests. The numerousflagsthatmaybeusedwiththisperlscriptare listedinTable 10 along with

a brief description of each flag's effect. More detailed descriptions of how these flags may be used are discussed here.

The suite of tests to be drawn from may be selected using the "-commit", "-install", and/or "-regress" flags. All of the commit and regression tests are selected by default, if none of these flags is specified. These flags may be used in combination. They specify which tests are loaded, but the user may limit which tests are executed using additional flags.

For users with a released version of the code that does not include RNG3, the tests should be executed using the "-rng1" and "-rng2" flags. Users who do not have a CAD capability with ITS should use the "-cg" flag. Users who do not have the PFF libraries should use the "-notpff" flag. Users who do not have MPI libraries should use the "-serial" flag. Except for RNG1 tests and a few tests of features not available in parallel, the same tests are executed in parallel as in serial.

On some platforms it may be necessary to compile the tests (perhaps on a front-end platform) and execute them separately. Separating the compile and execution steps can also be useful for code debugging. For this reason the "-pre" and "-post" flags are available. The "-pre" flag will causethe scriptto compilethe executablesrequiredand assemblethe testsin subdirectoriesofthe $HOME/tmp directory. After these tests have been run, the "-post" flag will cause the script to post process the output files. The runcalcs.pl script, discussed below, may be useful for executing the tests.

Many other flags are intended to be helpful while debugging specific executables. These flags only serve to limit the tests executed. They do not add tests. Two complementaryflags are the "-its" flag, which specifies only the ITS tests, and the "-mits" flag, which specifies only the MITS tests. Another pair of complementary flags are "-cad" and "-cg". The flags "-acc", "-cyl", and "-tig" specify the ACCEPT, CYLTRAN, and TIGER codes, respectively, all of which are included by the "-cg" flag. The flags "-pcodes" and "-mcodes" specify the PCODES and EBFIELDS tests, respectively. The "-std" flag specifies the standard code tests (meaning non-PCODES and non-EBFIELDS tests). The flags "-static" and "-dynamic" allow the user to determine the MPI mode to be used for all tests. By default, the MPI mode is set for each test as part of the test suite.

The flags for limiting the number of tests to be executed not only state which tests will be run but also imply that other tests will not be run. The "-its" flag states that ITS tests will be run, and implies that MITS tests will not be run. The "-cad" flag implies that CG tests will not be run, but this may be overruled by explicit use of another flag. Using the "-cad" and "-cg" flags will resultinalltestsbeingrun.A moreuseful combinationwouldbetoinvokethe"-cad"and"-acc" flags together, to include both subsets of tests. The "-pcodes", "-mcodes", and "-std" may also be

t
i
n
g
o
f

I
T
S

Table 10. Flags available with the “tests.pl” ITS testing script

Flag	**Effect of the Flag**
help	Displays the list of valid flags and their functions
install	Tests are drawn from the installation suite
commit	Tests are drawn from the commit suite
regress	Tests are drawn from the regression suite
its	The ITS tests will be run
mits	The MITS tests will be run
cad	The CAD tests will be run
cg	The CG tests will be run
acc	The ACCEPT tests will be run
cyl	The CYLTRAN tests will be run
tig	The TIGER tests will be run
pcodes	The PCODES tests will be run
mcodes	The EBFIELDS tests will be run
std	The standardcode tests will be run
pff	Only PFF tests will be run
notpff	No PFF tests will be run
rng1	The RNG1 tests will be run
rng2	The RNG2 tests will be run
rng3	The RNG3 tests will be run
serial	The tests will be run in serial
mpi	The tests will be run in parallel
static	The MPI mode will be static for all tests
dynamic	The MPI mode will be dynamic for all tests
mpicmd	Sets the command used to launch parallel calculations
sercmd	Sets the command used to launch serial calculations
platform	Set mpicmd and sercmd for a specific platform
compile	The user can specify the compile flags
pre	Tests will be created in $HOME/tmp, but not run
post	Tests that have been run will be post processed
check	Serial and parallel archive outputs will be compared
checknew	New outputs will be compared with archived outputs

list	The names of active tests will be listed
t=testcase	Runs a specific test case only
cover	Generate code coverage instrumentation and statistics
numdiff	Uses numerical differencing with tolerances
rel	Specifies the relative difference cutoff
abs	Specifies the absolute difference cutoff

used in any combination to invoke independent subsets of tests. The flag “-t=testcase” will run a specific test case (e.g., “tests.pl t=itsaccm1 3”).

The “-mpicmd”, “-sercmd”, and “-platform” flags may be set equal to some text string. The “-mpicmd” flag allows the user to specify the command to be used for launching parallel calcu

lations (examples: “-mpicmd=‘mpirun -machinefile $HOME/machines -np 5’ ” will launch cad calculations as “mpirun -machinefile $HOME/machines -np5its.x prmfile”, and “-mpicmd=‘yod -sz5’”willlaunchcg calculationsas“yod-sz5its.xmdatoutput”).Thetextstringthatonesets mpicmd equal to should be the string that a user would enter on the command line for launching the calculation including the specification of the number of processors (all MPI test benchmarks use 5 processors) and excluding the ITS execution command itself. The same is true of the “sercmd”flagforserial calculations.Thisislesslikelytoberequired,butmaybeneededtorequest a single processor on a parallel platform (example: “-sercmd=‘yod -sz 1’ ” will launch cad calculationsas“yod-sz1its.x prmfile”). The “-platform”flagmaybeusedtoselectfromasetof platforms for which the mpicmd and sercmd flags are known. Available platform options are

crater, pegasus, redstorm, white, and q. If the mpicmd or sercmd flag is used, the platform flag will be ignored.

The “-compile” flag allows the user to specify the compiler flag to be used with the tests. By default,the testswillberunwiththe-O3flag.The compilerflag shouldbe specifiedbysettingthe compile flag equal to the desired compiler flag (e.g., “-compile=-g” or “-compile=-O2”). The only flags currently availableare-g,-w,-O,-O1,-O2,-O3,and-O4.Onlyasingleflagmaybe specified.

The “-check” flag compares the serial output files in the Output directory with the parallel output files in the OutputMPI directory. The “-list” flag will simply list the tests that have been set to be executed. This can be useful for evaluating which tests are specified by a given set of flags. (The “-post” flag can be used to eliminate listing of “cleanup” tests, which merely indicate that a new executable must be compiled.)

The “-checknew” flag compares existing new output files in the NewOutput or NewOutputMPI directory with the archive output files. This flag can be useful to quickly evaluate whether differences observed when the new outputs were generated were due to minor numerical differences. We recommend that tests first be run with a simple difftest. If there are tests that fail the simple diff, the “-checknew” flag can be used with the “-numdiff”, “-rel”, and “-abs” flags to determine what level of numerical difference exists in the test failures. Small numerical differences are not unusual in outputs generated on different platforms and may be acceptable. When using the “-checknew” flag, the group of tests selected should be limited to those that have been newly generated, but a subset of those tests can also be selected. Individual tests can be examined with the “-t” flag specification.

As briefly mentioned earlier,the “runcalcs.pl” script can be used to execute tests that have been prepared using the “-pre” flag. This script accepts only a few flags. Aflag must be set specifying aninputfile,suchas “-file=example”.Thisinputfileshouldlisttheteststobeexecuted,bylisting the names of the directories in which the tests have been set up. This can be accomplished by simply listing the contents of the $HOME/tmp directory, and redirecting the list into a file, such as “ls $HOME/tmp > example”. The user may need to edit this file to remove any extraneous information.

Theruncalcs.plscriptwillrunserialtestsbydefault,butthe“-mpi”flagmaybeusedtospecify that parallel tests must be run. The runcalcs.pl script will accept flags for mpicmd and sercmd, with the same behavior as described for the tests.pl script.

By default the “diff”-erence in comparing the expected output to the actual output is strictly character comparison. The “-numdiff” flag causes the diffto ignore absolute and relative numerical differences of less than specified tolerances. This avoids identifying acceptable

variances in output due to numerical differences in compiler and/or processor floating point calculations. The numericaldifferencingtoolis includedintheITS distributionintheTools/diffUtilsdirectory.To build the tool, run “configure” and “make”. Copy diffUtils/src/diffto your $HOME/bin directory. The “-abs” flag allows the user to specify the cutoff for absolute differences, below which

differences will be suppressed. The default absolute tolerance is 1e-18. The “-rel” flag allows the user to specify the cutofffor numerical differences, below which differences will be suppressed. The defaultrelative toleranceis 1e-8.

The “-cover” flag generates code instrumentationin the executable, and instrumentation output when the code is executed. These are then collated using Intel code coverage utilities. Typically, this would be run for all test cases at once: “tests.pl -cover”. This generates complete code coverage statistics for all test cases. This is normally only performed by the developers. This option may only be used with the Intel compilers ifort and icc with versions 9.1.032 or above.

24.2 InstallationTesting

See the Installation section of the manual for a broader set of instructions on code installation. Before performing installation testing of ITS, it is recommended that the XGEN and CEPXS cross section generating codes be tested. Testing of those codes is described in the sections Running XGEN and Running CEPXS.

It is recommended that the user compile an enhanced version of the “diff” utility. This can becreatedby changingtotheTools/diffUtilsdirectory underITS,andrunning: “configure”and “make”. Once this is completed

successfully, copy diffUtils/src/diff to your $HOME/bin and ensure that the $HOME/bin directory is early in your search $PATH. This causes any numbers encountered with an absolute value less than 1e-18, or relative differences less than 1e-8 to be ignored. That is, extremely small numbers areignored, and only8significant digits are examined.

Most users are unlikely to be developing code modifications, so in this section we describe the steps required to test the installation of the code on a new platform. The tests should be executed using the base command “tests.pl -install”, but other flags are likely to be necessary. The specific flags used will depend upon the capabilities of the code being tested. Outside of Sandia, users will not have access to RNG3. Users without RNG3 should use the flags “-rng1 -rng2”. Only users who have purchased and installed the ACIS libraries will have the CAD capability. Users without the CAD capability should use the “-cg” flag. Users who do not have access to MPI libraries cannot test the parallel capability in ITS and should use the “-serial” flag. For MPI, the user will likely want to use the “-mpicmd” flag, or the “-pre”, runcalcs.pl, and “-post” capabilities described in the preceding section.

As the test script executes, a report will be written to the screen stating the number of lines in the differences from the expected outputs. If all tests report zero lines of difference, the tests have passed. If the script terminates with an error message,the tests have failed. If differences arereported, the user must inspect the differences (in files in the Diffs directory) to determine whether thesedifferences constituteacode failure. Oftenwhenportingthecodetoanewplatformornew compiler, insignificant differences will be reported. Insignificant differences may be large relative differences in very small values (-1E-19 instead of 2E-20) or small relative differences (1.218E+01 instead of 1.219E+01). Any difference in the number of random numbers used constitutes a test failure. Use the enhanced diffcapability mentioned above if too many rounding or small difference errors occur. This is invoked with the “-numdiff” flag.

If the test script exits with an error, the user should determine if an ohoh file is present in the RegTests directory, which may contain information about the cause of the failure. If there is no ohoh file, the user should inspect the test subdirectory of the $HOME/tmp directory. This directory may contain relevant information or allow the user to attempt to execute the test.

After the test script exits with an error, the user should execute “interrupt-cleanup” before attempting to run additional tests. When test failures are encountered, the user may wish to

f

I
T
S

attempt running the tests with the debug flag (“-compile=-g”) to determine whether compiler optimization is causing the failure.

A
.

R
u
n
n
i
n
g

X
G
E
N

Last Modified Date: 2008/04/17 21:39:02

A Running XGEN

This document contains outlines for running the XGEN code. Instructions are provided for running the code by using commands or by using scripts. Known platform dependencies of the code are discussed in the last section.

A.1 Running XGEN without Scripts

CHECKOUT: Do a “cvs checkout -P xgen” to acquire a copy of the code. If necessary, tar the directory and transfer it to the desired platform.

To run on any platform, either use the sendn script (described in the Running XGEN with Scripts section) or build an executable using the following commands.

MAKEFILE SETTINGS: In the directory xgen/Code, you must alter the Defines.mk settings to specify the PCODES definition or at least remove the OPTION1 placeholder for your build of the XGEN executable.

CONFIGURE: Execute “./configure”. If the platform and operating system are identified, but the proper config/mh-* and config/mt-* files are not present, then you must either identify the proper files to use or create the necessary files and alter the configure.in file accordingly. If the platform and operating

system are not identified, the config.sub file also must be altered (search for tflops as an example).

MAKE: If configure functions properly, a working Makefile will be produced. Execute this with the “make” command to produce the XGEN executable, xgen.x.

ATOMIC DATA: The code xgen.x will look for the atomic data file in the local directory in the Fortran unit9file. This fileisin the xgen directory under xgen/Code/XSdata. Dopplerbroadening data is located in the comp.dat file in the same directory. The LLNL data set, epdl97.all, must be acquired and placed in the xgen/Code/XSdata directory. Soft links may be created from the location of the files to one’s working directory as:

ln -s $HOME/xgen/Code/XSdata/x6.dat fort.9 ln -s $HOME/xgen/Code/XSdata/comp.dat comp.dat ln -s $HOME/xgen/Code/XSdata/epdl97.all epdl97.all

EXECUTION: The input and output files may be specified as command-line arguments. The first argument is the input file. The second argument is the output file (hence, there is no way to specify the output file without specifying the input file). If an input file is not specified, the file “xgen.inp” will be used. If this file does not exist, it will be created as an empty input deck, and thecodewillberunwith default settings.Ifanoutputfileisnot specified,thefile “xgen.out”will be used.

REGRESSION TESTS: Files for running regression tests are in xgen/Tests/RegTests. The tests can be run by executing “tests.pl”. The script will place cross section files in $HOME/cross3, outputandcross sectionfilesinthedirectory NewOutput,andresultsofdiffswiththerepository version of output and cross section files in the directory Diffs. If the LLNL data sets have not been acquired, the “-noLLNL” flag should be used.

A. Running XGEN

The default is to comparethe expected output to the actual output using strictly character comparison. The “-numdiff” flag causes the “diff”-erence to

ignore absolute andrelative numerical differences of less than “-abs=1e-18” and “-rel=1e-7”. This avoids identifying acceptable variances in output due to numerical differences in compiler and/or processor floating point calculations. The numericaldifferencingtoolis includedintheTools/diffUtilsdirectoryoftheITS distribution. To build the tool, run “configure” and “make”. The resulting executable should be copied from its/Tools/diffUtils/src/diffto your $HOME/bin directory.

A.2 Running XGEN with Scripts

ACVS checkoutmaybe performedon platformsthathavedirect accesstotheCVSrepository. On all other platforms, the directory of files must be tarred and transferred, and the directory options in the sendn script must always be used instead of the CVS options.

For working on platforms that do not have direct access to the repository, it is recommended that two copies of the repository checkout be set up: one copy to be used as a working directory in which code modifications and builds can be performed, and one unaltered copy that can be comparedwithto maintainarecordofchangesmadetotheworking version.The unalteredcopy should be give a name uniquely indicating the cvs version.

1. SCRIPTS: Use the command “cvs checkout -P xgen” to acquire a copy of the code and all associated files.

   (a) From the directory xgen/Scripts, copysendn and the nxCVS script to your $HOME/bin directory. Copy the files in the Subscripts directory to a $HOME/bin/Subscripts directory. sendn and the Subscripts are identical for ITS, XGEN, and CEPXS.

2. CUI FILE: Copy the xgen/Scripts/x.cui file to your working directory, and edit it for your specific problem. The sections of a cui file are:

   (a) DRIVER SCRIPT:ThescriptnxCVSis availableasa driverscript.Youmay substitutea customized script by including it in the first section of the cui file. Sendn will position the cui file, including the sections of the driver script. The driver script contains the commands necessary to build and execute the program, clean up after itself, and producerelevantresult informationinajobfile.(If necessary,itwillalso includeinthejob file information useful in determining the cause of a job failure).

   (b) DEFS: The PCODES preprocessor definition may be selected in the defs section of the

   cui file.
   There are several options for running the scripts that may be set in the defs section of
   the cui file. Compiler flags may be specified.

   Theusermayalsorequestan interactivescript.The interactivescriptshouldbeusedon systems that require job queuing or on systems that require cross compiling. The first

halfofthescriptwillbuildthe executable.Thefilesforrunningthejobwillbe locatedin $HOME/tmp/<jobname>. The executable is named “xgen.x”, the input file is “xdat”, and the output file should be named “output”. When the calculation is complete, the user may execute the “postproc” file in the $HOME/tmp/<jobname> directory. This will movethe fort.11cross sectionfileto $HOME/cross3directory,renameit according totheresponsethat wasgiventothe sendn script,andproduceajobfileinthedirectory from which the job was originally submitted.

A. Running XGEN

(c) DIFFS: Patches to be applied to the code should be inserted in the “diffs” portion of the xgen.cui file. These patches can be formatted as a cvs diff or as a directory diff. Diffs can be lifted out of job files from previous calculations. Alternatively, one can checkout a copy of the code, make modifications, and then generate a diffs file. On a platform with access to the CVS repository, one can use “cvs diff > diffs” from within the xgen/Code directory. On a platform without access to the CVS repository one can performadirectorydiff,butthismustbe performedfromwithinthe xgen/Codedirectory of the unaltered copy of the code, and it must use the -r and -b flags (in that order), such as “diff-r -b . $HOME/xgenaltered/Code> diffs”. Then, the diffs file can be used in the xgen.cui file. Multiple diff files (diffs from two different executions of the diff command) cannot be patched to a single cvs file.

Newfilescanbeaddedtoabuild.Withinthediffs section,alineofthe followingformat denotes the start of a new file:

New file: <Directory/Filename> Tobe includedinthe compilingandlinkingofthecode,thenewfilemustbereferenced in the Makefile.in list of sources. This change in the Makefile.in can also be included in the diffs section by making the desired change in a copy of Makefile.in and using the cvs diffprocedure described above.

(d) XDAT: The XGEN input should be included in the last section of the

cui file. Instructions for XGEN input are in Keywords for XGEN.

3. SENDN: Submit the job using the command “sendn x.cui <jobname>”. This script will prompt you for the following3piecesof information (and possibly the 4thdepending upon your responses to the first 3):

(a) CROSS SECTIONS: First, sendn will request the name of the cross section file <cross> to be created. Upon successful completion, the scripts will assign the given name to the fort.11 cross section file and place it at “$HOME/cross3/<cross>”.

(b) LOCAL COMPILE: Second, sendn will request the location of a checked-out copy of xgen that can be used for making the executable. Thus, you may use a version of the code that you have checked out and modified. You must give a complete pathname for the base xgen directory (e.g., /scratch/temporary/xgen). No files will be deleted from the make directory as a result of the calculation, but some files may be modified if requested in the diffs section of the cui file. Any new files that have been included in the directory (other than through the diffs section of the cui file)will not appear in the job file, but they may be used if the Makefile.in has been so modified (in which case, thejobfilewillshowthereferencetothenewfileasadifferenceinthe Makefile.in).

Be aware that any definitions specified in the “defs” section of the cui file and any modifications specified in the “diffs” section of the cui file will be applied to the files in this directory. Attempts to change the code definitions for a calculation using the same directory will not take effect unless “make clean” has been executed. Attempts to apply the diffs in a directory where the diffs have already been applied for a previous calculation will result in an error.

CVS COMPILE: To request a cvs checkout of the code, you may respond to this request with “none” (or press “Enter”). The cvs checkout will be performed in the $HOME/tmp/<jobname> directory and should not affect any other versions of the code.

A. Running XGEN

(c) DIRECTORYDIFF for LOCAL COMPILE: If your response to the

second request was a directory pathname, the script will request a local directory with which to perform a diff. The diffcommand will be used to compare the two directories and all subdirectories. The job file will not recordthe version number of either directory, therefore the user may not have enough information in the job file to duplicate a calculation unless the diffdirectory has a name corresponding to the tag used in the cvs checkout of the code. The directory name does appear in the job file.

VERSION for CVS COMPILE: If your response to the second request was “none”, the script will request the version for a cvs checkout. The command will be issued as “cvs checkout <options> xgen”. Theresponseto thisrequest willbe usedforthe <options> andanyvalidcvsflagsmaybeusedthatdonot containaslash,“/”. Examplesofvalid syntax for responses are: -D now, -D “March 28, 2001”, -D 4:00pm, -D “3 hours ago”, -D “2 fortnights ago”, -r 1, -r release-1.0, etc. Another valid response is to simply press “Enter” withno input, which willresultinthe checkoutofthe mostrecent versionof the code.

(d) CVSDIFFforLOCAL COMPILE:Ifyougaveapathnameformakingthe executablebut not for a directory diff, the script will request the version for a cvs diff. The syntax for responses to this request are the same as for specifying the cvs version for a checkout. The directory in which the executable is made will be compared with the repository using the cvs diff command. This version (that will appear in the job file) and the results of the cvs diff(that will also appear in the job file) can be used to reproduce a calculation.

CVSUPDATE/DIFFforCVS COMPILE:Ifyoudidnotgiveapathnameformakingthe executable, the script will request a version for a cvs update and diff. In this case, the scriptwill checkoutaversionofthecodeusingthe optionsinthethirdresponse,apply the “diffs” from the cui file, attempt to update to the version of the code specified in this response, and then compare the resulting code to the repository using the cvs diff command with the options specified in this response. The version requested here (that will appearinthejobfile)andtheresultsofthecvsdiff(thatwillalso appearinthejob file) canbe usedtoreproducea calculation.

For either of these options, if no option is specifiedfor the cvs diff, the option “-D now” will be used. The date and time of the calculation, which are recorded in the job file, andtheresultsofthecvsdiffcanbeusedtoreproducethe calculationatalaterdate.

A.3 Platform Dependencies

The following is a list of platforms on which XGEN has been successfully built and tested. Following the name of the platform is the system type and

operating system.

Crater (AMD Athlon, Linux)
Pegasus (64-bit Intel, Linux)
PC (MicrosoftVisual Studio)

See the Installation section for guidance on installing XGEN on a PC.

XGEN Code Options

Last Modified Date: 2006/10/06 19:49:57

B XGEN Code Options

Only one code option can be selected for compiling the XGEN codes. This option has been implemented as a preprocessor definition. It is necessary to choose between the standard cross section code and the PCODES version.

The code option must be specified in either the CUI file (if using the scripts to execute the code) or in the Defines.mk file (if building an executable). Refer to the documentation on Running XGEN for more details on applying definitions.

B.1 Preprocessor Definitions

PCODES (more ionization and relaxation)

C.
Summary
of XGEN
Keywords

Last Modified Date: 2007/03/22 18:49:21

C Summary of XGEN Keywords

Table C.1 contains a listing of keywords relevant to the XGEN code. The keywords are listed approximately in an order of importance. In addition, for each keywordthe default code behavior is listed. The default behavior will be employed by the code if the keywordis not found in the input deck. More detailed descriptions of the syntax, sub-keywords, and use of these keywords are contained in the XGEN Keywords section.

Table C.1. Summary of XGEN keywords and default settings

KEYWORD	**DEFAULT**
**** MATERIALS ****	
MATERIAL	AL
CONDUCTOR or NON-CONDUCTOR	see discussion under keywords

GAS	normal state (pure elements only) liquid/solid (compounds/mixture)
DENSITY	normal density (g/cc) (pure elements only)
DENSITY-RATIO	1.0
SUBSTEP	internal
**** ENERGY RANGE ****	
ENERGY	1.0 MeV
BELOW-1KEV	off
**** PHYSICS OPTIONS ****	
ELECTRON-GRID-LENGTH	64
DOPPLER	no Doppler
ELECTRON-ANGLE-BINS	33
STEP	8
**** OTHER OPTIONS ****	
ECHO	input echoing is on
INCLUDE-FILE	all data in single input file
PRINT-ALL	abbreviated cross-section tables are printed
TITLE	no title

Last Modified Date: 2008/05/14 15:55:20

D Keywords for XGEN

The input keywords must be specified in either the CUI file (if using the scripts to execute the code) or in the input file (if executing the code manually). Refer to the documentation on Running XGEN for more details on specifying the input file in the CUI file or otherwise. An overview of the keywords is available in the Summary of XGEN Keywords section.

D.1 Input Notation

Most keywords should be used once and not repeated in an input file. The exception to this is the MATERIAL keyword. Sub-keywords should be used once per use of their primary keyword.

Parameters are associated with the preceding keywordappearing on the same line. If parameters are omitted, they will be set to zero. Consideration should be given to the fact that in some situations this value is invalid and will trigger an error.

Comments may be inserted in the input deck. Anything

appearing to the right of an asterisk anywhere in the input deck will be treated as a comment and ignored by the code.

Input is not case sensitive, with one insignificant exception. Character input provided with the TITLE keywordwill appear in the output file exactly as provided, but the case will not affect the code in any way.

D.2 Keywords

1. **BELOW-1KEV**

   Syntax: BELOW-1KEV

   Default: Off

   If “BELOW-1KEV” is inserted in the input stream, energygrids for less than1keV will be generated from the LLNL Evaluated Photon Data Library (EPDL97).

   The new capability for attenuationof source photons below1 keVrequires the Livermore evaluated data library for photons, which is not distributed with ITS. It is available for download from the IAEA website at http://www-nds.iaea.org/epdl97/. ITS uses the EPDL97 in ENDF/B-VI format. This file should be located in the XGEN distribution at .../Code/XSdata/epdl97.all.

2. **DOPPLER**

   Syntax: DOPPLER

   Default: no Doppler

   If “DOPPLER” is inserted in the input stream, Compton profiles, elemental binding energies, and shell configurations will also be generated. Only single element materials are allowed with this option.

3. **ECHO**

   Syntax: ECHO0

D. Keywords fo

r
X
G
E
N

Default: Input is echoed to the output

If “ECHO 0” is inserted in the input stream, subsequent input will not be echoed. If “ECHO 1” is inserted in the input stream, subsequent input will be echoed to the output.

4. **ELECTRON-ANGLE-BINS**

Syntax:
ELECTRON-
ANGLE-
BINS
[parameter(1
)]
Example:
ELECTRON-
ANGLE-
BINS2
90 180
Default: 33
bins,
unevenly
distributed.

If one increases the number of substeps per step or decreases the step size, it may be necessary to use a finer binning structure for electron angular scattering to accurately represent the Goudsmit-Saunderson distribution.

This keyword must be followed by parameter(1) values. The angular binning begins with 0degrees. The user must specify in degrees where each bin ends and the next begins. The final bin must end at 180 degrees.

5. **ELECTRON-GRID-LENGTH**

Syntax: ELECTRON-GRID-LENGTH [parameter(1)]

Example: ELECTRON-GRID-LENGTH 80

Default: 64, which corresponds to 8 halvings (or 0.39%) of the maximum energy.

Regardlessofthevalue,thegridwillbetruncated below1keV.

parameter(1) must be a multiple of 8.
In problems requiring a broad energy range of electron transport, this keywordallows the
user to extend the energy range for which electron transport data is available. Alternatively,
iftheSTEPkeywordisusedtorefinethe electron energy-lossgrid,thiskeywordcanbeused
to preserve the energy range over which data is available.

6. **ENERGY**

Syntax: ENERGY [parameter(1)]

Example: ENERGY 2.5

Default: Maximum cross-section energy is 1.0 MeV

Maximum energy in MeV for which electron cross sections will be calculated. Values may range between 1.0 GeV and 1.0 keV.

7. **INCLUDE-FILE**

S
y
nt
a
x:
I
N
C
L
U
D
E
-
FI

LEEExample: INCLUDE-FILE filename.inc Default: All input

is
re
a
d
fr
o
m
a
si
n
gl
e
fil
e.

D
.
K
e
y
w
or
d
s
fo
r
X
G
E
N

This allows the user to include input parameters from other files.

8. **MATERIAL**

Syntax: MATERIAL [keyword] [parameter(1)] ...

Example: MATERIALTA0.25C0.75

Example: MATERIALH

Default: Aluminum

Identifies unique material (pure element, compound or homogeneous mixture) and the appropriate weight fractions for which electron and photon cross sections are to be calculated. This keywordis repeated for each

unique material. Each element symbol must be followed byasinglerealnumber,parameter(1),whichistheweight fractionofthat constituent,except for pure elements where a blank for parameter(1) will default to 1.0. The weight fractions must sum to 1.0.

DATAarrays containing 100 atomic symbols (e.g.,TAfor tantalum) along with corresponding default values for the electrical characterization (conductor/non-conductor), mass density, and state (solid/liquid or gas) at normal pressure and temperature (zero ◦C and one atm) are included in the code to simplify the input for purematerials. The default properties of elementsareprovidedinTableD.2.To override these defaultsorto construct compound materials the following secondary keywords associated with this primary keywordmay be used.

(a) **CONDUCTOR/NON-CONDUCTOR**

Syntax: NON-CONDUCTOR
Default:ApureelementwithaZof1,2,7,8,9,10,17,18,35 ,36,53,54,85,or86 is a non-conductor; otherwise, the element is a conductor. Acompound/mixture is a non-conductor if any one of its constituent elements is a non-conductor by default; otherwise,itisa conductor.

The only collective effect in the ITS Monte Carlo model is the density-effect correction to the electronic stopping power. The value of this correction depends on whether the transport region is a conductor or non-conductor. The user may explicitly define any material to be a conductor or a non-conductor via the appropriate keyword. However, if a material so defined as a conductor consists of constituent elements, all of which are non-conductors by default (e.g., pure water), the material will be redefined to be a non-conductor, and the user will be so informed via a message in the output file.

(b) **GAS**

Syntax:

GAS

Default:

Normal

state

for

elements

an

d

l
i
q
u
i
d
/
s
o
l
i
d

f
o
r

c
o
m
p
o
u
n
d
s

T
h
i
s

k
e
y

w
o
r
d
i
s

u
s
e
d

t
o

s
p
e
c
i
f
y

t
h
a
t

t
h
i
s

m
a
t
e

r
i
a
l

i
s

i
n

a

g
a
s
e
o
u
s

s
t
a
t
e

a
t

n
o
r
m
a
l

p
r
e
s
sure and temperature. The material state is used in calculating the density effect contribution to the electronic stopping power.

(c) **DENSITY**

Syntax: DENSITY [parameter(1)]

Example: DENSITY 2.0

D
.
K
e
y
w
or
d
s
fo
r
X
G
E
N

Default: Normal density for elements – no default for compounds! Density (in g/cm^3)of the target material at normal pressure and temperature.

(d)

D
E
N
S
I
T
Y
-
R
A
T
I
O

Syntax:

DENSITY-RATIO

[parameter(1)]

Example:

DENSITY-RATIO

0.5

Default:

De

n
s
i
t
y

r
a
t
i
o

i
s

1
.
0

R
a
t
i
o

o
f

t
h
e

a
c
t
u
a
l

density to the density of the target

material at normal pressure and

temperature(usedin calculating densityeffect contributiontoelectronic stopping powers).

(e) **SUB**

STEP

Syntax:

SUBSTEP [parameter(1)]

Ex

a
m
p
l
e
:

S
U
B
S
T
E
P

1
0

D
e
f
a
u
l
t
:

C
a
l
c
u
l
a
t
e
d

internally as a function of atomic num

b
e
r

N
u
m
b
e
r

o
f

r
a
n
d
o
m

w
a
l
k

s
u
b
s
t
e
p
s

i
n
t

o

w
h
i
c
h

e
a
c
h

m
a
c
r
o
s
c
o
p
i
c

e
l
e
c
t
r
o
n

s
t
e
p

i
s

s
u
b
d
i

vided. The default values have been empirically determined; other values should not be used without careful consideration of their effects on the condensed history model.

9. **PRINT-ALL**

Syntax: PRINT-ALL

Default: Abbreviated cross-section tables will be printed

This keywordwill cause all except differential electron cross-sections to be printed out.

10. **STEP**

Syntax: STEP [parameter(1)] Example: STEP 12 Default: Successive electron energies arerelatedby$E_{i+1}=2^{-1/8}E_i$

[parameter(1)] determines the spacing of the electron energy grid and the size of the macroscopic electron steps. Successive energies arerelatedby$E_{i+1}=2^{-(1/[parameter(1)])}E_i$. The default value has been empirically determined; other values should not be used without careful consideration of their effects on the condensed history model.

11. **TITLE**

Syntax:
TITLE
Exampl
e:
TITLE
Adjoint
Dose
calculat
ion in Al

box in
Satellite
GPS-4
Default:
no title

This keywordsignals that the next line of input contains a character string that is a title of up to 80 columns that will be written to the output file.

D. Keywords for XGEN

Table D.2. List of available elements and d

default properties in XGEN

Z	Element	Symbol	Atomic Weight	Density (g/cm3)	State	Conductor
1	Hydrogen	H	1.008	0.08988E-3	Gas	N
2	Helium	He	4.0026	0.1785E-3	Gas	N
3	Lithium	Li	6.94	0.53	S/L	Y
4	Beryllium	Be	9.01218	1.85	S/L	Y
5	Boron	B	10.81	2.34	S/L	Y
6	Carbon	C	12.011	2.26	S/L	Y
7	Nitrogen	N	14.0067	1.2506E-3	Gas	N
8	Oxygen	O	15.9994	1.429E-3	Gas	N
9	Fluorine	F	18.9984	1.696E-3	Gas	N
10	Neon	Ne	20.17	0.89990E-3	Gas	N
11	Sodium	Na	22.9898	0.97	S/L	Y
12	Magnesium	Mg	24.305	1.74	S/L	Y
13	Aluminum	Al	26.9815	2.70	S/L	Y
14	Silicon	Si	28.086	2.33	S/L	Y
15	Phosphorus	P	30.9738	1.82	S/L	Y
16	Sulfur	S	32.06	2.07	S/L	Y
17	Chlorine	Cl	35.453	3.214E-3	Gas	N
18	Argon	Ar	39.948	1.7837E-3	Gas	N
19	Potassium	K	39.102	0.86	S/L	Y
20	Calcium	Ca	40.08	1.55	S/L	Y
21	Scandium	Sc	44.9559	3.00	S/L	Y
22	Titanium	Ti	47.90	4.51	S/L	Y
23	Vanadium	V	50.941	6.10	S/L	Y
24	Chromium	Cr	51.996	7.19	S/L	Y
25	Manganese	Mn	54.938	7.43	S/L	Y
26	Iron	Fe	55.847	7.86	S/L	Y
27	Cobalt	Co	58.9332	8.90	S/L	Y
28	Nickel	Ni	58.71	8.90	S/L	Y

29	Copper	Cu	63.546	8.96	S/L	Y
30	Zinc	Zn	65.37	7.14	S/L	Y
31	Gallium	Ga	69.72	5.91	S/L	Y
32	Germanium	Ge	72.59	5.32	S/L	Y
33	Arsenic	As	74.9216	5.72	S/L	Y
34	Selenium	Se	78.96	4.79	S/L	Y
35	Bromine	Br	79.904	7.59E-3	Gas	N
36	Krypton	Kr	83.80	3.733E-3	Gas	N
37	Rubidium	Rb	85.467	1.53	S/L	Y
38	Strontium	Sr	87.62	2.60	S/L	Y
39	Yttrium	Y	88.9059	4.47	S/L	Y
40	Zirconium	Zr	91.22	6.49	S/L	Y

D. Keywords for XGEN

Table D.2 (continued).

Z	Element	Symbol	Atomic Weight	Density (g/cm3)	State	Conductor
41	Niobium	Nb	92.9064	8.40	S/L	Y
42	Molybdenum	Mo	95.94	10.2	S/L	Y
43	Technetium	Tc	98.9062	11.5	S/L	Y
44	Ruthenium	Ru	101.07	12.2	S/L	Y
45	Rhodium	Rh	102.9055	12.4	S/L	Y
46	Palladium	Pd	106.4	12.0	S/L	Y
47	Silver	Ag	107.868	10.5	S/L	Y
48	Cadmium	Cd	112.4	8.65	S/L	Y
49	Indium	In	114.82	7.31	S/L	Y
50	Tin	Sn	118.69	7.30	S/L	Y
51	Antimony	Sb	121.75	6.62	S/L	Y
52	Tellurium	Te	127.6	6.24	S/L	Y
53	Iodine	I	126.9045	4.94	S/L	N
54	Xenon	Xe	131.3	5.887E-3	Gas	N
55	Cesium	Cs	132.9055	1.90	S/L	Y
56	Barium	Ba	137.34	3.50	S/L	Y
57	Lanthanum	La	138.9055	6.17	S/L	Y
58	Cerium	Ce	140.12	6.67	S/L	Y
59	Praeseodymium	Pr	140.9077	6.77	S/L	Y
60	Neodymium	Nd	144.24	7.00	S/L	Y
61	Promethium	Pm	145.0	7.22	S/L	Y
62	Samarium	Sm	150.4	7.54	S/L	Y
63	Europium	Eu	151.96	5.26	S/L	Y
64	Gadolinium	Gd	157.25	7.89	S/L	Y
65	Terbium	Tb	158.9254	8.27	S/L	Y
66	Dysprosium	Dy	162.50	8.54	S/L	Y
67	Holmium	Ho	164.9303	8.80	S/L	Y
68	Erbium	Er	167.26	9.05	S/L	Y
69	Thulium	Tm	168.9342	9.33	S/L	Y
70	Ytterbium	Yb	173.04	6.98	S/L	Y
71	Lutetium	Lu	174.97	9.84	S/L	Y
72	Hafnium	Hf	178.49	13.1	S/L	Y
73	Tantalum	Ta	180.947	16.6	S/L	Y
74	Tungsten	W	183.85	19.3	S/L	Y
75	Rhenium	Re	186.2	21.0	S/L	Y
76	Osmium	Os	190.2	22.6	S/L	Y
77	Iridium	Ir	192.22	22.5	S/L	Y
78	Platinum	Pt	195.09	21.4	S/L	Y
79	Gold	Au	196.9665	19.3	S/L	Y
80	Mercury	Hg	200.59	13.6	S/L	Y

D
.
K

eywordsforXGEN

TableD.2(continued).

Z	Element	Symbol	Atomic Weight	Density (g/cm3)	State	Conductor
81	Thallium	Tl	204.37	11.85	S/L	Y
82	Lead	Pb	207.2	11.4	S/L	Y
83	Bismuth	Bi	208.9806	9.8	S/L	Y
84	Polonium	Po	209.0	9.2	S/L	Y
85	Astatine	At	210.0	No default	S/L	N

86	Radon	Rn	222.0	9.73E-3	Gas	N
87	Francium	Fr	223.0	No default	S/L	Y
88	Radium	Ra	226.0	5.00	S/L	Y
89	Actinium	Ac	227.0	10.07	S/L	Y
90	Thorium	Th	232.0381	11.7	S/L	Y
91	Protactinium	Pa	231.0359	15.4	S/L	Y
92	Uranium	U	238.029	19.07	S/L	Y
93	Neptunium	Np	237.0482	19.5	S/L	Y
94	Plutonium	Pu	244.0	19.84	S/L	Y
95	Americium	Am	243.0	11.7	S/L	Y
96	Curium	Cm	247.0	13.51	S/L	Y
97	Berkelium	Bk	247.0	14.0	S/L	Y
98	Californium	Cf	251.0	No default	S/L	Y
99	Einsteinium	Es	254.0	No default	S/L	Y
100	Fermium	Fm	257.0	No default	S/L	Y

E

.

R

u

n

n

i

n

g

C

E

P

X

S

Last Modified Date: 2008/04/17 21:39:26

E Running CEPXS

This document contains outlines for running the CEPXS code. Instructions are provided for running the code by using commands or by using scripts. Known platform dependencies of the code are discussed in the last section.

E.1 Running CEPXS without Scripts

CHECKOUT: Do a “cvs checkout -P cepxs” to acquire a copy of the code. If necessary, tar the directory and transfer it to the desired platform.

To run on any platform, either use the sendn script (described in the Running CEPXS with Scripts section) or build an executable using the following commands.

CONFIGURE: In the directory cepxs/Code, execute "./configure". If the platform and operating system are identified, but the proper config/mh-* and config/mt-* files are not present, then you must either identify the proper files to use or create the necessary files and alter the configure.in file accordingly. If the platform and operating system are not identified, the config.sub file also must be altered (search for tflops as an example).

MAKE: If configure functions properly, a working Makefile will be produced. Execute this with the "make" command to produce the CEPXS executable, xcepxs.

ATOMIC DATA: The code xcepxs will look for the atomic data files in the local directory. These files are in the cepxs directory under cepxs/Code/XSdata. Soft links may be created from the location of the atomic data files to one's working directory as:

ln -s $HOME/cepxs/Code/XSdata/elec3.dat elec3.dat

REGRESSION TESTS:Filesforrunningregression testsareincepxs/Tests/RegTests.The"runregtests" script will make an executable, run each problem, place output files in the directory NewOutput, and place results of diffs with the repository version of output files in the directory Diffs.

E.2 Running CEPXS with Scripts

ACVS checkoutmaybe performedon platformsthathavedirect accesstotheCVSrepository. On all other platforms, the directory of files must be tarred and transferred, and the directory options in the sendn script must always be used instead of the CVS options.

For working on platforms that do not have direct access to the repository, it is recommended that two copies of the repository checkout be set up: one copy to be used as a working directory in which code modifications and builds can be performed, and one unaltered copy that can be comparedwithto maintainarecordofchangesmadetotheworking version.The unalteredcopy should be give a name uniquely indicating the cvs version.

1. SCRIPTS: Use the command "cvs checkout -P cepxs" to acquire a copy of the code and all associated files.

E
.

R
u
n
n
i

n
g

C
E
P
X
S

(a) From the directory cepxs/Scripts, copy sendn and the ncCVS script to your $HOME/bin directory. Copy the files in the Subscripts directory to a $HOME/bin/Subscripts directory. sendn and the Subscripts are identical for ITS, XGEN, and CEPXS.

2. CUI FILE: Copy the cepxs/Scripts/cepxs.cui file to your working directory, and edit it for your specificproblem. The sectionsofa cui file are:

(a) DRIVER SCRIPT:The script ncCVSis availableasa driver script.Youmay substitutea customized script by including it in the first section of the cui file. Sendn will position the cui file, including the sections of the driver script. The driver script contains the commands necessary to build and execute the program, clean up after itself, and producerelevantresult informationinajobfile.(If necessary,itwillalso includeinthejob file information useful in determining the cause of a job failure).

(b) DEFS: There are currently no preprocessor definitions to be set for CEPXS. There are several options for running the scripts that may be set in the defs section of the cui file. Compiler flags may be specified. Theusermayalsorequestan interactivescript.The interactivescriptshouldbeusedon systems that require job queuing or on systems that require cross compiling. The first halfofthescriptwillbuildthe executable.Thefilesforrunningthejobwillbe locatedin $HOME/tmp/<jobname>. The executable is named “xcepxs”. When the calculation is complete, the user may execute the “postproc” file in the $HOME/tmp/<jobname> directory. This will move the fort.11 cross section file to $HOME/crossm directory, renameit accordingtotheresponsethatwasgiventothe sendn script,andproducea job file in the directory from which the job was originally submitted.

(c) DIFFS: Patches to be applied to the code should be inserted in the “diffs” portion of the cepxs.cui file. These patches can be formatted as a cvs diff or as a directory diff. Diffs can be lifted out of job files from previous calculations. Alternatively, one can checkout a copy of the code, make modifications, and then generate a diffs file. On a platform with access to the CVS repository, one can use “cvs diff > diffs” from within the cepxs/Code directory. On a platform without access to the CVS repository one can perform a directory diff, but this must be performed from within the cepxs/Code directory of the unaltered copy of the code, and it must use the -r and -b flags (in that order), such as “diff-r -b . $HOME/cepxsaltered/Code> diffs”. Then, the diffs file can be used in the cepxs.cui file. Multiple difffiles (diffs from two different executions of the diffcommand) cannot be patched to a single cvs file. Newfilescanbeaddedtoabuild.Withinthediffs section,alineofthe followingformat denotes the start of a new file: New file: <Directory/Filename> Tobe includedinthe compilingandlinkingofthecode,thenewfilemustbereferenced in the Makefile.in

list of sources. This change in the Makefile.in can also be included in the diffs section by making the desired change in a copy of Makefile.in and using the cvs diffprocedure described above.

(d) CEPINP: The CEPXS input should be included in the last section of the cui file. Instructions for CEPXS input are in Keywords for CEPXS.

3. SENDN: Submit the job using the command “sendn cepxs.cui <jobname>”. This script will prompt you for the following3piecesof information (and possibly the 4thdepending upon your responses to the first 3):

E
.

R
u
n
n
i
n
g

C
E
P
X
S

(a) CROSS SECTIONS: First, sendn will request the name of the cross section file <cross> to be created. Upon successful completion, the scripts will assign the given name to the fort.11 cross section file and place it at “$HOME/crossm/<cross>.11”.

(b) LOCAL COMPILE: Second, sendn will request the location of a checked-out copy of cepxsthat canbe usedfor makingthe executable. Thus,youmay usea versionofthe code that you have checked out and modified. You must give a complete pathname for the base cepxs directory (e.g., /scratch/temporary/cepxs). No files will be deleted from the make directory as a result of the calculation, but some files may be modified if requested in the diffs section of the cui file. Any new files that have been included in the directory (other than through the diffs section of the cui file)will not appear in the job file, but they may be used if the Makefile.in has been so modified (in which case, thejobfilewillshowthereferencetothenewfileasadifferenceinthe Makefile.in). Be aware that any definitions specified in the “defs” section of the cui file and any modifications specified in the “diffs” section of the cui file will be applied to the files in this directory. Attempts to change the code definitions for a calculation using the samedirectorywillnottakeeffectunless“makeclean”hasbeen executed. Attemptsto apply the diffs in a directory where the diffs have already been applied for a previous calculation will result in an error. CVS COMPILE: To request a cvs checkout of the code, you may respond to this request with “none” (or press “Enter”). The cvs checkout will be performed in the $HOME/tmp/<jobname> directory and should not affect any other versions of the code.

(c) DIRECTORYDIFF for LOCAL COMPILE: If your response to the

second request was a directory pathname, the script will request a local directory with which to perform a diff. This should be the unaltered copy that you made when you checked out. The diffcommand will be used to compare the two directories and all subdirectories. The job file will not recordthe version number of either directory, therefore the user may not have enough information in the job file to duplicate a calculation unless the diff directory has a name corresponding to the tag used in the cvs checkout of the code. The directory name does appear in the job file. VERSION for CVS COMPILE: If your response to the second request was “none”, the script will request the version for a cvs checkout. The command will be issued as “cvs checkout <options> cepxs”. The response to this request will be used for the <options> andanyvalidcvsflagsmaybeusedthatdonotcontainaslash,“/”. Examples of valid syntax for responses are: -D now, -D “March 28, 2001”, -D 4:00pm, -D “3 hours ago”, -D “2 fortnights ago”, -r 1, -r release-1.0, etc. Another valid response is to simplypress“Enter”withnoinput,whichwillresultinthe checkoutofthemostrecent version of the code.

(d) CVSDIFFforLOCAL COMPILE:Ifyougaveapathnameformakingthe executablebut not for a directory diff, the script will request the version for a cvs diff. The syntax for responses to this request are the same as for specifying the cvs version for a checkout. The directory in which the executable is made will be compared with the repository using the cvs diff command. This version (that will appear in the job file) and the results of the cvs diff(that will also appear in the job file) can be used to reproduce a calculation. CVSUPDATE/DIFFforCVS COMPILE:Ifyoudidnotgiveapathnameformakingthe executable, the script will request a version for a cvs update and diff. In this case, the

E
.

R
u
n
n
i
n
g

C
E
P
X
S

scriptwill checkoutaversionofthecodeusingthe optionsinthethirdresponse,apply the “diffs” from the cui file, attempt to update to the version of the code specified in this response, and then compare the resulting code to the repository using the cvs diff command with the options specified in this response. The version requested here (that will appearinthejobfile)andtheresultsofthecvsdiff(thatwillalso

appearinthejob file) canbe usedtoreproducea calculation.
For either of these options, if no option is specified for the cvs diff, the option “-D now” will be used. The date and time of the calculation, which are recorded in the job file, andtheresultsofthecvsdiffcanbeusedtoreproducethe calculationatalaterdate.

E.3 Platform Dependencies

The following is a list of platforms on which CEPXS has been successfully built and tested. Following the name of the platform is the system type and operating system.

Crater (AMD Athlon, Linux)
Pegasus (64-bit Intel, Linux)
PC (MicrosoftVisual Studio)
See the Installation section for guidance on installing CEPXS ona PC.

Summary of CEPXS Keywords

Last Modified Date: 2008/04/17 21:39:25

F Summary of CEPXS Keywords

TableF.3 contains a listingof keywordsforthe CEPXS codefor generatingcross sectionsto be used with the MITS code. The keywords are listed approximately in order of importance. In addition, for each keyword the default code behavior is listed. The default behavior will be employed by the code if the keywordis not found in the input deck. More detailed descriptions of the syntax, sub-keywords, and use of these keywords are contained in the CEPXS Keywords section.
F.
Summary
of CEPXS
Keywords

TableF.3. Summary of CEPXS keywords (for use with MITS) and default settings

KEYWORD	**DEFAULT**
MITS	ONELD
**** MATERIALS ****	
MATERIAL or MATNAM	required (repeated for each material)
**** ENERGY RANGE ****	
ENERGY	1.0 MeV

CUTOFF	1keV for photon sources 1% of source energy for electron sources
**** SPECIES/COUPLING ****	
ELECTRON-SOURCE or PHOTON-SOURCE	electron source, full-coupling with photons
NO-POSITRONS	positrons automatic if energy extends above 1.03 MeV and full-coupling
**** GROUP STRUCTURE ****	
ELECTRONS (or EGROUP)	50 logarithmic groups
PHOTONS (or PGROUP)	50 logarithmic groups
ANNIHILATION-LINE	automatic line if energy extends above 1.03 MeV
NO-LINES	annihilation line is automatic
LINES	relaxation lines are mixed with continuum
MONO-PHOTON	no source line groups
**** ALTERNATIVE PHYSICS MODELS ****	
ITS2P1	cross sections correspond to ITS 3.0
USCATand/or USCAT-ITS	Fokker-Planck scattering cosine = 0.95
NO-PCODE	ITS-PCODES ionization/relaxation physics
NO-SEC-ELEC-GLOBAL	secondary electrons generated
NO-COHERENT	coherent photon scattering included
NO-INCOH-BINDING	incoherent scattering includes binding effects
CSDA	restricted CSDA
KNOCKONS-WITH-PRIMARIES	secondaries w/o corresp. primary downscatter
LEGENDRE	Legendre order is 15
ELASTIC-LEGENDRE	Legendre order is 15
**** OTHER OPTIONS ****	
TITLE	no title
PRINT	cross section matrices not printed
PRINT-ALL	no “additional” information is printed
INCLUDE-FILE	all data in single input file
CONTRAST	compact fort.11 file
CEPXS-INFO-ONLY	cross sections are generated

G.
K
ey
w
or
ds
for
C
E
P
X
S

Last Modified Date: 2008/04/17 21:39:27

G Keywords for CEPXS

The input keywords must be specified in either the CUI file (if using the scripts to execute the code) or in the input file (if executing the code manually). Refer to the documentation on Running CEPXS for more details on specifying the input file in the CUI file or otherwise. An overview of the keywords is available in the Summary of CEPXS Keywords section.

G.1 Input Notation

Most keywords should be used once and not repeated in an input file. Exceptions to this are the MATERIAL, MATNAM, and MONO-PHOTON keywords. Most subkeywords should be used once per use of their primary keyword.

Comments may be inserted in the input deck. Lines with an asterisk in the first column will be ignored by the code. It is important to note that asterisks inserted anywhere other than the first column may not cause words that follow to be ignored by the code.

Input is not case sensitive, with one exception. Character input provided with the TITLE keywordwill appear in the output file exactly as provided, but the case does not affect the code in any way.

G.2 Keywords

Transport Code Keywords

1. **ONELD**

   Syntax: ONELD

   Default: ONELD

This keywordspecifies that the cross sections from CEPXS will be formatted for ONELD.

2. **MITS**

   Syntax: MITS

   Default: ONELD

This keywordspecifies that the cross sections from CEPXS will be formatted for MITS.

3. **BFP** *(No*t *MITS*, *No*t *ONELD*)

Syntax: BFP

Default: The Boltzmann-Fokker-Planck approximation is not used.

This keyword specifies that the Boltzmann-Fokker-Planck approximation is to be used for electron cross sections. This approximation is automatically used if the MITS keyword is used.

G. Keywords for CEPXS

Additional Keywords

1. **ANNIHILATION-LINE**

Syntax: ANNIHILATION-LINE Default: Annihilation line is included if energy grid extends above 1.03 MeV, unless keywordNO-LINES is used. (See the LINES keywordfor explanation of line groups.)

Specifies that the annihilation line will be separated from the continuum, if 0.511 MeV is within the energy range. This keywordoverrides the NO-LINES keywordand overrides the requirement that the energy grid extend above 1.03 MeV.

Note: This keywordis incompatible with PGROUP.

2. **BCD** *(NOT CURRENTLY FUNCTIONAL)*

Syntax: BCD

This keywordspecifies the format needed for MCNP.

3. **CEPXS-INFO-ONLY**

Syntax: CEPXS-INFO-ONLY

Default: Cross sections are calculated.

This keywordspecifies that only range, mfp, and other such information is to be generated, but cross sections will not be calculated.

4. **CONTRAST** *(MITS*)

Syntax: CONTRAST Default: A compact fort.11 file is produced for MITS. (No fort.11 file is produced unless MITS is specified.)

An expanded fort.11 file is produced that can be compared with fort.11 files from other versions of CEPXS using the contrast program. This fort.11 file is still functional for MITS, but it has larger memory requirements.

5. **CONDENSED-HISTORY** *(No*t *MITS*)

Syntax: CONDENSED-HISTORY

Default:Adiamond energy-differencing schemeis used for the CSDA operator.

A second order backward differencing of the CSDA operator is used, and a condensed-history based approximation is used.

6. **CSDA**

Syntax: CSDA

Default: Restricted CSDA is used.

G. Keywords for CEPXS

The Continuous-Slowing-Down Approximation (CSDA) is used to characterize all electron energy loss.

7. **CSDLD** *(Not MITS*)

   Syntax: CSDLD
   Default:Adiamond energy-differencing schemeis used for the CSDA operator.

8. **CUTOFF**

   Syntax: CUTOFF [parameter(1)] Example: CUTOFF 0.01 Default: 1 keV for photon sources. One percent of the source energy for electron

   sources.

   Cross sections extend to the cutoff energy. This energy is the lower bound of the lowest energy group for both electrons and photons. [parameter(1)] is the cutoffenergy in MeV,i.e., it specifies the lower energy bound of the lowest-energy group to be generated. The cutoff energy cannotbe less than1keV.

9. **EGROUP**

   Syntax: EGROUP
   Default: Fifty logarithmic electron groups.

   The electrongroupstructureis obtainedfromthe
   ASCIIfile,“egroup”.An“egroup”fileis
   generated on each CEPXS run and may be used on a subsequent CEPXS run. The structure of the “egroup” file is not important unless the user wishes to create this file from scratch. On the first line of the file, the number of electron groups is specified. Each subsequent line of the file is associated with a group (in descending order of energy.) Each line specifies, in order, the top energy of the group, the mid-point energy of the group, and the bottom energy of the group, all in MeV. The code will function provided that the midpoint energies are within the energy group bounds, even if they are not exact mid-point values.

10. **ELASTIC-LEGENDRE**

    Syntax: ELASTIC-LEGENDRE [parameter(1)]
    Example: ELASTIC-LEGENDRE5
    Default: 15, or Legendre order of cross sections set with LEGENDRE keyword.

This keywordspecifies the Legendreorder of electron elastic cross sections as [parameter(1)]. Thisorderisusedif between0andtheLegendreorderofthecross sections. Otherwise,itis reset to the Legendre order of the cross sections.

Note: This keywordshould be used with caution. For ONELD the extended transport correction is applied, and it is important to consider the quadrature set with which the cross sections will be used.

G. Keywords for CEPXS

11. **ELECTRONS**

Syntax: ELECTRONS Default: For MITS,50 logarithmic electrongroups. For other codes,upto50 automatic-structure electron groups.

If this primary keywordis used, one of the secondary keywords must be used to specify the electron group structure.

Note: This keywordis incompatible with EGROUP.

(a) **LINEA**R
Syntax: LINEAR [parameter(1)]
Example: LINEAR 40

Alinear electron group structure is created where [parameter(1)] is the number of electron groups.

(b) **LO**G
Syntax: LOG [parameter(1)]
Example: LOG 40

Alogarithmic electron group structure is created where [parameter(1)] is the number of electron groups.

(c) **USER**

Syntax: USER [parameter(1)]

[parameter(2)] ...
[parameter(parameter(1)+1)]
Example: USER 10
1.0 0.8 0.6 0.5 0.4 0.35 0.3 0.28 0.26 0.25

Auser-defined electron group structure is created where [parameter(1)] is the number of electron groups. The string that follows is composed of the lower boundary energies of each group arranged in decreasing order. The last energy must be the same as the cutoffenergy, i.e., [parameter(parameter(1)+1)] must equal CUTOFF [parameter(1)].

12. **ELECTRON-SOURCE**

Syntax: ELECTRON-SOURCE
Default: For MITS, a fully-coupled electron source. Asource must be specified for other codes.

The following sub-keywords are used to specify the coupling scheme for photon and electron cross sections.

(a) **NO-COUPLING**

Syntax: NO-COUPLING

Default: Full-coupling. This keywordindicates that photons will not be included in the calculation.

(b) **PARTIAL-COUPLIN**G

Syntax:PARTIAL-COUPLING
Default: Full-coupling.

This keywordindicates that electrons can produce photons but photons cannot produce electrons.

(c) **FULL-COUPLING**

Syntax: FULL-COUPLING
Default: Full-coupling.

This keywordindicates that electrons can produce photons and photons can produce electrons. Also, if the upper bound of the energy grid extends above 1.03 MeV,positrons will be included in the cross sections with full coupling to photons and partial coupling to electrons.

13. **ENERGY**

Syntax: ENERGY parameter(1)
Example: ENERGY 0.665
Default:1MeV

parameter(1) specifies the midpoint energy in MeV of the highest-energy group to be generated (but see description of energy grid generation for caveats). This energy cannot exceed 100 MeV.

14. **FIRST-ORDER** *(Not MITS*)

Syntax: FIRST-ORDER
Default:Adiamond energy-differencing schemeis used for the CSDA operator.

Cross sections associatedwiththeCSD operatorwillbe equivalenttoa first-orderdifferencing in energy.

15. **INCOH-BINDING**

Syntax: INCOH-BINDING Example: INCOH-BINDING Default: No incoherent binding for ONELD. Incoherent binding for MITS.

This keywordspecifies that incoherent binding effects will be included. Either of the keywords ITS2P1 or NO-INCOH-BINDING specifies that incoherent binding effects will not be included.

16. **INCLUDE-FILE**

S
y
n
t

a
x
:
I
N
C
L
U
D
E
-
F
I
L
E

E
x
a
m
p
l
e
:
I
N
C
L
U
D
E
-
F
I
L
E

fi
l
e
n
a
m
e
.
i
n
c

G.
K
ey
w
or
ds
for
C
E
P
X
S

Default: All input is read from a single file.

This allows the user to include input parameters from other files.

17. **ITS2P1**

Syntax: ITS2P1
Default: ITS version 3.0 physics.

Cross sections corresponding to ITS version 2.1 are used.

18. **KNOCKONS-WITH-PRIMARIES**

Syntax: KNOCKONS-WITH-PRIMARIES
Default: Knockon electrons are generated
without corresponding primary down-scatter.

Knockon electron production is associated with primary downscatter.

19. **LEGENDRE**

Syntax: LEGENDRE [parameter(1)]

Example: LEGENDRE7

Default: The Legendre order of the cross sections is 15.

[parameter(1)] is the Legendre order of the cross sections. If parameter(1) is zero or omitted, the default will be used.

20. **LINES**

Syntax: LINES [parameter(1)] [parameter(2)] ...

Example: LINES AL BE

Default: photon relaxation lines are mixed with the continuum.

Specifies that separate groups are generated for the photon relaxation radiation of the elements specified by the parameters. The binding energies of these elements will also fall on the photon group boundaries (the group structureis adjusted accordingly.) If no parameters are specified,relaxation linegroups willbe created for all elementsin theproblem.

Line groups have are assigned energy group widths corresponding to the continuum group with which they would otherwise be mixed and mid-point energies equal to the line energy. The cross sections for the line groups are calculated at the line energies.

Note: This keywordis incompatible with PGROUP.

21. **MATERIAL**

Syntax:
MATERIAL
[parameter(1
)]
parameter(2
) ...
Example:
MATERIALH
.1111
O.8889

G.

Default: None.At least oneMATERIALorMATNAM entryisrequired.

Specifies material composition. The mandatory string specifies either the name of a hard-
wired material or the chemical names of the constituent elements and their weight fractions. In the latter case, the first parameter in the string is mandatory and is the chemical name of an element in the material. The second parameter is the weight fraction of the element in the material. For single element materials, the weight fractionis not needed (and will be in error if not equal to 1.0).

Additional parameters may specify other elements and their weight fractions. If necessary, this parameter list may be extended to other lines by terminating a line with a dash. The weight fractions of elements in a material must sum to 1.0.

The defaultpropertiesof elementsareprovidedinTableG.4.Thehardwired materials availablein CEPXSandthe defaultpropertiesof those materialsareprovidedinTableG.5.

(a) **CONDUCTOR** Syntax: CONDUCTOR Default: Default valuesareprovidedforsingle element materials.Acompound is set as a conductor unless any one of its constituent elements is a non-conductor. The only collective effect in the CEPXS model is the density-effect correction to the electronic stopping power. The value of this correction can depend on whether a material is a conductor or a non-conductor. Note: If a compound is specified as a conductor, but none of its constituent elements
are conductors, then the material is automatically reset to a non-conductor.
Note: This keywordis incompatible with ITS2P1.

(b) **DENSITY** Syntax: DENSITY [parameter(1)] Example: DENSITY 1.001 Default: For single element materials and hard-wired materials, defaults are

provided. For a user-defined composition, this keywordis required.

The density of the material is set to [parameter(1)] in g/cm3.

(c) **GAS**

Syntax: GAS

Default: Solid or Liquid.

This keywordspecifies that the materialisa gas.

(d) **NON-CONDUCTOR** Syntax: NON-CONDUCTOR Default: Default valuesareprovidedforsingle element materials.Acompound is set as a conductor unless any one of its constituent elements is a non-conductor. The only collective effect in the CEPXS model is the density-effect correction to the electronic stopping power. The value of this correction can depend on whether a material is a conductor or a non-conductor.

Note: This keywordis incompatible with ITS2P1.

G. Keywords for CEPXS

(e) **NO-SEC-ELEC**

Syntax: NO-SEC-ELEC

Default: Secondary electrons are generated. This keywordspecifies that secondary electrons are not generated in this material.

22. **MATNAM**

Syntax: MATNAM [parameter(1)]
[parameter(2)] parameter(3) ...

Example: MATNAMWATER
H0.1111O0.8889

Default: None.At least oneMATERIALorMATNAM entryisrequired.

This keyword is an alternative to the MATERIAL keyword. It allows the user to assign a material name to any material whether an element,a hard-wired material, ora user-defined composition. The code will use the first six characters of the user-assigned material name, [parameter(1)]. This user-assigned name will only appear in the CEPXS output file, cepout, and is not available for use in the generated cross section files.

Unlike the MATERIAL keyword, the material composition is specified on the following line. However, all of the same secondary keywords apply.

23. **MONO-PHOTON**

Syntax: MONO-PHOTON [parameter(1)]

Example: MONO-PHOTON 1.311

Default: No source line groups.

Amono-energetic source line group is created at the energy specified by [parameter(1)] in MeV.Thiskeywordmustberepeatedforeachsourceline needed.Upto28linesareallowed.

If the source line is meant to be the highest energy in the calculation, it should be set exactly equal to the energy specified with the ENERGY keyword. This will result in the source line being the highest energy group, with the second highest energy group spanning the energy range between it and the third highest energy group. (ENERGY specified the mid-point energy of the highest energy group. This is still the case, but now the highest energy group is a line group.) If the source line is otherwise specified to fall within the highest energy group, it will still be the first group, but the second group will extend in energy both above and below the line source group and therefore, may not yield the desired results.

If the energy specified for the source line is higher or lower than the extent of the energy grid, it will be omitted.

Note: In MITS, source lines are not allowed in adjoint mode.

Note: For ONELD, PHOTON-SOURCE must be specified to use MONO-

PHOTON.

24. **NO-COHERENT**

Syntax: NO-COHERENT G. Keywords for CEPXS

Default: Coherent photon scattering will be included in the calculation.

This keyword deactivates the simulation of coherent photon scattering. This provides a functionality equivalent to the NO-COHERENT keywordin ITS.

25. **NO-INCOH-BINDING**

Syntax: NO-INCOH-BINDING

Default: Incoherent photon scattering will include binding effects.

This keyword causes incoherent photon scattering to be simulated in the Klein-Nishina or free-electron approximation. This provides a functionality equivalent to the NO-INCOHBINDING keywordin ITS.

26. **NO-LINES**

Syntax: NO-LINES

Default: An annihilation line is included if the energy grid extends above 1.03 MeV.

This keywordspecifies that no line groups will be included. The LINES keywordwill overridethiskeywordand causephotonrelaxationlinegroupstobe included.The ANNIHILATIONLINE keyword will override this keyword and cause an annihilation line group to be included.

27. **NO-PCODE**

Syntax: NO-PCODE Default: CEPXS essentially duplicates the ionization/relaxation physics of the ITS PCODES.

This keyword is used to select the ionization/relaxation physics of the standard (non-PCODES) ITS codes. This option allows only the K-shell to have non-zero binding energy. Asingle “average” energy for the fluorescencephotons and Auger electrons is used. This “average” energy is less than the K-shell binding energy.

28. **NO-POSITRONS**

Syntax: NO-POSITRONS
Default: Positrons are generated if the upper limit of the energy grid exceeds 1.03 MeV and cross sections are fully-coupled.

Positrons are not generated. Annihilation quanta are generated at the site of the pair interaction. The approximate treatment for positrons must be specified by one of the secondary keywords.

(a) **PEQE**

Syntax: PEQE
Default: None.

Pair secondaries areproduced and transported as electrons. That is, annihilation quanta are produced at the beginning rather than end of the positron path. This option providesareasonable estimatefordosebutan incorrectchargeprediction.

(b) **NO-PAI**R

Syntax: NO-PAIR
Default: None.

No pair secondaries areproduced. Energy and charge aredeposited locally. This option provides a reasonable estimate for charge but an incorrect dose prediction.

29. **NO-SEC-ELEC-GLOBAL**

Syntax: NO-SEC-ELEC-GLOBAL

Default: Secondary electrons are generated in all materials.

Neither photon-produced nor electron-produced secondary electrons are generated in any material.

30. **PGROUP**

Syntax: PGROUP

Default: Fifty logarithmic photon groups.

The photon group structure is obtained from the ASCII file, “pgroup”. A “pgroup” file is
generated on each CEPXS run and may be used on a subsequent CEPXS run. The structure of the “pgroup” file is not important unless the user wishes to create this file from scratch. On the first line of the file, the number of photon groups is specified. For MITS, the first line also contains the following (in order): the value of USCAT, a logical flag indicating whetherpositronsarepresent,andalogicalflag indicating whethertheNO-PAIR keywordis used. Each subsequent line of the file is associated with a group (in descending order of energy.) Each line specifies, in order, the top energy of the group in MeV, the midpoint energy of the group in MeV,the bottom energy of the groupin MeV,and an integer that is zero for a continuum group or one for a line group (relaxation, source, or annihilation). The code will function provided that the mid-point energies are within the energy group bounds, even if they are not exact mid-point values. For line groups, the energy bounds should be the energy bounds of the group in which the line would otherwise be included, and the mid-point energy should the line energy.

The line in the “pgroup” file following the energy group structure specifies the number of relaxation lines (not including annihilation and source lines) and the group that contains the annihilation line (zero if not applicable). On the following lines, one line gives the energy of arelaxationline followedbytheZvalueofthe associated elementandthe nextline contains three character fields (a3,a4,a4) that describe the relaxation line type.

31. **PHOTONS**

Syntax: PHOTONS Default: For MITS,50 logarithmic photongroups. For other codes,upto50 automatic-structure photon groups.

G.

K
ey
w
or
ds
for
C
E
P
X
S

If this primary keywordis used, one of the secondary keywords must be used to specify the
photon group structure.
Note: This keywordis incompatible with PGROUP.

(a) **LINEA**R
Syntax: LINEAR [parameter(1)]
Example: LINEAR 40

Alinear photon group structure is created where [parameter(1)] is the number of photon groups.

(b) **LO**G
Syntax: LOG [parameter(1)]
Example: LOG 40

Alogarithmic photon group structure is created where [parameter(1)] is the number of photon groups.

(c) **USER**

Syntax: USER [parameter(1)]

[parameter(2)] ...
[parameter(parameter(1)+1)]
Example: USER 10
1.0 0.8 0.6 0.5 0.4 0.35 0.3 0.28 0.26 0.25

Auser-defined photon group structure is created where [parameter(1)] is the number of photon groups. The stringthat follows is composed of the lower boundary energies of each group arranged in decreasing order. The last energy should be the same as the cutoffenergy, i.e., [parameter(parameter(1)+1)] must equal CUTOFF [parameter(1)].

32. **PHOTON-SOURCE**

Syntax: PHOTON-SOURCE Default:

For MITS only, transport is independent of the source specified and the cross sections are fully-coupledby default.A source mustbe specified for other codes.

The following sub-keywords are used to specify the coupling scheme for photon and electron cross sections.

(a) **NO-COUPLING**

Syntax: NO-COUPLING

Default: Full-coupling. This keywordindicates that electrons will not be included in the calculation.

(b) **PARTIAL-COUPLIN**G

Syntax:PARTIAL-COUPLING

Default: Full-coupling.

This keyword indicates that photons can produce electrons but electrons cannot produce photons.

G. Keywords for CEPXS

(c) **FULL-COUPLING**

Syntax: FULL-COUPLING

Default: Full-coupling.

This keywordindicates that photons can produce electrons and electrons can produce photons. Also, if the upper bound of the energy grid extends above 1.03 MeV,positrons will be included in the cross sections with full coupling to photons and partial coupling to electrons.

33. **PRINT**

Syntax: PRINT

Default: Multigroup Legendre cross section matrices are not

printed.

This keywordspecifies that the multigroup Legendrecross section matrices areto be printed to the CEPXS output file, CEPOUT.

(a) **LE**G

Syntax: LEG [parameter(1)]

Example: LEG3

Default: Only the zero Legendre order cross sections are printed.

This sub-keywordspecifies that multigroup Legendre cross sections are to be printed from Legendre order zero to Legendre order [parameter(1)].

(b) **ROWS**

Syntax: ROWS

Default: All cross section rows are printed.

This sub-keywordspecifies that multigroup Legendre cross sections are to be printed foronly selectedrowsofcross sections.Morespecifically,thetotalcross section,theenergy deposition cross section, etc., will be printed, but group-to-group scattering cross sections will not be printed except for self-scatter cross sections.

34. **PRINT-ALL**

Syntax: PRINT-ALL

Default: No “additional” information will be printed.

This keyword requests that additional information be printed out for each material. This information includes:

The collisional and radiative stopping powers and the density effect correction.

The collisional stopping power due to large-energy losses.

The exponent for the power-law extrapolation of the collisional stopping power below 10 keV.

35. **SECOND-ORDER** *(No*t *MITS*)

G.
K
ey
w

or
ds
for
C
E
P
X
S

Syntax: SECOND-ORDER Example: SECOND-ORDER
Default:Adiamond energy-differencing schemeis used for the CSDA operator.

Asecond order backwarddifferencing of the CSDA operator is used.

36. **TITLE**

Sy
nt
ax:
TI
TL
E
pa
ra
m
et
er(
1)
Ex
a
m
ple
:
TI
TL
E
3
M
eV
ele

ctr
on
s
on
gol
d
De
fa
ult:
No
titl
e.

This keywordspecifies the title of the calculation that will appear in output files. The title will be read from the following parameter(1) lines. If parameter(1) is zero or omitted, the following (one) line will be read as the title. The title cardmay contain any alphanumeric informationin columns1to72with whichthe user wishesto identifythe calculation.

37. **USCAT** *(MITS*)

Syntax: USCAT[parameter(1)] Example: USCAT0.99 Default: The cosine of the Fokker-Planck scattering angle is 0.95 for all energies and

materials.

This keywordsets the cosine of the Fokker-Planck scattering angle to [parameter(1)].

38. **USCAT-ITS** *(MITS*)

Syntax: USCAT-ITS Default: The cosine of the Fokker-Planck scattering angle is 0.95 for all energies and materials.

Energy-and material-dependent values of the Fokker-Planck scattering angle are internally calculated such that the electron mean free path is equal to the electron sub-step size in ITS. However, the automatic generation of Fokker-Planck scattering angles is over-ridden in cases where the scattering cosine would be less than the constant “uscat” parameter (which has a default value of 0.95 but may be changed with the

USCATkeyword).

G. Keywords for CEPXS

Table G. 4. List of available elements and default properties in CEPXS

Z	Element	Symbol	Atomic Weight	Density (g/cm3)	State	Conductor
1	Hydrogen	H	1.00794	0.08375E-3	Gas	N

2	Helium	He	4.002602	0.1663E-3	Gas	N
3	Lithium	Li	6.941	0.534	S/L	Y
4	Beryllium	Be	9.012182	1.848	S/L	Y
5	Boron	B	10.811	2.37	S/L	Y
6	Carbon	C	12.0107	1.70	S/L	Y
7	Nitrogen	N	14.0067	1.165E-3	Gas	N
8	Oxygen	O	15.9994	1.332E-3	Gas	N
9	Fluorine	F	18.9984	1.580E-3	Gas	N
10	Neon	Ne	20.1797	0.8385E-3	Gas	N
11	Sodium	Na	22.98977	0.971	S/L	Y
12	Magnesium	Mg	24.305	1.74	S/L	Y
13	Aluminum	Al	26.981538	2.699	S/L	Y
14	Silicon	Si	28.0855	2.33	S/L	Y
15	Phosphorus	P	30.97376	2.2	S/L	Y
16	Sulfur	S	32.066	2.0	S/L	Y
17	Chlorine	Cl	35.4527	2.995E-3	Gas	N
18	Argon	Ar	39.948	1.662E-3	Gas	N
19	Potassium	K	39.0983	0.862	S/L	Y
20	Calcium	Ca	40.078	1.55	S/L	Y
21	Scandium	Sc	44.95591	2.989	S/L	Y
22	Titanium	Ti	47.867	4.54	S/L	Y
23	Vanadium	V	50.9415	6.11	S/L	Y
24	Chromium	Cr	51.9961	7.18	S/L	Y
25	Manganese	Mn	54.93805	7.44	S/L	Y
26	Iron	Fe	55.845	7.874	S/L	Y
27	Cobalt	Co	58.9332	8.90	S/L	Y
28	Nickel	Ni	58.6934	8.902	S/L	Y
29	Copper	Cu	63.546	8.96	S/L	Y
30	Zinc	Zn	65.39	7.133	S/L	Y
31	Gallium	Ga	69.723	5.904	S/L	Y
32	Germanium	Ge	72.61	5.323	S/L	Y
33	Arsenic	As	74.9216	5.73	S/L	Y
34	Selenium	Se	78.96	4.5	S/L	Y
35	Bromine	Br	79.904	7.072E-3	Gas	N
36	Krypton	Kr	83.80	3.478E-3	Gas	N
37	Rubidium	Rb	85.4678	1.532	S/L	Y
38	Strontium	Sr	87.62	2.54	S/L	Y
39	Yttrium	Y	88.90585	4.469	S/L	Y
40	Zirconium	Zr	91.224	6.506	S/L	Y

G. Keyword

dsfor CEPXS

Table G.4 (continued).

Z	Element	Symbol	Atomic Weight	Density (g/cm3)	State	Conductor
41	Niobium	Nb	92.90638	8.57	S/L	Y
42	Molybdenum	Mo	95.94	10.22	S/L	Y
43	Technetium	Tc	98.0	11.5	S/L	Y
44	Ruthenium	Ru	101.07	12.41	S/L	Y
45	Rhodium	Rh	102.9055	12.41	S/L	Y
46	Palladium	Pd	106.42	12.02	S/L	Y
47	Silver	Ag	107.8682	10.5	S/L	Y
48	Cadmium	Cd	112.411	8.65	S/L	Y
49	Indium	In	114.818	7.31	S/L	Y
50	Tin	Sn	118.71	7.31	S/L	Y
51	Antimony	Sb	121.76	6.691	S/L	Y
52	Tellurium	Te	127.60	6.24	S/L	Y
53	Iodine	I	126.90447	4.93	S/L	N
54	Xenon	Xe	131.29	5.485E-3	Gas	N
55	Cesium	Cs	132.90545	1.873	S/L	Y
56	Barium	Ba	137.327	3.50	S/L	Y
57	Lanthanum	La	138.9055	6.154	S/L	Y
58	Cerium	Ce	140.116	6.657	S/L	Y
59	Praeseodymium	Pr	140.90765	6.71	S/L	Y
60	Neodymium	Nd	144.24	6.90	S/L	Y
61	Promethium	Pm	145.0	7.22	S/L	Y
62	Samarium	Sm	150.36	7.46	S/L	Y
63	Europium	Eu	151.964	5.243	S/L	Y

64	Gadolinium	Gd	157.25	7.90	S/L	Y
65	Terbium	Tb	158.92534	8.229	S/L	Y
66	Dysprosium	Dy	162.50	8.55	S/L	Y
67	Holmium	Ho	164.93032	8.795	S/L	Y
68	Erbium	Er	167.26	9.066	S/L	Y
69	Thulium	Tm	168.9342	9.321	S/L	Y
70	Ytterbium	Yb	173.04	6.73	S/L	Y
71	Lutetium	Lu	174.967	9.84	S/L	Y
72	Hafnium	Hf	178.49	13.31	S/L	Y
73	Tantalum	Ta	180.9479	16.65	S/L	Y
74	Tungsten	W	183.84	19.3	S/L	Y
75	Rhenium	Re	186.207	21.02	S/L	Y
76	Osmium	Os	190.23	22.57	S/L	Y
77	Iridium	Ir	192.217	22.42	S/L	Y
78	Platinum	Pt	195.078	21.02	S/L	Y
79	Gold	Au	196.96655	19.32	S/L	Y
80	Mercury	Hg	200.59	13.55	S/L	Y

G. Keywords for CEPXS

Tabl

e G. 4 (continued).

Z	Element	Symbol	Atomic Weight	Density (g/cm3)	State	Conductor
81	Thallium	Tl	204.3833	11.72	S/L	Y
82	Lead	Pb	207.2	11.35	S/L	Y
83	Bismuth	Bi	208.98038	9.747	S/L	Y
84	Polonium	Po	209.0	9.32	S/L	Y
85	Astatine	At	210.0	No default	S/L	N
86	Radon	Rn	222.0	9.066E-3	Gas	N
87	Francium	Fr	223.0	No default	S/L	Y
88	Radium	Ra	226.0	5.00	S/L	Y
89	Actinium	Ac	227.0	10.07	S/L	Y
90	Thorium	Th	232.0381	11.72	S/L	Y
91	Protactinium	Pa	231.03588	15.37	S/L	Y
92	Uranium	U	238.0289	18.95	S/L	Y
93	Neptunium	Np	237.0381	20.45	S/L	Y
94	Plutonium	Pu	244.0	19.82	S/L	Y
95	Americium	Am	243.0	13.671	S/L	Y
96	Curium	Cm	247.0	13.51	S/L	Y
97	Berkelium	Bk	247.0	14.78	S/L	Y
98	Californium	Cf	251.0	No default	S/L	Y
99	Einsteinium	Es	252.0	No default	S/L	Y
100	Fermium	Fm	257.0	No default	S/L	Y

G. Keywords for CEPXS

Table G.5. Listof availablematerials, compositions(inw/o),and densities in CEPXS

Material	Keyword	Composition	Density
Alumina	AL203	O.4708 AL .5292	3.99
Aluminum 6061	AL6061	MG .0080 AL .9725 SI .0040 TI .0015 CR .0015 MN .0015 FE .0070 CU .0015 ZN .0025	2.69
Aluminum 7075	AL7075	MG .0250 AL .9000 CR .0030 CU .0160 ZN .0560	2.80
Bakelite Phenolic	BAKELI	H.0645 C.7656 O.1699	1.45
Brass	BRASS	FE .0020 CU .6150 ZN .3520 PB .0310	8.40
Bronze	BRONZE	P.0030 CU .9470 SN .0050	8.86
CaF TLD	CAFTLD	F.4668 CA .51332 MN .01988	3.18
Kapton	KAPTON	H.0100 C.5600 N.1300 O.3000	1.42
Kevlar	KEVLAR	H.0428 C.6958 N.1166 O.1448	1.31
Kennertium	KENNER	CU .0200 W.9800	18.5
Kovar	KOVAR	MN .0030 FE .5370 CO .1700 NI .2900	7.90
Lithium Fluoride	LIF	LI .2675 F.7325	2.635
Mica	MICA	H.0051 O.4820 AL .2032 SI .2115 K.0982	2.80
Mylar	MYLAR	H.0519 C.6186 O.3295	1.38
Nylon	NYLON	H.0980 C.6368 N.1238 O.1414	1.13
Phenolic Linen	PHENOL	H.0800 C.7100 N.0600 O.1500	1.33
Polyethylene	POLYE	H.1437 C.8563	0.92
Polyimide PCB	POLYIM	H.0150 B.0120 C.2800 N.0350 O.3370 NA .0030 MG .0020 AL .0590 SI .1460 CA .1110	1.85
Polyurethane	POLYU	H.0402 C.5913 N.1398 O.2287	0.192
Silicone Dioxide	SIO2	O.5326 SI .4674	2.20
Silica Glass	GLASS	O.5257 MG .0006 AL .0048 SI .4475 K.0008 CA .0164 FE .0042	2.18
Solder	SOLDER	SN .6000 PB .4000	8.67
Stainless Steel 304	SS304	C.0008 SI .0100 P.0005 S.0003 CR .1900 MN .0200 FE .6784 NI .1000	8.03
Stainless Steel 410	SS410	C.0015 SI .0100 P.0004 S.0003 CR .1200 MN .0100 FE .8578	7.76
RTV630	RTV630	H.0490 C.2100 O.3250 AL .0010 SI .4130 CU .0020	1.28
Teflon	TEFLON	C.2401 F.7599	2.15
Water	WATER	H.1100 O.8900	1.0

H
.

Glossary of Terms

Last Modified Date: 2008/04/17 22:45:38

H GlossaryofTerms

ACCEPT:the three dimensional geometry capability of the ITS codes. **ACCEPTM**:Historically, the MCODES version of ACCEPT. **ACCEPTP**:the PCODES version of ACCEPT. **ACIS**:a particular formatof B-rep geometry developedby SpatialTechnology, Inc. **Adjoint**: the solution of the adjoint transport problem. This simulation method is similar to the physical processes going backwards in time. Particles begin at a radiation detector, and tallies are made at radiation source locations. **Biasing**: a distortion of natural analog processes to achieve variance reduction in certain desired output quantities.

Body: one of the geometric entities used to construct a zone. In CG, these are geometric primitives such as spheres, boxes, etc. In CAD, a body is not a geometric primitive, and a body and a zone refer to the same entity. **B-rep**: the boundary representation geometry description typically employed by CAD packages. **CAD**: computer aided design, also used to refer to the boundary representation (B-rep) geometries created by CAD. **CEPXS**: the Coupled Electron-Photon X-Section generation code used to create multi-group cross sections for ITS. **CG**:abbreviation for Combinatorial Geometry. The method of combining simple geometric entities, such as tori and spheres, into more complicated geometry descriptions. This method is also know as Constructive Solid Geometry (or CSG). **Code Options**: the set of valid preprocessor definitions used to select a version of the ITS codes to be compiled. **Code Zone**: unions in the CG description of an input zone are maintained internally as separate code zones for particle tracking purposes.

Collision Forcing: a biasing technique used for photons. Photons entering a zone are forcedto interactwitha specifiedprobability, whichmaybelargeror smallerthanthe natural interaction probability. The weight of the photon is modified accordingly. **Combinatorial Geometry**: see CG. **Constructive Solid Geometry**: see CG. **CSG**:see CG. **CUI**:the Combined User Inputs employed when running ITS with shell scripts.

H. Glossary of Terms

CutoffEnergy: the energy below which particles are not simulated in detail. The cutoff energies may be different for electrons and photons.

CYLTRAN: the axisymmetric cylindrical material geometry ITS codes, with fully three dimensional descriptions of particle trajectories.

CYLTRANM:Historically, the MCODES version of CYLTRAN.

CYLTRANP:the PCODES version of CYLTRAN.

Dose: energy absorbed by a material per unit mass, often used interchangeably with the term “energy deposition”, which is not normalized per unit mass.

ElectronTrapping: seeTrapping.

ETRAN:the one dimensional, single material electron/photon transport code developed by the National Institute of Standards andTechnology from which the ITS codes were developed.

Forward: the solution of the forwardtransport problem. This simulation method is similar to the physical processes going forwardin time. Particles begin at a radiation source, and tallies are made at radiation detector locations.

Group: a span of the particle energy domain within the multigroup approximation over which particles are assumed to interact with the same probabilities.

HYBRID Geometry: the geometry may be described by a combination of CAD and CG zones.

Input Body: see Body.

Input Zone: see Zone.

ITS:theIntegratedTIGERSeriescodes, sometimesreferringtoonlythe continuous-energy versions of the codes and not the MITS codes.

ITS-CAD:the ITS and MITS codes capable of tracking particles in three dimensional CAD geometries.

KERMA: Kinetic Energy Released to MAterial is the energy deposition calculated in the absence of electron transport with the assumption of electronic equilibrium. Various assumptionscanbemaderegardingthe radiativeenergythatwouldbeproducedbyelectrons. CEPXS assumes that radiative energy is not lost from the system and is locally deposited.

Keywords: the set of words that may be used in the input deck that the code keys upon to activate and deactivate code features.

MCODES: The ITS codes that were historically used for simulations involving electric and magnetic fields. Fields are now integrated into the standardcodes and activated with the EBFIELDS keyword.

MITS:the Multigroup Integrated TIGER Series codes.

H
.

G
I

Glossary of Terms

Multigroup: an approximation involving the discretization of the particle energy domain into groups. The approximation is only accurate if the cross sections do not vary significantly over each group.

Next Event Estimator:abiasing technique used for photons. Theprobability thataphoton will escape without interacting is used to record an escape tally, with the weight of the tally modified accordingly, regardless of whether the photon actually escapes in the particular simulation.

Overlay:atally structuresuperimposed uponazone for calculating the subzone distribution of energy and charge deposition and/or flux.

PCODES: the more elaborate ionization/relaxation model adapted from the SANDYL code.

Preprocessor: the compiler preprocessor used to apply the selection of the code version and include the necessary common block statements into the source code. The cpp program is used for preprocessing ITS.

Prmfile: the file containing CAD parameter input.

RNG:random number generator.

Russian Roulette: a biasing technique in which particles may be eliminated with some probability. If the particle is not eliminated from the simulation, the weight of the particle is modified accordingly.

Satfile: the file containing the CAD geometry in ACIS text format.

Scaling: a biasing technique based on scaling material cross sections (larger or smaller) and correspondingly modifying the weighting of any particle exposed to the scaled cross sections.

Sub-keyword: a secondary keywordthat will not be recognizedwithout the presence of the associated primary keyword.

Subzone:ageometryentity,usuallyasmall fractionofazone,inwhichenergyandcharge deposition and/or flux are tallied.

TIGER:the one dimensional geometry capability of the ITS codes.

TIGERP:the PCODES version of TIGER.

Trapping: a biasing technique used for electrons. Because electrons have a finite range withina material,itmaynotbepossibleforan electronto escapethe zoneor subzonethatit isin.Ifit cannot escape,itis “trapped.”Trappingmay neglecteffectsof secondary photons that can escape from the zone.

XGEN:the coupled electron-photon X-section GENeration code used to create continuous-energy cross sections for ITS.

Zone: a geometry entity that consists of a single material and density.

References

[1] M. J. Berger and S. M. Seltzer, ETRAN Monte Carlo code system for electron and photon transport through extended media, CCC-107, Radiation Shielding Information Center, Computer Code Collection, Oak Ridge National Laboratory, 1968.

[2] M. J. Berger, Monte Carlo calculation of the penetration and diffusion of fast charged particles, in *Method*s *i*n *Computationa*l *Physics*, edited by B. Adler, S. Fernbach, and M. Rotenberg, volume1, pages 135–215, Academic, NewYork, 1963.

[3] M. B. Emmett, The MORSE Monte Carlo radiation transport code system, Technical Report ORNL-4972, Oak Ridge National Laboratory, 1975.

[4] D.P. Sloan, A new multigroup Monte Carlo scattering algorithm suitable for neutral and charged-particle Boltzmann and Fokker-Planck calculations,Technical Report SAND83-7094, Sandia National Laboratories, 1983.

[5] J. A. Halbleib andW. H.Vandevender, EZTRAN–a user-oriented version of the ETRAN-15 electron-photon Monte Carlo technique, Technical Report SC-RR-71-0598, Sandia National Laboratories, 1971.

[6]J.A. HalbleibandW.H.Vandevender, EZTRAN2:A user-oriented versionofthe ETRAN18B electron-photon Monte Carlo technique,Technical Report SLA-73-0834, Sandia National Laboratories, 1973.

[7] J.A. Halbleib andW.H.Vandevender, Nuclear Science and Engineering **57**, 94 (1975).

[8] J.A. Halbleib andW.H.Vandevender, Nuclear Science and Engineering **61**, 288 (1976).

[9] J. A. Halbleib, Nuclear Science and Engineering **75**, 200 (1980).

[10]W. Guber,J. Nagel,R. Goldstein,P.S. Mettelman,andM.H. Kalos,Ageometric description technique suitable for computer analysis of both the nuclear and conventional vulnerability of armored military vehicles, Technical Report MAGI-6701, Mathematical Applications Group, Inc., 1967.

[11] E.A. Straker,J.W.H. Scott, andN.R. Byrn, The MORSE code with combinatorial geometry, Technical Report SAI-72-511-LJ (DNA 2860T), Science Applications, Inc., 1972.

[12] J. A. Halbleib and J. E. Morel, Nuclear Science and Engineering **70**, 219 (1979).

[13] J. A. Halbleib and J. E. Morel, CYLTRANP, Sandia National Laboratories, unpublished, 1981.

[14] H. M. Colbert, SANDYL:Acomputer code for calculating combined photon-electron transport in complex systems, Technical Report SLL-74-0012, Sandia National Laboratories, 1974.

[15] J.A. Halbleib,Sr. andW.H.Vandevender, Journalof Applied Physics **48**, 2312 (1977).

[16] L.F. Shampine,H.A.Watts, andS. Davenport, SIAM Rev. **18**, 376 (1976).

[17] K. L. Hiebert and L.F. Shampine, Implicitly defined output points for solutions of ODEs, Technical Report SAND80-0180, Sandia National Laboratories, 1980.

[18] J. A. Halbleib, ACCEPTM, Sandia National Laboratories, unpublished, 1981.

[19] J. A. Halbleib, Nuclear Science and Engineering **66**, 269 (1978).

[20] J. A. Halbleib, SPHEM, 1979, Sandia National Laboratories, unpublished.

[21] J.A. Halbleib andT.A. Mehlhorn, Nuclear Science and Engineering **92**, 338 (1986).

[22] T. A. Mehlhorn and J. A. Halbleib, Monte Carlo benchmark calculations of energy deposition by Electron/Photon showers up to1GeV, in *Proceedings of a Topical Meeting on Advances in Reactor Computations*, page 608, 1983, ISBN 0-89448-111-8.

[23] J. A. Halbleib, R.P. Kensek, G. D.Valdez, S. M. Seltzer, and M. J. Berger, IEEETransactions on Nuclear Science **39**, 1025 (1992).

[24] L. J. Lorence, Jr., R. P. Kensek, and J. A. Halbleib, Transactions of the American Nuclear Society **73**, 339 (1995).

[25] L. J. Lorence, R. P. Kensek, J. A. Halbleib, and J. E. Morel, IEEE Transactions on Nuclear Science **42**, 1895 (1995).

[26] J. E. Morel, L. J. Lorence, R.P. Kensek, J. A. Halbleib, and D.P. Sloan, Nuclear Science and Engineering **124**, 369 (1996).

[27] B. C. Franke et al., Adjoint charge deposition and CAD transport in ITS, in *Proceedings of the ANS International Meeting on Mathematical Methods for Nuclear Applications*, Salt Lake City, Utah, 2001.

[28] B. C. Franke et al., ITS version 5.0: The integrated TIGER series codes, Technical report, Sandia National Laboratories, 2002, unpublished draft.

[29] J. A. Halbleib and T. A. Mehlhorn, ITS: The Integrated TIGER Series of coupled Electron/Photon Monte Carlo transport codes, Technical Report SAND84-0573, Sandia National Laboratories, 1984.

[30] MacKichan Software, Inc., MacKichan software, http://www.mackichan.com, 2003.

[31] D. Price et al., Version management with CVS, http://www.cvshome.org, 2002.

[32] Amtec Engineering, Inc., Amtec Engineering home page, http://www.amtec.com, 2003.

[33] Computational Engineering International, CEI: Products, http://www.ceintl.com/products/ensight.html, 2005.

[34] SpatialTechnology Inc., 3D ACIS Modeler, http://www.spatial.com, 2005.

[35] J. A. Halbleib, J. Appl. Phys. **45**, 4103 (1974).

[36] J.A. Halbleib andW.H.Vandevender, IEEETrans. Nucl. Sci. **NS-22**, 2356 (1975).

[37] J. A. Halbleib, Bull. Am. Phys. Soc. **26**, 1062 (1981).

[38] J. S. Hendricks, Nuclear Science and Engineering **109**, 86 (1991).

[39] G. Marsaglia,A. Zaman, andW.W.Tsang, Statist.Prob. Lett. **9**, 35 (1990).

[40] F. James, Comput. Phys. Commun. **60**, 329 (1990).
[41]P.L'Ecuyer, INFORMSJ. on Computing **9**, 57 (1997).

[42] M. Matsumoto andT. Nishimura, ACMTransactions on Modeling and Computer Simulations (TOMACS) **8**,3(1998).

Index

DISTRIBUTION:

1 MS 1179
M. J. Crawford, 01341

1 MS 1179
B. C. Franke, 01341

1 MS 1179
R.P. Kensek, 01341

1 MS 1179
T.W. Laub, 01341

1 MS 1179
L. J. Lorence, 01341

1 MS 0899 Technical Library, 04536