

Simultaneous optimization of several tallies with FW-CADIS

SCALE 6.3.1

ORNL is managed by UT-Battelle, LLC for the US Department of Energy



U.S. DEPARTMENT OF
ENERGY

Learning goals

- Introduce attendees to the Forward-Weighted CADIS (FW-CADIS) variance reduction technique
- Identify the input and its syntax to use the FW-CADIS technique in MAVRIC

FW-CADIS



Problem

- Analog Monte Carlo tallies tend to have relative uncertainties that are inversely proportional to flux
 - Low flux areas hardest to converge
 - Computation time is controlled by worst uncertainty
- Biasing (typically weight windows) helps move particles to areas of interest
 - Spend more time on “important” particles
 - Sacrifice results in “unimportant” areas
 - Different biasing for different tallies
- CADIS (weight windows and source) is very effective for source–detector type problems

Mesh tallies

- Monte Carlo is used to calculate many tallies
 - Mesh tally: answers everywhere
 - Wide range in relative uncertainties
- Many applications need mesh tallies
 - Dose rate maps
 - Activation of surrounding materials
 - Burnup of fuel pins in reactors

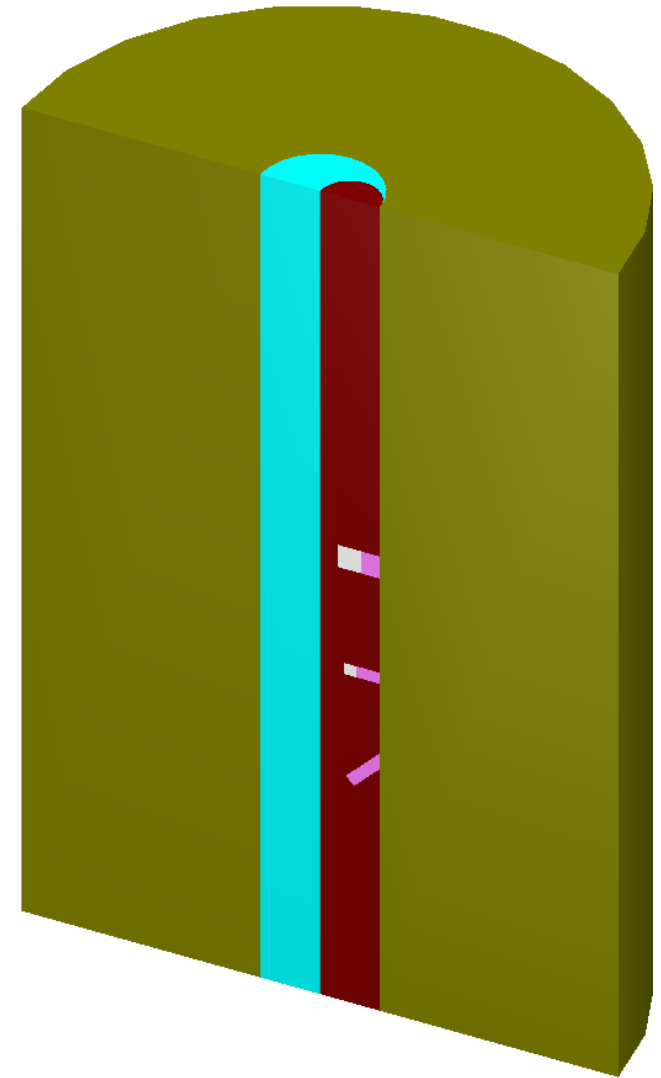


Gamma ray litho-density log

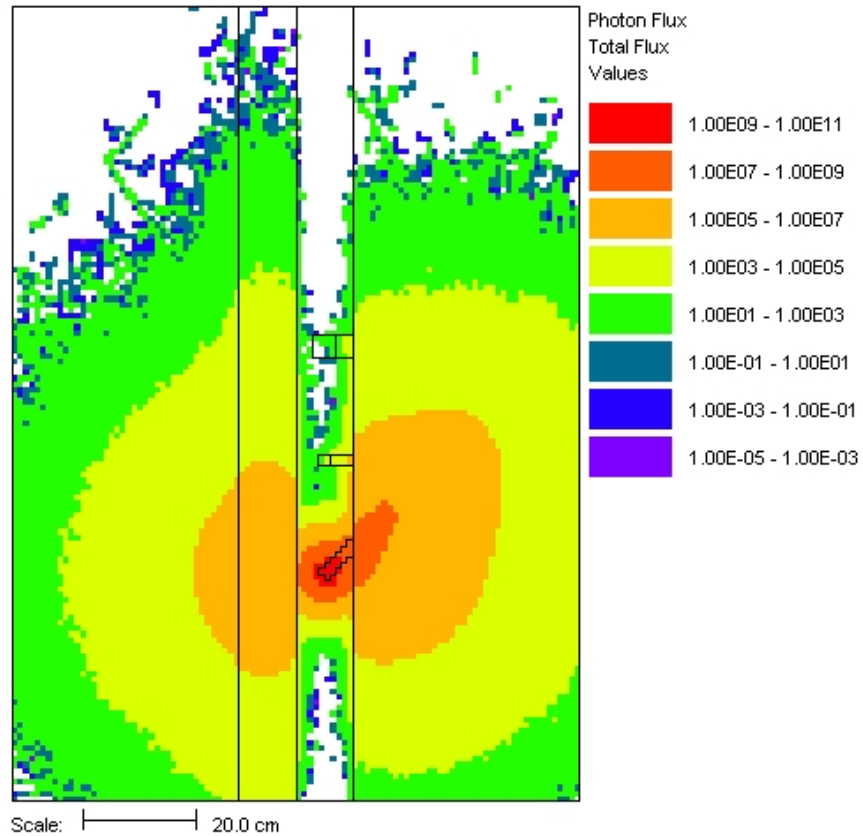
- Problem description from

Robin P. Gardner and Kuruvilla Verghese, "Monte Carlo Nuclear Well Logging Benchmark Problems with Preliminary Intercomparison Results," *Nuclear Geophysics* 5(4), 429-438 (1991).

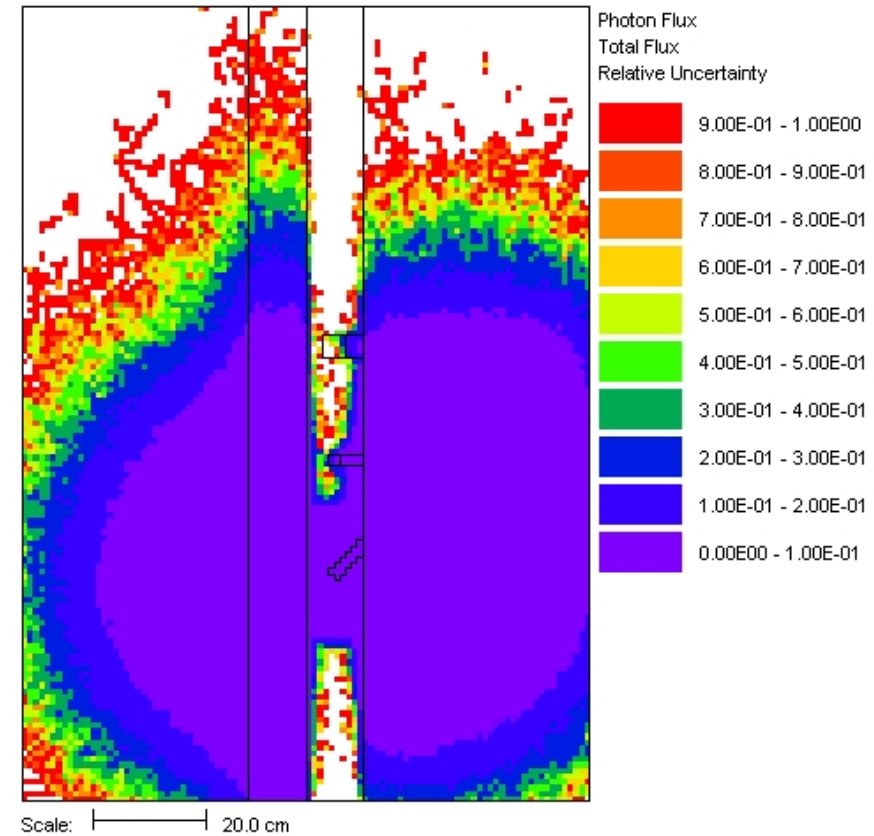
- Source: Cs-137, 2.7 Ci
- Detectors: NaI
 - Near: 2x2 at 20 cm
 - Far: 4x4 at 40 cm
- Borehole: 20 cm diameter
- Tool: 10 cm diameter



Analog



Total Photon Flux



Relative Uncertainty

Calculation	Minutes
Monte Carlo	2762

Results

Near 1.49×10^3 ($\pm 8.2\%$)
Far 6.13×10^1 ($\pm 19\%$)

How to improve uncertainties

- Weight windows/biased source with CADIS
 - Calculate adjoint fluxes from an adjoint source at one of the detector locations

$$q^+(\vec{r}, E) = \sigma_d(\vec{r}, E)$$

- Estimate response

$$R = \int_{V_s} \int_E q(\vec{r}, E) \phi^+(\vec{r}, E) dE dV$$

- Create importance map

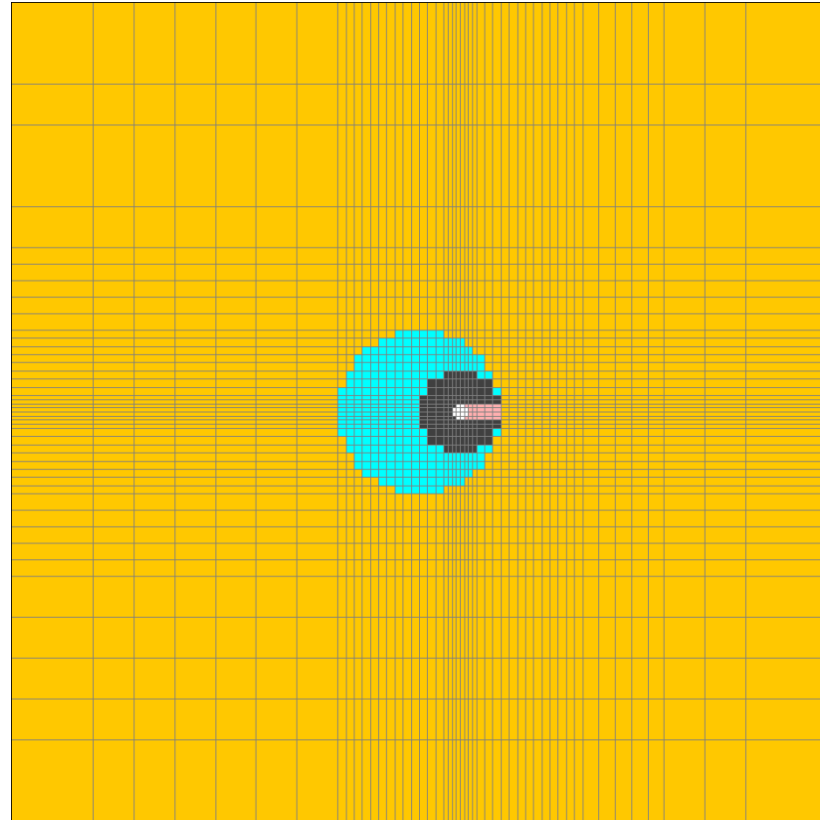
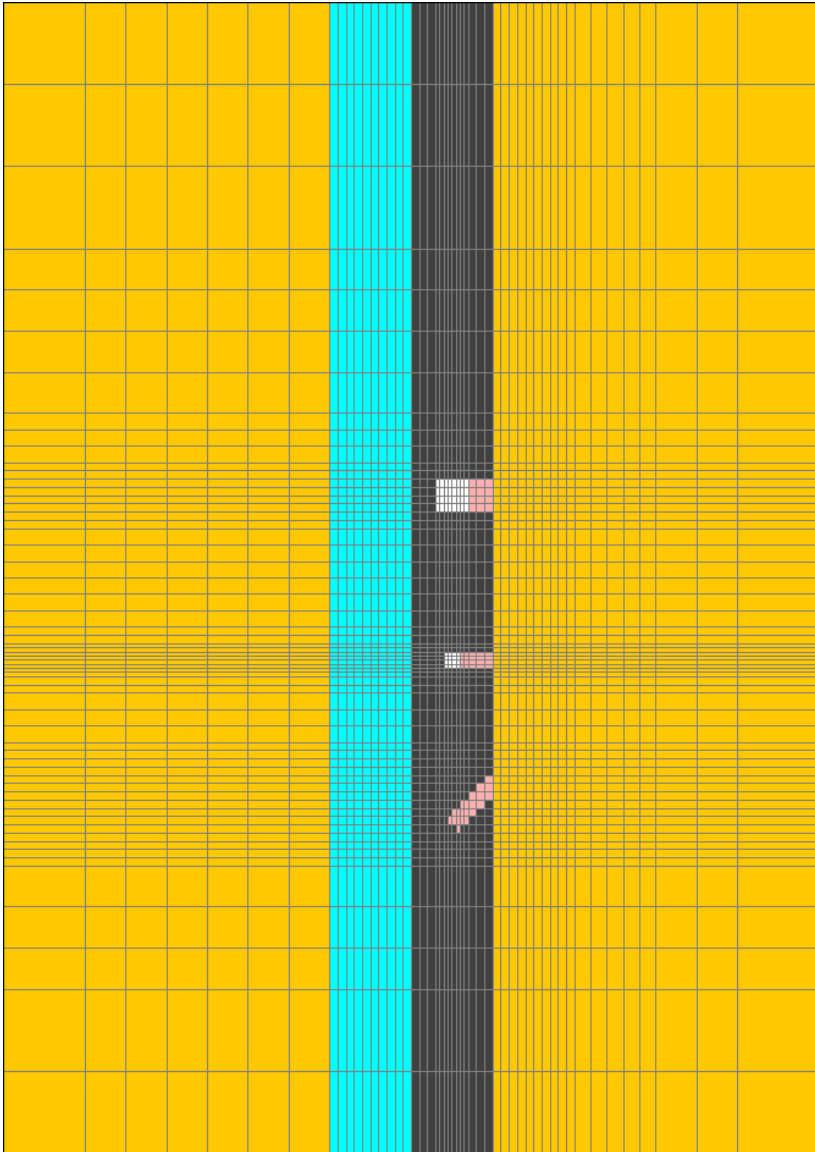
$$\bar{w}(\vec{r}, E) = \frac{R}{\phi^+(\vec{r}, E)}$$

- and consistent biased source

$$\hat{q}(\vec{r}, E) = \frac{1}{R} q(\vec{r}, E) \phi^+(\vec{r}, E)$$

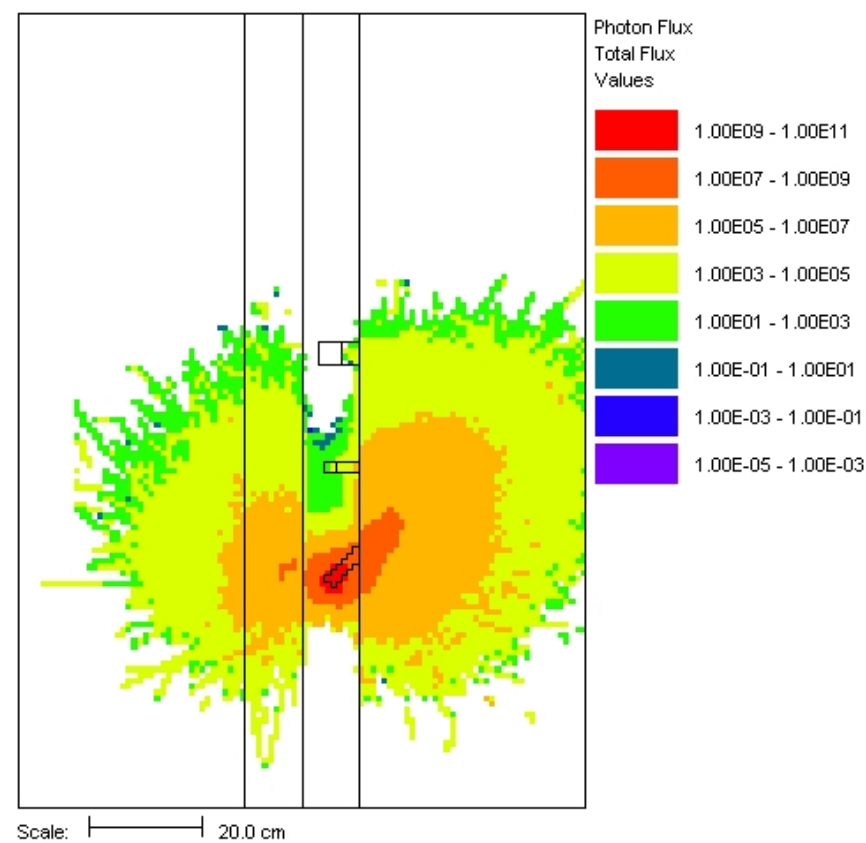
- Run the forward Monte Carlo

Discrete ordinates mesh

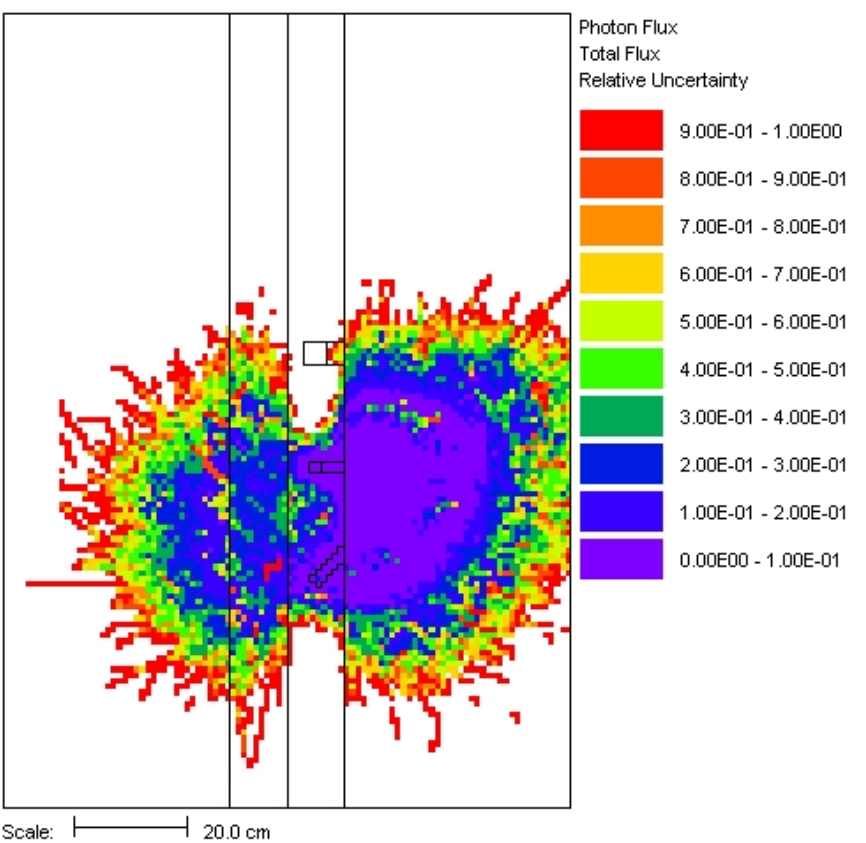


$$49 \times 43 \times 59 = 124,313$$

CADIS – for near detector



Total Photon Flux

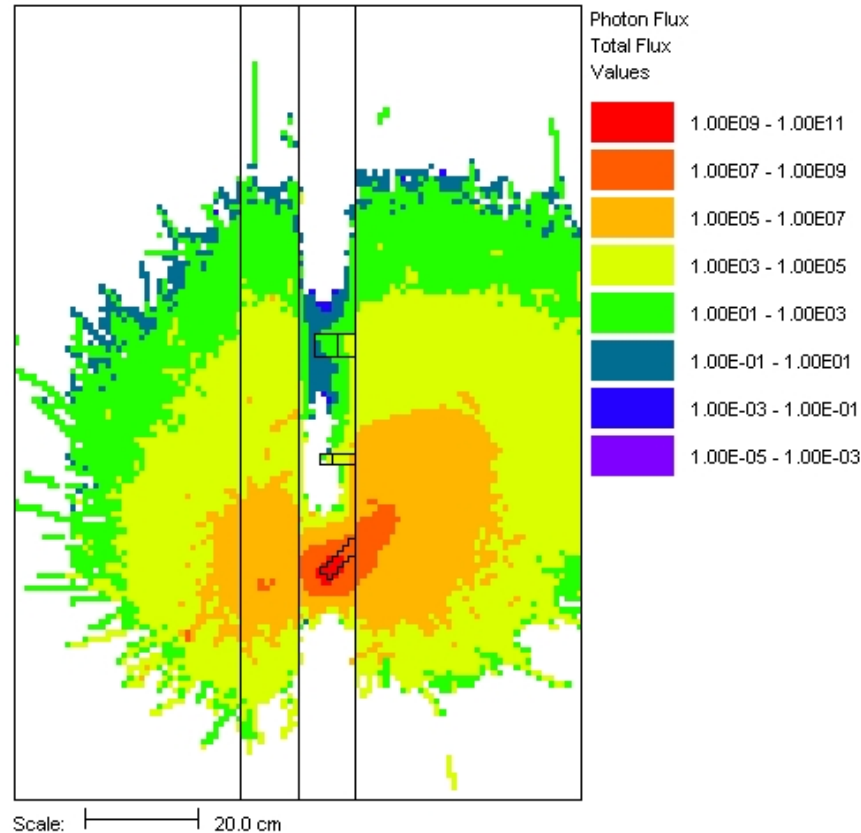


Relative Uncertainty

Calculation	Minutes
Adjoint DO	7
Monte Carlo	126

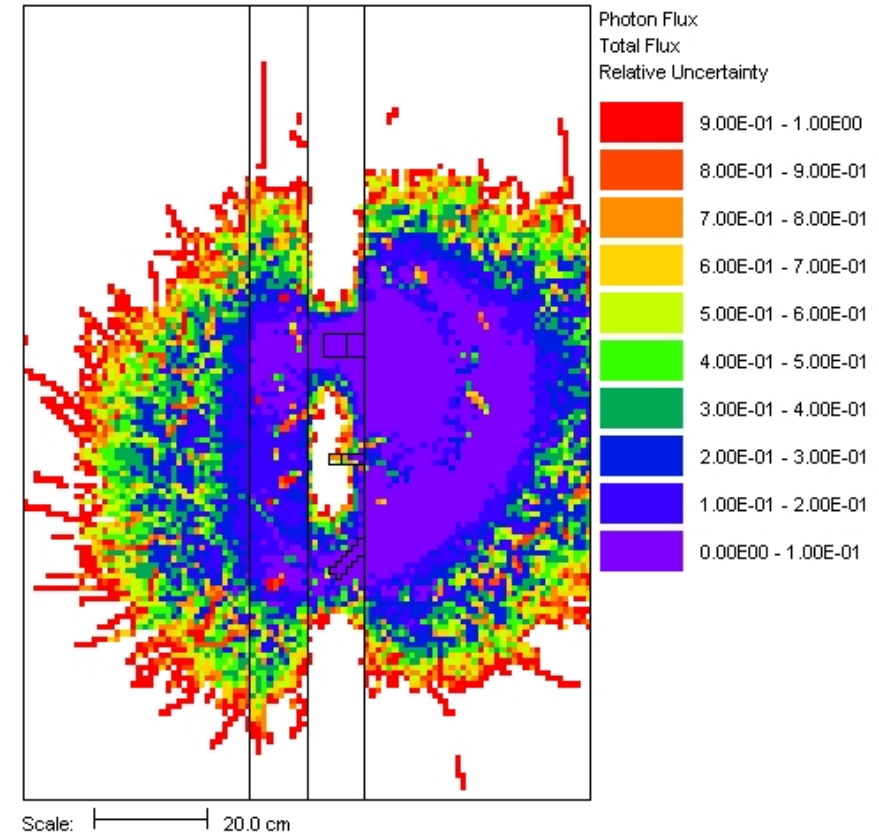
Results
Near 1.54×10^3 ($\pm 0.5\%$)
Far 0.00

CADIS – for far detector



Total Photon Flux

Calculation	Minutes
Adjoint DO	7
Monte Carlo	124



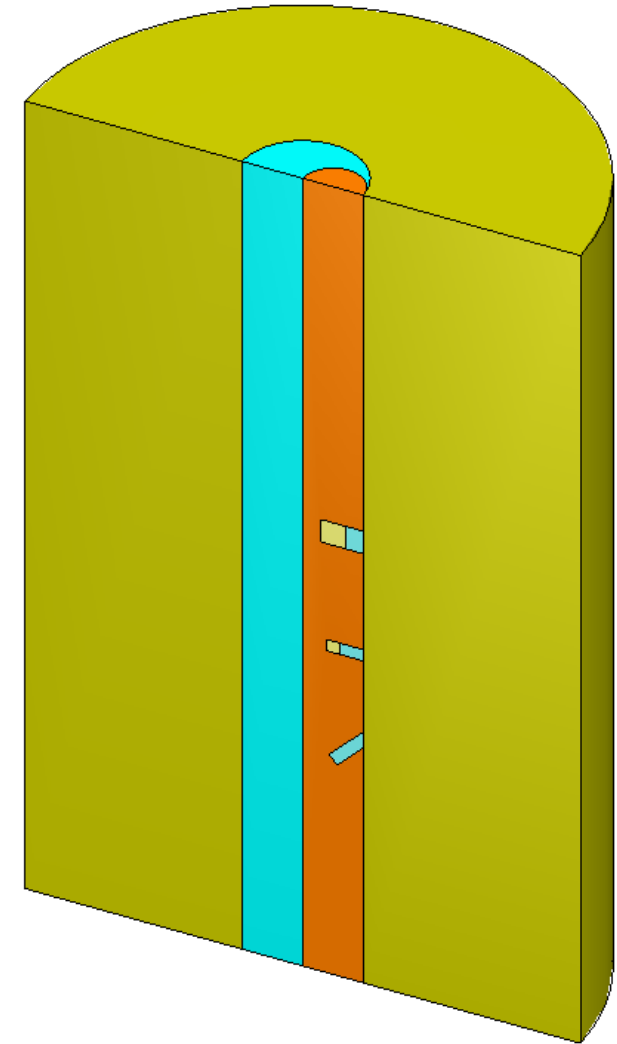
Relative Uncertainty

Results

Near 3.10×10^3 ($\pm 65\%$)
 Far 5.59×10^1 ($\pm 0.3\%$)

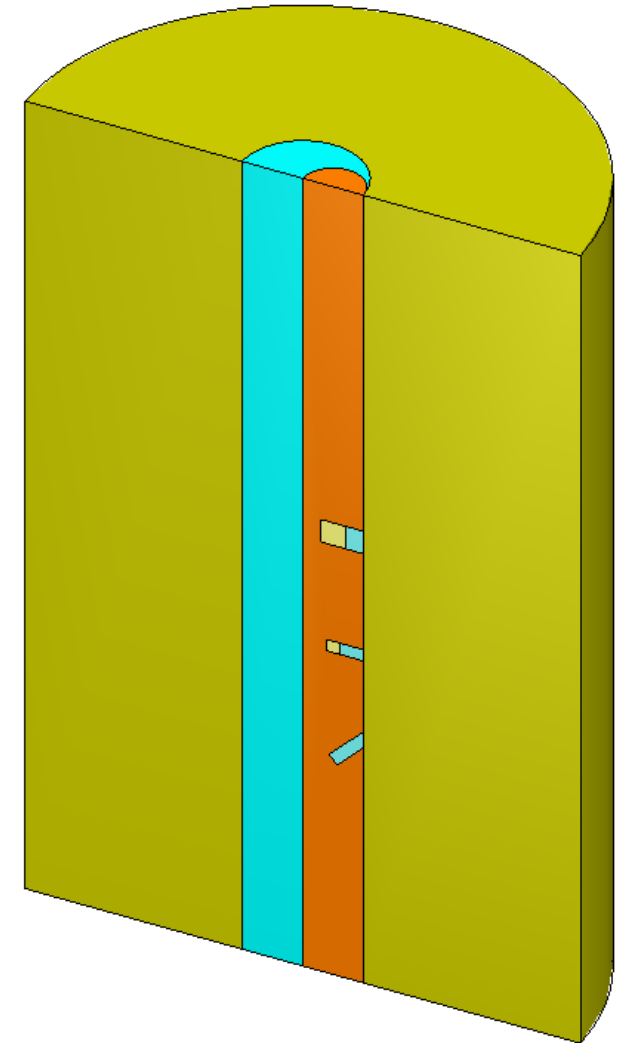
How to improve uncertainties

- CADIS works
 - Improves the figure of merit (FOM) for one detector
 - At the expense of tracking particles deep into the formation
 - At the expense of the other detector
- Can we get both detectors simultaneously?
- Try adjoint source in both detectors



How to improve uncertainties (cont.)

- Adjoint source in both detectors:
 - Recall that the adjoint source locations act like particle attractors in the MC simulations
 - MC particles will tend to go to the “easiest” adjoint source location
 - Need to put more adjoint source in the far detector so that the same number of particles get to each
- How much adjoint source strength to put into the far detector relative to the near?
- Ratio needs to be the same as ratio of the responses!



Litho-density log input file

```
read importanceMap
  gridGeometryID=1

  '
    near detector
    adjointSource 1
      boundingBox 6 4 1 -1 21 19
      unit=1 region=2
      responseID=1
      weight=1.00
    end adjointSource

    '
    far detector
    adjointSource 2
      boundingBox 7 3 2 -2 42 38
      unit=1 region=4
      responseID=1
      weight=27.55
    end adjointSource
  end importanceMap
```

```
response 1
  title="total photon flux"
  photon
  values 19r1.0 end
end response
```

Sure would be nice if this
could be done automatically...

FW-CADIS in MAVRIC

- Perform a forward discrete ordinates calculation
 - Estimate the responses $R(\vec{r}, E)$ everywhere
 - Construct the CADIS adjoint source but weight the source strength with $1/R(\vec{r}, E)$
-

- Perform the adjoint discrete ordinates calculation
- Create the weight windows and biased source
- Perform the Monte Carlo calculation

FW-CADIS

- To achieve the same relative uncertainties in both detectors, the amount of adjoint source in each detector location must be inversely proportional to the expected response

For the optimization of:	Use adjoint source:
$\phi(\vec{r}, E)$	$q^+(\vec{r}, E) = \frac{1}{\phi(\vec{r}, E)}$
$\int \phi(\vec{r}, E) dE$	$q^+(\vec{r}, E) = \frac{1}{\int \phi(\vec{r}, E) dE}$
$\int \phi(\vec{r}, E) \sigma_d(\vec{r}, E) dE$	$q^+(\vec{r}, E) = \frac{\sigma_d(\vec{r}, E)}{\int \phi(\vec{r}, E) \sigma_d(\vec{r}, E) dE}$

Litho-density log input file

```
read importanceMap
  gridGeometryID=1

  near detector
    adjointSource 1
      boundingBox 6 4 1 -1 21 19
      unit=1 region=2
      responseID=1
    end adjointSource

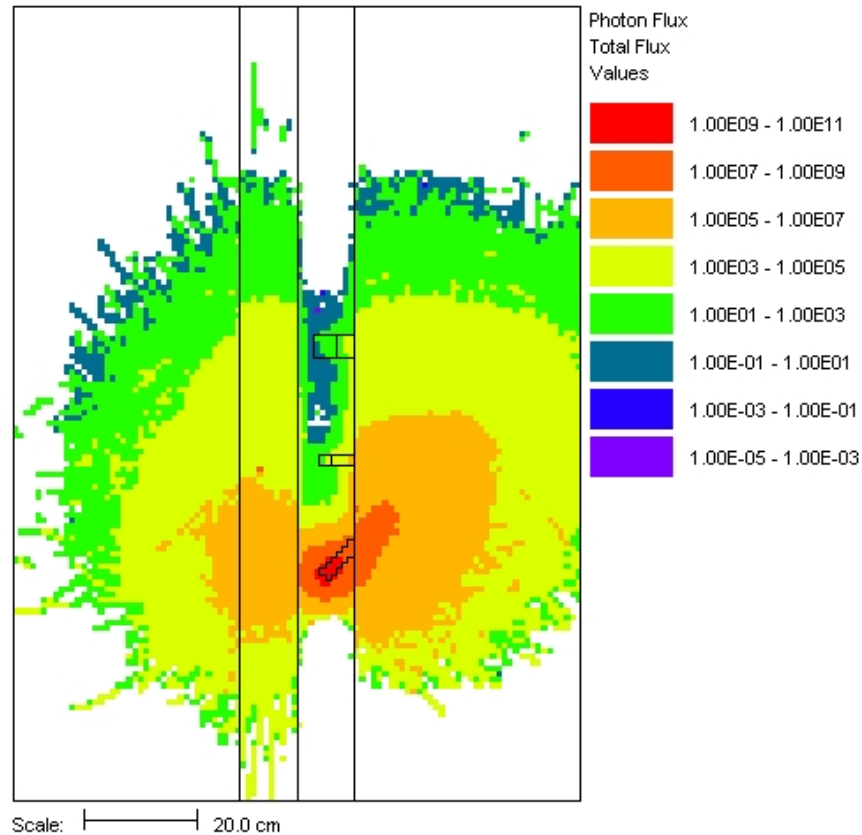
  far detector
    adjointSource 2
      boundingBox 7 3 2 -2 42 38
      unit=1 region=4
      responseID=1
    end adjointSource

  respWeighting
end importanceMap
```

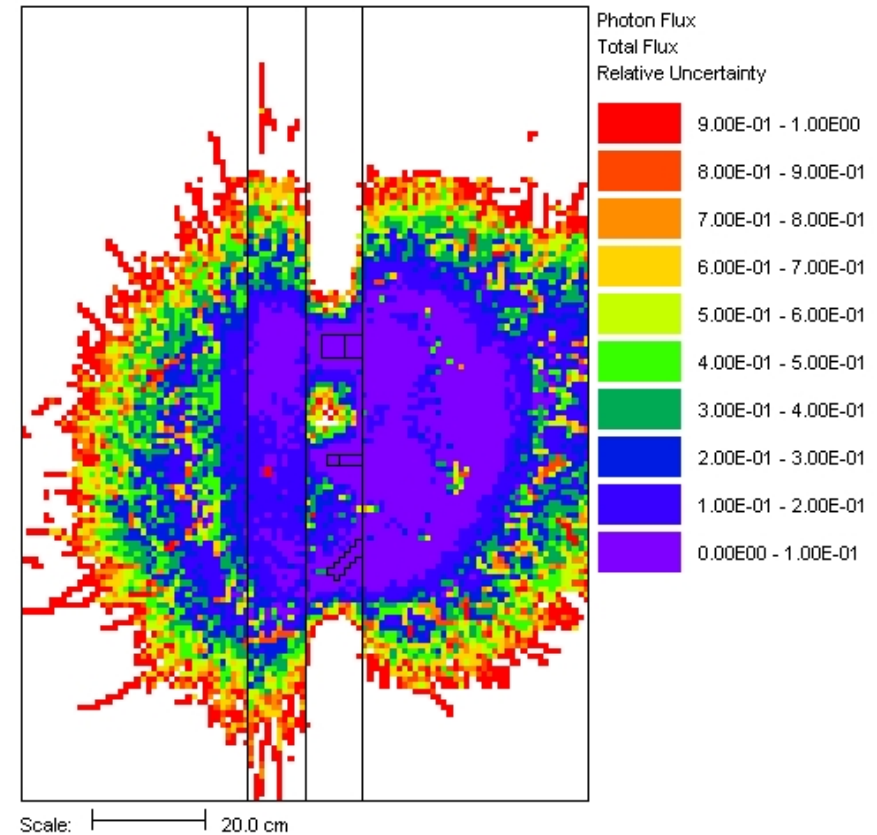
```
response 1
  title="total photon flux"
  photon
    values 19r1.0 end
end response
```

One keyword is almost automatic!
Use “fluxWeighting” or “respWeighting”.

FW-CADIS – for both detectors



Total Neutron Flux



Relative Uncertainty

Calculation	Minutes
Forward DO	6
Adjoint DO	7
Monte Carlo	126

Results

Near $1.54 \times 10^3 (\pm 0.6\%)$
Far $5.56 \times 10^1 (\pm 0.4\%)$

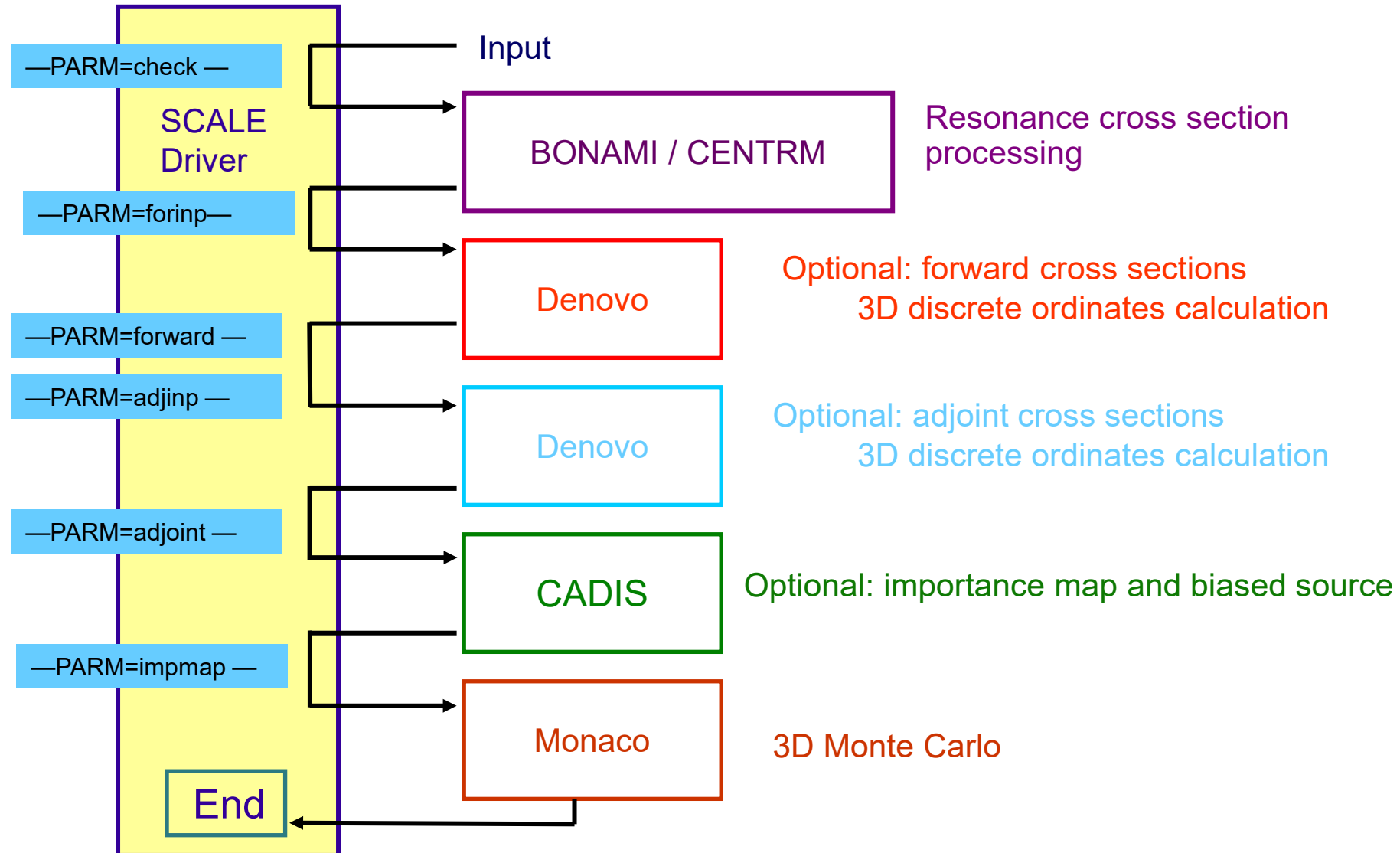
Photon methods comparison

	Analog	CADIS		FW-CADIS
		Near	Far	
Simulation times (min)				
Forward dis. ord.				6
Adjoint dis. ord.		7	7	7
Monte Carlo	2762	126	123	126
Near Detector				
Charged pairs/sec	1.494E+03	1.545E+03		1.543E+03
Relative uncertainty	8.15%	0.46%		0.61%
Far Detector				
Charged pairs/sec	6.130E+01		5.587E+01	5.564E+01
Relative uncertainty	19.14%		0.34%	0.43%
Monte Carlo FOM ^a ratio				
Near detector	1	6830		3867
Far detector	1		71821	43278
Time (min) required for 1%				
Near detector	183547	34		60
Far detector	1012136		21	36
Time (min) required				
For both detectors	1012136	55		60
Time improvement				
Speed-up over analog	1	18321		16748

FW-CADIS for mesh tally

- Mesh tally is just a large set of tallies for which we want roughly uniform relative uncertainties
- Instead of two tallies, there are $I \times J \times K$ tallies
- Adjoint source: the response function divided by the expected forward response in each cell of the mesh tally volume
 - Expected flux
 - Expected total flux
 - Expected dose rate

Sequence MAVRIC calls Monaco



Importance map block format

For weighting each adjoint source by space- and energy-dependent flux

```
read importanceMap
  adjointSource id ... end adjointSource
  adjointSource id ... end adjointSource
  ...
  adjointSource id ... end adjointSource

  gridGeometryID=g

  fluxWeighting
end importanceMap
```

For weighting each adjoint source by its energy-dependent response

```
read importanceMap
  ...
  respWeighting
end importanceMap
```

Importance map block example

```
read importanceMap
  adjointSource 33
    boundingBox 10 -20 30 -30 25 0
    unit=2 region=4
    responseID=2
  end adjointSource
  gridGeometryID=1
  respWeighting
end importanceMap
```

For the optimization of:	Use adjoint source:
$\phi(\vec{r}, E)$	$q^+(\vec{r}, E) = \frac{1}{\phi(\vec{r}, E)}$
$\int \phi(\vec{r}, E) dE$	$q^+(\vec{r}, E) = \frac{1}{\int \phi(\vec{r}, E) dE}$
$\int \phi(\vec{r}, E) \sigma_d(\vec{r}, E) dE$	$q^+(\vec{r}, E) = \frac{\sigma_d(\vec{r}, E)}{\int \phi(\vec{r}, E) \sigma_d(\vec{r}, E) dE}$

FW-CADIS summary

- Used for optimizing multiple tallies at the same time
 - Convenient for optimizing mesh tallies
 - Each adjoint source is weighted by the forward flux solution
 - Ideal for global solutions in a large region
- A single keyword in the importance map block changes optimization
 - Optimize flux (`fluxWeighting`) or response (`respWeighting`)