SAND2007-3528P

Linking INPAG-N to GoldSim

June 2007

Patrick Mattie
Don Kalinich

Sandia National Laboratories

Email: pdmatti@sandia.gov or dakalin@sandia.gov

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

Sandia National Laboratories

Overview

- Introduction
  - Requirements
  - Functional Design
  - Specifications
- Launch_INPEG DLL
- GoldSim model file that runs multiple realizations of INPEG-N

Sandia National Laboratories

Requirements

Link INPAG-N with GoldSim

1. Code must interface with GoldSim

   GoldSim requirements:

   - ✓ code compiled as a DLL
   - ✓ code written in C,C++, or FORTRAN

2. Code must run INPAG-N model

   - ✓ Launch the EXE
   - ✓ Iterate over multiple input files
   - ✓ Save output

Sandia National Laboratories

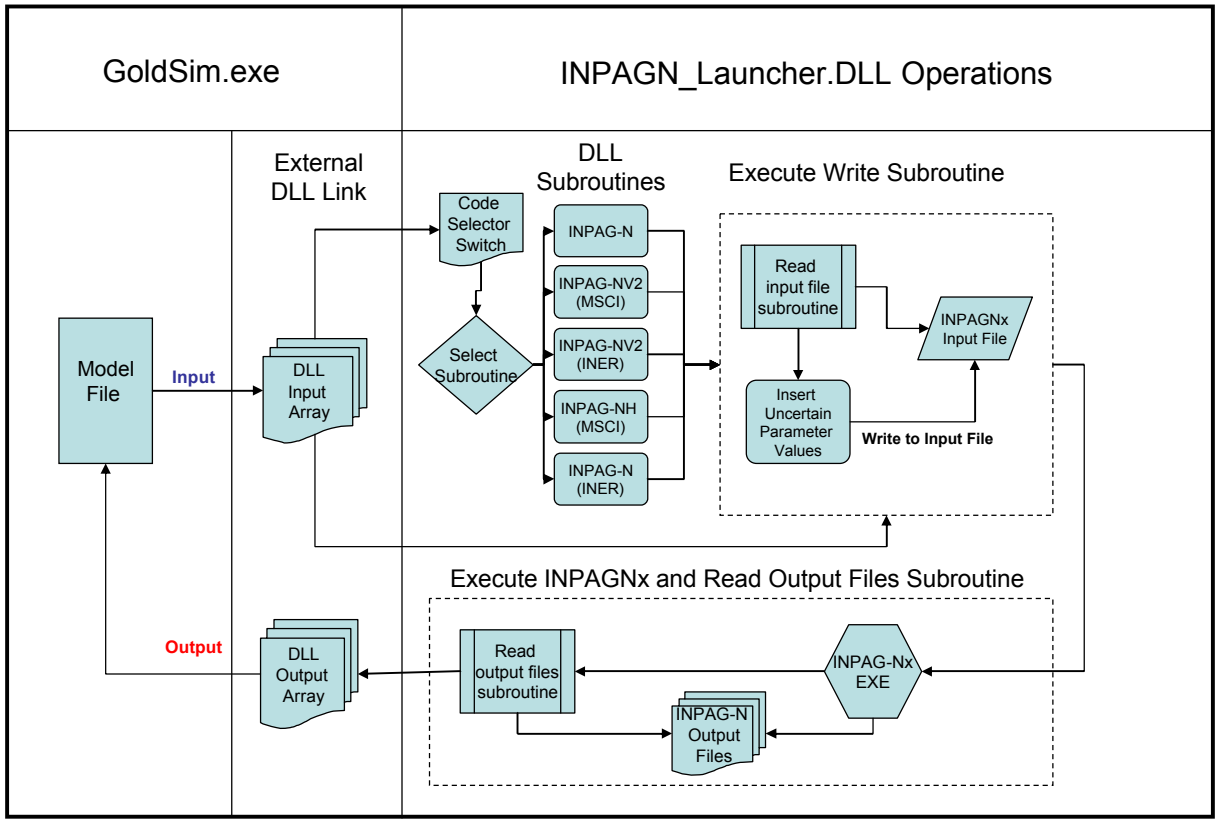
Functional Design

Need three parts:

1. GoldSim model file
2. DLL
3. INPAGN.EXE

Operations:

1. Input data is passed from the model file to the DLL
2. DLL Writes Input File
3. DLL Launches the EXE
4. DLL reads Output files and the data is loaded into an array read by the model file
5. GoldSim Saves Output



Sandia National Laboratories

Specifications

- GoldSim is a probabilistic code the DLL was designed to be executed over multiple realizations
- DLL coded to accept the input data required for five versions of INPAG-N
- DLL able to write five different input file formats for INPAG-N
- DLL able to read five different output file types for INAPG-N

Sandia National Laboratories

GoldSim External Functions –
Linking a Model File to a DLL

Sandia National Laboratories

Understanding External DLL's

Refer to GoldSim User's Guide – Appendix C: *Understanding External (DLL) Elements* (GTG, 2006)

- **The external functions are bound to the GoldSim executable code at run time using DLL technology.**
- **The function name and argument list (set of input and output data for the function) are specified in the GoldSim model file using an *External Element* function.**
- **Data are passed from GoldSim to the external function and back again via arrays of double percision floating point arguments.**

Sandia National Laboratories

External Function Format

- GoldSim calls the DLL in a defined sequence
  - Method = 0 *[XF_INITIALISE]*
    - Initialize DLL (Load into Memory)
  - Method = 2 *[XF_REP_VERSION]*
    - DLL Reports DLL Version
  - Method = 3 *[XF_REP_ARGUMENTS]*
    - DLL reports # of input arguments expected
    - DLL reports # of output argument expected
  - Method = 1 *[XF_CALCULATION]*
    - In() array arguments are loaded
    - Begin DLL Calculation
    - Pass out() array argument to GoldSim
  - Method = 99 *[XF_CLEANUP]*
    - Deallocate Arrays
    - Close DLL and Release from Memory

Sandia National Laboratories

Launch_INPAG DLL

Sandia National Laboratories

Launch_INPAGN DLL

- Source Code Written in Fortran 90
- Compiled using “Digital Visual Fortran” from Intel, as recommended in the GoldSim User Manual (Appendix C, p. 556)
- Source code could be translated and compiled in C or C++
- Code was developed in a modular format, using subroutines to read and write the INPAG-N input blocks

Sandia National Laboratories

Launch_INPAGN DLL Logic

- **Method = 0**
  - **Code Initializes**
- **Method = 2**
  - **Report Version = 3.0**
- **Method = 3**
  - **# inputs arguments = (57+(7*nrni)**
    - **Number of radionuclides read from RN.DAT file (nrni)**
    - **7 inputs arguments are vectors by # of RN's**
    - **57 input arguments are single values**
  - **# ouput arguments = (nrni*200*3)+200**
    - **200 maximum output times in INPAG-N**
    - **3 output files maximum for INPAG-N**
- **Method = 1**
  - **Calculation (details in following slides)**
- **Method =99**
  - **Deallocate Arrays**
  - **Close Files**

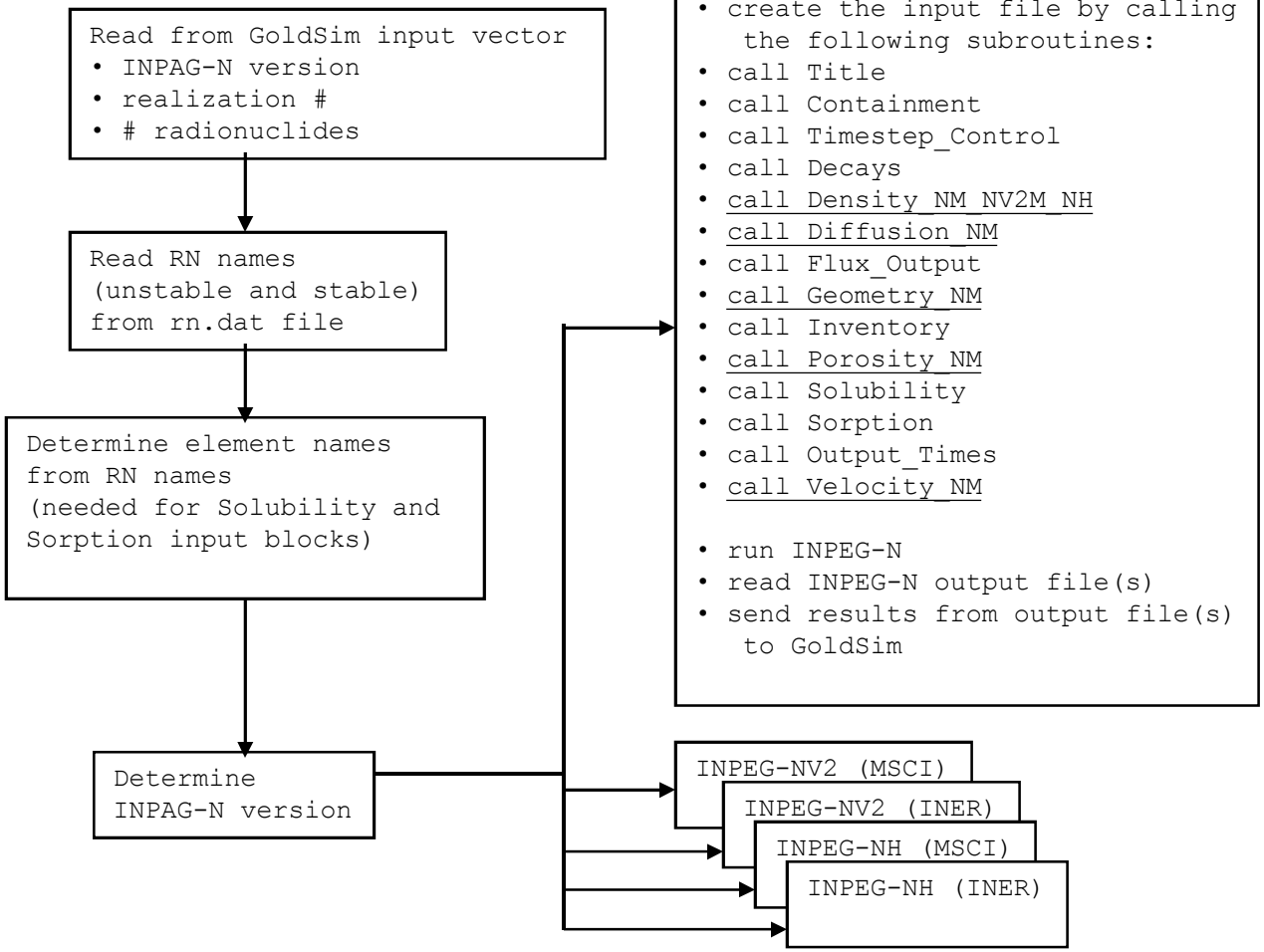
Sandia National Laboratories

Launch_INPAGN DLL Calculation Logic

- Method = 1 Calculation:
  - Determine which version of INPAG-N is to be run
    - based on flag value passed from GoldSim
  - Read input
    - RN names from the rn.dat file
    - constants and stochastics from the GoldSim input vector
  - Write the INPAG-N input to an input data file
  - Run INPAG-N using the input data file
  - Read INPAG-N results from its output file(s) and pass them into GoldSim

Sandia National Laboratories

DLL Logic Flowchart



Sandia National Laboratories

INPEG-N Input Blocks

- INPEG-N input is read in the following order:
  - Containment
  - Timestep_Control
  - Decays
  - Density
  - Diffusion
  - Flux Output
  - Geometry
  - Inventory
  - Porosity
  - Solubility
  - Output_Times
  - Velocity
- Blocks are read from the input vector and written to the input file by subroutines
- Underlined blocks have different inputs for different INPEG-N versions
- Separate subroutines have been written to handle the different input for different INPEG-N versions

Sandia National Laboratories

Input to DLL from GoldSim

- Control parameters
  - INPAG-N version to run
  - realization number
  - number of radionuclides in model problem
- Input for INPAG-N modules
  - some modules (e.g., DECAY) have the same input for all INPAG-N versions
  - other modules (e.g., GEOMETRY) have different inputs for different INPAG-N versions
- GoldSim input vector is sized to account for all potential INPAG-N inputs
  - 58 scalar inputs
  - 7 radionuclide vector inputs

Sandia National Laboratories

Example of DLL Input Vector

```
!  Description of DLL input vector
!                                                          INPAG-N version
! index           parameter                          N(M)   NV2(M)  NV2(I)  NH(M)  NH(I)
!  1        -- INPAG-N selector                       y      y       y       y      y
!  2        -- realization                            y      y       y       y      y
!  3        -- # of radionuclides (nRN)               y      y       y       y      y
!-CONTAINMENT-------------------------------------------------------------------------
!  4        -- FAIL-TIME                              y      y       y       y      y
!  5        -- INV-START-TIME                         y      y       y       y      y
!  6        -- ALTER-TIME                             y      y       y       y      y
!-TIMESTEP-CONTROL--------------------------------------------------------------------
!  7        -- TOLERANCE                              y      y       y       y      y
!  8        -- MIN-TIMESTEP                           y      y       y       y      y
!-DECAYS------------------------------------------------------------------------------
!  9        -- DECAYS(1)                              y      y       y       y      y
!  8+nRN    -- DECAYS(nRN)                            y      y       y       y      y
!-DENSITY-----------------------------------------------------------------------------
!  9+nRN    -- CANISTER                               y      y       y       y      y
! 10+nRN    -- BENTONITE                              y      y       y       y      y
! 11+nRN    -- ROCK                                   y      y       y       y      y
! 12+nRN    -- EDZ                                    NO     NO      y       NO     NO
! 13+nRN    -- TUNNEL                                 NO     NO      y       NO     NO
!-DIFFUSION---------------------------------------------------------------------------
! 14+nRN    -- CANISTER                               y      y       y       y      y
! 15+nRN    -- BENTONITE                              y      y       y       y      y
! 16+nRN    -- EDZ                                    y      y       y       y      y
! 17+nRN    -- ROCK-FRACTURE                          NO     y       y       y      y
! 18+nRN    -- ROCK-MATRIX                            NO     y       y       y      y
! 19+nRN    -- TUNNEL                                 NO     y       y       NO     NO
```

Sandia National Laboratories

Example DLL Input Vector

	parameter	input	i =
	INPAG-N selector	1	1
	Realization #	1	2
	# of radionuclides	4	3
Containment	FAIL-TIME		4
	INV-START-TIME		5
	ALTER-TIME		6
Timestep-Control	TOLERANCE		7
	MIN-TIMESTEP		8
Decay	DECAY(1)		9
	DECAY(2)		10
	DECAY(3)		11
	DECAY(4)		12
Density	CANISTER		13
	BENONITE		14
	ROCK		15
	EDZ	0	16
	TUNNEL	0	17
Diffusion	CANISTER		18
	BENONITE		19
	EDZ		20
	ROCK-FRACTURE	0	21
	ROCK-MATRIX	0	22
	TUNNEL	0	23

INPEG-N (MSCI)

four radionuclides in this example; therefore the DECAY subroutine reads in four values

a value of 0 is put in the input vector for inputs that do not exist in a given INPEG-N version

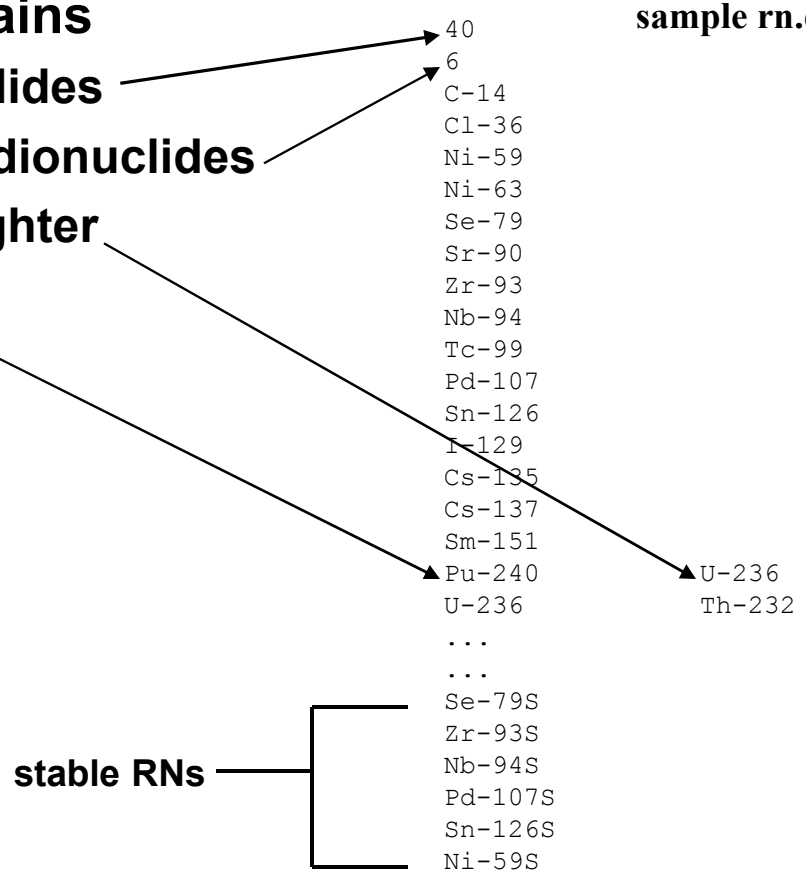
inputs 16 and 17 are skipped by Density_NM_NV2M_NH subroutine; counter is incremented to i=18

inputs 21, 22, and 23 are skipped by Diffusion_NM subroutine; counter is incremented to i=24

Sandia National Laboratories

rn.dat Input

- **rn.dat file contains**
  - **# of radionuclides**
  - **# of stable radionuclides**
  - **parent daughter**



Sandia National Laboratories

INPAG-N INPUT Files

- INPAG-N (MSCI)
  - inpag_n.dat
- INPAG-NV2 (MSCI)
  - inpag_nv2.dat
- INPAG-NV2 (INER)
  - inpag_nv2i.dv2
- INPAG-NH (MSCI)
  - inpag_nhm.dat
- INPAG-NH (INER)
  - inpag_nhi.dh1

Sandia National Laboratories

INPAG-N Output Files

- INPAG-N (MSCI)
  - basenf.flx
    - RN release rates
- INPAG-NV2 (MSCI)
  - inpag_nv2.flx
    - RN release rates (total, tunnel, rock)
- INPAG-NV2 (INER)
  - inpag_nv2i.fxn
    - RN release rates (Total)
  - inpag_nv2iN.fxn
    - RN release rates (Tunnel)
  - inpag_nv2iR.fxn
    - RN release rates (Rock)

Sandia National Laboratories

INPAG-N Output Files

- INPAG-NH (MSCI)
  - inpag_nhm.flx
    - RN release rates (total, fracture, rock)
- INPAG-NH (INER)
  - inpag_nhi.fxn
    - RN release rates (Total)
  - inpag_nhiF.fxn
    - RN release rates (Fracture)
  - inpag_nhiR.fxn
    - RN release rates (Rock)

Sandia National Laboratories

INPAG-N Output Files

- Separate subroutines used to read INPAG-N output files(s)
- Output passed back into GoldSim as a 2-D table(s) of RN release rate (mol/yr) vs. time

Sandia National Laboratories

Algorithm for Reading Output Times

- **Used for reading Monitor Scientific versions of INPAG-N, INPAG-NV and INPAG-NH:**

****

If # intervals = 0 then

 # time steps =
(log(end_time) – log(start_time)) * # per decade

else

 # time steps =
(end_time – start_time)/(# intervals) + 1

Sandia National Laboratories

INER Algorithm for added TS

- **Used for reading INER versions of INPAG-NV & INPAG-NH:**

```
!*******************************************************************************
subroutine INER_ADD_TS

        write (99,*) '************ INER_ADD_TS  SUBROUTINE **************'

! calculate number of timesteps in INER Modified Code Output files
  if(interval.eq.0) then
                        nb_timesteps = (( LOG10(end_time) - LOG10(start_time)) * no_per_decade)
                        write (99,*) 'NO_PER_DECADE: nb_timesteps= ', nb_timesteps
!  ADDED BY DR.JU TO INCREASE OUTPUT
                        IF (start_time.GE.999999.9) THEN
                          KSTEP=(no_per_decade*6)
                        ELSEIF (start_time.GE.99999.9) THEN
                          KSTEP=(no_per_decade*5)
                        ELSEIF (start_time.GE.9999.9) THEN
                          KSTEP=(no_per_decade*4)
                        ELSEIF (start_time.GE.999.9) THEN
                          KSTEP=(no_per_decade*3)
                        ELSEIF (start_time.GE.99.9) THEN
                          KSTEP=(no_per_decade*2)
                        ELSEIF (start_time.GE.9.9) THEN
                          KSTEP=no_per_decade
                        ELSE
                          KSTEP=0
                        ENDIF
                        write (99,*) 'NO_PER_DECADE: KSTEP(', KSTEP,') + (',nb_timesteps,') nb_timesteps = nb_timesteps(adjusted)'
                        nb_timesteps= nb_timesteps+KSTEP

  else
                        nb_timesteps = (((end_time - start_time)/interval))+1
                        write (99,*) 'INTERVAL: nb_timesteps= ', nb_timesteps
  endif

                        write (99,*) 'nb_timesteps (adjusted)= ', nb_timesteps
end subroutine INER_ADD_TS
```

Sandia National Laboratories

Array Deallocation and File Closure

- Once the program has completed its calculations
  - all allocated arrays are deallocated
  - input and output files are closed
- If arrays are not deallocated, the associated memory becomes inaccessable. Over multiple realizations (i.e., mutiple executions of the DLL) enough memory can be “lost” such that it impacts execution of the DLL, GoldSim, or other programs.
- Ensuring all files are closed prevents possible errors when attempting to reopen the files

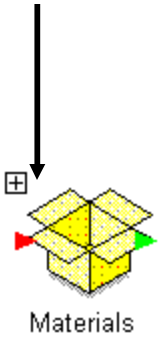
Sandia National Laboratories

GoldSim Model File That Runs INPAG-N

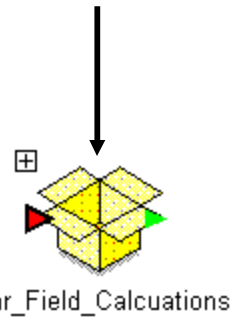
Sandia National Laboratories

“Root” Container

Radionuclide definition



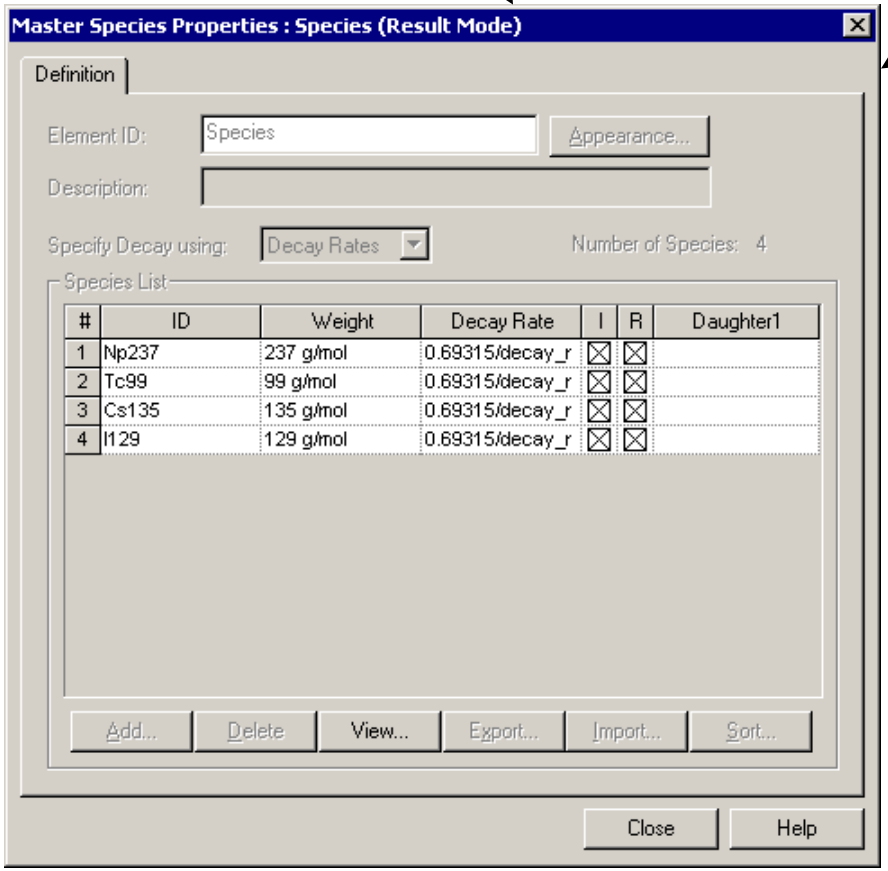
INPEG-N calculation elements

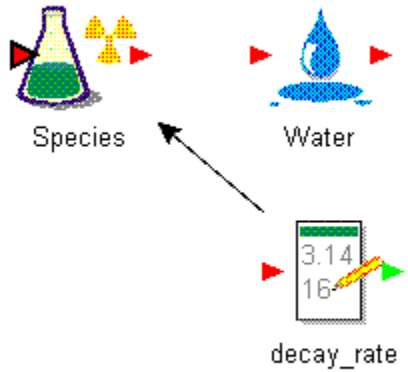


Sandia National Laboratories

Materials Container

Declaration of radionuclides

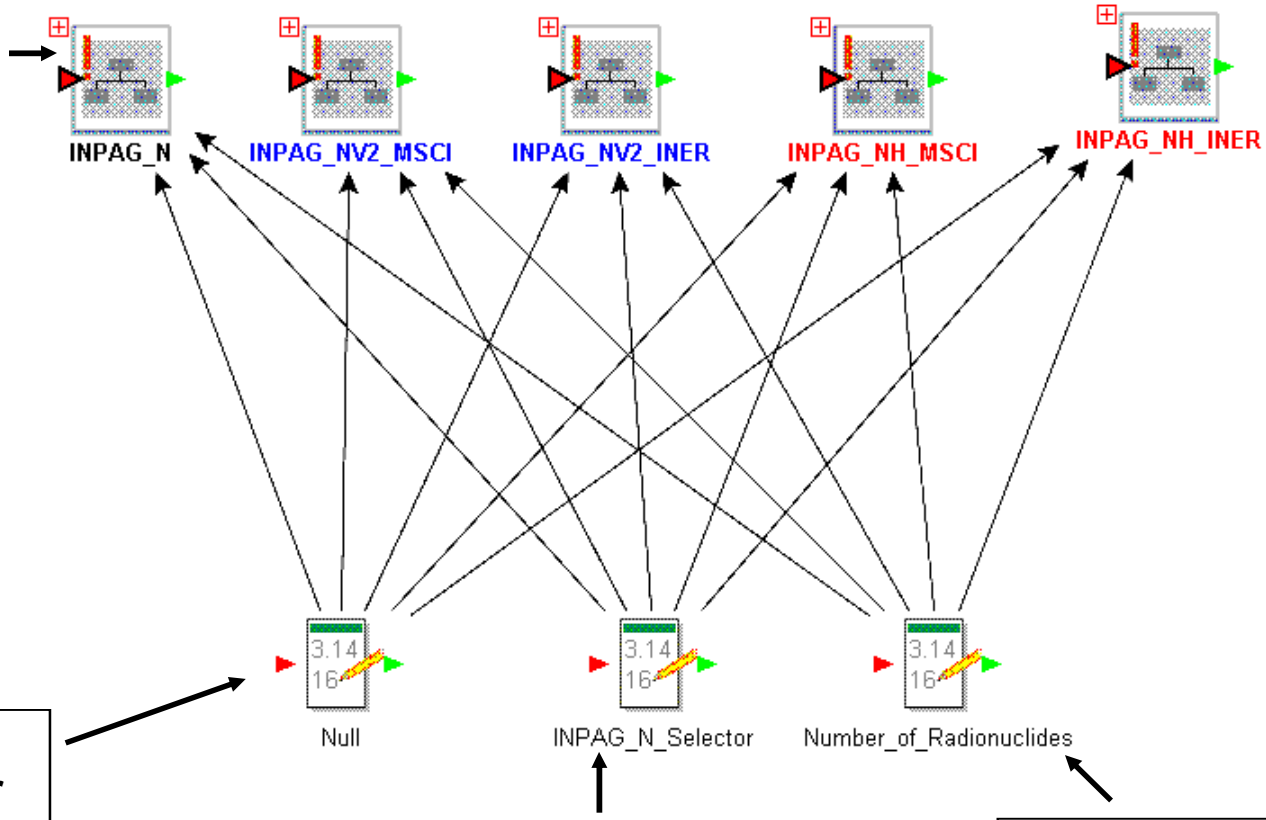




Radionuclide decay rates

Sandia National Laboratories

Near_Field_Calculations Container



Containers for individual INPAG versions

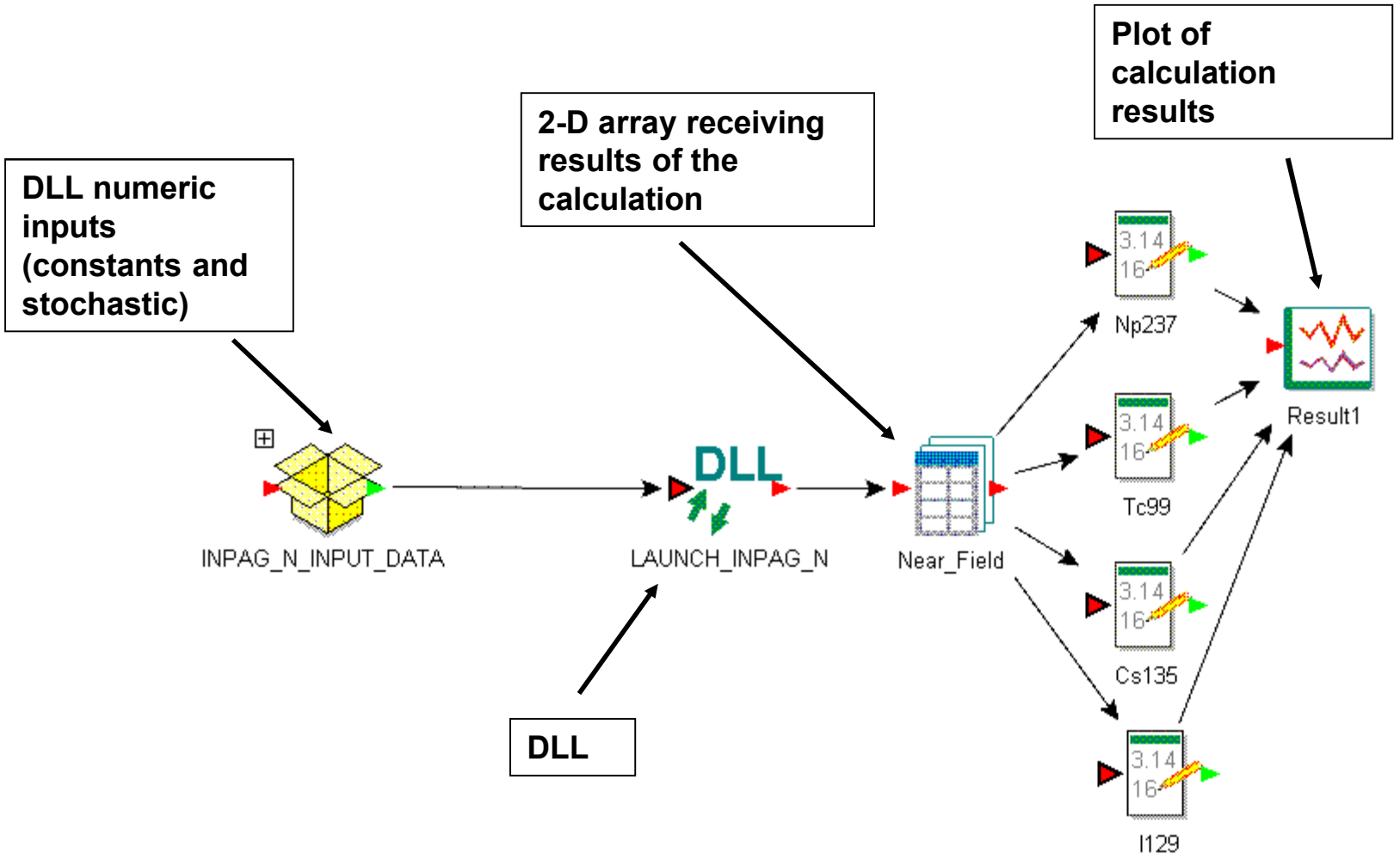
Null value (0) placeholder for non-existent input

INPAG-N flag
=1; INPAG-N (MSCI)
=2; INPAG-NV2 (MSCI)
=3; INPAG-NV2 (INER)
=4; INPAG-NH (MSCI)
=5; INPAG-NH (INER)

Number of radionuclides

Sandia National Laboratories

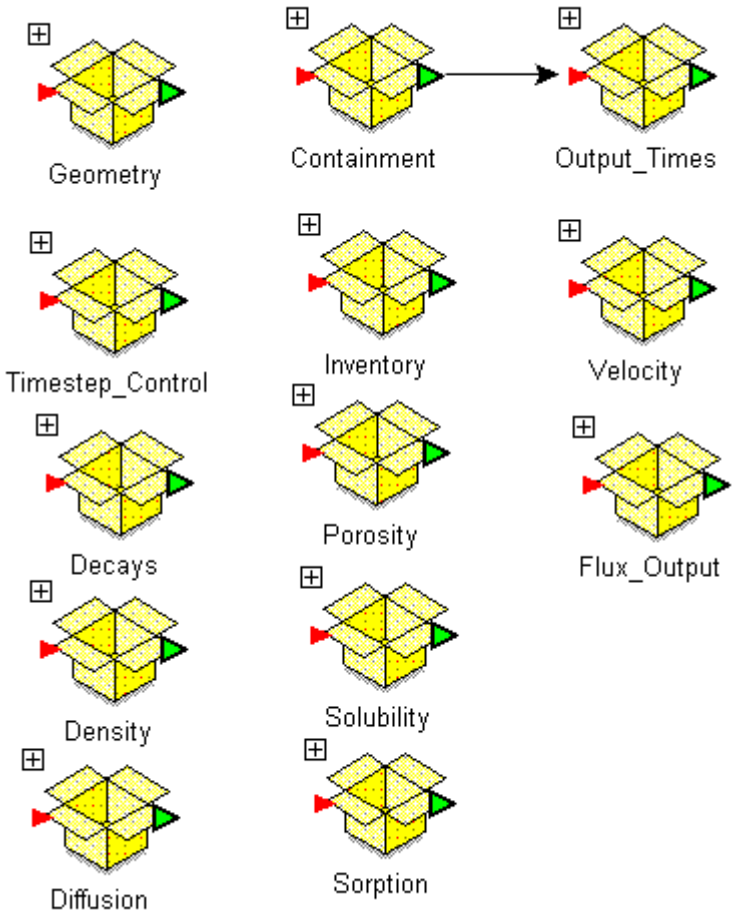
INPAG_N Container



Sandia National Laboratories

INPAG_N_Input_Data Container

Input for each block is in a separate container

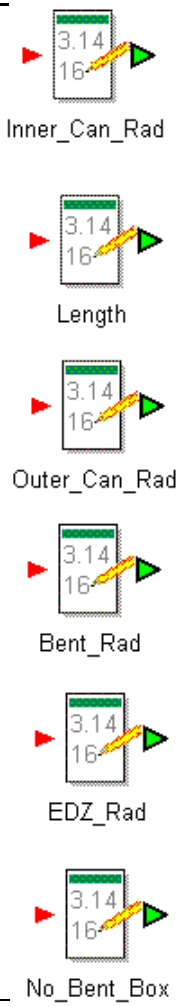


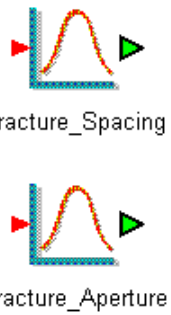
Sandia National Laboratories

Geometry Container

Geometry block inputs

Constant inputs



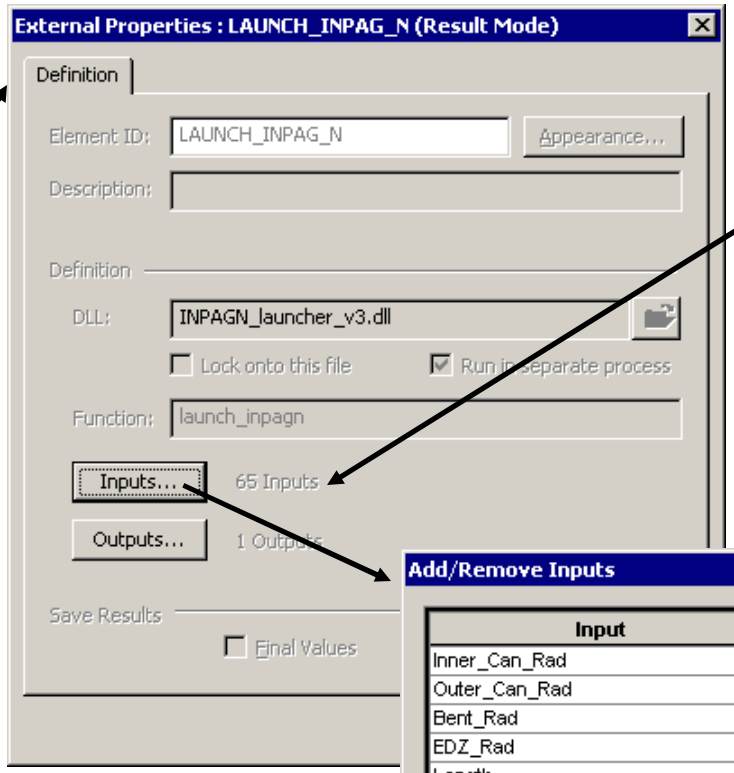


Stochastic inputs

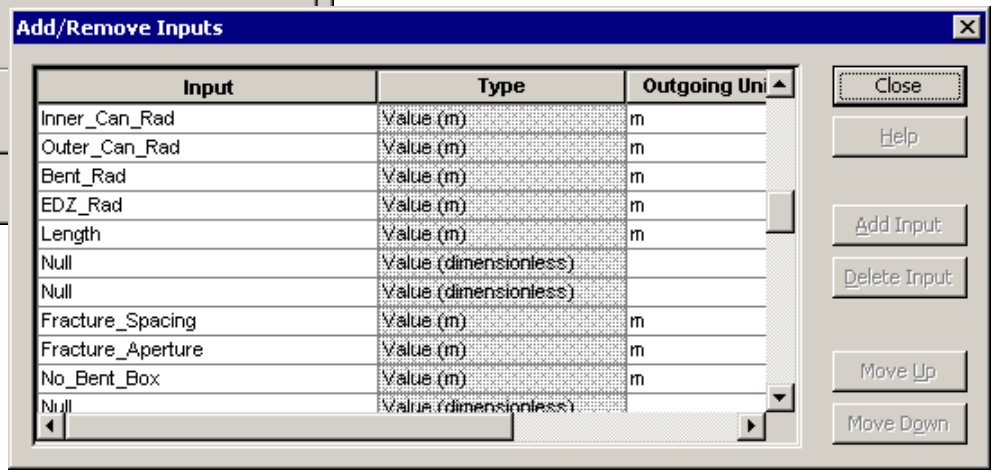
Note that input types (constant or stochastic) can be changed by deleting the element and replacing it with an element of the other type but with the same name

Sandia National Laboratories

DLL Element



65 inputs
- **58 scalar inputs**
- **7 vector inputs**
- **(by # of RN)**



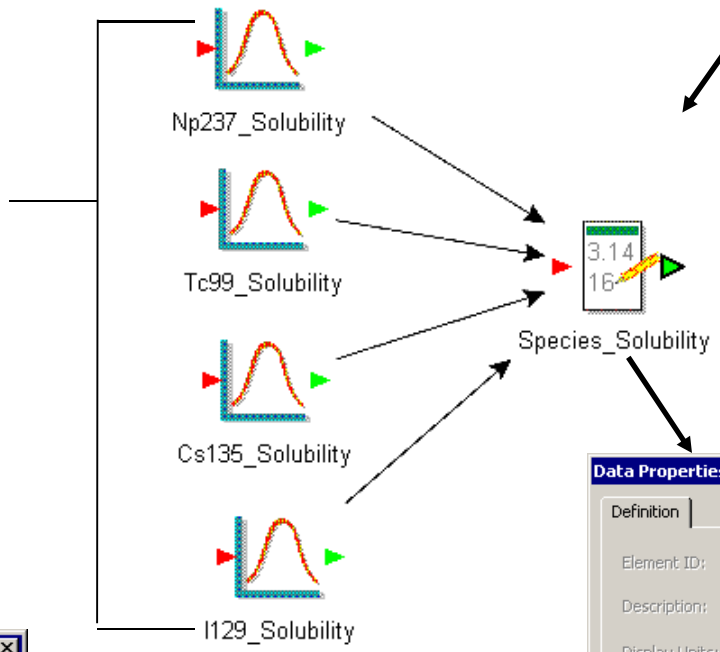
Showing Geometry block input (note Null values for non-existent inputs)

Sandia National Laboratories

Radionuclide Dependent Inputs

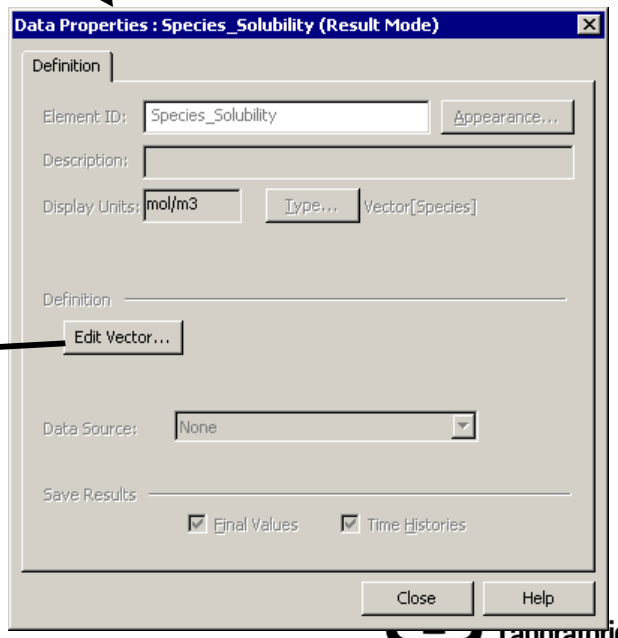
Radionuclide dependent inputs are declared for each radionuclide

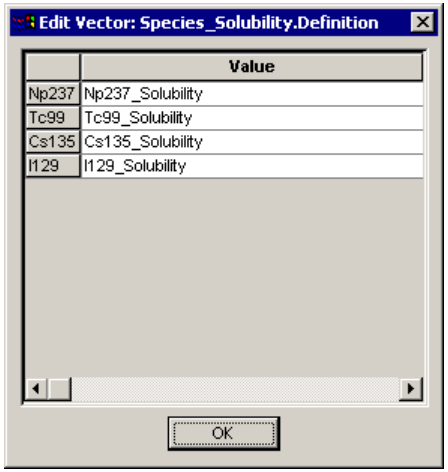
Elements must be added/deleted corresponding to the addition/deletion of radionuclides from the problem



Input values are grouped into an array, whose size is defined by the number of species

The links to this element must be updated if the number of radionuclides is changed

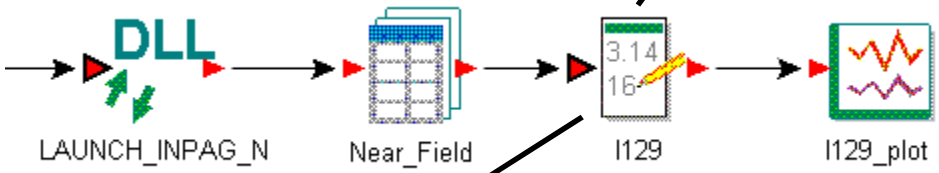


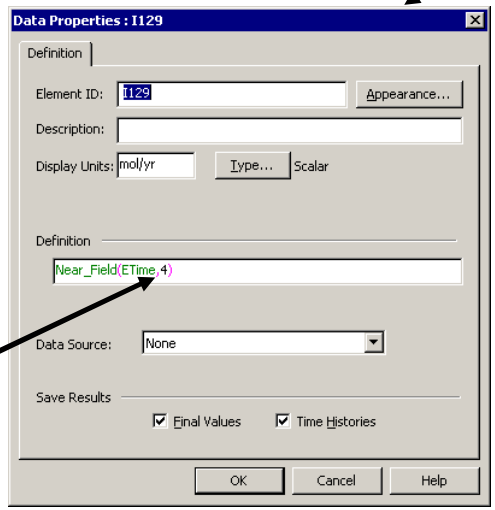


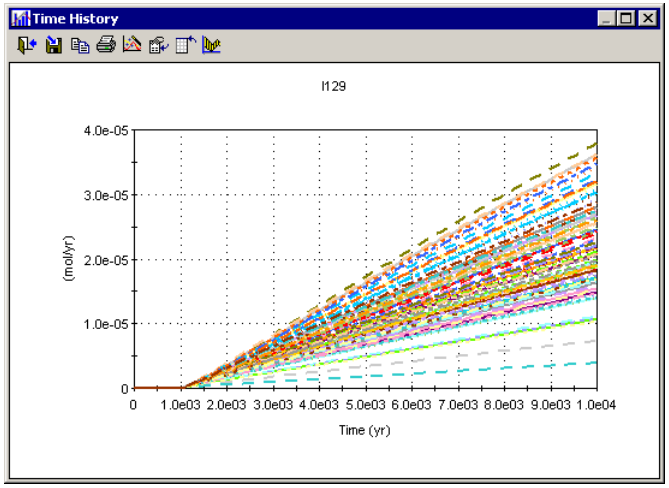
Plotting INPAG-N Output

INPAG-N output files are passed back into GoldSim as 2-d tables (release rate of RNs vs time)

Individual RN output is accessed from the 2-d table by referencing it in a data element





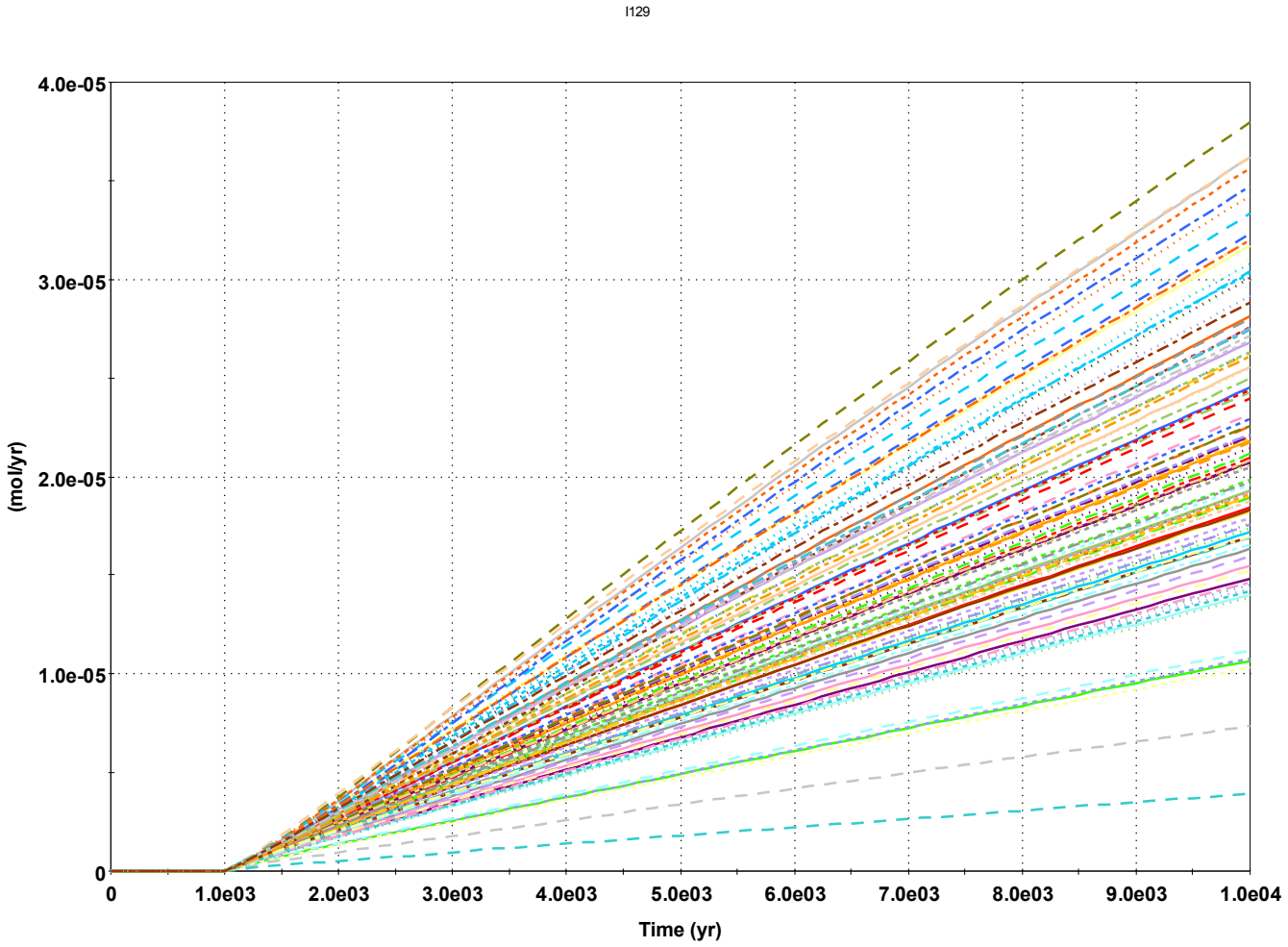


In this example, the I129 release rate (mol/yr) is in the 4th column of the 2-d table

Plot of 100 realizations of I129 release rate. Note that data can be accessed in tabular form from plot element

Sandia National Laboratories

INPAG-N Multiple Realization Output in GoldSim



Each colored line is the result from one realization

Sandia National Laboratories

Debugging

- scratch.txt file; contains information useful for debugging
  - INPAG-N version being run
  - DLL input vector
  - contents of the INPAG-N output file(s) that is read by the DLL
  - DLL output vector
  - comments indicating milestones in the program execution
- Additional information can be added to the file by editing the source code to write information to unit=99 and recompiling the DLL

Sandia National Laboratories

Debugging

- Output files for METHOD=0, METHOD=2, and METHOD=3
  - allows information from those method calls to be saved
  - lets the programmer know if method was/was not executed
- The function name in the DLL element dialog box must match exactly with the name in the !DEC$ ATTRIBUTES dllexport, c command (i.e., launch_inpagn)
- Changes to the number of input and output parameters (scalars or vectors by RN) have to be reflected in the METHOD=3 coding for the number of input and output parameters

Sandia National Laboratories

Acknowledgements

Lockheed Martin

Sandia National Laboratories:

Don Kalinich – Co-Developer of DLL
Hong-Nian Jow – Program Manager
Cedric Sallaberry – Software development
Jerry McNeish, Mgr. – Technical Editing and Approval

Sandia National Laboratories