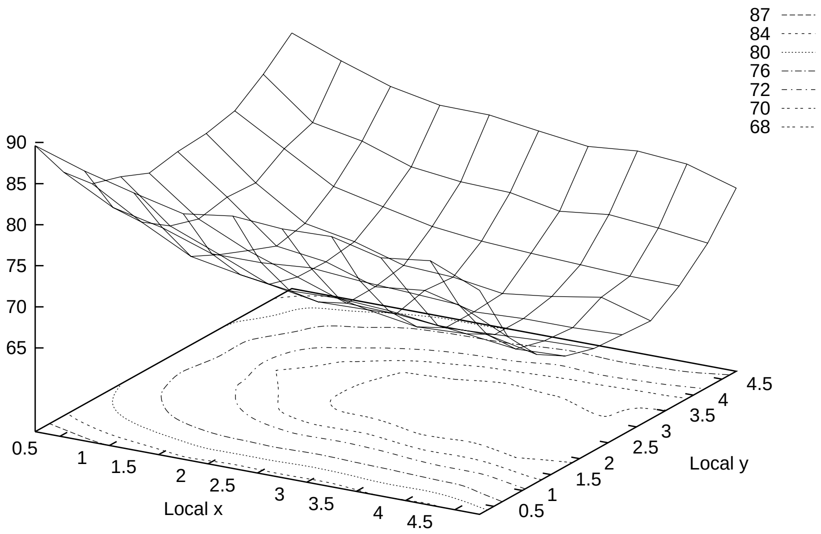
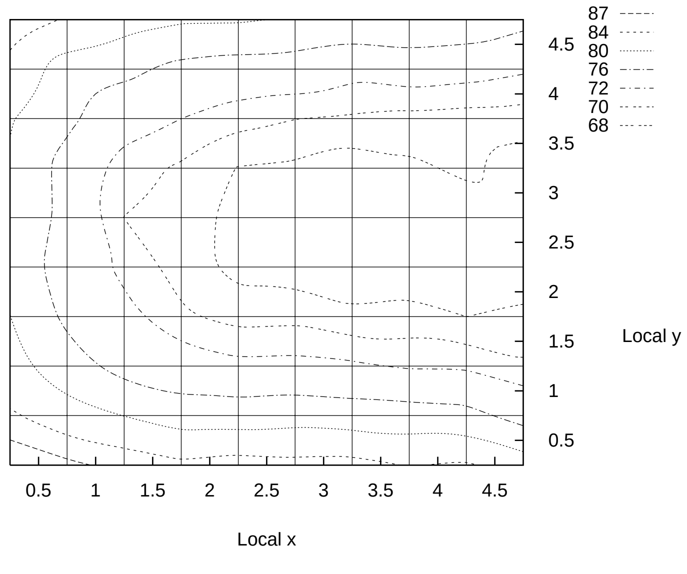
GnuRad

(**Gnu** **R**adation **A**ided **D**esigner)





Introduction

Welcome to GnuRad; GnuRad is a **Rad**iation heat transfer **C**omputer **a**ided **m**odeler. It allows a user to model a variety of radiation heat transfer problems without having to write a computer program. GnuRad can model something simple as energy radiating from a plane surface, or something complex where the model is a series of connected enclosures containing a participating media. The types of models that GnuRad can run is quite extensive, currently limited to only orthogonal plane **gray surfaces** with a media having isotropic scattering. Some examples of what GnuRad can model are:

- The radiant energy which is absorbed by a furnace wall.
- Radiant energy distribution of a room with windows and a fireplace.
- The radiation exchange factor of one surface to another.
- The temperature distribution in a participating media based on radiant energy.

GnuRad is a computer program which simulates the radiation heat transfer in an enclosure or series of connected enclosures. The basic enclosure is 'box' shaped (regular parallelepiped) having arbitrary dimensions and containing a media. Depending on the requirements of the model, boxes can be joined together, walls can be removed, and the media can be participating or non-participating. A wall or media temperature/emitting flux can be initially specified in the input or GnuRad can solve for an unknown temperature. Wall and media properties are specified in the input file. Wall properties are: temperature or emitting flux, emissivity, diffuse reflectivity, specular reflectivity, and number of grid elements which form the wall. Media properties are: temperature, extinction coefficient, single scattering albedo, type of scattering, and the number of 'brick' elements which form the media. GnuRad provides five different types of output files which presents the results in a tabulated format and a text-graphic format. The other output files are used as inputs to spreadsheets and plotting programs.

GnuRad is a direct physical model where radiant energy is sent and tracked to and from all participating sources. More precisely, a photon bundle containing a specific amount of energy is sent from a wall or the media, and it is tracked until it is absorbed by a wall or the media. This process is repeated until all surfaces and media have sent their required number of bundles. Direction and location for sending the bundles is determined by applying the Monte Carlo technique. Details of the Monte Carlo technique can be found in several of the books listed in the Reference section of this document.

GnuRad is written in C++ using object oriented programming practices. The source code and user manual of GnuRad are free to be copied or modified as long as any changes or enhancements continue to be free. GnuRad is protected by the GNU public license. This philosophy will allow GnuRad to continue to grow in options and features. The latest version of GnuRad can be done by sending email to gnurad@u.washington.edu.

It is recommended that the first five sections of this manual be read before attempting to use GnuRad. Sections two through four give details on how to construct a model, create an input file, and run a model. Finally, section five contains nine different sample input files that have descriptive comments.

Building A Model

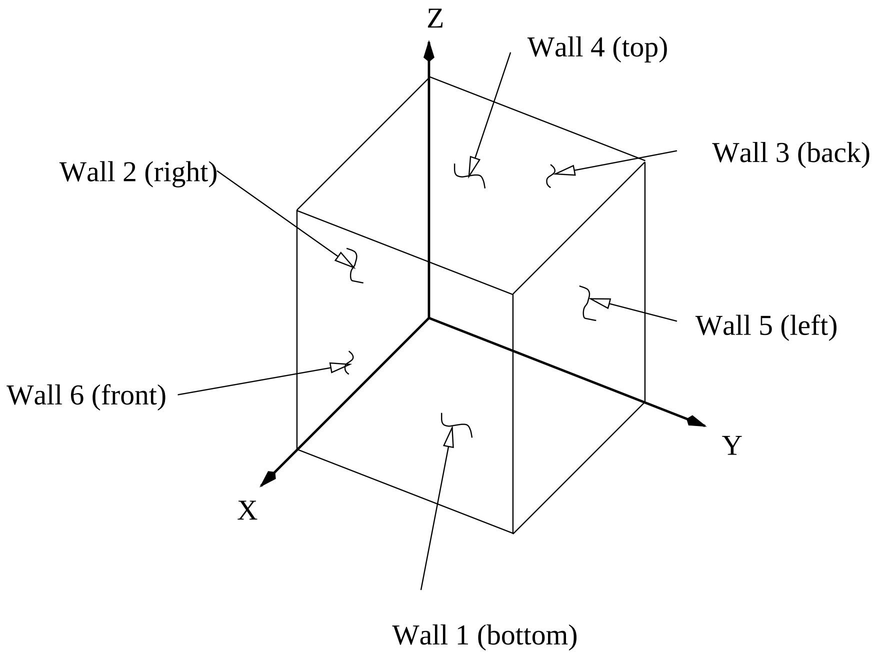
Introduction

In order to run Gnurad, the user must create an input file of what is to be modeled. The minimum input consist of what **mode** to run Gnurad, the bundle energy or the number of bundles to send, and the model description. The minimum model is a box containing a media. All six walls of the box and the media must have their properties described in the input. In this section, we will focus on describing the geometric properties of a model, and the coordinate systems which are involved in building a model. Details of the input language and samples of input files are given in later sections.

Coordinates

There are three coordinate systems used in GnuRad; they are: global, box, and local coordinates. Global coordinates are used for the entire model, box coordinates are used for the box and its media, and local coordinates are used for each of the six walls. The importance of the global coordinate system will be discussed in the "Joining of Boxes" section.

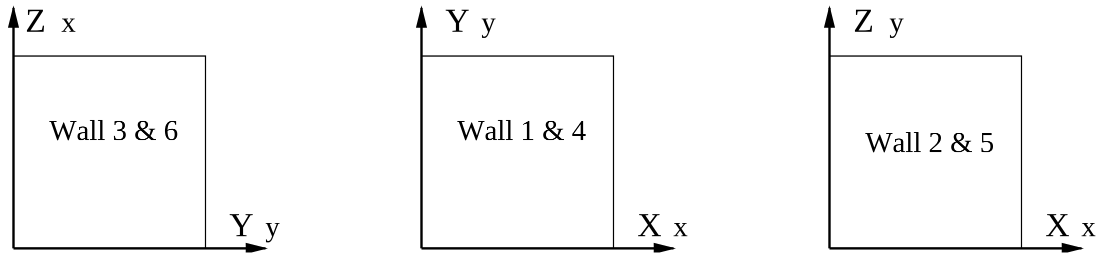
Box coordinates are needed to tell the numbering system of the walls and the dimension of the box and its media. Below is a figure showing the box coordinate system and the wall numbering scheme. This numbering scheme is the same for all boxes in the model.



Box Coordinates

Local Coordinates

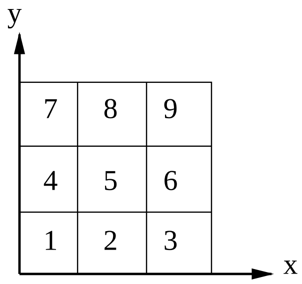
As previously stated, all walls have their own local coordinate systems. Shown below are how the six local coordinate systems (**lower case**) of the walls line up with the box coordinate system (**upper case**).



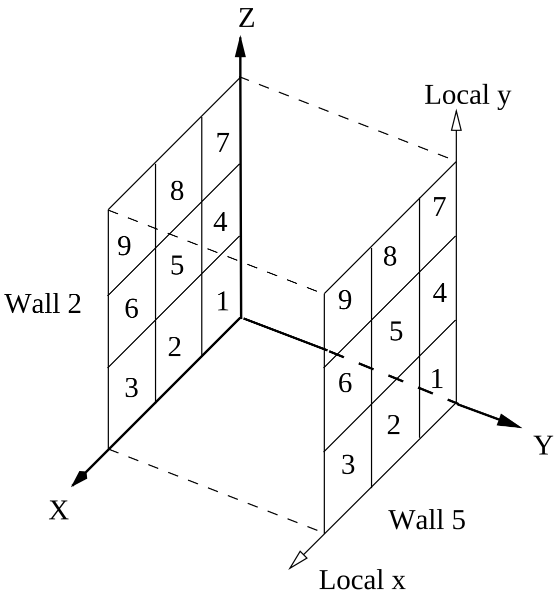
Local Coordinates

Meshing A Wall

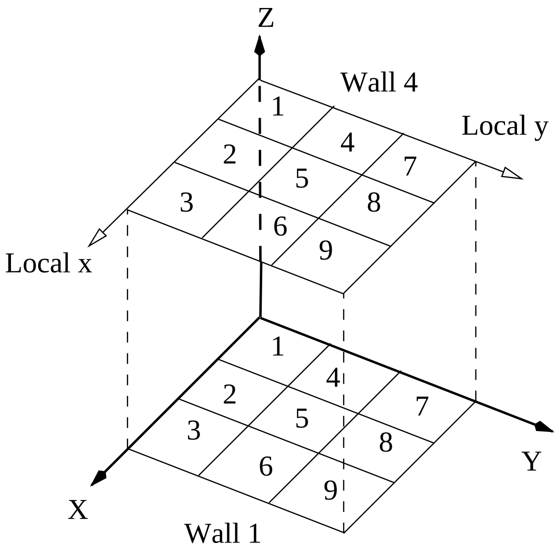
A wall is divided into a 1x1 mesh by default. However, it is sometimes desirable to divide a wall into many smaller elements (i.e. 5x4 mesh). This may be done to view a temperature profile of an unknown wall or to account for varying properties of the wall. When a wall is divided into smaller elements, each element is tagged with a specific number. Shown below and on the following page is the numbering scheme of the wall elements, with respect to a wall's local coordinate system and the box coordinate system.



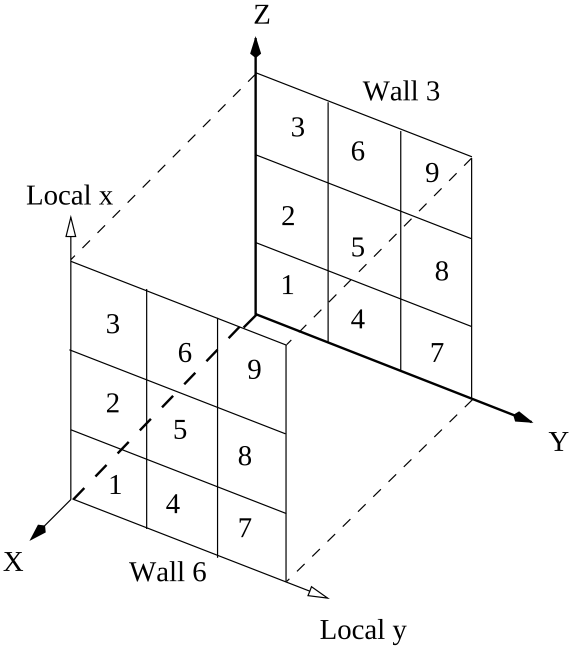
Wall Element Numbering (Local Coordinates)



Walls 2 & 5



Walls 1 & 4

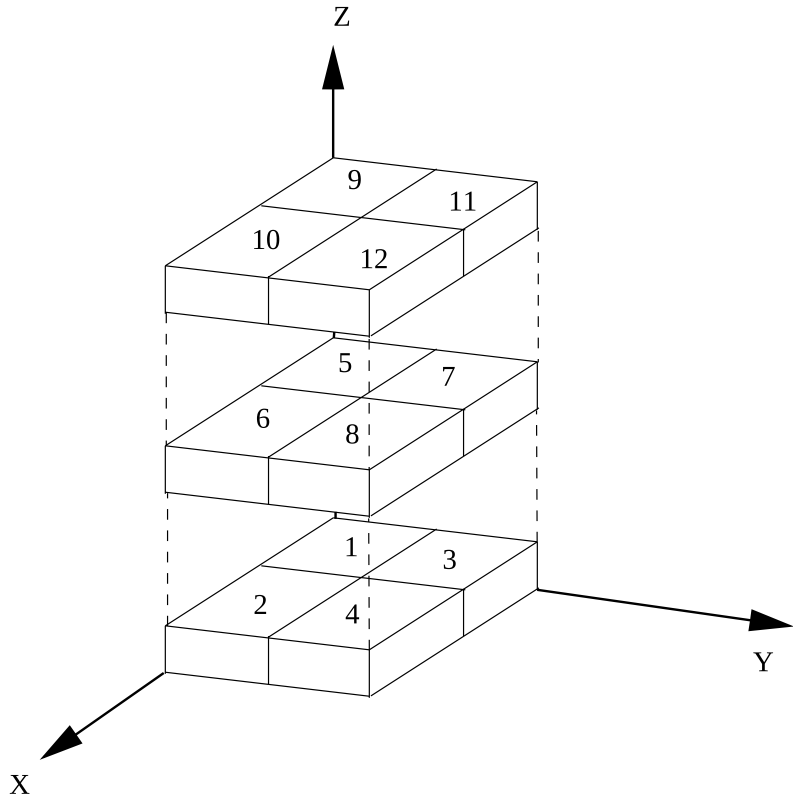


Walls 3 & 6

Wall Element Numbering (Local & Box Coordinates)

Meshing A Media

The media of a box can also be divided into smaller 'brick' elements. Each 'brick' element also has a unique numerical identifier as shown below.



Media Elements (Box Coordinates)

Joining Boxes

When a model consists of more than one box, it becomes necessary to join boxes together to either form a new shape and/or to have varying properties of the media. There are three rules which must be adhered to when joining boxes; they are:

i) The individual boxes must have their coordinates aligned with the global coordinates.
ii) Joining walls must have the same global dimensions.
iii) Both joining walls must be declared by using the keyword "**join**" in the **WALL** statement.

Units

The following units are used in GnuRad.

Dimensions - Meters
Energy - Joule
Flux - Watts per square meter
Power - Watts
Temperature - Degrees Kelvin
Time - Seconds

Input Language

GnuRad uses an input language to create a model. The language has keywords to describe the various properties of the model. The keywords are listed below. They may be lower, mixed, or upper case.

Box	**Bundle**	**Iterate**	**Media**	**Melement**
Mode	**Model**	**Particle**	**Seed**	**Wall**
Welement	**#**			

Separators

Input data is separated by any of the following characters which are enclosed in single quotes:
’**,**’ ’**:**’ ’**;**’ ’**(**’ ’**)**’ ’*space*’ ’*Carriage return*’

Keyword Order

Each keyword has it’s own list of parameters and data associated with it. In regards to order, GnuRad expects to see the following order of keywords. Note, a decimal indicates that this keyword is either optional or only used with certain modes.

1)	**Mode**
2)	**Bundle**
2.1)	**{Seed}**
3)	**Model**
First box set (Denote by E1)	
E1-1)	**Box**
E1-2)	**Wall** 1 through 6
E1-2.1)	**{Welement}**
E1-2.2)	**{Iterate}**
E1-3)	**Media**
E1-3.1)	**{Melement}**
E1-3.2)	**{Iterate}**
Second box set (Denoted by E2)	
E2-1)	**Box**
E2-2)	**Wall** 1 through 6
E2-2.1)	**{Welement}**
E2-2.2)	**{Iterate}**
E2-3)	**Media**
E2-3.1)	**{Melement}**
E2-3.2)	**{Iterate}**

Next set, continue until all box are entered

Keyword Description

Box(*bnum*, *Xdim*, *Ydim*, *Zdim*)

bnum - The box number; this should be a positive integer value greater than zero. The elements must appear in sequential order.
Xdim - x axis dimension which coincides with the box coordinate system.
Ydim - y axis dimension which coincides with the box coordinate system.
Zdim - z axis dimension which coincides with the box coordinate system.

Bundle(*value*)

value - Value can have two possible choices. If GnuRad is in temperature mode, *value* is a positive floating point number greater than zero which represents the energy of a photon bundle per unit time. If GnuRad is in View factor/ Radiation exchange factor (Script *F*) mode, *value* is a positive integer number greater than zero which represents the number of bundles to send per unit area.

Iterate(*maximum, percent, factor*, q_{net}(i), ...q_{net}(nelements))

maximum - The maximum number of iteration loops, an integer value. Note, setting maximum to a small value may prematurely stop convergence.
percent - The maximum error ratio specified in percentage.
factor - The iteration update factor, 1.0 is a nominal value.
q_{net}(i) - The final flux value the ith element should reach after iterating.

Media(*bnum, Media keyword*)

bnum - The box number containing the media.

<u>*Media keyword* - The following are for use in the Temperature mode.</u>
Media(*element*, **clear**)
Media(*element*, **element**, *meshx, meshy, meshz*, K_e, *ssal, scatter*)
{**Melement** Keyword must follow on the next line if **element** is used in **Media**}
Media(*element*, **flux**, *flux*, K_e, *ssal, scatter*)
Media(*element*, **temp**, *temperature*, K_e, *ssal, scatter*)
Media(*element*, **unknown**, *meshx, meshy, meshz*, K_e, *ssal, scatter*)

{**Iterate** Keyword must follow immediately if **unknown** is used}

Media keywords - View Factor or Radiation Interchange Factor Mode (RIF)
Media(*element*, **participate,** K_e, *ssal, scatter*)
Media(*element*, **clear**)

clear - Use when the Media is non participating.
element - Use when each media meshed element should have its own temperature or flux value.
flux - Energy per unit volume per unit time sent out by the media.
participate - The media participates in the radiation exchange factor calculation.
temp - Temperature of the media.
unknown - The temperature of the media is unknown.

K_e - Extinction coefficient
meshx - The number of meshed elements in the x direction.
meshy - The number of meshed elements in the y direction.
meshz - The number of meshed elements in the z direction.
scatter - Type of scattering choices are currently, "**Isotropic**".
ssal - Single scattering albedo.
temperature - Media temperature.

Melement(*value, first element property, ... last element property*)
value - It can be **flux** or **temp**.
element property - Depending on *value*, it is either the energy per unit volume per unit time or the temperature.

Mode(*mode*)

mode - The execution mode, the two choices are, "**Temperature**" or "**View**".
Temperature mode is for solving for temperatures and fluxes where **View** is solving for view factors or radiation exchange factors.

Model(*num Boxes*)

num Boxes - The number of boxes that build the model.

Particle(*p1,p2,p3,p4,p5,p6*)

p1- p6 Partial parameters, all are real values.

This keyword is to be used by future scattering models where the properties of the media's particles may need to be known. Please see the "Future Enhancement" section at the end of this manual for more detail.

Seed(*unsigned int*)

unsigned int - An unsigned integer value which seeds the random number generator. This Keyword is optional.

Wall(*num, Wall keyword*)

num - The wall number, an integer between one and six.

Wall keywords - The following are for use in the Temperature Mode
Wall(*num*, **element**, *local meshx, local meshy*)
{**Welement** Keyword must follow on the next line if **element** is used in **Wall**}
Wall(*num*, **flux**, *flux, emissivity,* $reflect_d$, $reflect_s$)
Wall(*num*, **join**, *jbox, jwall*)
Wall(*num*, **remove,** *far_temp*)
Wall(*num*, **temp**, *temperature, emissivity,*$reflect_d$, $reflect_s$)
Wall(*num*, **unknown**, *local meshx, local meshy, emissivity,* $reflect_d$, $reflect_s$)
{**Iterate** Keyword must follow immediately if **unknown** is used}

Wall keywords - View Factor or Radiation Exchange Factor Mode
Wall(*num*, **participate,** *emissivity*)
Wall(*num*, **receive,** *emissivity*)
Wall(*num*, **remove**)
Wall(*num*, **send,** *emissivity*)

element - Divides the wall into a meshed wall where each element has it's own temperature or flux, emissivity, and reflectivity.
flux - Energy per unit area per unit time sent from the wall.
join - The wall joins two boxes. It becomes a virtual wall.
participate - The wall participates in the radiation exchange factor calculation.
receive - The receiving wall in the view factor or radiation exchange factor mode.
remove - The wall has been removed. It does not participate.
send - The sending wall in the view factor or radiation exchange factor mode.

temp - The initial temperature of the wall.
unknown - The temperature or flux in unknown.

emissivity - The emissivity of the wall.
far_temp - The outside blackbody temperature.
flux - The energy per unit area per unit time sent out by the wall.
jbox - The ID number of the box which is joined to wall.
jwall - The wall number of the *jbox*.
local meshx - The number of meshed units in the wall local x coordinate system.
local meshy - The number of meshed units in the wall local y coordinate system.
$reflect_d$ - The diffuse reflection component.
$reflect_s$ - The specular reflection component.
temperature - The initial temperature of the wall.

Welement(*value, [temp or flux, emissivity,* $reflect_d$*,* $reflect_s$ *],* ..... for all elements)

value - value is either "temp" or "flux".
temp or flux - The element's temperature or flux depending on value.
$reflect_d$ - The diffuse reflection component.
$reflect_s$ - The specular reflection component.

 # - Comment

If the '#' symbol is used in the input file, the rest of the line is considered to be comments.

Program Execution

GnuRad can be executed three different ways; they are:

i) Program's name (i.e. gnurad)
ii) Program's name + input file (i.e. gnurad i*nput*)
iii) Program's name + input file + output file (i.e. gnurad *input output*)

If choice one is selected, GnuRad will prompt you for an input file name. Both choice one and two will write the output information directly to standard output, which is typically the terminal screen. When mode is TEMPERATURE, the output will be the tabulated format file, and when the mode is VIEW, the output will be the resulting view or radiation interchange factor. Choice three will send the output to file name specified with the appropriate file extension. Note, if the output file already exists, it will be over written without warning.

Sample Input Files

Introduction

A good way to become familiar with GnuRad is to run the *sample1.in* input file. This file and other sample files have been included on the distribution disk and are also listed in this manual. These files make a good way of introducing the GnuRad model building language. It is recommended that *sample1.in* be used since it is fully commented and it's output files are used in the description of the output files. Once GnuRad and the sample file has been placed in the same directory, type in the following at the command prompt:

C >gnurad *sample1.in* *sample1*

GnuRad will display the current process, such as building the model, sending the photon bundles, or iterating. Once the command prompt is back, four files will have been created. They are *sample1.tbl*, *sample1.tgr*, *sample1.csv* and *sample.plt*. Two of these output files (*.tbl* and *.tgr*) give their reports on the model in a tabular and text-graphic format and are self explanatory. The *.csv* and *.plt* files are used as input to spreadsheets and plotting programs. These output files are described further in the manual.

When comparing the output files from your run of *sample1.in* to the ones in the "Output" section of this manual, do not be alarmed at small variations in the answers. These small variations are not errors, but a result of the Monte Carlo process which GnuRad uses. For further information, please see the Program Theory section.

Sample One - Unknown Wall Temperature

```
#       Example one, a box enclosure with one unknown wall
#       temperature. All other walls have known properties
#       (i.e. temperature, emissivity, and reflectivity).
#       The media in this example is non participating
#       (i.e. clear).

Mode(Temperature)  # Model Mode is either temperature or view
Bundle(0.001)      # Bundle energy
Model(1)           # Number of enclosures in the model


  # This is Box one , X = 1, Y = 5, Z = 5 meters
  # Note, ':'  ';' and ',' can be used to separate data
Box(1;1,5,5)


    # For wall numbering, think of yourself as being inside of the box.
    # Wall 1 is on the bottom
    # temp = 100 K, emis = .5, diffuse reflection = 0.5, specular ref. = 0
Wall(1;temp,100,.5,.5,0)


    # Left wall 1 meter long 5 meters high
    # temp = 100 K, emis = 1.0,diffuse reflection = 0, specular ref. = 0
Wall(2;temp,100,1.0,0,0)


    # Back wall, 5 meters by 5 meters.
    # temp = 50 K, emis = 0.7, diffuse ref = 0.3, specular ref. = 0
Wall(3;temp,50,0.7,.3,0)


    # Top wall 1 meter long 5 meters high
Wall(4;temp,70,1.0,0,0)


    # Right wall 1 meter long 5 meters high
Wall(5;temp,100,0.6,.4,0)


   # Front wall with an unknown temperature
   # Emissivity = 0.4,diffuse reflection = .6, specular ref. = 0 , 5X5 grid
Wall(6;unknown,5,5,0.4,.6,0)

   # Iterate a maximum of 40 times or to a sent/receive ratio of 0.5 %
   # update factor = 1.0, and iterate to a net flux of 0.0 watts
Iterate(40,0.5,1.0,    0,0,0,0,0
                       0,0,0,0,0
                       0,0,0,0,0
                       0,0,0,0,0
                       0,0,0,0,0)

   # The media is non-participating, clear

Media(1,clear)
```

Sample Two - Unknown Media Temperature

```
#       Example two, a box enclosure with an unknown media
#       temperature. All walls have known properties
#       (i.e. temperature, emissivity, and reflectivity).
#       The media is participating.

Seed(12345)        # Apply the optional random number seed
Mode(Temperature)  # Model Mode is either temperature or view
Bundle(0.001)      # Bundle energy
Model(1)           # Number of enclosures in the model


  # This is Box one , X = 1, Y = 5, Z = 5 meters
  # Note, ':'  ';'  '{blank}' and ',' can be used to separate data

Box(1;1,5,5)


    # For wall numbering, think of yourself as being inside of the box.
    #Bottom
Wall(1;temp,100,.5,.5,0)


    # Left wall 1 meter long 5 meters high
    # temp = 100 K, emis = 1.0, diffuse reflection = 0, specular ref. = 0
Wall(2;temp,100,1.0,0,0)


    # Back wall, 5 meters by 5 meters.
    # temp = 50 K, emis = 0.7, diffuse reflection = .3, specular ref. = 0
Wall(3;temp,100,0.7,.3,0)


    # Top wall 1 meter long 5 meters high
Wall(4;temp,70,1.0,0,0)


    # Right wall 1 meter long 5 meters high
Wall(5;temp,100,0.6,.4,0)


   # Front wall is defined by using the element keyword
   # It is broken into a 3X3 grid
Wall(6; element,3,3)

   # Wall elements have a temperature of 100 K, diffuse scattering
   # and an emissivity of 0.4 or 0.9
Welement(temp, 100,0.9,.1,0  100,0.4,.6,0  100,0.4,.6,0
               100,0.4,.6,0  100,0.9,.1,0  100,0.4,.6,0
               100,0.4,.6,0  100,0.4,.6,0  100,0.9,.1,0)


Media(1;unknown,3,3,2,0.7,0.5,isotropic)

   # Iterate a maximum of 30 times or to a sent/receive ratio of 0.9 %
   # update factor = 1.0, and iterate to a net flux of 0.0 watts/m/m/m/element
Iterate(30,0.9,1.0,    0,0,0
                       0,0,0
                       0,0,0

                       0,0,0
                       0,0,0
                       0,0,0)
```

Sample Three - Unknown Wall & Media Temperature

```
#       Example three, a box enclosure with one wall and the media
#       temperature unknown. All other walls have known properties
#       (i.e. temperature, emissivity, and reflectivity).

Mode(Temperature)  # Model Mode is either temperature or view
Bundle(0.001)      # Bundle energy
Model(1)           # Number of enclosures in the model

  # This is Box one , X = 1, Y = 5, Z = 5 meters
  # Note, ':'  ';' and ',' can be used to separate data
Box(1;1,5,5)

    # For wall numbering, think of yourself as being inside of the box.
    #Bottom
Wall(1;temp,100,.5,.5,0)

    # Left wall 1 meter long 5 meters high
    # temp = 100 K, emis = 1.0
Wall(2;temp,100,1.0,0,0)

    # Back wall, 5 meters by 5 meters.
    # temp = 50 K, emis = 0.7, Diffuse ref =.3
Wall(3;temp,50,0.7,.3,0)

    # Top wall 1 meter long 5 meters high
Wall(4;temp,70,1.0,0,0)

    # Right wall 1 meter long 5 meters high
Wall(5;temp,100,0.6,.4,0)

   # Front wall with an unknown temperature
   # Emissivity = 0.4, Diffuse ref =.6, 5X5 grid
Wall(6;unknown,5,5,.4,.6,0)

   # Iterate a maximum of 30 times or to a sent/receive ratio of 0.8 %
   # update factor = 1.0, and iterate to a net flux of 0.0 watts
Iterate(30,0.8,1.0,    0,0,0,0,0
                       0,0,0,0,0
                       0,0,0,0,0
                       0,0,0,0,0
                       0,0,0,0,0)

   # The temperature of the media is unknown. Ke = 0.7, omega = 0.5
   # Isotropic scatter.
Media(1;unknown,3,3,3,0.7,0.5,isotropic)

   # Iterate a maximum of 30 times or to a sent/receive ratio of 0.9 %
   # update factor = 1.0, and iterate to a net flux of
   # 0.0 watts/m/m/m/element
   # except the center element which is emitting 20 watts/m/m/m
Iterate(30,0.9,1.0,    0,0,0
                       0,0,0
                       0,0,0

                       0,0,0
                       0,20,0
                       0,0,0

                       0,0,0
                       0,0,0
                       0,0,0)
```

Sample Four - Modest Problem 5.7

```
# All four walls have the same temperature and emissivity
# The top & bottom have the same temp. and emis. but they
# do not need to be the same as the walls.
#
#   q(bottom) = 4sigma(Tb^4 - Tw^4)/(4/eb + 2/ew -1)
#
#   q(wall) = - 1/2q(bottom)
#
#   Tb = 400 K (Temp of bottom/top) Tw = 200 K (Wall temp)
#   eb = 0.8 (Emissivity of bottom/top) ew = 0.4 (Wall emiss)
#   sigma = 5.67051x10^-8 (Stefan_Boltzmann constant)
#
#   q(top/bottom) = 604.85  q(per wall) = -302.42

# Note, comments are more concise, see lower sample numbers for
# more descriptive detail.

Seed(5146)
Mode(temperature)
 Bundle(0.01)
 Model(1)
  Box(1,1,1,1)
    Wall(1,temp,400,0.8,.2,0) # Bottom

    Wall(2,temp,200,0.4,.6,0) # Side
    Wall(3,temp,200,0.4,.6,0) # Side

    Wall(4,temp,400,0.8,.2,0) # Top

    Wall(5,temp,200,0.4,.6,0) # Side
    Wall(6,temp,200,0.4,.6,0) # Side

    Media(1,clear)
```

Sample Five - Modest Example 5.4

```
#       The actual problem is 2-D, to simulate, the length will be much
#       larger than the width or height. In this case, the length will be 100
#       meters, and the width will be 0.4 meters (40 cm), and the height will
#       be 0.3 meters (30 cm). The direct solution yields a power exchange of
#       4230 Watts per meter between the adjacent walls.

Mode(temperature)
 Bundle(5)
 Model(1)
  Box(1,100,0.4,0.3)

    Wall(1,temp,1000,0.3,.7,0) # Bottom ( T = 1000 K, Emis = 0.3 )
    Wall(2,temp,600,0.8,.2,0)  # Side   ( T = 600 K, Emis = 0.8 )
    Wall(3,temp,000,0.0,0,1)  # End    ( T = 0 K, Emis = 0.0 )

    Wall(4,temp,1000,0.3,.7,0) # Top    ( T = 1000 K, Emis = 0.3 )
    Wall(5,temp,600,0.8,.2,0)  # Side   ( T = 600 K, Emis = 0.8 )
    Wall(6,temp,000,0.0,0,1)  # End    ( T = 0 K, Emis = 0.0 )

    Media(1,clear)
```

Sample Six - View Factor

```
    # In this case, two surfaces form a 90 degree corner where their length
    # to width ratio is 1:3.  Direct value is  F1->2 = 0.12

    # The mode is set to View/ Radiation Interchange Factor
Mode(view)

   # The number of bundles to be sent per unit area
Bundle(20000)

Model(1)

    # The box is 1X3X3
Box(1,1.0,3.0,3.0)

    # Wall 1 will send the bundles, Emis = 1.0
Wall(1,send,1.0,0,0)

    # Wall 2 will receive the bundles, Emis = 1.0
Wall(2,receive,1.0,0,0)

    # Wall 3 - 6 have been removed
Wall(3,remove)
Wall(4,remove)
Wall(5,remove)
Wall(6,remove)

    # Since this is a view factor problem, the media will
    # be clear
Media(1,clear)
```

Sample Seven - Radiation Exchange Factor

```
#       In this example the radiation exchange between opposite walls of
#       a cube. All surfaces have an emissivity less than unity and the
#       media is also participating.

Mode(view)
Bundle(20000)
Model(1)
Box(1,1.0,1.0,1.0)

  # Wall 1 is the receiving wall and is opposite to wall 3
Wall(1,receive,0.9,.1,0)


  # Wall 2 is a participating wall, but the bundles it absorbs are
  # not used in the REF calculation
Wall(2,participate,0.5,.5,0)


   # Wall 3 is the sending wall
Wall(3,send,0.9,.1,0)


  # Wall 4 - 6 are participating walls
Wall(4,participate,0.5,.5,0)
Wall(5,participate,0.5,.5,0)
Wall(6,participate,0.5,.5,0)


  # The media is participating
Media(1,participate,0.7,0.8,isotropic)
```

Sample Eight - Joining Several Boxes

```
# Three boxes have been joined together, all the surfaces have
# an initial temperature of 100 K.
#  ^ Z
#  |
#  |--------- -------------
#  |         |             |
#  | Box 1   |    Box 3    |
#  |---------|-------------
#  | Box 2   |
#  |         |
#  |         |
#  |         |
#  X--------------> Y

Seed(4321)
Mode(Temperature)
Bundle(0.0001)

   # ’3‘ implies this model is built from three boxes
 Model(3)

    # Enclosure 1 is a box where X = 1, Y = 2, and Z = 1
Box(1;1,2,1)

 # Wall 1 is a joining wall to box 2, wall 4. It is not a real wall, but virtual
Wall(1;join,2,4)

# Wall 2 is described in terms of flux sent into the system 4.536 Watts/m/m =>100K
Wall(2;flux,4.536,0.8,.2,0)
Wall(3;temp,100,0.5,.5,0)
Wall(4;temp,100,0.5,.5,0)
   # Wall 5 joins Box 1 to Box 3
Wall(5;join,3,2)
Wall(6;temp,100,0.2,.8,0)

    # Box 1 has a Media which has a temp. = 100 K, Ke = 1.0 and ssal = 0.2 with
    # isotropic scattering.
Media(1,temp,100,1.0,0.2,isotropic)

#---------- Second Enclosure ------------
Box(2;1,2,4)

Wall(1;temp,100,0.5,.5,0)
Wall(2;temp,100,0.3,.7,0)
Wall(3;temp,100,0.3,.7,0)

   # Wall 4 joins Box 1 to Box 2
Wall(4;join,1,1)
Wall(5;temp,100,1.0,0,0)
Wall(6;temp,100,1.0,0,0)
Media(2,temp,100,0.3,0.8,isotropic)

#--------- Third Enclosure --------------
Box(3;1,3,1)

Wall(1;temp,100,0.5,.5,0)

   # Wall 2 joins Box 3 to Box 1
Wall(2;join,1,5)
Wall(3;temp,100,0.5,.5,0)
Wall(4;temp,100,0.9,.1,0)
Wall(5;temp,100,1.0,0,0)
Wall(6;temp,100,1.0,0,0)
Media(3;temp,100,0.5,0.7,isotropic)
```

Sample Nine - Using All Features

```
#  ^ Z
#  |                This is an example of two enclosures (boxes) joined
#  |---------       together to form a 1X1X1 cube. The walls which join
#  |         |      the two boxes together become virtual to the model
#  | Box 1   |      and do not interact in the radiation heat transfer.
#  |---------|      Note, Both boxes have a participating media with
#  | Box 2   |      different properties.
#  |         |
#  X-----------> Y Finally, all known temperatures and fluxes are based on
#                a value of 100 K. If the model runs correctly, all walls
#                and media should have a final temperature of 100 K and a
#                net flux of 0 Watts.

Seed(11516)     # The random number seed is 11516 this keyword is optional
Mode(Temperature) # The model will operate in the temperature mode
Bundle(0.0001)    # Each photon bundle send out has 0.0001 Watts of energy
Model(2)          # '2' implies this model is built from two boxes

    # Enclosure 1 is a box where X = 1, Y = 1, and Z = 1
Box(1,1,1,0.5)

   # Wall 1 is a joining wall to box 2, wall 4. It is not a real wall, but
   # virtual
Wall(1,join,2,4)

    # Wall 2 is described in terms of flux sent into the system
    # 4.536 Watts/m/m => 100 K
Wall(2,flux,4.536,0.8,.2,0)

    # Wall 3 uses the initial temperature mode
    # Emissivity = 0.5 and diffuse reflectivity = 0.5
Wall(3,temp,100,0.5,.5,0)

# Wall four has an unknown temperature, its value will be determined
# by iteration. It will be meshed into a 5X5 grid with an emis. of 0.6
Wall(4,unknown,5,5,.6,.4,0)

    # The keyword 'Iterate' allows for the following information
    # Max num. of iterations = 40 or stop at 5% error, Update factor = 1.0
    # Final flux value for the 5X5 mesh is 0.0
Iterate(40,0.5,1.0, 0,0,0,0,0 0,0,0,0,0 0,0,0,0,0 0,0,0,0,0 0,0,0,0,0)

Wall(5,temp,100,1.0,Diffuse)

# Wall 6 is broken down into a 3X3 mesh where each element property can be
# described separately.
Wall(6,element,3,3)
Welement(temp, 100,1.0,0,0   100,0.7,.3,0   100,0.1,0,.9
               100,0.5,0,.5  100,1.0,0,0    100,1.0,0,0
               100,0.8,.2,0  100,0.1,0.9    100,0.3,.7,0)

# Box 1 has a Media which has a temp.= 100 K, Ke = 1.0 and ssal = 0.2 with
# isotropic scattering.
Media(1,temp,100,1.0,0.2,isotropic)

#---------- Second Enclosure ------------
Box(2,1,1,0.5)

    # Wall 1 is broken into a 2X2 grid with unknown temperatures
Wall(1,Element,2,2)
WELEMENT(Unknown  1.0,0,0,  0.5,.5,0,
                  0.1,0,.9, 0.9,.1,0 )

    #Iteration parameters are the same as the other box's wall
iterate(40,0.5,1.0, 0, 0,
```

```
                    0, 0)

    # Walls 2 - 6 use the temp keyword except wall 4 which joins box 2 to
    # box 1.
Wall(2,temp,100,0.3,.7,0)

   # Wall three has been removed from the model, the far field temperature
   # is 100 K
Wall(3,remove,100)

Wall(4,join,1,1)
Wall(5,temp,100,1.0,0,0)
Wall(6,temp,100,1.0,0,0)

 # Box 2 has an unknown Media temperature, notice only a 0.7 % tolerance is
 # given in the iteration statement.
Media(2,unknown,2,2,4,0.7,0.9,isotropic)
Iterate(40,0.7,1.0,  0,0,
                     0,0,

                     0,0,
                     0,0,

                     0,0,
                     0,0,

                     0,0,
                     0,0 )
```

Output

GnuRad has five standard output files. Four are created in the **TEMPERATURE** mode, and one is created in the **VIEW** mode. All of the files can be identified by the three letter extension which is automatically placed on by GnuRad.

TEMPERATURE mode
output.tbl - Tabular format.
output.tgr - Text graphics format.
output.csv - Comma separated variable format.
output.plt - Plot file format.

VIEW mode
output.vwf - View factor or radiation interchange factor summary format.

Tabular Format

The Tabular format is always printed when GnuRad is run in the **TEMPERATURE** mode. When no output file name is specified, the tabular format is printed to the screen. The tabular format gives complete information about the model; first the basic information is reported, such as the local geometry, wall or media type, and the energy of a photon bundle. If the wall or media temperature were initially unknown, information about iteration parameters are also given. Next, information about the elements which make a wall or the media are reported in a tabular format. This includes the initial temperature, the number of bundles sent, the number of bundles received, the temperature based on number of bundles received, and the net flux. After the tabular information of the elements, the net flux of the wall or media is reported. Finally, The net flux of the entire model is reported at the end of the file. If all went well, this value should be approximately zero. This indicates that all the bundles were properly tracked. An example of the tabular format file is shown below. Note, descriptive comments have been placed inside the sample file and are shown in italic character.

(File - sample1.tbl)

```
 -------- BOX 1,  WALL 1 PROPERTIES --------
```

Basic wall information is listed first. For the wall type, it may have one of the following: F - Flux mode, T - Temperature mode, U - Unknown temp/flux, and V - virtual wall, it is where two boxes join.

```
 Wall type = T  Model element number = 1
 Local 'x' meshed to 1, section(s)
 Local 'x' is       1 meters
 Local 'y' meshed to 1, section(s)
 Local 'y' is       5 meters
 Bundle energy =   0.001
```

Each participating wall has the following tabulated print out for each of it's elements. The columns are marked as the following: Elem - Wall element number, Emis - Emissivity, Ref_D - Diffuse reflection component, Ref_S - Specular reflection component, #Sent - Number of bundles sent, S_temp - Temperature based on the number of bundles sent, S_Energy - Energy based on the number of bundles sent per unit time, # Received - The number of bundles received by the wall when the model was ran, R_temp -Temperature based on the number of bundles received, R_Energy - Energy based on the number of bundles received per unit time, Net Energy - The net energy of the element

```
 Energy, #Sent, and  #Received are per unit time (seconds).

 Elem Emis Ref_D   Ref_S    # Sent   S_temp    S_Energy  # Receive R_temp    R_Energy   Net
Energy
  1   0.500 0.500 0.000     14176 99.9996  14.1760        5543 79.0762  5.54300  8.63300

>>>> Net Energy for the wall =     8.633
```

(The net energy per unit time, of the entire wall is printed)

```
 -------- BOX 1,  WALL 2 PROPERTIES --------

 Wall type = T  Model element number = 1
 Local 'x' meshed to 1, section(s)
 Local 'x' is       1 meters
 Local 'y' meshed to 1, section(s)
 Local 'y' is       5 meters
 Bundle energy =   0.001


 Energy, #Sent, and  #Received are per unit time (seconds).

 Elem Emis Ref_D   Ref_S    # Sent   S_temp    S_Energy  # Receive R_temp    R_Energy   Net
Energy
  1   1.000 0.000 0.000     28352 99.9996  28.3520       10113 77.2809  10.1130  18.2390

>>>> Net Energy for the wall =    18.239


 -------- BOX 1,  WALL 3 PROPERTIES --------

 Wall type = T  Model element number = 1
 Local 'x' meshed to 1, section(s)
 Local 'x' is       5 meters
 Local 'y' meshed to 1, section(s)
 Local 'y' is       5 meters
 Bundle energy =   0.001


 Energy, #Sent, and  #Received are per unit time (seconds).

 Elem Emis Ref_D   Ref_S    # Sent   S_temp    S_Energy  # Receive R_temp    R_Energy   Net
Energy
  1   0.700 0.300 0.000      6202 49.9998  6.20200       41258 80.2994  41.2580  -35.0560

>>>> Net Energy for the wall =   -35.056


 -------- BOX 1,  WALL 4 PROPERTIES --------

 Wall type = T  Model element number = 1
 Local 'x' meshed to 1, section(s)
 Local 'x' is       1 meters
 Local 'y' meshed to 1, section(s)
 Local 'y' is       5 meters
 Bundle energy =   0.001
```

```
 Energy, #Sent, and  #Received are per unit time (seconds).

 Elem Emis Ref_D   Ref_S    # Sent   S_temp    S_Energy   # Receive R_temp    R_Energy   Net
Energy
  1   1.000 0.000 0.000      6807 69.9989  6.80700        9968 77.0024  9.96800  -3.16100

>>>> Net Energy for the wall =    -3.161


 -------- BOX 1,  WALL 5 PROPERTIES --------

 Wall type = T  Model element number = 1
 Local 'x' meshed to 1, section(s)
 Local 'x' is       1 meters
 Local 'y' meshed to 1, section(s)
 Local 'y' is       5 meters
 Bundle energy =   0.001


 Energy, #Sent, and  #Received are per unit time (seconds).

 Elem Emis Ref_D   Ref_S    # Sent   S_temp    S_Energy   # Receive R_temp    R_Energy   Net
Energy
  1   0.600 0.400 0.000     17011 99.9993  17.0110        5771 76.3180  5.77100  11.2400

>>>> Net Energy for the wall =     11.24


 -------- BOX 1,  WALL 6 PROPERTIES --------

 Wall type = U  Model element number = 1
 Local 'x' meshed to 5, section(s)
 Local 'x' is       5 meters
 Local 'y' meshed to 5, section(s)
 Local 'y' is       5 meters
 Bundle energy =   0.001
```

Iteration limits, minimum sent/received percent cutoff, and desired element flux values are given for unknown walls.

```
 Number of Iterations  40
 Max Iteration no. allowed 40
 Final flux error was set to maximum of  0.5 percent.
 Iteration update factor =    1
 Element number 1, Desired flux     0 W/m/m
 Element number 2, Desired flux     0 W/m/m
 Element number 3, Desired flux     0 W/m/m
 Element number 4, Desired flux     0 W/m/m
 Element number 5, Desired flux     0 W/m/m
 Element number 6, Desired flux     0 W/m/m
 Element number 7, Desired flux     0 W/m/m
 Element number 8, Desired flux     0 W/m/m
 Element number 9, Desired flux     0 W/m/m
 Element number 10, Desired flux     0 W/m/m
 Element number 11, Desired flux     0 W/m/m
 Element number 12, Desired flux     0 W/m/m
 Element number 13, Desired flux     0 W/m/m
 Element number 14, Desired flux     0 W/m/m
 Element number 15, Desired flux     0 W/m/m
 Element number 16, Desired flux     0 W/m/m
 Element number 17, Desired flux     0 W/m/m
 Element number 18, Desired flux     0 W/m/m
 Element number 19, Desired flux     0 W/m/m
 Element number 20, Desired flux     0 W/m/m
 Element number 21, Desired flux     0 W/m/m
```

Element number 22, Desired flux   0 W/m/m
Element number 23, Desired flux   0 W/m/m
Element number 24, Desired flux   0 W/m/m
Element number 25, Desired flux   0 W/m/m

Energy, #Sent, and #Received are per unit time (seconds).

Elem	Emis	Ref_D	Ref_S	# Sent	S_temp	S_Energy	# Receive	R_temp	R_Energy	Net Energy
1	0.400	0.600	0.000	1297	86.9590	1.29700	1249	86.1431	1.24900	0.0480000
2	0.400	0.600	0.000	1095	83.3553	1.09500	997	81.4242	0.997000	0.0980000
3	0.400	0.600	0.000	1043	82.3476	1.04300	990	81.2809	0.990000	0.0530000
4	0.400	0.600	0.000	1039	82.2685	1.03900	965	80.7628	0.965000	0.0740000
5	0.400	0.600	0.000	912	79.6303	0.912000	877	78.8551	0.877000	0.0350000
6	0.400	0.600	0.000	830	77.7766	0.830000	936	80.1491	0.936000	-0.106000
7	0.400	0.600	0.000	658	73.3899	0.658000	620	72.3065	0.620000	0.0380000
8	0.400	0.600	0.000	570	70.8025	0.570000	573	70.8954	0.573000	-0.00300000
9	0.400	0.600	0.000	564	70.6154	0.564000	557	70.3953	0.557000	0.00700000
10	0.400	0.600	0.000	555	70.3320	0.555000	521	69.2292	0.521000	0.0340000
11	0.400	0.600	0.000	781	76.6024	0.781000	797	76.9918	0.797000	-0.0160000
12	0.400	0.600	0.000	517	69.0959	0.517000	519	69.1626	0.519000	-0.00200000
13	0.400	0.600	0.000	493	68.2797	0.493000	491	68.2103	0.491000	0.00200000
14	0.400	0.600	0.000	429	65.9468	0.429000	449	66.7024	0.449000	-0.0200000
15	0.400	0.600	0.000	432	66.0618	0.432000	472	67.5406	0.472000	-0.0400000
16	0.400	0.600	0.000	791	76.8464	0.791000	823	77.6121	0.823000	-0.0320000
17	0.400	0.600	0.000	600	71.7162	0.600000	598	71.6564	0.598000	0.00200000
18	0.400	0.600	0.000	521	69.2292	0.521000	566	70.6779	0.566000	-0.0450000
19	0.400	0.600	0.000	481	67.8603	0.481000	508	68.7932	0.508000	-0.0270000
20	0.400	0.600	0.000	477	67.7188	0.477000	472	67.5406	0.472000	0.00500000
21	0.400	0.600	0.000	1052	82.5246	1.05200	1045	82.3870	1.04500	0.00700000
22	0.400	0.600	0.000	886	79.0566	0.886000	881	78.9448	0.881000	0.00500000
23	0.400	0.600	0.000	819	77.5177	0.819000	851	78.2640	0.851000	-0.0320000
24	0.400	0.600	0.000	790	76.8221	0.790000	778	76.5287	0.778000	0.0120000
25	0.400	0.600	0.000	734	75.4230	0.734000	726	75.2166	0.726000	0.00800000

>>>> Net Energy for the wall =   0.105

-------- BOX 1 MEDIA PROPERTIES ---------

The Media is Clear !

For the media type, it may have one of the following: C - Clear (non participating), F - Flux mode, T - Temperature mode, U - Unknown temp/flux
When the media is non-participating, no information other than the line shown below is printed.

Current types of scattering by the media are only isotropic, it is denoted by 'I'.

>>>>>>> Net Energy of the model = -5.88418e-15 <<<<<<<<

A check of the model is made by summing all the walls and media flux, if all bundles have been accounted for, Net Flux should be approximately zero (-5.9×10^{-15} *is approx. zero).*

Text-Graphic Format

The text-graphic format is an ASCII text file which is formatted to look like graphics. When viewing this file or printing it, the text characters should be fixed width. A sample of the file is shown below. Note, descriptive comments have been placed inside the sample file and are shown in italic character.

(File - sample1.tgr)

```
***** Element description *****
Element Number - The element number as described in the User Manual
[Resulting Temp - Printed only if it was previously unknown]
Net Energy/time - (Bundle Energy)*( #Received - #Sent)
```

```
 ----------------
| Element Number |
| Resulting Temp |
| Net Flux       |
 ----------------
```

Wall 1 was described as a 1X1 element in the input file. The net flux on element 1 which is also the net flux of the wall is listed as 8.633, these values compare directly with the tabular format file.

```
 -------- BOX 1, Wall 1 -------------

 local y

 ^
 |
 -----------
 | El#   1  |
 |    8.633 |
 ---------------> local x


 -------- BOX 1, Wall 2 -------------

 local y

 ^
 |
 -----------
 | El#   1  |
 |   18.239 |
 ---------------> local x


 -------- BOX 1, Wall 3 -------------

 local y

 ^
 |
 -----------
 | El#   1  |
 |  -35.056 |
 ---------------> local x


 -------- BOX 1, Wall 4 -------------

 local y

 ^
 |
 -----------
 | El#   1  |
 |   -3.161 |
 ---------------> local x


 -------- BOX 1, Wall 5 -------------

 local y

 ^
 |
 -----------
 | El#   1  |
 |    11.24 |
```

```
 ---------------> local x


 -------- BOX 1, Wall 6 -------------
```

Wall 6 was defined as a 5X5 grid, which means it is made up of 25 elements. Notice the element numbering order. Since wall 6 temperature was initially unknown, the final temperature based on the number of ***bundles received*** *is also printed along with the local flux.*

```
 local y

 ^
 |
 -------------------------------------------------------
 | El#  21  | El#  22  | El#  23  | El#  24  | El#  25  |
 |   82.387 |  78.9448 |   78.264 |  76.5287 |  75.2166 |
 |    0.007 |    0.005 |   -0.032 |    0.012 |    0.008 |
 -------------------------------------------------------
 | El#  16  | El#  17  | El#  18  | El#  19  | El#  20  |
 |  77.6121 |  71.6564 |  70.6779 |  68.7932 |  67.5406 |
 |   -0.032 |    0.002 |   -0.045 |   -0.027 |    0.005 |
 -------------------------------------------------------
 | El#  11  | El#  12  | El#  13  | El#  14  | El#  15  |
 |  76.9918 |  69.1626 |  68.2103 |  66.7024 |  67.5406 |
 |   -0.016 |   -0.002 |    0.002 |    -0.02 |    -0.04 |
 -------------------------------------------------------
 | El#   6  | El#   7  | El#   8  | El#   9  | El#  10  |
 |  80.1491 |  72.3065 |  70.8954 |  70.3953 |  69.2292 |
 |   -0.106 |    0.038 |   -0.003 |    0.007 |    0.034 |
 -------------------------------------------------------
 | El#   1  | El#   2  | El#   3  | El#   4  | El#   5  |
 |  86.1431 |  81.4242 |  81.2809 |  80.7628 |  78.8551 |
 |    0.048 |    0.098 |    0.053 |    0.074 |    0.035 |
 ------------------------------------------------------------> local x


 The Media is Clear.


>>>>>>> Net Energy of the model = -5.88418e-15 <<<<<<<<
```

Comma Separated Variable Format

The "csv" formatted file allows output from GnuRad to be used as input to a spreadsheet program, such as Xspread, MS Excel, or Lotus 123. A sample of this file is shown below. This file is not designed for direct viewing, but should be loaded first into a spreadsheet where the commas are used to separate the information into the proper cell. Once imported into a spreadsheet, the format is similar to the tabular format file.

```
*** Box 1 Wall 1 Information***
Bundle Energy
     0.001
Loc x dim,      1 , Loc y dim,      5
Meshx,1, Meshy,1
  Elem,Emis.,  Reflect_S,  Reflect_D,    #  Sent,  S_temp,  S_Energy,  #  Receive,  R_temp,
R_Energy, Net Energy
   1, 0.500, 0.000, 0.500,      14176, 99.9996,   2.8352,        5543, 79.0762,   1.1086,
1.7266
```

```
*** Box 1 Wall 2 Information***
Bundle Energy
     0.001
Loc x dim,      1 , Loc y dim,      5
Meshx,1, Meshy,1
  Elem,Emis.,  Reflect_S,  Reflect_D,    #  Sent,  S_temp,  S_Energy,  #  Receive,  R_temp,
R_Energy, Net Energy
   1, 1.000, 0.000, 0.000,       28352, 99.9996,   5.6704,       10113, 77.2809,   2.0226,
3.6478

*** Box 1 Wall 3 Information***
Bundle Energy
     0.001
Loc x dim,      5 , Loc y dim,      5
Meshx,1, Meshy,1
  Elem,Emis.,  Reflect_S,  Reflect_D,    #  Sent,  S_temp,  S_Energy,  #  Receive,  R_temp,
R_Energy, Net Energy
   1, 0.700, 0.000, 0.300,        6202, 49.9998, 0.24808,       41258, 80.2994, 1.65032,
-1.40224

*** Box 1 Wall 4 Information***
Bundle Energy
     0.001
Loc x dim,      1 , Loc y dim,      5
Meshx,1, Meshy,1
  Elem,Emis.,  Reflect_S,  Reflect_D,    #  Sent,  S_temp,  S_Energy,  #  Receive,  R_temp,
R_Energy, Net Energy
   1, 1.000, 0.000, 0.000,        6807, 69.9989,   1.3614,        9968, 77.0024,   1.9936,
-0.6322

*** Box 1 Wall 5 Information***
Bundle Energy
     0.001
Loc x dim,      1 , Loc y dim,      5
Meshx,1, Meshy,1
  Elem,Emis.,  Reflect_S,  Reflect_D,    #  Sent,  S_temp,  S_Energy,  #  Receive,  R_temp,
R_Energy, Net Energy
   1, 0.600, 0.000, 0.400,       17011, 99.9993,   3.4022,        5771,   76.318,   1.1542,
2.248

*** Box 1 Wall 6 Information***
Bundle Energy
     0.001
Loc x dim,      5 , Loc y dim,      5
Meshx,5, Meshy,5
 Elem,Emis., Reflect_S, Reflect_D,  # Sent, S_temp, S_Energy, # Receive, R_temp, R_Energy, Net Energy
  1, 0.400, 0.000, 0.600,      1297,  86.959,   1.297,      1249, 86.1431,   1.249,    0.048
  2, 0.400, 0.000, 0.600,      1095, 83.3553,   1.095,       997, 81.4242,   0.997,    0.098
  3, 0.400, 0.000, 0.600,      1043, 82.3476,   1.043,       990, 81.2809,    0.99,    0.053
  4, 0.400, 0.000, 0.600,      1039, 82.2685,   1.039,       965, 80.7628,   0.965,    0.074
  5, 0.400, 0.000, 0.600,       912, 79.6303,   0.912,       877, 78.8551,   0.877,    0.035
  6, 0.400, 0.000, 0.600,       830, 77.7766,    0.83,       936, 80.1491,   0.936,   -0.106
  7, 0.400, 0.000, 0.600,       658, 73.3899,   0.658,       620, 72.3065,    0.62,    0.038
  8, 0.400, 0.000, 0.600,       570, 70.8025,    0.57,       573, 70.8954,   0.573,   -0.003
  9, 0.400, 0.000, 0.600,       564, 70.6154,   0.564,       557, 70.3953,   0.557,    0.007
 10, 0.400, 0.000, 0.600,       555,  70.332,   0.555,       521, 69.2292,   0.521,    0.034
 11, 0.400, 0.000, 0.600,       781, 76.6024,   0.781,       797, 76.9918,   0.797,   -0.016
 12, 0.400, 0.000, 0.600,       517, 69.0959,   0.517,       519, 69.1626,   0.519,   -0.002
 13, 0.400, 0.000, 0.600,       493, 68.2797,   0.493,       491, 68.2103,   0.491,    0.002
 14, 0.400, 0.000, 0.600,       429, 65.9468,   0.429,       449, 66.7024,   0.449,    -0.02
 15, 0.400, 0.000, 0.600,       432, 66.0618,   0.432,       472, 67.5406,   0.472,    -0.04
 16, 0.400, 0.000, 0.600,       791, 76.8464,   0.791,       823, 77.6121,   0.823,   -0.032
 17, 0.400, 0.000, 0.600,       600, 71.7162,     0.6,       598, 71.6564,   0.598,    0.002
 18, 0.400, 0.000, 0.600,       521, 69.2292,   0.521,       566, 70.6779,   0.566,   -0.045
 19, 0.400, 0.000, 0.600,       481, 67.8603,   0.481,       508, 68.7932,   0.508,   -0.027
 20, 0.400, 0.000, 0.600,       477, 67.7188,   0.477,       472, 67.5406,   0.472,    0.005
 21, 0.400, 0.000, 0.600,      1052, 82.5246,   1.052,      1045,  82.387,   1.045,    0.007
 22, 0.400, 0.000, 0.600,       886, 79.0566,   0.886,       881, 78.9448,   0.881,    0.005
 23, 0.400, 0.000, 0.600,       819, 77.5177,   0.819,       851,  78.264,   0.851,   -0.032
 24, 0.400, 0.000, 0.600,       790, 76.8221,    0.79,       778, 76.5287,   0.778,    0.012
 25, 0.400, 0.000, 0.600,       734,  75.423,   0.734,       726, 75.2166,   0.726,    0.008


 The Media is Clear !
```

Plot Format

This file allows the unknown wall or walls to have their temperatures and or fluxes plotted by the program "GNUPLOT". The original output file will need to be separated into two files. The first file should contain the information on how GNUPLOT is to plot the data; it is currently set to plot a 3-D contour plot. The second file should contain only the numerical data; a commented line in the output file will tell you where the data begins and ends. Finally, the reader should obtain the GNUPLOT user manual, since it will tell much more information on what types of plots can be produced.

(File - sample1.plt)

This is the part which describes plotting information, Note # symbol is comments in GNUPLOT.

```
  # This file is designed to be used in GNUPLOT plotting program.
 # GNUPLOT is available on many platforms, please find more
 # information on GNUPLOT (via Internet) for more detail.
 # BOX 1, Wall 1
 # Wall properties are not unknown or the number
 # elements is not FOUR or greater.
 # BOX 1, Wall 2
 # Wall properties are not unknown or the number
 # elements is not FOUR or greater.
 # BOX 1, Wall 3
 # Wall properties are not unknown or the number
 # elements is not FOUR or greater.
 # BOX 1, Wall 4
 # Wall properties are not unknown or the number
 # elements is not FOUR or greater.
 # BOX 1, Wall 5
 # Wall properties are not unknown or the number
 # elements is not FOUR or greater.
 # BOX 1, Wall 6
set parametric
set data style lines
set contour
set cntrparam bspline
set xlabel 'Local x'
set ylabel 'Local y'
 # Values are per unit time (seconds)
 # By default, the temp. plotted is based on the number of bundles
 # received. If the plot should be based on the number sent,
 #  change 'using 1:2:3' to 'using 1:2:4'
 # If the plot should be based on energy received,
 #  change 'using 1:2:3' to 'using 1:2:5'
 # If the plot should be based on energy sent,
 #  change 'using 1:2:3' to 'using 1:2:6'
 # If the plot should be based on net energy,
 #  change 'using 1:2:3' to 'using 1:2:7'
splot '<file.dat>' using 1:2:3 title 'Title'

 # Cut and paste the rest of this file into a specific file
 # name and replace <file.dat> with that file name.
 # Note; keep the file name and title in single quotes
```

This part is the data Local x, Local y, Temperature based on bundles received, Temperature based on bundles sent, Flux received, Flux Sent, and Net Flux. ***Note, Local x & y are based on the center of the***

element.

```
#---------------- <file.dat> -------------

      0.5        0.5     82.387    82.5246      1.249      1.297      0.048
      1.5        0.5    78.9448    79.0566      0.997      1.095      0.098
      2.5        0.5     78.264    77.5177       0.99      1.043      0.053
      3.5        0.5    76.5287    76.8221      0.965      1.039      0.074
      4.5        0.5    75.2166     75.423      0.877      0.912      0.035

      0.5        1.5    77.6121    76.8464      1.249      1.297      0.048
      1.5        1.5    71.6564    71.7162      0.997      1.095      0.098
      2.5        1.5    70.6779    69.2292       0.99      1.043      0.053
      3.5        1.5    68.7932    67.8603      0.965      1.039      0.074
      4.5        1.5    67.5406    67.7188      0.877      0.912      0.035

      0.5        2.5    76.9918    76.6024      1.249      1.297      0.048
      1.5        2.5    69.1626    69.0959      0.997      1.095      0.098
      2.5        2.5    68.2103    68.2797       0.99      1.043      0.053
      3.5        2.5    66.7024    65.9468      0.965      1.039      0.074
      4.5        2.5    67.5406    66.0618      0.877      0.912      0.035

      0.5        3.5    80.1491    77.7766      1.249      1.297      0.048
      1.5        3.5    72.3065    73.3899      0.997      1.095      0.098
      2.5        3.5    70.8954    70.8025       0.99      1.043      0.053
      3.5        3.5    70.3953    70.6154      0.965      1.039      0.074
      4.5        3.5    69.2292     70.332      0.877      0.912      0.035

      0.5        4.5    86.1431     86.959      1.249      1.297      0.048
      1.5        4.5    81.4242    83.3553      0.997      1.095      0.098
      2.5        4.5    81.2809    82.3476       0.99      1.043      0.053
      3.5        4.5    80.7628    82.2685      0.965      1.039      0.074
      4.5        4.5    78.8551    79.6303      0.877      0.912      0.035

------------- End of <file.dat> -------------
```

View Factor &Radiation Exchange Factor Format

The view factor or radiation exchange factor format file is created when GnuRad is in **VIEW** mode and an output file is specified when GnuRad is executed. The format of this file is similar to the tabular format but with the following exceptions. The wall type is either ’R’, ’S’, or ’P’. ’R’ signifies it is a receiving wall and it is used in the calculation of the view or radiation exchange factor. ’S’ signifies it is s sending wall and it is used in the calculation of the view or radiation exchange factor. Finally, ’P’ signifies the wall participates in the model but the bundles it receives are not used in the calculation of the view or radiation exchange factor.

(File - sample7.vwf)

Basic wall information is listed first. The wall type, may have one of the following: P - Participating, R - Receiving wall, S - Sending wall, and V - virtual wall, where two boxes join.

```
-------- BOX 1,  WALL 1 PROPERTIES --------

Wall type = R  Model element number = 1
Local ’x’ meshed to 1, section(s)
Local ’x’ is       1 meters
Local ’y’ meshed to 1, section(s)
```

```
Local 'y' is       1 meters

Elem  Emis  Ref_D Ref_S     # Sent     # Receive
  1   0.900 0.100 0.000          0        4735


-------- BOX 1,  WALL 2 PROPERTIES --------

Wall type = P  Model element number = 1
Local 'x' meshed to 1, section(s)
Local 'x' is       1 meters
Local 'y' meshed to 1, section(s)
Local 'y' is       1 meters

Elem  Emis  Ref_D Ref_S     # Sent     # Receive
  1   0.500 0.500 0.000          0        2471


-------- BOX 1,  WALL 3 PROPERTIES --------

Wall type = S  Model element number = 1
Local 'x' meshed to 1, section(s)
Local 'x' is       1 meters
Local 'y' meshed to 1, section(s)
Local 'y' is       1 meters

Elem  Emis  Ref_D Ref_S     # Sent     # Receive
  1   0.900 0.100 0.000      20000        3332


-------- BOX 1,  WALL 4 PROPERTIES --------

Wall type = P  Model element number = 1
Local 'x' meshed to 1, section(s)
Local 'x' is       1 meters
Local 'y' meshed to 1, section(s)
Local 'y' is       1 meters

Elem  Emis  Ref_D Ref_S     # Sent     # Receive
  1   0.500 0.500 0.000          0        2569


-------- BOX 1,  WALL 5 PROPERTIES --------

Wall type = P  Model element number = 1
Local 'x' meshed to 1, section(s)
Local 'x' is       1 meters
Local 'y' meshed to 1, section(s)
Local 'y' is       1 meters

Elem  Emis  Ref_D Ref_S     # Sent     # Receive
  1   0.500 0.500 0.000          0        2516


-------- BOX 1,  WALL 6 PROPERTIES --------

Wall type = P  Model element number = 1
Local 'x' meshed to 1, section(s)
Local 'x' is       1 meters
Local 'y' meshed to 1, section(s)
Local 'y' is       1 meters

Elem  Emis  Ref_D Ref_S     # Sent     # Receive
  1   0.500 0.500 0.000          0        1770


-------- BOX 1 MEDIA PROPERTIES ---------
```

For the media type, it may have one of the following: C - Clear (non participating), P - Participating. In this case, the media is participating and has isotropic scattering.

```
 Media Type = P  Scatter = I Model element number = 1
 'X' meshed to 1, section(s)
 'X' is       1 meters
 'Y' meshed to 1, section(s)
 'Y' is       1 meters
 'Z' meshed to 1, section(s)
 'Z' is       1 meters
 Extinction Coefficient (Ke) =     0.7
 Single scattering albedo (omega) = 0.800

 Elem    # Sent    # Receive
   1         0        2607

View or Radiation Exchange Factor =   0.23675
```

Future Enhancements

The source code of GnuRad includes the structure for two additional types of scattering. The proposed types are, **S2** and **S3** scattering. They can be invoked by specifying their names as the type of scatter in the **MEDIA** keyword.

i.e. **Media**(1: **temp**, 300, 0.7, 0.5, **S2**) or **Media**(1: **temp**, 300, 0.7, 0.5, **S3**)

What is needed to complete the S2 or S3 scatter type is the selection of the scattering model (Mie, Rayleigh, etc.). Once a model has been determined, the proper equations can be added into the "S2_scat" or "S3_scat" function. Both of these functions appear in the MEDIA **class** (source code file name is "media.C"). Expecting that additional particle data may be required, the keyword **PARTICLE** has been added to the global variable list of the MEDIA **class**. This will allow six additional parameters to be used by the scattering functions. All six of these parameters are of **double** type (Double precision).

Particle(*p1,p2,p3,p4,p5,p6*) where *p1 - p6* are double precision

If additional particle parameters are needed, the MEDIA header file and PARSER **class** must be changed.

References

Brewster M. Q. (1992), "Thermal Radiative Transfer & Properties", Wiley-Interscience, New York.
Modest ...

Appendix A: Model Theory

The proceeding sections give details on how various physical characteristics are handled by GnuRad.

Bundle Energy (per unit time)

E_{bundle} - Specified in the input file, it is constant for the entire run.

Number of Bundles to Send per Wall or Media Element (per unit time)

$$Num_{wall} = \frac{A\varepsilon\sigma T^4}{E_{bundle}} \qquad Num_{wall} = \frac{Aq}{E_{bundle}}$$

$$Num_{media} = \frac{4V1\text{-}\omega K_e \sigma T^4}{E_{bundle}} \qquad Num_{media} = \frac{Vq}{E_{bundle}}$$

A - element area	ε - emissivity
K_e - extinction coefficient	q - heat flux
V - element volume	ω - single scattering albedo
σ - Stefan-Boltzmann constant	T - Temperature

Note: $K_a = (1-\omega)K_e$ & $K_e = K_s + K_a$

Location to Send a Bundle

$x = x_i + Random_1 \times delta_x$
$y = y_i + Random_2 \times delta_y$
$z = z_i + Random_3 \times delta_z$

$Random_n$ - A random number between zero and one inclusively. 'n' is to indicate that a random number is only used once.

x_i - 'x' location of the i^{th} element	$delta_x$ - 'x' length of the element
y_i - 'y' location of the i^{th} element	$delta_y$ - 'y' length of the element
z_i - 'z' location of the i^{th} element	$delta_x$ - 'z' length of the element

Only x & y are needed for the Wall elements and they are in local coordinates.

X, y, & z are required for the Media and they are in global/ box coordinates.

Direction to Send Bundle

Wall (in local coordinates):

$\theta = \sin^{-1}([Random_1]^{0.5})$ $\varphi = 2\pi \times Random_2$
Polar and Azimuthal angles

$n_i = \sin(\theta)\cos(\varphi)$ $n_j = \sin(\theta)\sin(\varphi)$ $n_i = \cos(\theta)$
Direction vector components

Media (in global/ box coordinates):

$\theta = \cos^{-1}(1\text{-}2 \times Random_1)$ $\varphi = 2\pi \times Random_2$
Polar and Azimuthal angles

$n_i = \sin(\theta)\cos(\varphi)$ $n_j = \sin(\theta)\sin(\varphi)$ $n_i = \cos(\theta)$
Direction vector components

Absorption with the Wall or Media

Wall: Absorbed if *Random* $< \varepsilon$, otherwise reflected.
Media: Absorbed if *Random* $> \omega$, otherwise it is scattered.

Bundle Direction When Reflected Off Of The Wall

Diffuse reflectivity: A new random direction is calculated as stated in the "Direction to Send Bundle" section.

Specular reflective: The bundles direction is reflected off the wall.

$\theta_r=\theta$ $\varphi_r=\varphi+\pi$

Bundle Direction When Scattered by the Media

Isotropic scattering: A new random direction is calculated as stated in the "Direction to Send Bundle"section.

Bundle Interaction

τ_{int} = - log(*Random*)
$\tau_{\text{wall}} = l_{wall} \times K_e$

l_{wall} - Length to the intersecting wall
τ_{int} - Optical length to media interaction
τ_{wall} - Optical length to the intersecting wall

If τ_{int} is less than or equal to τ_{wall}, the bundle will interact with the media
If τ_{int} is greater than τ_{wall}, the bundle will interact with the wall

When a bundle has an interaction with the media or the wall and it is not absorbed, a new τ_{int}, τ_{wall}, and direction are calculated.

Wall Element Temperature

$$T = [(E_{bundle} \times n_{received})/(A\varepsilon\sigma)]^{0.25}$$

$n_{received}$ - the number of bundles received

Media Element Temperature

$$T = [(E_{bundle} \times n_{received})/(4VK_e(1-\omega)\sigma)]^{0.25}$$

$n_{received}$ - the number of bundles received

Wall & Media Element Incoming Flux

$$q = E_{bundle} \times n_{received}$$

Unknown Wall or Media Temperature

The following steps are used to determine an unknown wall or media temperature:

1. Initially, all of the elements of the unknown wall and/or media are set to send zero bundles.

$n_{send_1}=0$

2. The model is ran and the number of bundles received per element is recorded.
3. The percent error of an element based on net flux is calculated.

$Q_{net} = q_{final} \times A$

$Q_{sent} = n_{sent} \times A$

$Q_{received} = n_{received} \times A$

$Percent = 100 \times ABS(ABS(Q_{sent} - Q_{net})/Q_{received} - 1)$

ABS - Absolute value

4. The number of bundles to send for the next iteration is calculated.

$n_{send_{i+1}} = n_{send_i} + F_I\,(n_{received} - n_{send_i}) + (Aq_i/E_{bundle})$
Wall element

$n_{send_{i+1}} = n_{send_i} + F_I\,(n_{received} - n_{send_i}) + (Vq_i/E_{bundle})$
Media element

F_I - iteration factor (default value is one)
i - iteration number

5. The model is ran again.
6. Steps three through five are repeated until one of the following conditions is true:
i) The percent error for all of the elements is less than the maximum specified value.
ii) The iteration number is equal or greater than maximum specified value.

Appendix B: Programming Environment

GnuRad was developed on an Intel 486 platform using the RedHat Linux operating system (a Unix environment). Tools used in this environment included many of the Free Software Foundation programs such as GCC compiler, GNU Emacs, GDB (Debugger), RCS (Revision Control System, and many of the other miscellaneous items (grep, tar, ls, cp, etc). Other tools used in this project were the X11 R6 window environment and the AUIS word processor. The advantages of this system were: a true 32 bit multitasking OS, stability of Unix (I never encountered the system crashing), economics (all of the tools and the OS were free or near free), and portability. If further development of GnuRad is done, it is recommended that a Unix environment be used, such as Linux, since it will save many hours of work on the developer.