LLNL-PRES-434714

Lawrence Livermore National Laboratory

COG – Special Features of Interest to Criticality Safety Practitioners

Rich Buck, Dave Heinrichs, Allan Krass, and Ed Lent

Presented at the American Nuclear Society 2010 Annual Meeting in San Diego, California
Tuesday, June 15, 2010

Lawrence Livermore National Laboratory, P.O. Box 808, L-198, Livermore, CA 94551-0808
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

Topics

- What is COG?
- How do I get COG?
- Why use COG?
- Some user-friendly features
- On-going R&D
- Conclusion

What is COG?

- High-fidelity multi-particle transport code
- Extensively used in criticality safety applications at LLNL
- Financial support provided by the US DOE NCSP
- Maintained by the Nuclear Criticality Safety Division

COG10

- **Initial public release (January 2006)**
- ICNC 2007

  [205] COG - Publicly Available Now to Criticality Safety Practitioners

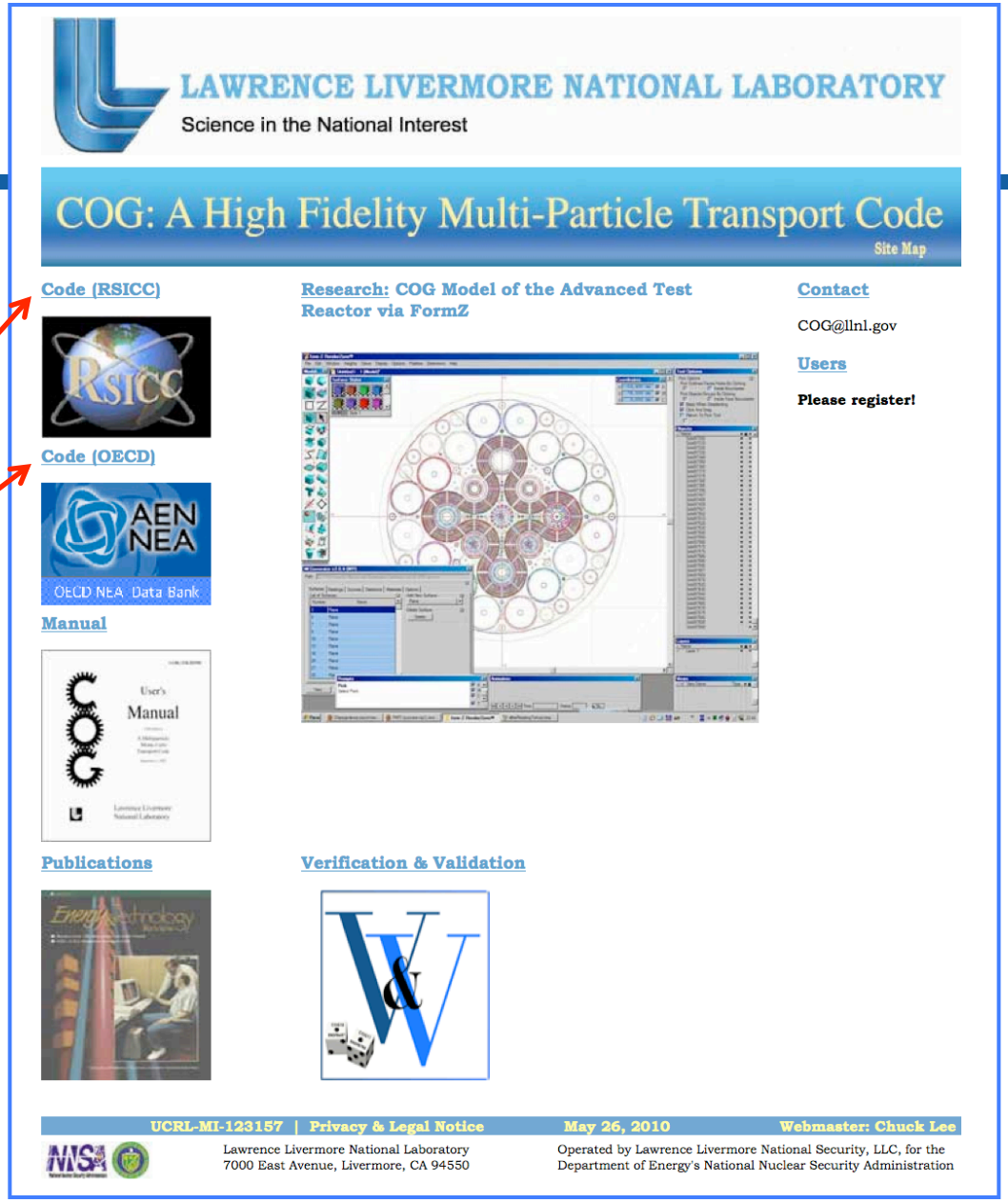
  Richard BUCK, Dermott CULLEN, David HEINRICHS, Edward LENT, Dale NIELSEN Jr., Kenneth SALE (Lawrence Livermore National Laboratory, USA)

- **http://cog.llnl.gov**

COG Website

How do I get COG10?

- http://cog.llnl.gov
- Click here
- or here



Why use COG?

It is DOE O 414.1C compliant safety software!

User: heinrichs1 | Timeout: 3:49 pm, 05/28/2010

SQA Software Project/Product Risk Grading and Inventory

Home -> Project Specific

Nuclear Criticality Safety - COG [O&B-Nuclear Operations] Edit

Privacy Option:
Grading results for this project can be viewed by other projects: Yes No

Software Development Control:
How much control is there of this Software Development Project? None Minor Major

Project Comments: Edit

Safety Software:
Is this Safety Software as defined in DOE Order 414.1C? Yes No

Matrix Basis:
Safety Software (Current)

Project Grading Access: Edit

David P Heinrichs (heinrichs1)	Nuclear Criticality Safety Division Leader
Mark Andrew Lee (lee73)	LSO Criticality Safety SME

Risk Consequence Reports

New Pending Report [Version: 05-Nov-07]

Submitted Reports

Matrix Basis	Matrix Version	Created	Submitted	Tier	
Safety Software	05-Nov-07	15-Jan-08	15-Jan-08	2	View Report

Likelihood of Failure Reports

New Pending Report [Version: 17-Sep-08]

Submitted Reports

Matrix Basis	Matrix Version	Created	Dev Control	Submitted	Score	
Safety Software	05-Nov-07	15-Jan-08	Major	15-Jan-08	2.4	View Report

Matrix Grade
[Version: 27-Aug-08]

Tier 0	RL1	RL1	RL1
Tier 1	RL3	RL2	RL2
Tier 2	RL4	RL3	RL3
Tier 3	RL4	RL4	RL3
Tier 4	RL4	RL4	RL4
Likelihood of Failure Rating:	0-2	2-8	8-16

Grading Summary Reports

- Full Report
- Printable Report
- Grade History
- Report History
- View Required Practices

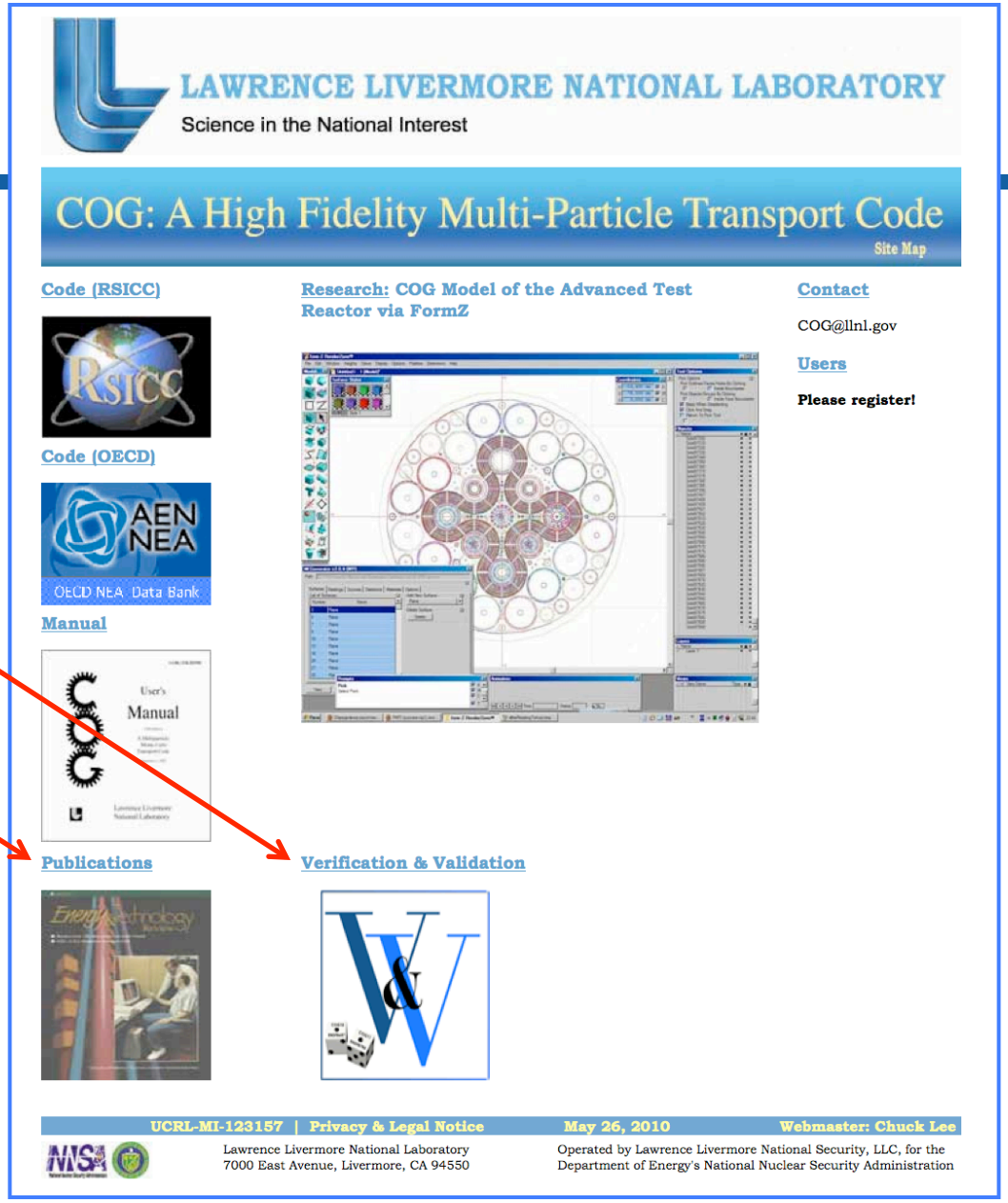
Home -> Project Specific

Administrative Information
Website Problems/Comments: webmaster-car@llnl.gov
Content Questions: SQA-Manager@llnl.gov

Why use COG?

It is benchmarked!

- Click here
- or here



Why use COG?

COG: A High Fidelity Multi-Particle Transport Code

Home	Manual	Publications	Contact	Users

REPORT # (PDFs)	DOCUMENT TITLE	AUTHOR(S)	PUBLISHED
LLNL-PRES-418844	^{233}U Benchmarks with a Comparison to COG and MCNP Results Using ENDF/B-VII.0 *(click here to download COG ^{233}U Inputs)*	David P. Heinrichs	11/03/2009
LLNL-TR-433976	Validation of COG10 and ENDFB6R7 on the *Surya* Workstation for General Application to Highly Enriched Uranium Systems *(click here to download COG HEU Inputs)*	Allan W. Krass and David P. Heinrichs	09/16/2009
README file	COG10 software installation README file	Rich Buck	06/02/2005

90 COG U-233 ICSBEP input decks

358 COG HEU ICSBEP input decks

More coming soon!

UCRL-MI-123157 | Privacy & Legal Notice May 26, 2010 Webmaster: Chuck Lee

Lawrence Livermore National Laboratory
7000 East Avenue, Livermore, CA 94550

Operated by Lawrence Livermore National Security, LLC, for the Department of Energy's National Nuclear Security Administration

COG user-friendly features

- Data block structure
- Options
  - ASSIGN-MC
  - SWEEP
  - VOLUME
  - SABLIB
  - MIX
  - REVOLUTION
  - PRISM
  - UNIT and FILL
  - TR

Data block structure

TITLE

```
PU-MET-FAST-001:  JEZEBEL (17.020 kg Pu(95.48)-1.02Ga @ 15.61 g/cc)
```

neutron only calculation with prompt and delayed multiplicities and spectra

BASIC

```
BASIC
 neutron delayedn
```

length in cm (default)

SURFACES

```
SURFACES
 1 sphere 6.3849 $ per Section 3.2
```

comments

GEOMETRY

```
GEOMETRY
 sector 1 alloy -1
 boundary vacuum 1
 picture cs material -7  0  7  -7  0 -7   7 0 -7
 volume  -7 -7 -7  7 -7 -7  -7 7 -7 14 14 14
```

Optional (default vacuum boundary condition)

(default resolution w/o titles)

"color" not specified (default B&W picture)

CRITICALITY

```
CRITICALITY
 npart=5000 nbatch=5005 sdt=0.0001 nfirst=6 norm=1.
 nsource=1  0. 0. 0.
```

MIX

```
MIX
 nlib=ENDFB6R7 $ Atom Densities per Table 3
 mat=1 bunches ga 1.3752-3 pu239 3.7047-2 pu240 1.7512-3 pu241 1.1674-4
```

Point-wise continuous cross-section library (nlib2 not used)

atoms b^{-1} cm^{-1} (1 of 4 options)

```
END
```

Any amount of comments may follow the end flag

ASSIGN-MC data block (optional)

ASSIGN-MC— Assign Plotting Colors to Materials

PICTUREs of the geometry may have areas filled with colors, which are keyed to the sector materials. See section **PICTURES of the Geometry** for information on the color option for cross-section and perspective pictures.

By default, colors are chosen by the code from a palette of 20 colors. To assign specific colors to materials, use the ASSIGN-MC statement:

ASSIGN-MC mat-ID#$_1$ c_1 { mat-ID#$_2$ c_2 } . . .

Where:

mat-ID#$_1$ is the ***mat-ID#*** to be assigned a color;

c_1 is the ASCII ***color name*** to be assigned to material ***mat-ID#$_1$***.

Example:

```
ASSIGN-MC
1 SKY                $ ASSIGN TO MATERIAL 1, COLOR SKY
2 YELLOW             $ ASSIGN TO MATERIAL 2, COLOR YELLOW
3 ROSE               $ ASSIGN TO MATERIAL 3, COLOR ROSE
```

When PICTURES are subsequently drawn, areas in the picture which represent various materials will be drawn in the specified color.

GEOMETRY data block – SWEEP (optional)

The SWEEP Statement

The SWEEP statement sweeps a line through the user's geometry between two specified points. All sectors and their boundary surfaces which intersect the SWEEP line will be listed in the output file, along with the distance of each intercept from the "start" point. This statement is very helpful in analyzing the exact COG placement of surfaces in a complex region of the geometry.

The format is:

SWEEP x_0 y_0 z_0 x_1 y_1 z_1

where:

x_0 y_0 z_0 represents the starting point of the sweep;

x_1 y_1 z_1 represents the ending point of sweep.

GEOMETRY data block – SWEEP

***Example** of a SWEEP statement:*

```
SWEEP 5. -10. 3.9813    15. -10. 3.9813
```

***Example** of the output of the above statement:*

```
SWEEP  5.        -10.      3.9813    15.       -10.      3.9813
```

X	Y	Z	DIST	CROSS SURF	LEAVE SECTOR	NAME	MAT	LEV
5.0000E+00	-1.0000E+01	3.9813E+00						
6.2221E+00	-1.0000E+01	3.9813E+00	1.2221E+00	40	0	FILL	0	0
7.1783E+00	-1.0000E+01	3.9813E+00	2.1783E+00	30	104	SPH4	1	0
8.2789E+00	-1.0000E+01	3.9813E+00	3.2789E+00	20	103	SPH3	6	0
9.4000E+00	-1.0000E+01	3.9813E+00	4.4000E+00	91	0	FILL	0	1
9.6434E+00	-1.0000E+01	3.9813E+00	4.6434E+00	92	911	SPHU1	2	1
9.8009E+00	-1.0000E+01	3.9813E+00	4.8009E+00	93	0	FILL	0	2
1.0199E+01	-1.0000E+01	3.9813E+00	5.1991E+00	93	201	SPH5	3	2
1.0356E+01	-1.0000E+01	3.9813E+00	5.3565E+00	92	0	FILL	0	2
1.0600E+01	-1.0000E+01	3.9813E+00	5.6000E+00	91	911	SPHU1	2	1
1.1721E+01	-1.0000E+01	3.9813E+00	6.7210E+00	20	0	FILL	0	1
1.2000E+01	-1.0000E+01	3.9813E+00	7.0000E+00	52	103	SPH3	6	0
1.2037E+01	-1.0000E+01	3.9813E+00	7.0374E+00	60	0	FILL	0	0
1.2822E+01	-1.0000E+01	3.9813E+00	7.8217E+00	30	109	CONE1	5	0
1.3778E+01	-1.0000E+01	3.9813E+00	8.7779E+00	40	109	CONE1	5	0
1.4000E+01	-1.0000E+01	3.9813E+00	9.0000E+00	60	109	CONE1	5	0
1.5000E+01	-1.0000E+01	3.9813E+00	1.0000E+01	0	0	FILL	0	0

GEOMETRY data block – VOLUME

VOLUME Calculations

COG can compute the VOLUME of user's sectors. In this option, you specify a box-shaped volume within your geometry. COG performs a Monte Carlo calculation of the volume and the mass of each MATERIAL, REGION, or SECTOR within that box. This can be an expensive calculation, but it is also the best method for finding errors associated with overlapping sectors. It is the only way to detect geometry errors which result in volume or mass discrepancies from the original physical model. You may request any number of VOLUME calculations. Each VOLUME specification has this form:

$$\textit{VOLUME} \begin{bmatrix} \textbf{SECTOR} & \textbf{or SEC} & \textbf{or S} \\ \textbf{MATERIAL} & \textbf{or MAT} & \textbf{or M} \\ \textbf{REGION} & \textbf{or REG} & \textbf{or R} \end{bmatrix} \{\textit{RES nres}\}\ x_0\ y_0\ z_0\ \ x_1\ y_1\ z_1$$

$$x_2\ y_2\ z_2\ \ \textit{length-x'}\ \ \textit{length-y'}\ \ \textit{length-z'}\ \ \{\textit{TITLE} = \text{"}\ldots\text{"}\}$$

where:

SECTOR (or **MATERIAL** or **REGION**) (or the shorter aliases) specifies what volume will be determined;

x_0, y_0, z_0 is a point in the reference corner of the box (in problem coordinates);

x_1, y_1, z_1 is any point along one edge, or its extension, of the box. This, along with the reference corner, defines the +x'–axis;

x_2, y_2, z_2 is any point along another edge, or its extension, of the box. This, along with the reference corner, defines the y'–axis. The z'–axis is constructed to form a right–handed box coordinate system;

length-x', ***length-y'***, and ***length-z'*** are the lengths of the defined box along each of the three box axes;

RES ***nres*** specifies an optional higher precision for the volume calculation (***nres*** a positive integer). When ***nres*** is specified, the standard error of the calculation is decreased by a factor of $1/\sqrt{nres}$.

RES allows users to increase the precision of the volume calculation

MIX Data Block

Four options:

Format of a **MIX** Data Block, showing the alternative means of specifying component amounts by 1 density, 2 atomic fraction, 3 weight percent/fraction, and bunches 4 (atoms/barn-cm). Weight percent and atom fraction are relative values (unnormalized).

1
- **MIX**
- **MAT = mat-*ID#*$_1$**
- ***component-1*** ***density-1***
- {***component-2*** ***density -2*** }
- . . .

2
- **MAT = mat-*ID#*$_2$**
- **ATOM-FRACTION** ***material-density***
- ***component-1*** ***atom-fraction-1***
- {***component-2*** ***atom-fraction-2*** }
- . . .

3
- **MAT = mat-*ID#*$_3$**
- **WEIGHT-PERCENT** ***material-density***
- ***component-1*** ***weight-percent-1***
- {***component-2*** ***weight-percent-2*** }
- . . .

4
- **MAT = mat-*ID#*$_4$**
- **BUNCHES**
- ***component-1*** ***bunches-1***
- {***component-2*** ***bunches-2*** }
- . . .
- {**NLIB =** ***libname***}

MIX Data Block – Component Name Options

1134 COGLEX dictionary entries *for*

Isotopes

- 26056 – ZAID
- Fe56 – Isotope name
- Iron56 – Alternate name

Elements

- 26000 – ZAID
- Fe – Chemical name (natural abundance of isotopes)
- Iron – Alternate name

Compounds

- SS304

Some codes require a user to specify 13+ isotopes for SS304 !!

COG will build up elements from isotopes and compounds from elements.

MIX Data Block – NLIB Statement

Mix and match nuclear data libraries

Secondary Neutron Libraries (NLIB2 and NLIB3)

COG has the capability of reading from a second and/or third neutron data file. To use this option insert statement(s) of the following form:

NLIB2 = ***filename2*** ***isotope2.1*** ***isotope2.2*** ...
NLIB3 = ***filename3*** ***isotope3.1*** ***isotope3.2*** ...

where:

filename2, filename3 are names of COG neutron libraries (e.g., **ENDL90**);

isotope2.1 ***isotope2.2*** ... are the names (or ZAIDs) of the desired isotopes to be read from the specified library.

Examples of NLIB2/NLIB3 option:

```
MIX
  NLIB ENDFB6R7
  NLIB2 RED2002 SN
  NLIB3 ENDL90 AL27
  MAT 1 U235 10.
  MAT 2 AL 1. SN 1.
```

MIX Data Block – SABLIB Statement

Mixtures may have multiple S(a,b) specifications.

Example: MAT=1 w-p 1.85 (be) 98 (beo) 2

Material Names that Invoke S(a,b) Thermal Neutron Cross Sections

Component Name	Treat as if it were:
(C6H6)	H and C bound in C6H6
(H.CH2)	H bound in CH2
(H.H2O)	H bound in H2O
(H.ZrH)	H bound in ZrH
(D.D2O)	D bound in D2O
(Be)	Be metal
(BeO)	Be and O bound in BeO
(C)	C as graphite
(O.UO2)	O bound in UO2
(Zr.ZrH)	Zr bound in ZrH

SURFACE Data Block – REVOLUTION

Curve Made of Straight-Line Segments Rotated About the X'-axis

$$\textit{surf-ID\#} \begin{bmatrix} \text{REVOLUTION} \\ \text{REV} \\ \text{R} \end{bmatrix} \textit{number-points} \quad x'_1\, r'_1 \quad x'_2\, r'_2 \quad \ldots x'_n\, r'_n$$

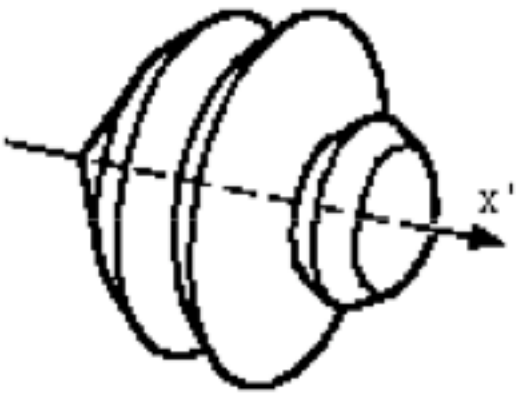
(TR ....)

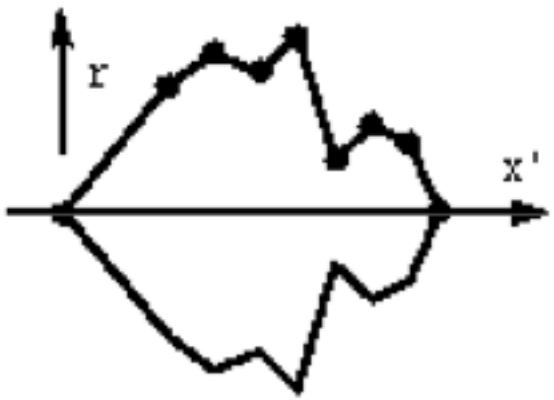
OR

$$\textit{surf-ID\#} \begin{bmatrix} \text{REVOLUTION} \\ \text{REV} \\ \text{R} \end{bmatrix} \textit{number-points} \quad \textit{POLAR} \quad r_1\, q_1 \quad r_2\, q_2 \quad \ldots r_n\, q_n$$

(TR ....)

SURFACE Data Block – REVOLUTION



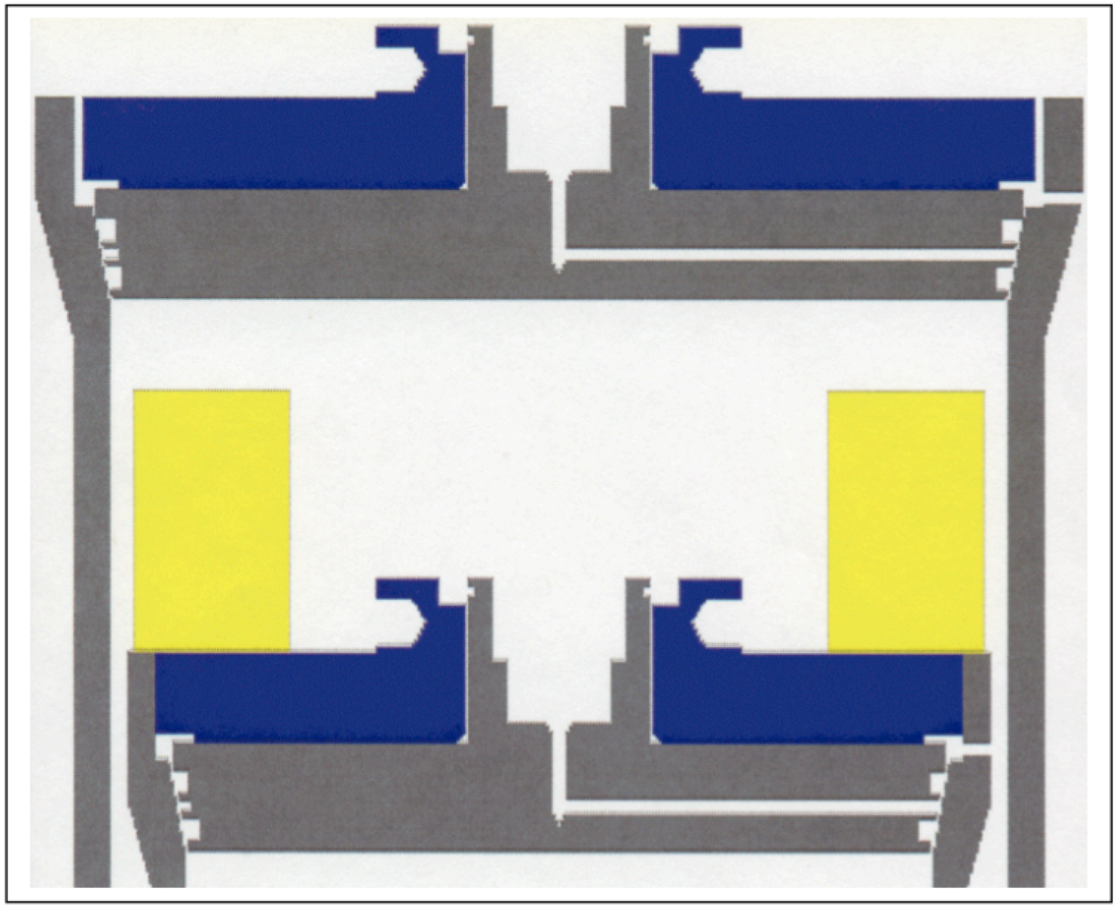


Example *of a Surface of Revolution defined by seven pairs of points.*

```
64 REVOLUTION 7  -2.4 0  -2.3999 1.5  -2 2.3  0 2.3
                 0.5 1.0  1.7 1.0  2.4 0    TR 37.5 -2.5 0
```

SURFACE Data Block – REVOLUTION

COG model of the containment vessels of the 9975 Type B shipping container



SURFACE Data Block – PRISM

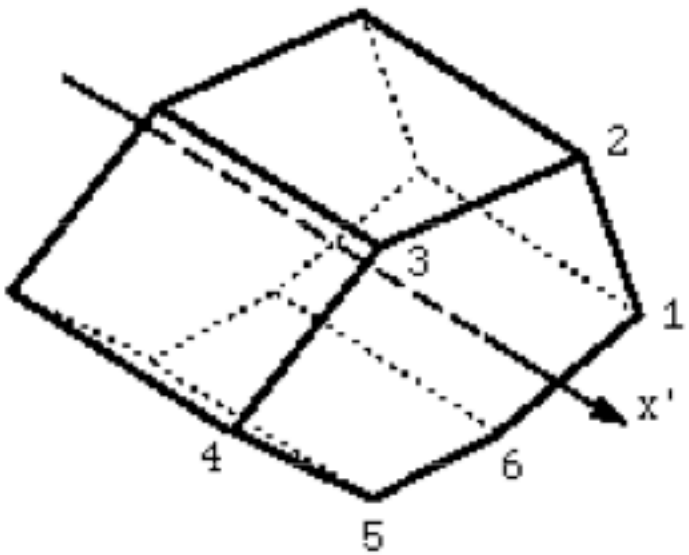
Right Prism

A general right prism with its axis parallel to the x'-axis is specified by:

$$\textit{surf-ID\#} \quad \begin{bmatrix} \textbf{PRISM} \\ \textbf{PRI} \end{bmatrix} \quad \textit{number-points} \quad y'_1\, z'_1 \quad y'_2\, z'_2 \quad \ldots \quad y'_n\, z'_n$$

$$\{x'_{b1} \;\; x'_{b2}\} \;\; (TR \ldots.)$$

SURFACE Data Block – PRISM



Example *of a PRISM of six points located at (y' z') =(4 0), (3 4), (-2 4), (-4 2), (-3 -1), (0 -2). Via a TR command, the prism is translated to a new origin (-3 -6 0), and rotated to point along the z-axis.*

```
15 PRISM 6   4 0   3 4   -2 4   -4 2   -3 -1   0 -2
            TR -3 -6 0   -3 -6 1
```

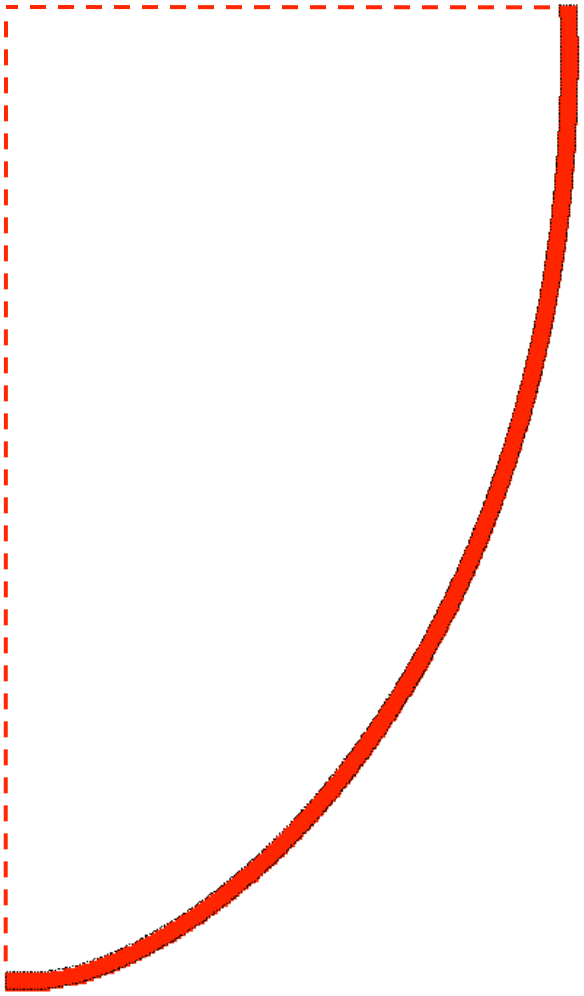
SURFACE Data Block – PRISM – Example

COG model of an involute shape

```
$ ---------------------------------------------------------
  1 prism 40 $ Involute of a circle with 2.7215-inch radius
$              with (Y,Z) pairs and arclength (inner)
$ ---------------------------------------------------------
  0.2604500E+01  0.5000000E-01 $  Xtra-Endpoint
  0.2604501E+01  0.0000000E+00 $  Non-Involute
  0.2721500E+01  0.0000000E+00 $  0.0000000E+00
  0.2819670E+01  0.1794009E-01 $  0.1000000E+00
  0.2914211E+01  0.5036954E-01 $  0.2000000E+00
  0.3005166E+01  0.9185364E-01 $  0.3000000E+00
  0.3092581E+01  0.1403748E+00 $  0.4000000E+00
  0.3176498E+01  0.1947296E+00 $  0.5000000E+00
  0.3256962E+01  0.2540825E+00 $  0.6000000E+00
  0.3334016E+01  0.3178036E+00 $  0.7000000E+00
  0.3407702E+01  0.3853928E+00 $  0.8000000E+00
  0.3478064E+01  0.4564385E+00 $  0.9000000E+00
  0.3545142E+01  0.5305926E+00 $  0.1000000E+01

         . . . . . . . . . . . . . . . . . .

  0.4259705E+01  0.2364418E+01 $  0.3000000E+01
  0.4267066E+01  0.2464144E+01 $  0.3100000E+01
  0.4271997E+01  0.2564020E+01 $  0.3200000E+01
  0.4274534E+01  0.2663986E+01 $  0.3300000E+01
  0.4274711E+01  0.2763983E+01 $  0.3400000E+01
  0.4272564E+01  0.2863958E+01 $  0.3500000E+01
  0.4272258E+01  0.2872953E+01 $  0.3509000E+01
  0.4222289E+01  0.2871201E+01 $  Xtra-Endpoint
  -100. 100.
```



GEOMETRY Data Block – UNIT

In the definition of a DEFINE UNIT, you can USE one or more other DEFINE UNITS. Thus, UNIT 4 could be defined as:

```
DEFINE UNIT 4
  SECTOR      17  BB1  . . .
  SECTOR      18  BB2  . . .
  USE UNIT 30 unit30  -300     TRU  0 112. -22.
  . . .
  FILL      3
```

The user would, of course, have to provide a definition for UNIT 30. UNITs may be nested 50 levels deep. When an error occurs in the geometric setup, the level number is also printed in the resulting fatal error statement. In COG terminology, level–0 is the level of the normal problem geometry. If a particle enters a UNIT USEd in the level–0 geometry, it passes to level–1. If the particle enters anther UNIT nested within the first one, it passes to level–2, etc.

GEOMETRY Data Block – FILL

In the definition of a DEFINE UNIT, you can USE one or more other DEFINE UNITS. Thus, UNIT 4 could be defined as:

```
DEFINE UNIT 4
  SECTOR      17  BB1  . . .
  SECTOR      18  BB2  . . .
  USE UNIT 30 unit30  -300     TRU  0 112. -22.
  . . .
  FILL      3
```

Not having to specify all sectors (e.g., air) can be a huge time saver !!

The user would, of course, have to provide a definition for UNIT 30. UNITs may be nested 50 levels deep. When an error occurs in the geometric setup, the level number is also printed in the resulting fatal error statement. In COG terminology, level–0 is the level of the normal problem geometry. If a particle enters a UNIT USEd in the level–0 geometry, it passes to level–1. If the particle enters anther UNIT nested within the first one, it passes to level–2, etc.

Each unit may specify a different FILL material, which is different from the overall problem FILL specification. The default FILL is "void".

SURFACE Data Block – TR

Translation and Rotation (TR) of Surfaces and Units

To place a surface in an orientation or location other than its initial one, the user must add to the surface specification a TR (translation/ rotation) specification of the form:

TR ***x0 y0 z0 (x1 y1 z1) (x2 y2 z2)***

x0 y0 z0 is the new origin. This is the translation part of the specification.

x1 y1 z1 is any point on the new positive x-axis. This is the first rotation.

x2 y2 z2 is any point on the new positive y-axis. This is the second rotation.

On-going R&D

COG11 is nearing completion

- **More physics**
- **More data libraries**
- **More geometry enhancements**
- **More user-friendly features**
- **RSICC release planned in conjunction with ICNC2011**

Conclusion

- **Give COG a try!**
- **LLNL is available to provide user support and training**
- **http://cog.llnl.gov**
-