The EGSnrc g application

Report PIRS-3100

E. Mainegra-Hing, D.W.O. Rogers, R.W. Townson, B.R.B. Walters, F. Tessier and I. Kawrakow

Chapter 1

Main Page

1.1 Introduction

This `EGSnrc` application calculates $\overline{g}$, the average kinetic energy fraction lost to radiative events for primary photon interactions (radiative loss fraction) or the radiative yield Y in the case of charged particle beams as they slow down. For photons, quantities such as kerma, K, collision kerma, K_{col}, energy-fluence-averaged mass energy-transfer coefficient, $\overline{\mu}_{tr}/\rho$, and mass energy-absorption coefficient, $\overline{\mu}_{en}/\rho$, are also calculated for the incident photons.

The g application was written by Iwan Kawrakow in the late 90's using the Mortran 3 language, well before a C/C++ interface was available. Important contributions and refinements were made by Dave Rogers in the early 2000's. Version 1.0 was released in January 2000 for the calculation of the average radiative fraction $\overline{g}$ or the radiative yield Y. Later on, in March 2002, the code was extended to compute total and collision kerma (version 1.1). In the summer of 2002 the calculation of average mass-energy transfer and absorption coefficients was added (version 1.2) and a bug corrected to properly account for fluorescent photons(version 1.3). The $\overline{g}$ value used during the 2003 update of the Canadian Co-60 air kerma primary standard was calculated with this code. It was first released as part of EGSnrc in 2004 and has since been maintained by the EGSnrc Monte Carlo group at the NRC and the Carleton Laboratory for Radiotherapy Physics (CLRP).

Motivated by the need to generate databases of mass-energy absorption coefficients for `EGSnrc` applications that allow calculating collision kerma, a photon-beam-only calculation (calculation type = 1) was implemented in April 2006. With this calculation type the efficiency of μ_{en} calculations is increased significantly. For instance, an efficiency increase of about a factor of 70 is obtained for a 600 keV calculation in air. At around the same time, the ability to run the calculation for more than one energy on a linear or a logarithmic energy grid was added. Since the code calculates μ_{en}/ρ and μ_{tr}/ρ values, databases for calculating collision and total kerma can be generated.

In January 2017, version 1.4 introduced refinements to the `AUSGAB` routine which combined with several changes in EGSnrc allow proper classification of sub-threshold events as either *radiative* or *non-radiative* during atomic relaxations following photoeffect, Compton scattering, and electron impact ionization (EII). According to this scheme, sub-threshold relaxation Auger electrons following photoeffect or Compton scattering are added to the kerma calculation (non-radiative) and sub-threshold relaxation fluorescent photons after EII are included as part of the energy lost to radiative events. An exhaustive discussion of these issues can be found in a detailed paper by Rogers and Townson (`Med.Phys.46 2019`). Version 1.5 fixes a bug in the calculation of $\overline{g}$ for both calculation types when the photon energy is sampled from an energy spectrum. It also provides a cleaner output and an option for more verbose output. The code has been extensively documented and a `Doxygen` based user's manual created.

In the current version (1.6) the original algorithm of a `calculation type 1` has been slightly adjusted to ensure μ_{en}/ρ is calculated to a user-requested relative precision σ and it has been made independent of the number of histories. The target statistical precision now defaults to 1%. The possibility to further optimize `calculation type 1` in the megavoltage energy range is introduced. The calculation progress messages during a type 1 calculation were made clearer and the documentation has been updated to reflect these changes.

1.2 Calculation method

A description of the physical quantities and the equations used for their calculation is presented in this section. Most of these definitions can be found in basic radiation physics textbooks, hence users more interested with practical aspects of using the code can skip to the next section Calculation types.

Although this app uses the variable g to refer indistinctively to both the radiative yield for charged particles, Y, and the radiative fraction for photons, $\overline{g}$, these quantities are conceptually different. Radiative yield is defined as the fraction of a charged particle's kinetic energy lost to radiation as it slows down in a medium, while radiative fraction is defined as the average fraction of a photon's kinetic energy transferred to charged particles and subsequently lost to radiation as these charged particles slow down in the medium. The radiative fraction $\overline{g}$ can be obtained from the integration of Y(E) over the secondary electron spectrum.

Note

> Particle production thresholds and transport cutt-offs *should be set to the 1 keV low energy limit in EGSnrc* (PCUT=AP=0.001 MeV, ECUT=AE=0.512MeV) to avoid potentially significant accuracy losses.

1.2.1 Radiation yield

Radiation yield Y(T_0) of a charged particle is defined as the fraction of its initial kinetic energy T_0 emitted as electromagnetic radiation through the slowing down process of that particle in a medium. For light charged particles, bremsstrahlung, fluorescence after impact ionization and in-flight annihilation (in the case of positrons) are the interactions responsible for the emission of electromagnetic radiation.

For the calculation of Y, the g application follows a number of charged particles with energy T_0 starting from the origin (0,0,0) with initial direction along the positive z-axis (0,0,1) as they travel across an infinite geometry filled with a user-defined medium. $\overline{E}_{\mathrm{rad}}$, the average energy radiated by these particles as they slow down in the medium is scored to compute Y using the expression

$$Y = \frac{\overline{E}_{\mathrm{rad}}}{T_0}. \tag{1.1}$$

In addition to Y, the application also outputs Y_{brems}, the radiative yield due to bremsstrahlung events to assess the approximation incurred when ignoring all other interactions as has been customarily done in the past.

1.2.1.1 Scoring algorithm for charged particle sources

- Charged particle transport starts at the origin of an infinite geometry filled with a user-defined medium and the charged particles are followed as they slow down in the medium until their energy falls below the transport cut-off energy ECUT.

- The average initial energy of charged particles is scored in $\overline{E}_{\mathrm{tot}}$. In the case of a monoenergetic source of energy T_0, $\overline{E}_{\mathrm{tot}} = T_0$, while for a spectrum $T(E)$, $\overline{E}_{\mathrm{tot}} = \overline{T}$, the average spectrum energy.

- Radiative photons (bremsstrahlung, annihilation photons, fluorescent photons after EII) are immediately discarded after their energy is scored as part of $\overline{E}_{\mathrm{rad}}$.

- Energy of sub-threshold fluorescent photons after EII are scored as part of $\overline{E}_{\mathrm{rad}}$.

1.2.2 Radiative fraction, mass energy transfer and absorption coefficients

A derivation of the equations used in the g application for the calculation of several physical quantities for photon beams is presented here. Expressions for monoenergetic beams are first derived, followed by a generalization to polyenergetic sources. A summary at the end of the section details the general scoring algorithm for photon sources.

1.2.2.1 Monoenergetic beams

The mass energy transfer coefficient, μ_{tr}/ρ for a monoenergetic photon beam of energy E can be calculated using

$$\left(\frac{\mu_{\mathrm{tr}}}{\rho}\right) = \frac{\overline{E}_{\mathrm{tr}}}{E} \cdot \left(\frac{\mu}{\rho}\right), \tag{1.2}$$

which follows from the `definition` of μ_{tr}/ρ, and where $\overline{E}_{\mathrm{tr}}$ is the average photon energy transferred to charged particles after photon interactions (photoelectric, Compton, and pair production), and μ/ρ is the mass attenuation coefficient for a photon of energy E.

The mass energy absorption coefficient for photons of energy E, μ_{en}/ρ, is defined as the non-radiative part of μ_{tr}/ρ and can be calculated using the expression

$$\left(\frac{\mu_{\mathrm{en}}}{\rho}\right) = (1 - \overline{g}) \cdot \left(\frac{\mu_{\mathrm{tr}}}{\rho}\right) = \frac{\overline{E}_{\mathrm{ab}}}{E} \cdot \left(\frac{\mu}{\rho}\right), \tag{1.3}$$

where $\overline{E}_{\mathrm{ab}}$ is the average fraction of $\overline{E}_{\mathrm{tr}}$ absorbed in the medium and $\overline{g}$ the average fraction of energy lost to radiative events.

$\overline{g}$ is obtained as the ratio of the average energy lost to radiative events as the electrons slow down in a medium, $\overline{\overline{E}}_{\mathrm{rad}}$, to $\overline{E}_{\mathrm{tr}}$:

$$\overline{g} = \frac{\overline{\overline{E}}_{\mathrm{rad}}}{\overline{E}_{\mathrm{tr}}}, \tag{1.4}$$

where

$$\overline{\overline{E}}_{\mathrm{rad}} = \overline{E}_{\mathrm{tr}} - \overline{E}_{\mathrm{ab}}. \tag{1.5}$$

The estimation of $\overline{\overline{E}}_{\mathrm{rad}}$ follows the approach described in section 1.2.1 above for obtaining radiation yield, Y, however averaging is over all secondary charged particles.

Once the coefficients are known, kerma per unit fluence is obtained as

$$\frac{K}{\phi} = E \cdot (\mu_{\mathrm{tr}}/\rho), \tag{1.6}$$

and similarly, collision kerma per unit fluence is given by

$$\frac{K_{\mathrm{col}}}{\phi} = E \cdot (\mu_{\mathrm{en}}/\rho). \tag{1.7}$$

1.2.2.2 Polyenergetic beams

In the case of a polyenergetic source of photons, the definition of $\overline{g}$ follows from the definition of kerma and collision kerma. Kerma per unit fluence is defined by

$$\frac{K}{\phi} = \int \phi(E) \cdot E \cdot \left[\frac{\mu_{\mathrm{tr}}(E)}{\rho} \right] \cdot dE / \int \phi(E) \cdot dE. \tag{1.8}$$

The right-hand side of Eq.(1.8) can be interpreted as the energy fluence weighted average of μ_{tr}/ρ times the fluence averaged spectrum energy or equivalently, as the fluence weighted average of the product of photon energy E and μ_{tr}/ρ:

$$\frac{K}{\phi} = \langle \mu_{\mathrm{tr}}/\rho \rangle_{\psi} \cdot \langle E \rangle_{\phi} = \langle E \cdot \mu_{\mathrm{tr}}/\rho \rangle_{\phi} . \tag{1.9}$$

Similarly, substituting μ_{tr} with μ_{en}, collision kerma per unit fluence is given by

$$\frac{K_{\mathrm{col}}}{\phi} = \langle \mu_{\mathrm{en}}/\rho \rangle_{\psi} \cdot \langle E \rangle_{\phi} = \langle E \cdot \mu_{\mathrm{en}}/\rho \rangle_{\phi} . \tag{1.10}$$

Eqs. (1.9) and (1.10) are generalizations of Eqs. (1.6) and (1.7) for polyenergetic sources. As in the case of monoenergetic beams, collision kerma and kerma are linked via the relationship

$$\frac{K_{\mathrm{col}}}{\phi} = (1 - \overline{g}) \frac{K}{\phi} = (1 - \overline{g}) \langle E \cdot \mu_{\mathrm{tr}}/\rho \rangle_{\phi} , \tag{1.11}$$

thus defining $\overline{g}$ as the fraction of energy transferred by all photons to electrons, lost to radiative events. With this definition the expression for $\overline{g}$ becomes

$$\overline{g} = 1 - \frac{\langle E \cdot \mu_{\mathrm{en}} \rangle_{\phi}}{\langle E \cdot \mu_{\mathrm{tr}} \rangle_{\phi}}, \tag{1.12}$$

which from Eqs. (1.2), (1.3) and (1.5) is equivalent to

$$\overline{g} = 1 - \frac{\langle \overline{E}_{\mathrm{en}} \cdot \mu \rangle_{\phi}}{\langle \overline{E}_{\mathrm{tr}} \cdot \mu \rangle_{\phi}} = \frac{\left\langle \overline{\overline{E}}_{\mathrm{rad}} \cdot \mu \right\rangle_{\phi}}{\langle \overline{E}_{\mathrm{tr}} \cdot \mu \rangle_{\phi}} \tag{1.13}$$

Eq. (1.13) generalizes the expression for $\overline{g}$ obtained in the monoenergetic case (Eq. 1.4) and is the expression used to compute $\overline{g}$.

The energy fluence weighted average $\langle \mu_{\mathrm{tr}}/\rho \rangle_{\psi}$ is computed using the expression

$$\langle \mu_{\mathrm{tr}}/\rho \rangle_{\psi} = \frac{\langle E \cdot \mu_{\mathrm{tr}}/\rho \rangle_{\phi}}{\langle E \rangle_{\phi}} = \frac{\langle \overline{E}_{\mathrm{tr}} \cdot \mu \rangle_{\phi}}{\langle E \rangle_{\phi}}, \tag{1.14}$$

and the energy fluence weighted average $\langle \mu_{\mathrm{en}}/\rho \rangle_{\psi}$ with

$$\langle \mu_{\mathrm{en}}/\rho \rangle_{\psi} = \frac{\langle E \cdot \mu_{\mathrm{en}}/\rho \rangle_{\phi}}{\langle E \rangle_{\phi}} = \frac{\langle \overline{E}_{\mathrm{en}} \cdot \mu \rangle_{\phi}}{\langle E \rangle_{\phi}}, \tag{1.15}$$

or alternatively using

$$\langle \mu_{\mathrm{en}}/\rho \rangle_{\psi} = (1 - \overline{g}) \langle \mu_{\mathrm{tr}}/\rho \rangle_{\psi} , \tag{1.16}$$

from which kerma and collision kerma per unit fluence are calculated using Eqs. (1.9) and (1.11) respectively.

1.2.2.3 Scoring algorithm for photon sources

- Initial photons are placed at the origin of an infinite geometry filled with a user-defined medium.

- All photons, primaries and secondaries, are discarded after an interaction

- The photon energy transferred to charged particles after these interactions is scored to estimate $\overline{E}_{\mathrm{tr}}$.

- After atomic relaxations only energy transferred to charged particles (Auger or Coster-Kronig) is scored as part of $\overline{E}_{\mathrm{tr}}$.

- Fluorescence photons from atomic relaxation events after electron impact ionization (EII) are classified as part of the radiative loss.

- Radiative photons (bremsstrahlung, annihilation photons, fluorescent photons after EII) are immediately discarded after their energy is scored as part of $\overline{\overline{E}}_{\mathrm{rad}}$.

- Electrons are followed as they slow down in the medium until their energy falls below the transport cut-off energy ECUT.

- Special attention is paid to sub-threshold relaxation particles:

  - Energy of sub-threshold Auger electrons after Compton or photoeffect is scored as part of $\overline{E}_{\mathrm{tr}}$.

  - Energy of sub-threshold fluorescent photons after EII are scored as part of $\overline{\overline{E}}_{\mathrm{rad}}$.

  - Energy of vacancies below an energy defined by the internal parameter $RELAXCUTOFF, set to 1 keV by default, is treated as if deposited by Auger electrons. This is an approximation which is reasonable accurate. See the paper by Rogers and Townson (`Med.Phys.46 2019`) for a detailed discussion.

1.3 Calculation types

In its default implementation, the g code loops over a fixed number of histories for a source of *photons* or *charged particles* sampling the initial particle parameters and invoking the electromagnetic shower routine. This calculation can be explicitly requested by setting the `calculation type` input to 0. Collision kerma is calculated using Eq.(1.10) and the energy fluence averaged mass energy absorption coefficient $\langle \mu_{\mathrm{en}}/\rho \rangle_\psi$ is calculated using Eq.(1.15)

A `calculation type` 1, for use with photon sources only, is tailored to the efficient calculation of the mass energy absorption coefficient $\langle \mu_{\mathrm{en}}/\rho \rangle_\psi$. In this mode the g application takes advantage of the faster convergence of μ_{tr} and 1- $\overline{g}$ (when $\overline{g}$ is small) to a desired statistical precision and collision kerma is calculated with Eq.(1.11) and the energy fluence averaged mass energy absorption coefficient $\langle \mu_{\mathrm{en}}/\rho \rangle_\psi$ is calculated using Eq.(1.16). The calculation finishes as soon as a user-requested relative uncertainty σ is reached, which by default is set to 0.01 (1%). For this calculation type the `number of histories` input entry is ignored. By default μ_{tr}/ρ is calculated to a precision of $\sigma/\sqrt{4/3}$ and 1- $\overline{g}$ of $\sigma/2$, respectively, ensuring the precision in μ_{en}/ρ will be σ. This approach can be modified for instance to speed up the calculation at higher energies as described in section Speeding up calculations above 1 MeV .

The advantage of using a `calculation type` 1, tailored to the efficient computation of μ_{en}/ρ, can be observed in Figure 1.1 where the efficiency gain of a `calculation type` 1 over type 0 as function of photon energy in air is quantitatively shown for the original and the current implementation. The main difference is that the current implementation guarantees that the uncertainty in μ_{en}/ρ is equal to or less than the user-requested value σ, while the original implementation computed μ_{en}/ρ to a statistical precision of 1.1 σ since μ_{tr}/ρ was calculated to the user-requested precision σ and 1- $\overline{g}$ to $\sigma/2$. Calculation type 1 can be 20 times more efficient in the kilovoltage range and up to 80 times more efficient around 1 MeV. The efficiency gains decrease abruptly above the pair production threshold.

The inputs for a `calculation type` can be anywhere in the input file and are not required to be placed between delimiters as is customary in most EGSnrc applications. An example of a calculation type 1 input block that should
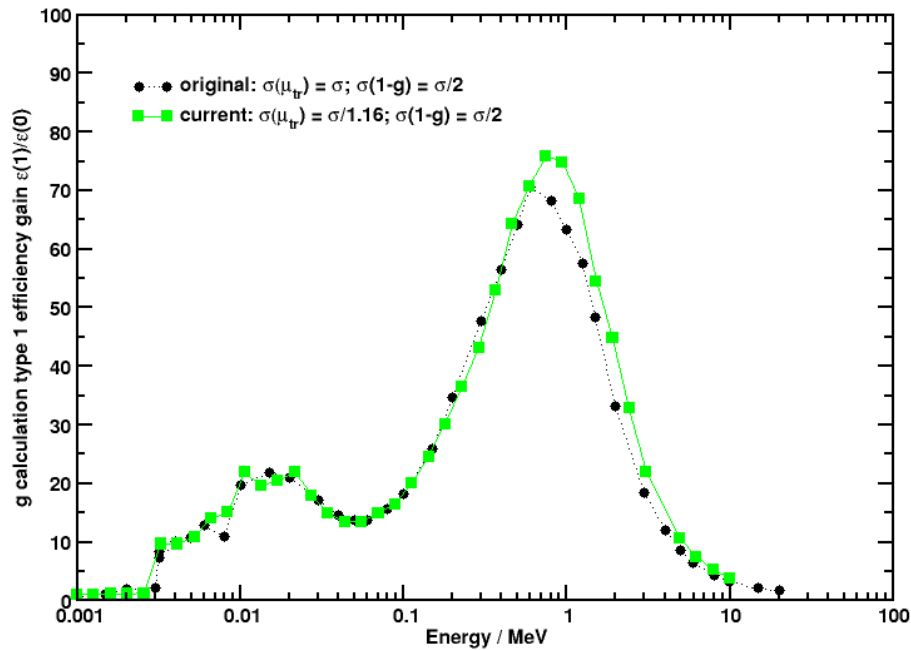
Figure 1.1 Efficiency gain of calculation type 1 over type 0 in air.

run until a 0.1% precision is reached is shown in the following code snippet:

```
#################################################################
# calculation types
#
# 0 (default)            : Executes fixed number of histories
#
# 1 (photon source only) : Runs to desired precision
#
#################################################################
#
calculation type = 1
#######################
# 0.1% desired precision
#######################
precision = 0.001    # relative precision
#############
```

1.4 Source definition

The input block for defining source parameters must be placed between `start` and `stop` delimiters. For instance, to define a monoenergetic source emitting particles of energy E, and charge q, the input keys required are:

```
 :Start Source Input:
  Incident Charge       = q # 0 for photons -1 or 1 for e- or e+
  INCIDENT SPECTRUM     = mono-energy
  INCIDENT KINETIC ENERGY = E # in MeV
:Stop Source Input:
```

Several individual energies can be requested by adding comma separated energy values:

```
 :Start Source Input:
  Incident Charge       = q # 0 for photons -1 or 1 for e- or e+
  INCIDENT SPECTRUM     = mono-energy
  INCIDENT KINETIC ENERGY = E1,...,EN # in MeV
:Stop Source Input:
```

In this case the code will loop over individual calculations for each energy.

To generate a database of μ_{tr} or μ_{en} values as function of energy, one can define a linear or a logarithmic energy grid. A linear (equidistant) energy grid is defined using the input keys

```
 :Start Source Input:
  Incident Charge         = q # 0 for photons -1 or 1 for e- or e+
  INCIDENT SPECTRUM       = mono-energy-lin-range
  INCIDENT KINETIC ENERGY = Emin, Emax, deltaE
:Stop Source Input:
```

where the input key `deltaE` defines the bin width ΔE defining a grid of energies: $E_{\text{min}}, ..., E_{\text{min}} + i \cdot \Delta E, ..., E_{\text{max}}$

Similarly, a logarithmic energy grid is defined using the input keys

```
 :Start Source Input:

 INCIDENT SPECTRUM=        mono-energy-log-range
 INCIDENT KINETIC ENERGY= Emin, Emax, N

:Stop Source Input:
```

where the logarithmic bin width Δ_{log} is obtained using

$$\Delta_{\text{log}} = log\left(\frac{E_{\text{max}}}{E_{\text{min}}}\right)/(N-1) \tag{1.17}$$

defining a logarithmic grid of N energy values

$$E_i = e^{log(E_{\text{min}})+i\cdot\Delta_{\text{log}}} \tag{1.18}$$

Note

> These calculations will be limited to one CPU thread since there is no parallel execution capability in g. As the energy increases, CPU time will also increase. If a computer cluster or multiprocessor PC is available, it is more efficient to perform these calculations individually by submitting them to the available CPU threads. A strategy for creating large μ_{en} databases is discussed below in section Building large databases.

A source of particles with a tabulated energy distribution (spectrum) can also be defined. A polyenergetic Co-60 beam is defined using the keys:

```
########################################################################
### SOURCE DEFINITION
########################################################################
 :Start Source Input:
#####################
 Incident Charge=   0 # -1 or 1 for e- or e+
 #############################
 # Co-60 polyenergetic source
 #############################
 INCIDENT SPECTRUM = SPECTRUM
 SPECTRUM FILE      = $HEN_HOUSE/spectra/egsnrc/co60.spectrum
 :Stop Source Input:
#####################
```

The format for spectrum files is described `at the end of section 2.7 of report PIRS-702`, *NRC User Codes for EGSnrc*. Further examples of spectrum files can be found in EGSnrc's default spectrum location:

$HEN_HOUSE/spectra/egsnrc/

1.5 Usage

The EGSnrc application g can be started from the command line using

```
g -i input_file [-p pegs_file] [-o output_file] [-b]
```

where the arguments in square brackets are optional. When the g application is run interactively, results of a calculation are written to standard output. If one wants to print the results of an interactive calculation to an output file, one can redirect the output to a file using

```
g -i input_file [-p pegs_file] > output_file
```

On Linux, one can make use of the **tee** command to route the results to multiple outputs (standard and a file) :

```
g -i input_file [-p pegs_file] | tee output_file
```

Note

Beware of overwriting an existing file!

1.5.1 Optional arguments

- With the **-b** option one specifies a batch run, *i.e.* the output is redirected to **output_file.egslog**.

- If material information is obtained via a PEGS4 data file, the option **-p** tells the application which file to use. Alternatively, materials can be defined in the input file via a media definition input block, in which case this option must be omitted.

- With the **-o** option one can change the name of the output files (by default **input_file.xxx** is used, where **xxx** is **.egslog** for the log file, **.egsdat** for the data file, etc.).

Remember that any application can be executed from the EGSnrc GUI egs_gui!!!

1.6 Input example: Monoenergetic 30 keV photon beam

```
#########################################
#
# Example input file for the g application:
#
#  - Monoenergetic 30 keV photon beam
#  - Calculation type 1 (photon beam only)
#  - Verbose output
#  - Desired precision 0.1%
#
#########################################
#############################
# Application specific inputs
#############################
 media=  air0o
 INITIAL RANDOM NO. SEEDS= 97, 33
 Number of HISTORIES= 2000000000000
 precision= 0.001
 calculation type= 1
 verbose = yes
####################
 :Start Source Input:
####################
 INCIDENT SPECTRUM=        mono-energy
```

```
 INCIDENT KINETIC ENERGY= 0.030
 Incident Charge=          0
#####################
 :Stop Source Input:
#####################
#############################
 :Start MC Transport Parameter:
#############################
 Global ECUT=                    0.512
 Global PCUT=                    0.001
 Global SMAX=                    1e10
 Spin effects=                   On
 Rayleigh scattering   = On
 Photon cross sections  = mcdf-xcom
 Pair cross sections = nrc
 Triplet production = On
 Bound Compton Scattering = norej
 Radiative Compton corrections = ON
 Atomic relaxations=             On
 Photoelectron angular sampling= On
 Brems angular sampling=         KM
 Brems cross sections= nrc
 Pair angular sampling=          KM
 ESTEPE=                         0.25
 XIMAX=                          0.5
 Electron Impact Ionization=     ik
 Skin depth for BCA=             3
 Boundary crossing algorithm=    exact
 Electron-step algorithm=        default
#############################
 :Stop MC Transport Parameter:
#############################
#########################
:start media definition:
#########################
    ### energy transport cutoffs
    ae = 0.512
    ue = 2.511
    ap = 0.001
    up = 2.00
    :start air0o:
      density correction file    = air_dry_nearsealevel
      bremsstrahlung correction= NRC
    :stop air0o:
#########################
:stop media definition:
#########################
```

1.7   Building large databases

A consequence of not having an option for parallel execution is that lengthy calculations cannot be run on multiple cores, for instance on a computer cluster or a multicore processor PC. This limitation makes the creation of large μ_{en}/ρ or μ_{tr}/ρ databases spanning an energy range from threshold up to several MeV very time consuming. As the energy of the initial particle increases, more energetic electrons are created which have to be followed in an infinite geometry until they have lost all their kinetic energy. It is hence more practical to submit individual type 1 calculations for each energy. Furthermore, one can take advantage of the energy dependence of the mass energy transfer and absorption coefficients and use a logarithmic energy grid to reduce the number of high energy data points.

A simple Bash script is provided here as an example of what one could do to submit multiple calculations on a logarithmic energy grid. By default it runs type 1 calculations in air for 10 energies between 1 keV and 1 MeV on a logarithmic energy grid in the available cores of a PC. Precision is set to 0.1%. Water is also available and further media can be added to the media definition block of the input file template. By passing different arguments a user can change certain parameters of the simulations. It would be simple to modify the script to allow other parameters to be changed.

```bash
#!/bin/bash
################################################################################
#
#  Linux script to create input files on-the-fly for the g-app
#  to generate databases of muen or mutr values on a logarithmic grid.
#
#  Copyright (C) 2020 National Research Council Canada
#
```

```
############################################################################
#
#  Author:          Ernesto Mainegra-Hing
#
#  Contributors:
#
############################################################################
#
#
# Defaults
#
n=10
Emin=0.001
Emax=1.0
accu=0.001 # 0.1% stat. precision
prefix=muen_rho
pe_xsections=xcom
medium=air
E=0.01
#-------------------------------
# Parse command line for arguments
#-------------------------------
while test $# -gt 0; do
    case "$1" in
      min=*)  Emin=`echo $1  | sed 's/min=//'` ;;
      max=*)  Emax=`echo $1  | sed 's/max=//'` ;;
        n=*)     n=`echo $1  | sed 's/n=//'` ;;
     accu=*)  accu=`echo $1  | sed 's/accu=//'` ;;
   medium=*) medium=`echo $1 | sed 's/medium=//'` ;;
   prefix=*) prefix=`echo $1 | sed 's/prefix=//'` ;;
       pe=*) pe_xsections=`echo $1 | sed 's/pe=//'` ;;
    esac
    shift
done
#--------------------
# get energy limits
#--------------------
AP=$(echo "scale=3; $Emin" | bc)
UP=$(echo "scale=3; $Emax" | bc)
AE=$(echo "scale=3; $AP+0.511" | bc)
UE=$(echo "scale=3; $UP+0.511" | bc)
#--------------------
# prepend leading zero
#--------------------
printf -v AP "%0.3f" ${AP}
printf -v UP "%0.3f" ${UP}
printf -v AE "%0.3f" ${AE}
printf -v UE "%0.3f" ${UE}
echo 'lower photon energy cut-off:' $AP
echo 'upper photon energy cut-off:' $UP
echo 'lower electron energy cut-off:' $AE
echo 'upper electron energy cut-off:' $UE
# Let's be explicit
typeset -i i
let "n -= 1"
Dlog1=$(echo "l(${Emax}) - l(${Emin})" | bc -l)
Dlog=$(echo "${Dlog1} / $n" | bc -l)
printf 'Dlog = %.5f\n' "$Dlog"
for ((i=0;i<=n;i++))
do
  E=$(echo "e(l(${Emin})+$i*${Dlog})" | bc -l)
  # Protect against round-off at 1 keV
  if (( $(echo "$E == 0.001" | bc -l) )); then
    E=$(echo "e(l(${Emin})+0.0000000001+$i*${Dlog})" | bc -l)
  fi
  printf -v Ename "%0.8f" ${E}
  echo E = ${E}
  name=${prefix}_${medium}_${pe_xsections}_${Ename}_MeV
    cat >${name}.egsinp <<EOF
###########################
:Start MC Transport Parameter:
Global ECUT=                    $AE
Global PCUT=                    $AP
Global SMAX=                    1e10
Spin effects=                   On
Rayleigh scattering   = On
Photon cross sections  = $pe_xsections
Pair cross sections = nrc
Triplet production = On
Bound Compton Scattering = norej
Radiative Compton corrections = ON
Atomic relaxations=             On
Photoelectron angular sampling= On
Brems angular sampling=         KM
Brems cross sections= nrc
Pair angular sampling=          KM
ESTEPE=                         0.25
```

```
XIMAX=                        0.5
Electron Impact Ionization=   On
Skin depth for BCA=           3
Boundary crossing algorithm=  exact
Electron-step algorithm=      default
:Stop MC Transport Parameter:
############################
media=  $medium
INITIAL RANDOM NO. SEEDS= 97, 33
Number of HISTORIES= 2000000000000
precision= $accu # relative precision
calculation type= 1
verbose = yes
:Start Source Input:
-------------------
INCIDENT SPECTRUM=        mono-energy
INCIDENT KINETIC ENERGY= ${E}
Incident Charge=          0
:Stop Source Input:
------------------
###########################################################################
### PEGS PARAMETERS
###########################################################################
:start media definition:
    ### energy transport cutoffs
    ae = $AE
    ue = $UE
    ap = $AP
    up = $UP
    :start air:
      density correction file    = air_dry_nearsealevel
      bremsstrahlung correction= NRC
    :stop air:
    :start water:
      density correction file    = water_icru90
      bremsstrahlung correction= NRC
    :stop water:
:stop media definition:
EOF
######################################################################
#
# Command to submit the calculations to the cores of a Unix/Linux
# computer cluster running a PBS job scheduler.
#
# Uncomment the two lines below and comment out the subsequent lines to
# submit to a cluster:
#
#echo Submitting $name to the queue user1...
#${HEN_HOUSE}scripts/run_user_code_batch g ${name} pegsless batch=pbs user1
######################################################################
######################################################################
#
# Submission to the available logical threads of a multicore PC:
#
# Comment out the 4 lines below if using the command above for submission
# to a cluster!
#
######################################################################
command="g -i $name"
$command -b -P 1 -j 1 >/dev/null 2>&1 &
processid=`printf %5d $!`
echo "LAUNCHED $processid: $command -b -P 1 -j 1 &"
######################################################################
done
# wait for completion and report
# Comment out the 4 lines below if using the command for submission
# to a cluster!
wait
echo -----------------------------------------------------------------------------
echo "SIMULATION COMPLETED"
echo
######################################################################
exit 0
```

1.7.1   Speeding up calculations above 1 MeV

As the energy of the photons increases over the pair production threshold, more high energy charged particles are produced and computing 1- $\overline{g}$ down to $\sigma/2$ becomes increasingly time consuming. To reduce the calculation time in this energy range, one could calculate 1- $\overline{g}$ to a higher uncertainty and correspondingly $\mu_{\rm tr}/\rho$ to a smaller uncertainty so that the statistical precision of $\mu_{\rm en}/\rho$ remains σ. Further optimization of a calculation type 1 in

the megavoltage energy range is possible by modifying the target uncertainty in μ_{tr}/ρ and $1-\overline{g}$ according to the relationships:

$$\sigma(\mu_{\text{tr}}) = \frac{\sigma}{\sqrt{m}} \qquad \text{and} \qquad \sigma(1-\overline{g}) = \frac{\sigma}{\sqrt{m/(m-1)}} \tag{1.19}$$

which guarantee a user-desired precision σ in μ_{en}/ρ and where m = 2, 3, 4, .., etc. The efficiency gain of a calculation type 1 over a type 0 in air as function of energy is shown in figure 1.2 for different m values. A value of m = 2 is recommended for calculations around 1 MeV.
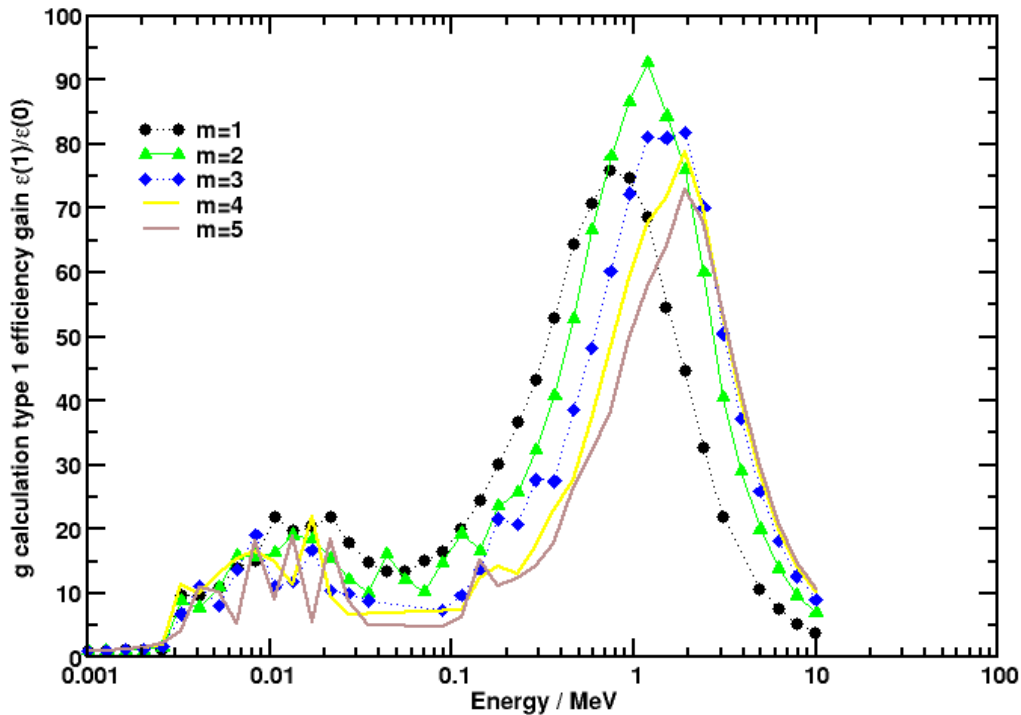
Figure 1.2 Efficiency gain of calculation type 1 over type 0 in air for different m values.

The input key for modifying m is `precision balance` as shown in the input block example below

```
###################################
  calculation type = 1
 ###################################
 # Desired precision for muen
 # (relevant only for calculation type 1)
 ###################################
 precision = 0.0001    # relative precision
 precision balance = 2 # same precision accu/sqrt(2) for mutr and 1-g
 #############
```

Note that omitting m or setting m = 1 invokes the default approach.

1.8 CVS log history

The g app was ported to the multiplatform EGSnrc system in 2004 and placed under revision control using the Concurrent Versioning System (CVS). The CVS log messages until EGSnrc was moved to the Git distributed version control system in 2015 are included verbatim below for reference:

```
RCS file: /home/cvsroot/HEN_HOUSE/user_codes/g/g.mortran,v
Working file: g.mortran
head: 1.6
branch:
locks: strict
access list:
symbolic names:
        port-to-git: 1.6
        export-to-git: 1.6
        Rio-egsnrc-course: 1.6
        V4-2-3-2: 1.6
        beam2009course: 1.5
        V4-r2-3-0: 1.5
        beam2007course: 1.5
        beam2007: 1.5
        V4-r2-2-5: 1.5
        beam2006course: 1.5
        V4-r2-2-3: 1.3
        beam2006: 1.3
        V4-r2-2-2: 1.3
        oz-course-2005: 1.3
        V4-r2-2-1: 1.3
        V4-r2-2-0: 1.3
        egs-course-2005: 1.3
        V4-r2-0-0: 1.2
keyword substitution: kv
total revisions: 6;     selected revisions: 6
description:
----------------------------
revision 1.6
date: 2010/02/17 19:04:40;  author: mainegra;  state: Exp;  lines: +20 -8
- Described the new calculation type = 1 which is much more efficient.
- Improved messages describing the calculation.
- Message about stopping calculation when a desired accuracy is reached only
  applies to calc type = 1.
----------------------------
revision 1.5
date: 2006/04/18 20:01:38;  author: mainegra;  state: Exp;  lines: +150 -19
- Added ability to run the calculation for more than one
  energy. One can now enter individual energies, or request
  linear or log energy grids. Useful for fast calculations,
  but unpractical for very long calculations.

We should update the header of g.mortran to remind us of
the latest changes ... will do it soon ...
----------------------------
revision 1.4
date: 2006/04/07 16:14:43;  author: iwan;  state: Exp;  lines: +159 -23
Implemented a type=1 calculation that can run until a
prescribed accuracy is reached.
In a type=1 calculation mu_tr is calculated first, then
mu_en is obtained from mu_en = mu_tr*(1-g), where g is
the fraction lost to radiation from slowing down electrons.
The advantage is that when g is small, mu_en converges much
faster to the prescribed accuracy compared to a type=0
calculation. We needed this to be able to obtain mu_en
tables more quickly than before.
----------------------------
revision 1.3
date: 2005/02/25 16:09:07;  author: iwan;  state: Exp;  lines: +12 -1
Count radiative losses due to fluorescence after
EII.
----------------------------
revision 1.2
date: 2005/01/05 13:33:43;  author: iwan;  state: Exp;  lines: +14 -4
Added a proper header.
----------------------------
revision 1.1
date: 2004/11/26 22:34:47;  author: mainegra;  state: Exp;
We had forgotten to port the little g code to the new system.
```

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 g.doxy File Reference

Documents the g application.

3.1.1 Detailed Description

Documents the g application.

Contains the main body of the user's manual for the g application.

Author

Ernesto Mainegra-Hing, NRC

3.2 g.f File Reference

Application for radiative energy loss calculations.

Functions/Subroutines

- program calculate_g
- subroutine inputs

  Subroutine to read user inputs and transport parameters.
- subroutine howfar
- subroutine ausgab (iarg)
- subroutine source
- subroutine test_brems
- subroutine test_compton

3.2.1 Detailed Description

Application for radiative energy loss calculations.

EGSnrc application to calculate $\overline{g}$, the average fraction of kinetic energy transferred to charged particles by photons lost to radiation when electrons slow down, or the radiative yield Y for electron beams. For photons, quantities such as kerma, collision kerma, $\langle \mu_{tr}/\rho \rangle_\psi$ and $\langle \mu_{en}/\rho \rangle_\psi$ are also calculated.

Implemented a type=1 calculation that can run until a prescribed precision is reached at any stage during the calculation. In a type=1 calculation μ_{tr}/ρ is calculated first, then μ_{en}/ρ is obtained as $\mu_{tr}/\rho \cdot (1 - \overline{g})$, where $\overline{g}$ is the fraction lost to radiation from slowing down electrons. The advantage is that when $\overline{g}$ is small, μ_{en}/ρ converges much faster to the desired precision compared to a type=0 calculation.

Revision History

Version 1.0 (I. Kawrakow, January 2000): Initial version.

Version 1.1 (I. Kawrakow, March 2002): Added $E \cdot \mu_{tr}/\rho$ and $E \cdot \mu_{en}$.

Version 1.2 (D. Rogers, June 2002): Get μ_{tr}/ρ and μ_{en}/ρ.

Version 1.3 (D. Rogers, Aug 2002): Account for fluorescent photons in $\overline{g}$ and μ_{tr}/ρ correctly.

Version 1.4 (R. Townson, December 2016): Energy depositions for kerma calculations below the cut-off are now sorted out using AUSGAB to include only Auger from relaxations, not fluorescence.

Version 1.5 (E. Mainegra-Hing, January 2020): The $\overline{g}$ calculation for spectra was missing a factor μ/ρ for both calculation types. This factor is now included when scoring e_{tot} and e_{rad}. Bug reported by D. Rogers. The output has been made more informative and a verbose option added for type 1 calculation. Code has been extensively documented and a doxygen based user's manual created.

Version 1.6 (E. Mainegra-Hing, August 2020): Calculation type 1 has been made independent of NCASE and the target precision now defaults to 1%. Algorithm now ensures μ_{en}/ρ is calculated to the user-requested precision σ by computing μ_{tr}/ρ down to $\sigma/\sqrt{4/3}$ and 1- $\overline{g}$ to $\sigma/2$. For calculations in the megavoltage range (for Ir-192, Co-60) one can modify the target precision of mutr and 1-g to save substantial time in the fairly time consuming calculations above 1 MeV. To this end the uncertainty in μ_{tr}/ρ and 1- $\overline{g}$ can be modified so that the μ_{en}/ρ value has an precision σ according to:

$$\sigma(\mu_{tr}) = \frac{\sigma}{\sqrt{m}} \qquad \text{and} \qquad \sigma(1-\overline{g}) = \frac{\sigma}{\sqrt{m/(m-1)}}$$

where m can be 2, 3, 4, etc. A value of m =1 indicates the original implementation should be used. For megavoltage energies around 1 MeV, m = 2 is recommended. Calculation progress messages during a type 1 calculation have been made clearer and the code has been further documented. The documentation has been also updated to reflect these changes.

Author

Iwan Kawrakow, NRC

Date

2000

Copyright

National Research Council Canada

Parameters

out	*g*	average fraction of the kinetic energy of secondary charged particles (produced in all the types of interactions) that is subsequently lost in radiative (photon-emitting) energy-loss processes as the particles slow to rest in the medium. (Taken from the NIST web page)	
	e_brem	Average energy lost to bremsstrahlung	
	ebrem_tmp	As above for the current history	
	e_rad	Average energy lost to bremsstrahlung & kinetic energy transferred in annihilation radiation (ie only annihilation in flight. Note this used to include fluorescent photons, but they should be excluded from e_tot as well.	
	erad_tmp	As above for the current history	
	e_tot	average energy released per particle includes subthreshold energy (iarg=4)	
	etot_tmp	As above for the current history	
	e_radc	= sum (etot_tmp∗erad_tmp)	
	e_bremc	= sum (etot_tmp∗ebrem_tmp)	
out	*e_brem/e_tot*	the average fraction of the kinetic energy subsequently lost in bremsstrahlung (photon-emitting) events	
out	*e_rad/e_tot*	same as g above	
	anorm		
	npgi		
	npei		
	E_ave	Average spectrum energy	
	factor	Converts e_mutr and e_muen scored as $\mathrm{MeV\,cm^2/g}$ to $\mathrm{Gy\,cm^2}$	
	de_pulsei		
	mutr	The correct definition for a spectrum is that given by Attix of an energy fluence weighted average mass transfer coefficient $$\langle \mu_{\mathrm{tr}}/\rho \rangle_\psi = \int \psi(E)\mu_{\mathrm{tr}}(E)/\rho dE / \int \psi(E) dE$$ This is equivalent to the ICRU60/ICRU85 definition as long as one sums the total energy transferred from the spectrum and then divides by the average energy incident. For a while we scored <mu_tr/rho> directly, but this can differ substantially from the correct results and amounts to averaging over the fluence, not the energy fluence	
	muen	Similarly, the energy fluence weighted average mass absorption coefficient is given by $$\langle \mu_{\mathrm{en}}/\rho \rangle_\psi = \int \psi(E)\mu_{\mathrm{en}}(E)/\rho dE / \int \psi(E) dE = \langle \mu_{\mathrm{tr}}/\rho \rangle \, (1 - \overline{g})$$	
	e_mutr	Fluence averaged product $\langle E\,\mu_{\mathrm{tr}}/\rho \rangle_\phi$	
	e_muen	Fluence averaged product $\langle E\,\mu_{\mathrm{en}}/\rho \rangle_\phi$	
	Eave	average energy of incident spectrum: either actual or sampled	

3.2.2 Function/Subroutine Documentation

3.2.2.1 ausgab()

```
subroutine ausgab (
          integer*4 iarg )
```

3.2.2.2 calculate_g()

```
program calculate_g
```

3.2.2.3 howfar()

```
subroutine howfar
```

3.2.2.4 inputs()

```
subroutine inputs
```

Subroutine to read user inputs and transport parameters.

3.2.2.5 source()

```
subroutine source
```

3.2.2.6 test_brems()

```
subroutine test_brems
```

3.2.2.7 test_compton()

```
subroutine test_compton
```

Index