

LA-UR-24-26694

Approved for public release; distribution is unlimited.

Title: PEM-EOS tools Useful tools for interacting with SESAME

Author(s): Sheppard, Daniel Glen
Pimentel, David A.
Peterson, Joe Walker

Intended for: Report

Issued: 2024-07-05



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.



PEM-EOS tools

Useful tools for interacting with SESAME

D. Sheppard, D. Pimentel, J. Peterson

X-Computational Physics, Los Alamos National Laboratory

EOS Theory meeting, 27-June-2024

LA-UR-24-NNNNN



Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA.

Enabling tools on various machines

- Install locations:
 - LANL HPC: Rocinante, Fog, Darwin, Tycho
(DATAROOTDIR=/usr/projects/data)
 - XLAN: Mccoy, Orga, Scotty
(DATAROOTDIR=/opt/local/codes/data)

Environment setup:

```
module use $DATAROOTDIR/modulefiles
```

```
module load python ← for the python tools; module is not called python on every machine
```

```
module load eospac
```

```
module load gnuplot ← this is not always necessary or available; your mileage may vary
```

What you'll find

- Command line tools
 - *interp_sesame_data* : interpolate SESAME data for provided inputs
 - *get_sesame_data* : fetch SESAME data to STDOUT
 - *convert_sesame_data* : convert between standard SESAME file formats
 - *hugoniot.pl* : calculate various thermodynamic data along a hugoniot
 - *release.pl* : calculate various thermodynamic data along an adiabatic shock release
 - *mphase.pl* : plot the phase diagram represented by the SESAME 321 table
 - *init-eos.pl* : calculate the EOS initialization data for a specified MATID at STP or alternative pressure and/or temperature
 - *sesplot* : plot SESAME data and other various SESAME queries/operations
- Python based (new FY20/21)
 - *SOPES* : an API layer that provides python wrapper to the EOSPAC6 compiled library
 - *eosplot* : plot SESAME data that is interpolated using EOSPAC6 via the SOPES API layer
 - *validate.py* : validate SESAME data for prescribed tests (under development)

Embedded help / Documentation

The majority of scripts are added to one's PATH via the `eospac` module. Each tool's online help can be accessed passing either the `-h` and/or `--help` option.

Example

```
interp_sesame_data -h  
get_sesame_data -h  
.....  
eosplot --help  
validate.py --help
```

interp_sesame_data

- Interpolate SESAME data for provided inputs

Example

```
interp_sesame_data -d -g -n 5 3720 EOS_Pt_DT 2:10 298.15
```

will evaluate 5 double precision total pressure (Pt) and partial derivative values for a density range of 2-10 g/cc at 298.15 K, and stream the metadata as comments.

```
#
# Unclassified
#
# RATIONAL INTERPOLATION
# EOS_Pt_DT: Total Pressure (GPa) (Density (Mg/m^3)- and Temperature (K)-dependent) :: /local/data/eos/sesame
#          D          T          Pt          dPt/dD          dPt/dT
2.672150173567271e+00  2.981500000000000e+02  -7.409367862351063e-01  2.528329207288347e+01  4.910223009169588e-03
3.915506341381815e+00  2.981500000000000e+02  6.155783392191446e+01  7.341338573462292e+01  4.287915008821412e-03
5.826479378861022e+00  2.981500000000000e+02  2.714278422737816e+02  1.446064373185576e+02  2.645894788555483e-03
7.438752026632138e+00  2.981500000000000e+02  5.536510346516640e+02  2.065563398690592e+02  1.701926427036065e-03
9.129317886179621e+00  2.981500000000000e+02  9.605906977335721e+02  2.752249347876867e+02  1.341716707340586e-03
```

Common EOSPAC datatypes for Total EOS (301): Used in interp_sesame_data, sesplot

<i>EOS_Pt_DT</i>	$P(\rho, T)$
<i>EOS_Ut_DT</i>	$U(\rho, T)$
<i>EOS_At_DT</i>	$A(\rho, T)$
<i>EOS_St_DT</i>	$S(\rho, T)$
<i>EOS_Gt_DT</i>	$G(\rho, T)$
<i>EOS_Ht_DT</i>	$H(\rho, T)$
<i>EOS_CVt_DT</i>	$C_V(\rho, T)$
<i>EOS_CP_DT</i>	$C_P(\rho, T)$
<i>EOS_BSt_DT</i>	$B_S(\rho, T)$
<i>EOS_BTt_DT</i>	$B_T(\rho, T)$
<i>EOS_ALPHAt_DT</i>	$\alpha(\rho, T)$

Table: iso-choric/iso-thermal

<i>EOS_D_PtT</i>	$\rho(P, T)$
<i>EOS_T_DPt</i>	$T(\rho, P)$
<i>EOS_Ut_DPt</i>	$U(\rho, P)$

Table: iso-baric

<i>EOS_T_DUt</i>	$U(\rho, T)$
<i>EOS_Pt_DUt</i>	$A(\rho, T)$

Table: iso-energy

Complete list of tabletypes see
EOSPAC6 documentation Appendix B
<https://xweb.lanl.gov/projects/data/eos/>

get_sesame_data

- Fetch SESAME data to STDOUT

Example

```
get_sesame_data -g 3720 301 1
```

will fetch the total pressure table for material 3720 as a three-column stream with the metadata as comments.

```
# Material Number:      3720
# Classification:      Unclassified
# Table Number:        301
# Data file name:      /local/data/eos/sesame
# Creation date:       8/28/2003
# Latest update date:  2/27/2004
# NR:                  111
# NT:                  78
#
# Total Pressure (GPa) (Density (Mg/m^3)- and Temperature (K)-dependent) :: /local/data/eos/sesame
#      D                Pt                T
0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
2.700000000000000E-06  -1.237993607594306E-07  0.000000000000000E+00
5.400000000000000E-06  -3.102655959337747E-07  0.000000000000000E+00
1.350000000000000E-05  -1.045208686262876E-06  0.000000000000000E+00
```

get_sesame_data: continued

- Query all available material ID's

```
get_sesame_data id
```

```
693 materials found
```

```
1540 2020 2022 2023 2024 2030 2100 2110 2140 2145 2160 2290 2291 2292 2293 2330 2360 2370 2441 2448 2460 2550 2551 2552 2680 2700
2705 2720 2721 2740 2810 2860 2880 2961 2962 2963 2980 2981 2982 2983 2984 3050 3070 3100 3101 3120 3140 3180 3200 3210 3280 3332
3333 3334 3336 3337 3510 3520 3541 3560 3660 3713 3715 3716 3717 3718 3719 3720 3730 3731 3810 3811 3830 3950 4100 4210 4270 4271
4272 5000 5001 5010 5011 5030 5040 5171 5172 5173 5180 5181 5190 5191 5210 5211 5212 5250 5251 5263 5265 5266 5267 5271 5272 5280
5300 5410 5411 5500 5501 5502 5520 5530 5531 5540 5550 5560 5760 5761 5762 5770 5780 5781 6013 6020 7010 7020 7030 7081 7082 7100
7102 7111 7112 7120 7121 7122 7130 7150 7152 7153 7154 7155 7160 7171 7173 7180 7190 7230 7244 7245 7246 7247 7252 7270 7271 7281
7282 7283 7330 7331 7343 7353 7360 7361 7362 7363 7371 7372 7373 7380 7381 7383 7385 7386 7387 7390 7391 7410 7411 7432 7440 7450
7460 7470 7480 7510 7520 7521 7530 7541 7542 7550 7560 7561 7570 7580 7590 7591 7592 7593 7601 7602 7603 7610 7611 7612 7613 7660
7740 7741 7750 7760 7761 7770 7771 7830 7831 7832 7833 7834 7930 7931 7940 7941 7970 7971 7980 7981 8010 8020 8200 8210 8211
12020 12021 12023 12025 12026 12030 12031 12032 12081 12100 12110 12120 12140 12142 12143 12160 12181 12182 12290 12291 12293 ...
```

- Query all available material ID's in a specific SESAME file

```
get_sesame_data id $EOSDEV/Al_multiphase/Al6061/Al6061_bin
```

```
5 materials found
```

```
90001 90002 90003 90004 93721
```

convert_sesame_data

- Convert a SESAME file from one supported format to another (ASCII, ASCII2, big endian ieee64 binary, little endian ieee64 binary, and HELIOS). For Example,

```
convert_sesame_data -q 3720.bin 3720.ascii  
convert_sesame_data -q -t A 3720.bin
```

both convert a SESAME binary file (3720.bin) to a legacy SESAME fixed-format ASCII file where the output files are named 3720.ascii and 3720.bin.ascii respectively.

Likewise the new ASCII2 and the legacy HELIOS formats are supported (including the splitting into one material per file):

```
convert_sesame_data -q 3720.bin 3720.ascii2  
convert_sesame_data -q -t 2 3720.bin  
convert_sesame_data -q -t H 3720.bin  
convert_sesame_data -q -t H -s 3720.bin
```

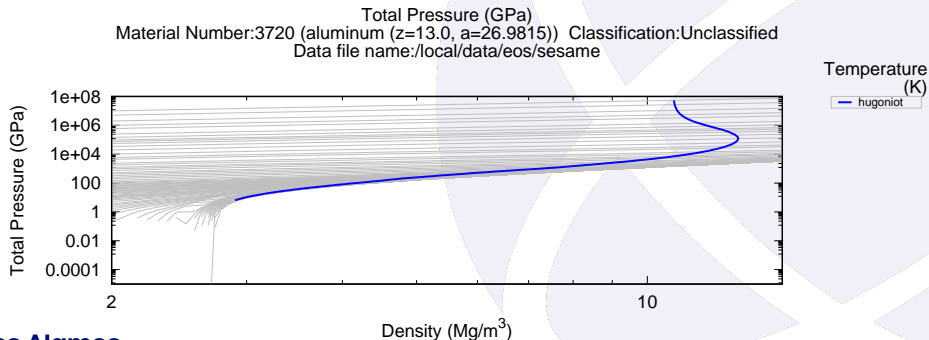
hugoniot.pl

Calculate various thermodynamic data along a hugoniot

Basic usage: `hugoniot.pl MATID optional initial conditions`

The data calculated by this tool include: *Density*[g/cc], *Temperature*[K], *Pressure*[GPa], *Energy*[MJ/kg], *Helmholtz*[MJ/kg], *Entropy*[MJ/kg/K], *(Sound Speed)²*[m²/s²] (i.e., $c^2 = \frac{dP}{d\rho}$ and $c^2 = -\frac{dS}{d\rho} / \frac{dS}{dP}$), *Compression*, *Us*[km/s], and *Up*[km/s]

```
hugoniot.pl 3720 -T 298.15 -n 200
```



release.pl

Calculate various thermodynamic data along an adiabatic shock release

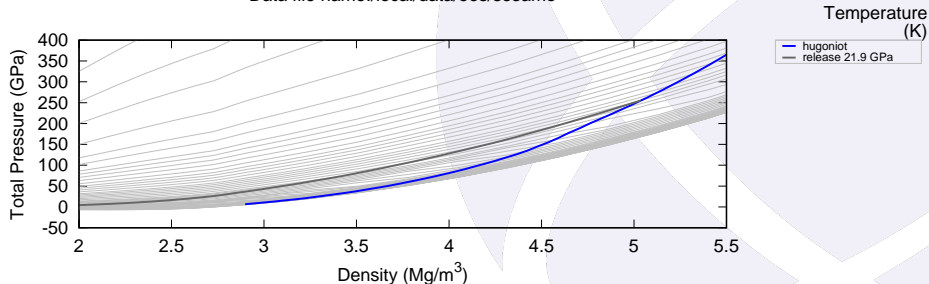
Basic usage: `release.pl MATID initial conditions`

The data calculated by this tool include: *Density*[g/cc], *Temperature*[K], *Pressure*[GPa], *Energy*[MJ/kg], *Helmholtz*[MJ/kg], *Entropy*[MJ/kg/K], and *(Sound Speed)²*[m²/s²] (i.e., $c^2 = \frac{dP}{d\rho}$ and $c^2 = -\frac{dS}{d\rho} / \frac{dS}{dP}$)

```
release.pl 3720 -r 5.03666115040678 -T 1.158262107553417e+04
```

Total Pressure (GPa)

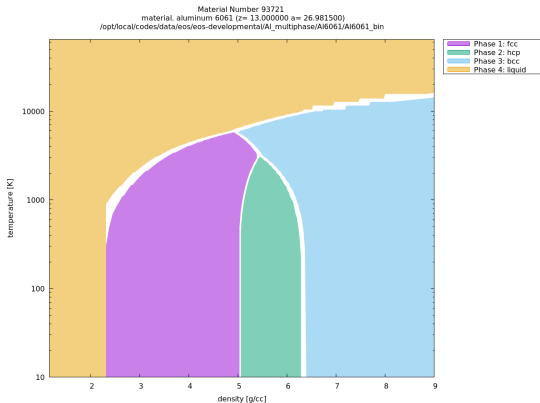
Material Number:3720 (aluminum (z=13.0, a=26.9815)) Classification:Unclassified
Data file name:/local/data/eos/sesame



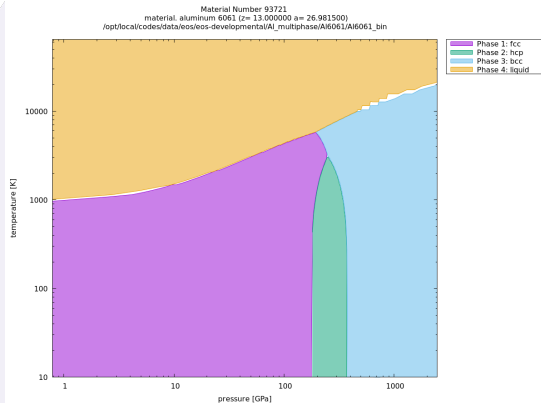
mpphase.pl

Plot the 321 table for a multiphase material (ρ, T) , (P, T)

```
mpphase.pl -F Al6061_bin 93721 -rt
```



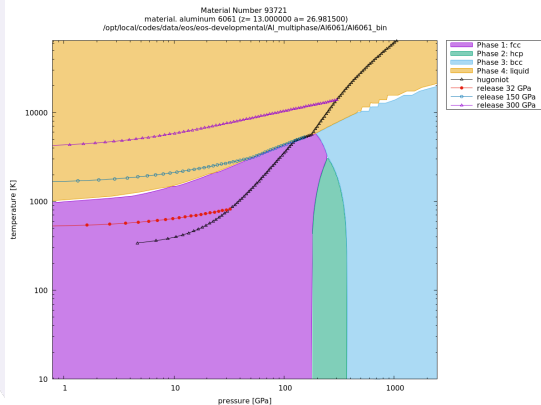
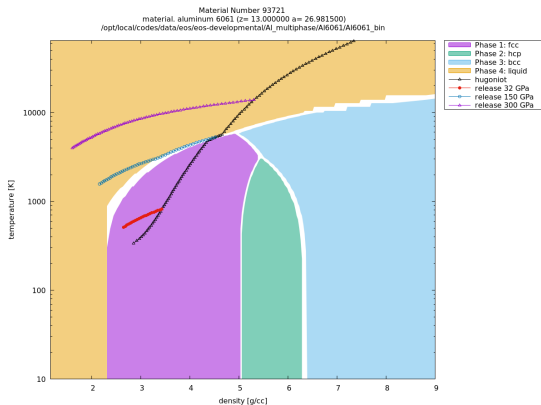
```
mpphase.pl -F Al6061_bin 93721 -pt
```



mphase.pl add hugoniot and release trajectories to plots

```
mphase.pl -F Al6061_bin 93721 -pt -rt -keepFiles
hugoniot.pl -F Al6061_bin 93721 -T 298.15 -Umax 10000 -n 200 -0
release.pl -F Al6061_bin 93721 -i ./hugoniot.dat

... gnuplot script magic happens ...
```



init-eos.pl

Calculate the EOS initialization data for a specified MATID at STP (or alternative pressure and/or temperature)

Example

```
init-eos.pl 3720
```

performs several calculations using `interp_sesame_data` to yield the following output:

```

1. fetch normal density at P=0.0001 GPa and 298.15 K => 2.700003583974167e+00 g/cc
2. fetch pressure at 2.700003583974167e+00 g/cc and 298.15 K
  2a. since $perr=|(9.99999999276213e-05 - 0.0001) / 0.0001|=7.23787620549726e-11 > 1.45060785288059e-11, iteratively solve
for density at 0.0001 GPa and 298.15 K
  WARNING! This is the best convergence possible: abs((9.99999999276213e-05 - 0.0001) / 0.0001) = 7.23787620549726e-11
  revert rho0 => 2.700003583974167e+00 g/cc
  P0 => 9.99999999276213e-05 GPa
  2b. pressure at 2.700003583974167e+00 g/cc and 298.15 K => 9.99999999276213e-05 GPa
3. fetch internal energy at 2.700003583974167e+00 g/cc and 298.15 K => -2.348837580943511e-06 MJ/kg
4. fetch pressure at 2.700003583974167e+00 g/cc and -2.348837580943511e-06 MJ/kg => 9.99999999276213e-05 GPa
5. fetch internal energy at 2.700003583974167e+00 g/cc and 298.15 K
6. fetch internal energy at 2.70000358397417 g/cc and 298.15 K
7. fetch helmholtz free energy at 2.700003583974167e+00 g/cc and 298.15 K
8. fetch helmholtz free energy at 2.70000358397417 g/cc and 298.15 K

```

init-eos.pl: continued

UNCLASSIFIED

Calculate the EOS initialization data for a specified MATID at STP (or alternative pressure and/or temperature)

SUMMARY -----

Material ID : 3720
Scaling ratio : 1
pressure : 9.999999999276213e-05 GPa
temperature : 298.15 K

Tabular -----
density : 2.700003583974167e+00 g/cc
internal energy : -2.348837580943511e-06 MJ/kg
helmholtz free energy : -3.136816885561139e-01 MJ/kg

Example

```
init-eos.pl --pressure .0001 --temperature 298.15 3720
```

will produce the identical output shown above.

There are other options available that are described in the online help page:

--density, --scaling_ratio, --DataFile, ...

sesplot

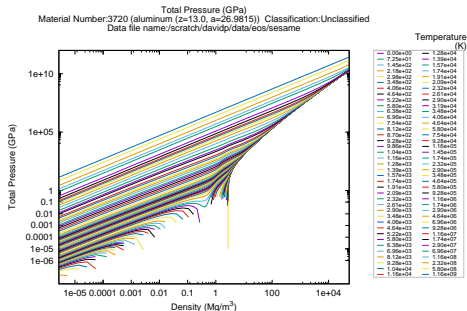
UNCLASSIFIED

Plot SESAME data using gnuplot and other various SESAME queries/operations

There is a very large number of options available in this tool, some of which are demonstrated below:

```
sesplot 3720 301 -ps \  
-Font "Helvetica" 14' -keyFont ',9'
```

```
sesplot 3720 301 -asinh y -xrange 1:3 \  
-ps sesplot.3720.301.1.asinh.ps \  
-Font "Helvetica" 14' -keyFont ',9'
```



Sopes

- Sopes is a little bit more than a python wrapper for EOSPAC6. It includes:
 - Common set of structures
 - Handles allocation and deallocation
 - Perhaps intuitive automatic type casting
 - Optional object oriented style

How can Sopes help?

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

15 #include <stdio.h>
16 #include <stdlib.h>
17 #include "eos_interface.h"
18
19 int main ()
20 {
21     enum
22     { nTablesE = 5 };
23     enum
24     { nXYPairsE = 4 };
25     enum
26     { nInfoItemsE = 12 };
27
28     EOS_INTEGER i, j;
29     EOS_REAL X[nXYPairsE], Y[nXYPairsE], F[nXYPairsE], dF[nXYPairsE],
30     dFy[nXYPairsE];
31     EOS_INTEGER tableType[nTablesE], numIndVars[nTablesE];
32     EOS_INTEGER matID[nTablesE];
33     EOS_INTEGER tableHandle[nTablesE];
34     EOS_INTEGER errorCode;
35     EOS_INTEGER tableHandleErrorCode;
36     EOS_INTEGER nTables;
37     EOS_INTEGER nXYPairs;
38     EOS_REAL infoVals[nInfoItemsE];
39     EOS_INTEGER nInfoItems;
40     EOS_INTEGER infoItems[nInfoItemsE] = {
41     EOS_cmnt_Len,
42     EOS_Exchange_Coeff,
43     EOS_F_Convert_Factor,
44     EOS_Log_Val,
45     EOS_Material_ID,
46     EOS_Team_Atomic_Bass,
47     EOS_Mean_Atomic_Num,
48     EOS_Modulus,
49     EOS_Normal_Density,
50     EOS_Table_Type
51 };
52 }
53
54 #eospac/eospac6/Source/Examples/TestC.c 12/2/26 13h
normal! zc
scotty !! 0-$ -bash 15 -bash 2*$-bash | |2024.06.24 16:42

```

Example of use found in EOSPAC6 in C

How can Sopes help? (Continued)

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

67 };
68 EOS_CHAR *tableTypeLabel[nTables] = {
69     "EOS_Pt_DT",
70     "EOS_Dv_T",
71     "EOS_Ogb",
72     "EOS_Comment",
73     "EOS_Info"
74 };
75 EOS_CHAR errorMessage[EOS_MaxErrMsgLen];
76
77 EOS_INTEGER one = 1;
78
79 nTables = nTablesE;
80 nXPairs = nXPairsE;
81 nInfoItems = nInfoItemsE;
82
83 /*
84  * EOS_Pt_DT, material 2148 works for Sesame table 381 (record type 1)
85  * EOS_Dv_T, material 2148 works for Sesame table 481 (record type 2)
86  * EOS_Ogb, material 2148 works for Sesame table 581 (record type 3)
87  * EOS_Comment, material 2148 works for Sesame tables 181-199 (record type 4)
88  * EOS_Info, material 2148 works for Sesame table 281 (record type 5)
89  */
90 tableType[0] = EOS_Pt_DT;
91 tableType[1] = EOS_Dv_T;
92 tableType[2] = EOS_Ogb;
93 tableType[3] = EOS_Comment;
94 tableType[4] = EOS_Info;
95
96 numIndVars[0] = 2;
97 numIndVars[1] = 1;
98 numIndVars[2] = 0;
99 numIndVars[3] = 0;
100 numIndVars[4] = 0;
101
102 matID[0] = 2148;
103
104 #eospac/eospac6/Source/Examples/TestC.c 84/236 35%
105 [scotty] 0-$ -bash 1$ -bash 2*$-bash | |2024.06.24 16:45
  
```

Example of use found in EOSPAC6 in C

How can Sopes help? (Continued)

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

103 matID[1] = 2140;
104 matID[2] = 12140;
105 matID[3] = 2140;
106 matID[4] = 2140;
107
108 errorCode = EOS_OK;
109 for (i = 0; i < nTables; i++) {
110     tableHandle[i] = 0;
111 }
112
113 /*
114  * initialize table data objects
115  */
116
117 eos.CreateTables (&nTables, tableType, matID, tableHandle, ErrorCode);
118 if (errorCode != EOS_OK) {
119     for (i = 0; i < nTables; i++) {
120         tableHandleErrorCode = EOS_OK;
121         eos.GetErrorCode (&tableHandle[i], &tableHandleErrorCode);
122         eos.GetErrorMessage (&tableHandleErrorCode, errorMessage);
123         printf ("eos.CreateTables ERROR %i: %s\n", tableHandleErrorCode,
124             errorMessage);
125     }
126 }
127
128 /*
129  * set some options
130  */
131
132 for (i = 0; i < nTables; i++) {
133     /* enable smoothing */
134     eos.SetOption (&tableHandle[i], &EOS_SMOOTH, EOS_NullPtr, ErrorCode);
135     if (errorCode != EOS_OK) {
136         eos.GetErrorMessage (ErrorCode, errorMessage);
137         printf ("eos.SetOption ERROR %i: %s\n", ErrorCode, errorMessage);
138     }
139 }
140
141 #eospac6/Source/Examples/TestC.c 178/236 50%
142 normal zc
143 scotty zc 0-5 -bash 15 -bash 2*5-bash | 1/20/24 06:24 16:47

```

Example of use found in EOSPAC6 in C

How can Sopes help? (Continued)

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

141 /*
142  * load data into table data objects
143  */
144
145 eos_LoadTables (&nTables, tableHandle, ErrorCode);
146 if (errorCode != EOS_OK) {
147     eos_GetErrorMessage (&errorCode, errorMessage);
148     printf ("eos_LoadTables ERROR %i: %s\n", errorCode, errorMessage);
149     for (i = 0; i < nTables; i++) {
150         tableHandleErrorCode = EOS_OK;
151         eos_GetErrorCode (&tableHandle[i], &tableHandleErrorCode);
152         eos_GetErrorMessage (&tableHandle[i], &tableHandleErrorCode, errorMessage);
153         printf ("eos_LoadTables ERROR %i (%i): %s\n", tableHandleErrorCode,
154             tableHandle[i], errorMessage);
155     }
156 }
157 /*
158  * interpolate -- errors codes are intentionally produced
159  */
160
161 X[0] = 3000 ;
162 X[1] = 6000 ;
163 X[2] = 0200 ;
164 X[3] = 0300 ;
165
166 Y[0] = 20000 0;
167 Y[1] = 620000 0;
168 Y[2] = 4000000 0;
169 Y[3] = 20000000 0;
170
171 for (i = 0; i < nTables; i++) {
172     printf ("in -- interpolate using tableType %s --\n", tableTypeLabel[i]);
173     eos_interpolate (&tableHandle[i], &nXYPairs, X, Y, F, dFx, dFy,
174         ErrorCode);
175     printf ("%s Interpolation Results\n", tableTypeLabel[i]);
176     if (errorCode != EOS_OK) {
177         eos_GetErrorMessage (&errorCode, errorMessage);
178         printf ("eos_interpolate ERROR %i (%i): %s\n", errorCode,
179             tableHandle[i], errorMessage);
180     }
181 }
182
183 #endif
184
185 #endif
186
187 #endif
188
189 #endif
190
191 #endif
192
193 #endif
194
195 #endif
196
197 #endif
198
199 #endif
200
201 #endif
202
203 #endif
204
205 #endif
206
207 #endif
208
209 #endif
210
211 #endif
212
213 #endif
214
215 #endif
216
217 #endif
218
219 #endif
220
221 #endif
222
223 #endif
224
225 #endif
226
227 #endif
228
229 #endif
230
231 #endif
232
233 #endif
234
235 #endif
236
237 #endif
238
239 #endif
240
241 #endif
242
243 #endif
244
245 #endif
246
247 #endif
248
249 #endif
250
251 #endif
252
253 #endif
254
255 #endif
256
257 #endif
258
259 #endif
260
261 #endif
262
263 #endif
264
265 #endif
266
267 #endif
268
269 #endif
270
271 #endif
272
273 #endif
274
275 #endif
276
277 #endif
278
279 #endif
280
281 #endif
282
283 #endif
284
285 #endif
286
287 #endif
288
289 #endif
290
291 #endif
292
293 #endif
294
295 #endif
296
297 #endif
298
299 #endif
300
301 #endif
302
303 #endif
304
305 #endif
306
307 #endif
308
309 #endif
310
311 #endif
312
313 #endif
314
315 #endif
316
317 #endif
318
319 #endif
320
321 #endif
322
323 #endif
324
325 #endif
326
327 #endif
328
329 #endif
330
331 #endif
332
333 #endif
334
335 #endif
336
337 #endif
338
339 #endif
340
341 #endif
342
343 #endif
344
345 #endif
346
347 #endif
348
349 #endif
350
351 #endif
352
353 #endif
354
355 #endif
356
357 #endif
358
359 #endif
360
361 #endif
362
363 #endif
364
365 #endif
366
367 #endif
368
369 #endif
370
371 #endif
372
373 #endif
374
375 #endif
376
377 #endif
378
379 #endif
380
381 #endif
382
383 #endif
384
385 #endif
386
387 #endif
388
389 #endif
390
391 #endif
392
393 #endif
394
395 #endif
396
397 #endif
398
399 #endif
400
401 #endif
402
403 #endif
404
405 #endif
406
407 #endif
408
409 #endif
410
411 #endif
412
413 #endif
414
415 #endif
416
417 #endif
418
419 #endif
420
421 #endif
422
423 #endif
424
425 #endif
426
427 #endif
428
429 #endif
430
431 #endif
432
433 #endif
434
435 #endif
436
437 #endif
438
439 #endif
440
441 #endif
442
443 #endif
444
445 #endif
446
447 #endif
448
449 #endif
450
451 #endif
452
453 #endif
454
455 #endif
456
457 #endif
458
459 #endif
460
461 #endif
462
463 #endif
464
465 #endif
466
467 #endif
468
469 #endif
470
471 #endif
472
473 #endif
474
475 #endif
476
477 #endif
478
479 #endif
480
481 #endif
482
483 #endif
484
485 #endif
486
487 #endif
488
489 #endif
490
491 #endif
492
493 #endif
494
495 #endif
496
497 #endif
498
499 #endif
500
501 #endif
502
503 #endif
504
505 #endif
506
507 #endif
508
509 #endif
510
511 #endif
512
513 #endif
514
515 #endif
516
517 #endif
518
519 #endif
520
521 #endif
522
523 #endif
524
525 #endif
526
527 #endif
528
529 #endif
530
531 #endif
532
533 #endif
534
535 #endif
536
537 #endif
538
539 #endif
540
541 #endif
542
543 #endif
544
545 #endif
546
547 #endif
548
549 #endif
550
551 #endif
552
553 #endif
554
555 #endif
556
557 #endif
558
559 #endif
560
561 #endif
562
563 #endif
564
565 #endif
566
567 #endif
568
569 #endif
570
571 #endif
572
573 #endif
574
575 #endif
576
577 #endif
578
579 #endif
580
581 #endif
582
583 #endif
584
585 #endif
586
587 #endif
588
589 #endif
590
591 #endif
592
593 #endif
594
595 #endif
596
597 #endif
598
599 #endif
600
601 #endif
602
603 #endif
604
605 #endif
606
607 #endif
608
609 #endif
610
611 #endif
612
613 #endif
614
615 #endif
616
617 #endif
618
619 #endif
620
621 #endif
622
623 #endif
624
625 #endif
626
627 #endif
628
629 #endif
630
631 #endif
632
633 #endif
634
635 #endif
636
637 #endif
638
639 #endif
640
641 #endif
642
643 #endif
644
645 #endif
646
647 #endif
648
649 #endif
650
651 #endif
652
653 #endif
654
655 #endif
656
657 #endif
658
659 #endif
660
661 #endif
662
663 #endif
664
665 #endif
666
667 #endif
668
669 #endif
670
671 #endif
672
673 #endif
674
675 #endif
676
677 #endif
678
679 #endif
680
681 #endif
682
683 #endif
684
685 #endif
686
687 #endif
688
689 #endif
690
691 #endif
692
693 #endif
694
695 #endif
696
697 #endif
698
699 #endif
700
701 #endif
702
703 #endif
704
705 #endif
706
707 #endif
708
709 #endif
710
711 #endif
712
713 #endif
714
715 #endif
716
717 #endif
718
719 #endif
720
721 #endif
722
723 #endif
724
725 #endif
726
727 #endif
728
729 #endif
730
731 #endif
732
733 #endif
734
735 #endif
736
737 #endif
738
739 #endif
740
741 #endif
742
743 #endif
744
745 #endif
746
747 #endif
748
749 #endif
750
751 #endif
752
753 #endif
754
755 #endif
756
757 #endif
758
759 #endif
760
761 #endif
762
763 #endif
764
765 #endif
766
767 #endif
768
769 #endif
770
771 #endif
772
773 #endif
774
775 #endif
776
777 #endif
778
779 #endif
780
781 #endif
782
783 #endif
784
785 #endif
786
787 #endif
788
789 #endif
790
791 #endif
792
793 #endif
794
795 #endif
796
797 #endif
798
799 #endif
800
801 #endif
802
803 #endif
804
805 #endif
806
807 #endif
808
809 #endif
810
811 #endif
812
813 #endif
814
815 #endif
816
817 #endif
818
819 #endif
820
821 #endif
822
823 #endif
824
825 #endif
826
827 #endif
828
829 #endif
830
831 #endif
832
833 #endif
834
835 #endif
836
837 #endif
838
839 #endif
840
841 #endif
842
843 #endif
844
845 #endif
846
847 #endif
848
849 #endif
850
851 #endif
852
853 #endif
854
855 #endif
856
857 #endif
858
859 #endif
860
861 #endif
862
863 #endif
864
865 #endif
866
867 #endif
868
869 #endif
870
871 #endif
872
873 #endif
874
875 #endif
876
877 #endif
878
879 #endif
880
881 #endif
882
883 #endif
884
885 #endif
886
887 #endif
888
889 #endif
890
891 #endif
892
893 #endif
894
895 #endif
896
897 #endif
898
899 #endif
900
901 #endif
902
903 #endif
904
905 #endif
906
907 #endif
908
909 #endif
910
911 #endif
912
913 #endif
914
915 #endif
916
917 #endif
918
919 #endif
920
921 #endif
922
923 #endif
924
925 #endif
926
927 #endif
928
929 #endif
930
931 #endif
932
933 #endif
934
935 #endif
936
937 #endif
938
939 #endif
940
941 #endif
942
943 #endif
944
945 #endif
946
947 #endif
948
949 #endif
950
951 #endif
952
953 #endif
954
955 #endif
956
957 #endif
958
959 #endif
960
961 #endif
962
963 #endif
964
965 #endif
966
967 #endif
968
969 #endif
970
971 #endif
972
973 #endif
974
975 #endif
976
977 #endif
978
979 #endif
980
981 #endif
982
983 #endif
984
985 #endif
986
987 #endif
988
989 #endif
990
991 #endif
992
993 #endif
994
995 #endif
996
997 #endif
998
999 #endif
1000
1001 #endif
1002
1003 #endif
1004
1005 #endif
1006
1007 #endif
1008
1009 #endif
1010
1011 #endif
1012
1013 #endif
1014
1015 #endif
1016
1017 #endif
1018
1019 #endif
1020
1021 #endif
1022
1023 #endif
1024
1025 #endif
1026
1027 #endif
1028
1029 #endif
1030
1031 #endif
1032
1033 #endif
1034
1035 #endif
1036
1037 #endif
1038
1039 #endif
1040
1041 #endif
1042
1043 #endif
1044
1045 #endif
1046
1047 #endif
1048
1049 #endif
1050
1051 #endif
1052
1053 #endif
1054
1055 #endif
1056
1057 #endif
1058
1059 #endif
1060
1061 #endif
1062
1063 #endif
1064
1065 #endif
1066
1067 #endif
1068
1069 #endif
1070
1071 #endif
1072
1073 #endif
1074
1075 #endif
1076
1077 #endif
1078
1079 #endif
1080
1081 #endif
1082
1083 #endif
1084
1085 #endif
1086
1087 #endif
1088
1089 #endif
1090
1091 #endif
1092
1093 #endif
1094
1095 #endif
1096
1097 #endif
1098
1099 #endif
1100
1101 #endif
1102
1103 #endif
1104
1105 #endif
1106
1107 #endif
1108
1109 #endif
1110
1111 #endif
1112
1113 #endif
1114
1115 #endif
1116
1117 #endif
1118
1119 #endif
1120
1121 #endif
1122
1123 #endif
1124
1125 #endif
1126
1127 #endif
1128
1129 #endif
1130
1131 #endif
1132
1133 #endif
1134
1135 #endif
1136
1137 #endif
1138
1139 #endif
1140
1141 #endif
1142
1143 #endif
1144
1145 #endif
1146
1147 #endif
1148
1149 #endif
1150
1151 #endif
1152
1153 #endif
1154
1155 #endif
1156
1157 #endif
1158
1159 #endif
1160
1161 #endif
1162
1163 #endif
1164
1165 #endif
1166
1167 #endif
1168
1169 #endif
1170
1171 #endif
1172
1173 #endif
1174
1175 #endif
1176
1177 #endif
1178
1179 #endif
1180
1181 #endif
1182
1183 #endif
1184
1185 #endif
1186
1187 #endif
1188
1189 #endif
1190
1191 #endif
1192
1193 #endif
1194
1195 #endif
1196
1197 #endif
1198
1199 #endif
1200
1201 #endif
1202
1203 #endif
1204
1205 #endif
1206
1207 #endif
1208
1209 #endif
1210
1211 #endif
1212
1213 #endif
1214
1215 #endif
1216
1217 #endif
1218
1219 #endif
1220
1221 #endif
1222
1223 #endif
1224
1225 #endif
1226
1227 #endif
1228
1229 #endif
1230
1231 #endif
1232
1233 #endif
1234
1235 #endif
1236
1237 #endif
1238
1239 #endif
1240
1241 #endif
1242
1243 #endif
1244
1245 #endif
1246
1247 #endif
1248
1249 #endif
1250
1251 #endif
1252
1253 #endif
1254
1255 #endif
1256
1257 #endif
1258
1259 #endif
1260
1261 #endif
1262
1263 #endif
1264
1265 #endif
1266
1267 #endif
1268
1269 #endif
1270
1271 #endif
1272
1273 #endif
1274
1275 #endif
1276
1277 #endif
1278
1279 #endif
1280
1281 #endif
1282
1283 #endif
1284
1285 #endif
1286
1287 #endif
1288
1289 #endif
1290
1291 #endif
1292
1293 #endif
1294
1295 #endif
1296
1297 #endif
1298
1299 #endif
1300
1301 #endif
1302
1303 #endif
1304
1305 #endif
1306
1307 #endif
1308
1309 #endif
1310
1311 #endif
1312
1313 #endif
1314
1315 #endif
1316
1317 #endif
1318
1319 #endif
1320
1321 #endif
1322
1323 #endif
1324
1325 #endif
1326
1327 #endif
1328
1329 #endif
1330
1331 #endif
1332
1333 #endif
1334
1335 #endif
1336
1337 #endif
1338
1339 #endif
1340
1341 #endif
1342
1343 #endif
1344
1345 #endif
1346
1347 #endif
1348
1349 #endif
1350
1351 #endif
1352
1353 #endif
1354
1355 #endif
1356
1357 #endif
1358
1359 #endif
1360
1361 #endif
1362
1363 #endif
1364
1365 #endif
1366
1367 #endif
1368
1369 #endif
1370
1371 #endif
1372
1373 #endif
1374
1375 #endif
1376
1377 #endif
1378
1379 #endif
1380
1381 #endif
1382
1383 #endif
1384
1385 #endif
1386
1387 #endif
1388
1389 #endif
1390
1391 #endif
1392
1393 #endif
1394
1395 #endif
1396
1397 #endif
1398
1399 #endif
1400
1401 #endif
1402
1403 #endif
1404
1405 #endif
1406
1407 #endif
1408
1409 #endif
1410
1411 #endif
1412
1413 #endif
1414
1415 #endif
1416
1417 #endif
1418
1419 #endif
1420
1421 #endif
1422
1423 #endif
1424
1425 #endif
1426
1427 #endif
1428
1429 #endif
1430
1431 #endif
1432
1433 #endif
1434
1435 #endif
1436
1437 #endif
1438
1439 #endif
1440
1441 #endif
1442
1443 #endif
1444
1445 #endif
1446
1447 #endif
1448
1449 #endif
1450
1451 #endif
1452
1453 #endif
1454
1455 #endif
1456
1457 #endif
1458
1459 #endif
1460
1461 #endif
1462
1463 #endif
1464
1465 #endif
1466
1467 #endif
1468
1469 #endif
1470
1471 #endif
1472
1473 #endif
1474
1475 #endif
1476
1477 #endif
1478
1479 #endif
1480
1481 #endif
1482
1483 #endif
1484
1485 #endif
1486
1487 #endif
1488
1489 #endif
1490
1491 #endif
1492
1493 #endif
1494
1495 #endif
1496
1497 #endif
1498
1499 #endif
1500
1501 #endif
1502
1503 #endif
1504
1505 #endif
1506
1507 #endif
1508
1509 #endif
1510
1511 #endif
1512
1513 #endif
1514
1515 #endif
1516
1517 #endif
1518
1519 #endif
1520
1521 #endif
1522
1523 #endif
1524
1525 #endif
1526
1527 #endif
1528
1529 #endif
1530
1531 #endif
1532
1533 #endif
1534
1535 #endif
1536
1537 #endif
1538
1539 #endif
1540
1541 #endif
1542
1543 #endif
1544
1545 #endif
1546
1547 #endif
1548
1549 #endif
1550
1551 #endif
1552
1553 #endif
1554
1555 #endif
1556
1557 #endif
1558
1559 #endif
1560
1561 #endif
1562
1563 #endif
1564
1565 #endif
1566
1567 #endif
1568
1569 #endif
1570
1571 #endif
1572
1573 #endif
1574
1575 #endif
1576
1577 #endif
1578
1579 #endif
1580
1581 #endif
1582
1583 #endif
1584
1585 #endif
1586
1587 #endif
1588
1589 #endif
1590
1591 #endif
1592
1593 #endif
1594
1595 #endif
1596
1597 #endif
1598
1599 #endif
1600
1601 #endif
1602
1603 #endif
1604
1605 #endif
1606
1607 #endif
1608
1609 #endif
1610
1611 #endif
1612
1613 #endif
1614
1615 #endif
1616
1617 #endif
1618
1619 #endif
1620
1621 #endif
1622
1623 #endif
1624
1625 #endif
1626
1627 #endif
1628
1629 #endif
1630
1631 #endif
1632
1633 #endif
1634
1635 #endif
1636
1637 #endif
1638
1639 #endif
1640
1641 #endif
1642
1643 #endif
1644
1645 #endif
1646
1647 #endif
1648
1649 #endif
1650
1651 #endif
1652
1653 #endif
1654
1655 #endif
1656
1657 #endif
1658
1659 #endif
1660
1661 #endif
1662
1663 #endif
1664
1665 #endif
1666
1667 #endif
1668
1669 #endif
1670
1671 #endif
1672
1673 #endif
1674
1675 #endif
1676
1677 #endif
1678
1679 #endif
1680
1681 #endif
1682
1683 #endif
1684
1685 #endif
1686
1687 #endif
1688
1689 #endif
1690
1691 #endif
1692
1693 #endif
1694
1695 #endif
1696
1697 #endif
1698
1699 #endif
1700
1701 #endif
1702
1703 #endif
1704
1705 #endif
1706
1707 #endif
1708
1709 #endif
1710
1711 #endif
1712
1713 #endif
1714
1715 #endif
1716
1717 #endif
1718
1719 #endif
1720
1721 #endif
1722
1723 #endif
1724
1725 #endif
1726
1727 #endif
1728
1729 #endif
1730
1731 #endif
1732
1733 #endif
1734
1735 #endif
1736
1737 #endif
1738
1739 #endif
1740
1741 #endif
1742
1743 #endif
1744
1745 #endif
1746
1747 #endif
1748
1749 #endif
1750
1751 #endif
1752
1753 #endif
1754
1755 #endif
1756
1757 #endif
1758
1759 #endif
1760
1761 #endif
1762
1763 #endif
1764
1765 #endif
1766
1767 #endif
1768
1769 #endif
1770
1771 #endif
1772
1773 #endif
1774
1775 #endif
1776
1777 #endif
1778
1779 #endif
1780
1781 #endif
1782
1783 #endif
1784
1785 #endif
1786
1787 #endif
1788
1789 #endif
1790
1791 #endif
1792
1793 #endif
1794
1795 #endif
1796
1797 #endif
1798
1799 #endif
1800
1801 #endif
1802
1803 #endif
1804
1805 #endif
1806
1807 #endif
1808
1809 #endif
1810
1811 #endif
1812
1813 #endif
1814
1815 #endif
1816
1817 #endif
1818
1819 #endif
1820
1821 #endif
1822
1823 #endif
1824
1825 #endif
1826
1827 #endif
1828
1829 #endif
1830
1831 #endif
1832
1833 #endif
1834
1835 #endif
1836
1837 #endif
1838
1839 #endif
1840
1841 #endif
1842
1843 #endif
1844
1845 #endif
1846
1847 #endif
1848
1849 #endif
1850
1851 #endif
1852
1853 #endif
1854
1855 #endif
1856
1857 #endif
1858
1859 #endif
1860
1861 #endif
1862
1863 #endif
1864
1865 #endif
1866
1867 #endif
1868
1869 #endif
1870
1871 #endif
1872
1873 #endif
1874
1875 #endif
1876
1877 #endif
1878
1879 #endif
1880
1881 #endif
1882
1883 #endif
1884
1885 #endif
1886
1887 #endif
1888
1889 #endif
1890
1891 #endif
1892
1893 #endif
1894
1895 #endif
1896
1897 #endif
1898
1899 #endif
1900
1901 #endif
1902
1903 #endif
1904
1905 #endif
1906
1907 #endif
1908
1909 #endif
1910
1911 #endif
1912
1913 #endif
1914
1915 #endif
1916
1917 #endif
1918
1919 #endif
1920
1921 #endif
1922
1923 #endif
1924
1925 #endif
1926
1927 #endif
1928
1929 #endif
1930
1931 #endif
1932
1933 #endif
1934
1935 #endif
1936
1937 #endif
1938
1939 #endif
1940
1941 #endif
1942
1943 #endif
1944
1945 #endif
1946
1947 #endif
1948
1949 #endif
1950
1951 #endif
1952
1953 #endif
1954
1955 #endif
1956
1957 #endif
1958
1959 #endif
1960
1961 #endif
1962
1963 #endif
1964
1965 #endif
1966
1967 #endif
1968
1969 #endif
1970
1971 #endif
1972
1973 #endif
1974
1975 #endif
1976
1977 #endif
1978
1979 #endif
1980
1981 #endif
1982
1983 #endif
1984
1985 #endif
1986
1987 #endif
1988
1989 #endif
1990
1991 #endif
1992
1993 #endif
1994
1995 #endif
1996
1997 #endif
1998
1999 #endif
2000
2001 #endif
2002
2003 #endif
2004
2005 #endif
2006
2007 #endif
2008
2009 #endif
2010
2011 #endif
2012
2013 #endif
2014
2015 #endif
2016
2017 #endif
2018
2019 #endif
2020
2021 #endif
2022
2023 #endif
2024
2025 #endif
2026
2027 #endif
2028
2029 #endif
2030
2031 #endif
2032
2033 #endif
2034
2035 #endif
2036
2037 #endif
2038
2039 #endif
2040
2041 #endif
2042
2043 #endif
2044
2045 #endif
2046
2047 #endif
2048
2049 #endif
2050
2051 #endif
2052
2053 #endif
2054
2055 #endif
2056
2057 #endif
2058
2059 #endif
2060
2061 #endif
2062
2063 #endif
2064
2065 #endif
2066
2067 #endif
2068
2069 #endif
2070
2071 #endif
2072
2073 #endif
2074
2075 #endif
2076
2077 #endif
2078
2079 #endif
2080
2081 #endif
2082
2083 #endif
2084
2085 #endif
2086
2087 #endif
2088
2089 #endif
2090
2091 #endif
2092
2093 #endif
2094
2095 #endif
2096
2097 #endif
2098
2099 #endif
2100
2101 #endif
2102
2103 #endif
2104
2105 #endif
2106
2107 #endif
2108
2109 #endif
2110
2111 #endif
2112
2113 #endif
2114
2115 #endif
2116
2117 #endif
2118
2119 #endif
2120
2121 #endif
2122
2123 #endif
2124
2125 #endif
2126
2127 #endif
2128
2129 #endif
2130
2131 #endif
2132
2133 #endif
2134
2135 #endif
2136
2137 #endif
2138
2139 #endif
2140
2141 #endif
2142
2143 #endif
2144
2145 #endif
2146
2147 #endif
2148
2149 #endif
2150
2151 #endif
2152
2153 #endif
2154
2155 #endif
2156
2157 #endif
2158
2159 #endif
2160
2161 #endif
2162
2163 #endif
2164
2165 #endif
2166
2167 #endif
2168
2169 #endif
2170
2171 #endif
2172
2173 #endif
2174
2175 #endif
2176
2177 #endif
2178
2179 #endif
2180
2181 #endif
2182
2183 #endif
2184
2185 #endif
2186
2187 #endif
2188
2189 #endif
2190
2191 #endif
2192
2193 #endif
2194
2195 #endif
2196
2197 #endif
2198
2199 #endif
2200
2201 #endif
2202
2203 #endif
2204
2205 #endif
2206
2207 #endif
2208
2209 #endif
2210
2211 #endif
2212
2213 #endif
2214
2215 #endif
2216
2217 #endif
2218
2219 #endif
2220
2221 #endif
2222
2223 #endif
2224
2225 #endif
2226
2227 #endif
2228
2229 #endif
2230
2231 #endif
2232
2233 #endif
2234
2235 #endif
2236
2237 #endif
2238
2239 #endif
2240
2241 #endif
2242
2243 #endif
2244
2245 #endif
2246
2247 #endif
2248
2249
```

How can Sopes help? (Continued)

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

197
198 for (i = 0; i < nTables; i++) {
199     printf ("Name: Table Information for tableType %s , tableHandle %s\n",
200            tableTypeLabel[i], tableHandle[i]);
201     for (j = 0; j < nInfoVals; j++) {
202         EOS_BOOLEAN equal;
203         eos_GetTableInfo (&(tableHandle[i]), &one, &(infoVals[j]),
204                           &(infoVals[j]), &errorCode);
205         eos_ErrorCodesEqual((EOS_INTEGER*)&EOS_INVALID_ID_INFO_FLAG, &errorCode, &equal);
206
207         if (errorCode == EOS_OK) {
208             printf ("%21s %25s %33.6f\n", j + 1, infoVals[j],
209                    infoVals[j]);
210         }
211         else if (!equal) {
212             /* Ignore EOS_INVALID_ID_INFO_FLAG since not all infoVals are currently
213                applicable to a specific tableHandle. */
214             eos_GetErrorMessage (&errorCode, &errorMessage);
215             printf ("eos_GetTableInfo ERROR %i: %s\n", errorCode, errorMessage);
216         }
217     }
218 }
219 /*
220  * Destroy all data objects
221  */
222
223 eos_DestroyAll (&errorCode);
224 if (errorCode != EOS_OK) {
225     for (i = 0; i < nTables; i++) {
226         tableHandleErrorCode = EOS_OK;
227         eos_GetErrorCode (&tableHandle[i], &tableHandleErrorCode);
228         eos_GetErrorMessage (&tableHandleErrorCode, &errorMessage);
229         printf ("eos_DestroyAll ERROR %i: %s\n", tableHandleErrorCode,
230                errorMessage);
231     }
232 }
233
234 //eospac6/eospac6/Source/Examples/TestC.c 213/236 90%
235
236 [scotty] 0-5 -bash 1$ -bash 2*5 -bash | 2024-06-24 16:54

```

Example of use found in EOSPAC6 in C

How can Sopes help? (Continued)

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

jospateron ~ ssh - bash connect@centy -- 07:39
1 import sopes
2 tableTypes = ["Pt_DT", "Dv_T", "Dgh", "Comment", "Info"]
3 matId = [2140, 2140, 12140, 2140, 2140]
4 Options= {"None"} * 5
5 Options[0]="EOS_SHOOTH"
6
7 A = sopes.load_tables(tableTypes, matId)
8 print(A[0])
9 for entry in A:
10     print(entry)
11     try:
12         sopes.option(A[entry], "SHOOTH")
13     except:
14         print(f"[entry] is not something that can be smoothed")
15
16 X = [3000, 6000, 0200, 0300]
17 Y = [20000.0, 620000.0, 4000000.0, 20000000.0]
18
19 for interp_entry in A:
20     try:
21         B = sopes.interpolate(interp_entry, X, Y)
22
23         for entry in B:
24             print(entry)
25     except:
26         print(f"[interp_entry] not interpolatable")
27
28 infoItems = [
29     "EOS_Cmnt_Len",
30     "EOS_Exchange_Coeff",
31     "EOS_T_Convert_Factor",
32     "EOS_Lng_Val",
33     "EOS_Material_ID",
34     "EOS_Mean_Atomic_Bass",
35     "EOS_Mean_Atomic_Rm",
36 ]
37
38 /test/testSopes.py 1/34 13
39
40 [scotty] 0*5-bash) 15 -bash 2-5 -bash ||20/24.96.24 17:02
  
```

Equivalent Sopes example

How can Sopes help? (Continued)

- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)

```

1 import sopes
2
3 tableTypes = ["Pt_DT", "Dv_T", "Bgh", "Comment", "Info"]
4 matid = [2140, 2140, 12140, 2140, 2140]
5
6 Tabs = []
7 for TType, mat in zip(tableTypes, matid):
8     try:
9         Tabs.append(sopes.Table(TType, mat, options={"SMOOTH":None}))
10    except:
11        Tabs.append(sopes.Table(TType, mat))
12
13 X = [1000, 6000, 8200, 8300]
14 Y = [20000.0, 62000.0, 400000.0, 20000000.0]
15
16 for interp in Tabs:
17     try:
18         print(interp)
19         B = interp.interpolate_many(X, Y)
20     except:
21         print(entry)
22     except:
23         print(f"{interp} not interpolatable")
24
25 infotems = [
26     "EDS_Cent_Len",
27     "EDS_Exchange_Coeff",
28     "EDS_F_Convert_Factor",
29     "EDS_Lag_Val",
30     "EDS_Material_ID",
31     "EDS_Mean_Atomic_Mass",
32     "EDS_Mean_Atomic_Num",
33     "EDS_Modulus",
34     "EDS_Normal_Density",
35     "EDS_Table_Type",
36 ]
37
38 ~/test/testTable.py 1/32 1%
39 ~/test/testTable.py* 52L, 2137C
40 (scotty) 0*5-bash) 15 -bash 2-5 -bash | |20/24.96.24 17:05

```

Equivalent Sopes example

Learning To Use Sopes

- Sopes does use a different set of calls from EOSPAC6
- For more information https://xweb.lanl.gov/projects/data/eos/sopes/html_wrapper.php?doc/index.html

The screenshot shows the website for the 'sopes' module. The main heading is 'sopes, a Python 2 and 3 Interface for EOSPAC 6' and 'sopes module'. Below this is a photograph of a taco. The page contains several sections of text, including a 'Table of Contents' with links to 'Categories of functions/methods' and 'Stateful (object-oriented)'. A 'NOTE' at the bottom states: 'NOTE: If numpy is available, it will return array values as numpy arrays, rather than wrap.array.'

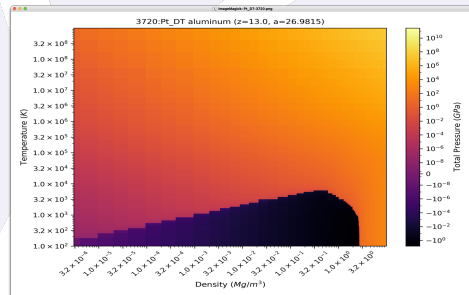
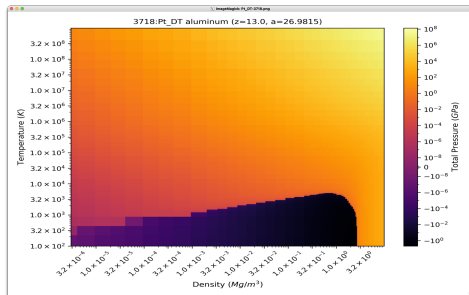
EOSplot

- Reinventing the wheel. Command line tools and scripting tool again. But for python

EOSplot Example

Command from within a script:

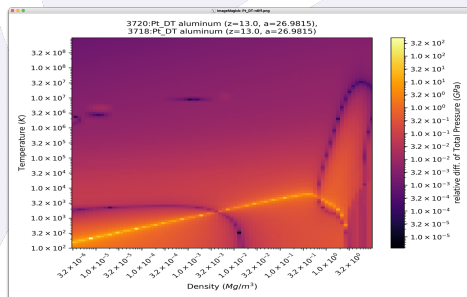
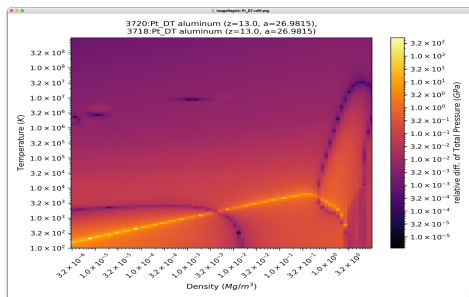
```
table1 = sopes.Table("Pt_DT", 3720)
plot_config = eosplot.plot_table(table1, xmin=3e-6, xmax=10, ymin=1e2, ymax=1e9,
style=eosplot.PlotStyle.COLORMAP)
plt.show()
```



EOSplot Example

Command from terminal:

```
ml python
ml eosplot
eosplot 3720:Pt_DT-3718:Pt_DT -xmin 3e-6 -xmax 10 -ymin 1e2 -ymax 1e9
```



validate.py

under development

Basic usage: `validate.py -f sesame_file -m MATID`

- Tests for completeness of material
 - Does 101 exist?
 - Does 201 exist?
 - ...
- ASCII2 format checking
 - Header lines correct? Date correct format, YYYYMMDD? Created date \leq Modified date?
 - Are commit tables well formed (80 ASCII printable characters per line, Does NWORDS match)?
 - Do numeric tables have correct number of words? Is all data numeric? 5 per line?
 - Does the gird extents and NWORDS math add up
(301 table $NWORDS = 2 + NR + NT * 3(NR * NT)$)

validate.py: continued

under development

- Thermodynamic table consistency checks
 - Do we have any crossing isotherms the Energy, Free energy projection?
 - Grid density checks, is grid dense enough that EOSPAC's pre-inversion equals default inversion to some tol?
 - Are Helmholtz and internal Energy consistent at $T=0$?
 - Tolerance checks for thermodynamic derivations? eg. $P \approx -\left(\frac{\partial A}{\partial V}\right)_T$
 - Are 411 and 412 unique? Are they consistent with each-other? $P_l(T_l) \approx P_m(T_m)$
 - Are melt tables consistent with 301? $P_{301}(\rho_m, T_m) \approx P_m(\rho_m)$, $U_{301}(\rho_m, T_m) \approx U_m(\rho_m)$
 - Are 300 series tables consistent? $301 = 304 + 305 + 306$
 - Does P increase as ρ increases for a single phase?
 - Does P increase as T increases for a single phase?
 - Order of phases in 321 table, Is the last phase listed the high temperature phase (liquid)?

validate.py: continued 2

under development

- Multi-phase Consistency checks
 - Does the flattened total 301 table match phase-specific 301 table (eq 30101) in relevant (ρ , T) space?
 - Do 421XXYY and 422XXYY phase boundary tables fully enclose a single phase?
 - Is the gridding of the phase boundary tables at least as dense as the phase-specific 301 tables?
 - Is last phase-specific identifier in the series the high temperature phase (liquid)?
 - Are thermodynamic values tabulated on the boundary tables consistent with their relative phase-specific 301 table?

validate.py: continued 3

under development

- One can't underestimate the importance of plotting
- `sesplot -F ce90601.bin 90601 301 1 -xrange 6:10 -yrange -1:6 -nolog10=xy`

