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

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

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

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

Table: iso-choric/iso-thermal

EOS_D_PtT	$\rho(P, T)$
EOS_T_DPt	$T(\rho, P)$
EOS_Ut_DPt	$U(\rho, P)$

Table: iso-baric

EOS_T_DUt	$U(\rho, T)$
EOS_Pt_DUt	$A(\rho, T)$

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
```

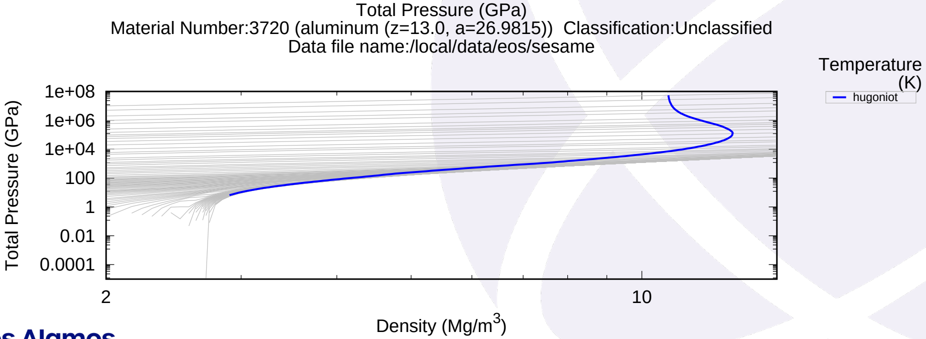
hugoniot.pl

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[m^2/s^2] (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`



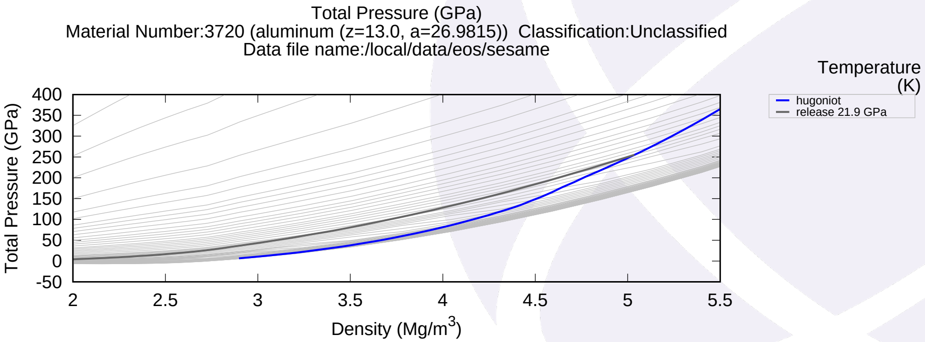
release.pl

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 $(Sound\ Speed)^2[m^2/s^2]$ (i.e., $c^2 = \frac{dP}{d\rho}$ and $c^2 = -\frac{dS}{d\rho} / \frac{dS}{dP}$)

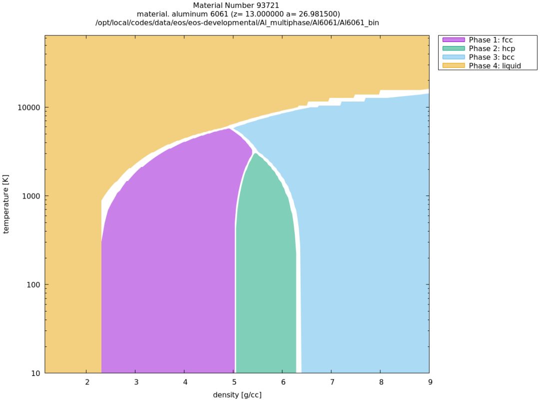
```
release.pl 3720 -r 5.03666115040678 -T 1.15826210755341 7e+04
```



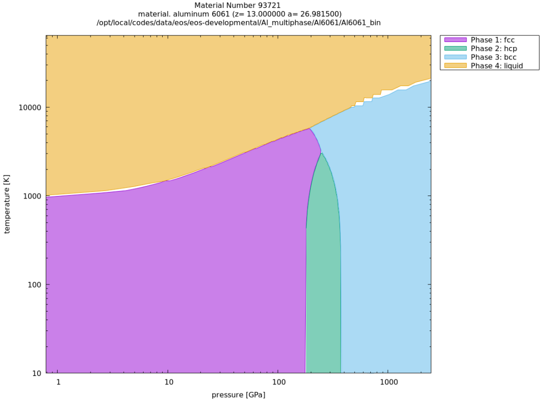
mphase.pl

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

`mphase.pl -F Al6061_bin 93721 -rt`



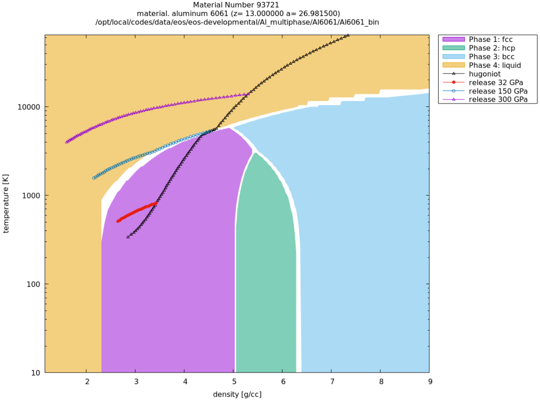
`mphase.pl -F Al6061_bin 93721 -pt`

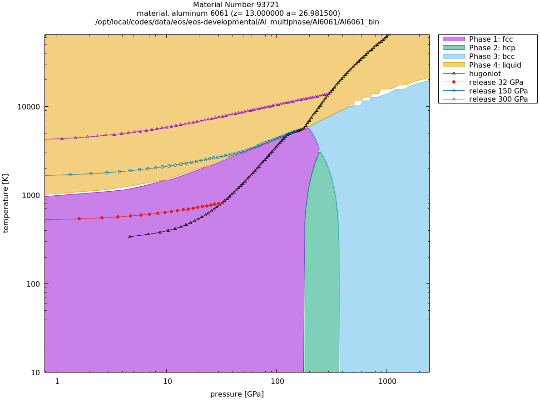


mphase.pl add hugoniot and release trajectories to plots

```
mphase.pl -F Al6061_bin 93721 -pt -rt -keepFiles
hugoniot.pl -F Al6061_bin 93721 -T 298.15 -Umax 10000 -n 200 -O
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=|(9.999999999276213e-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((9.999999999276213e-05 - 0.0001) / 0.0001) = 7.23787620549726e-11
       revert rho0 => 2.700003583974167e+00 g/cc
       P0 => 9.999999999276213e-05 GPa
   2b. pressure at 2.700003583974167e+00 g/cc and 298.15 K => 9.999999999276213e-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 => 9.999999999276213e-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

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, ...`

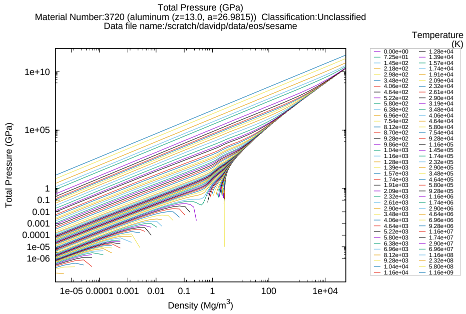
sesplot

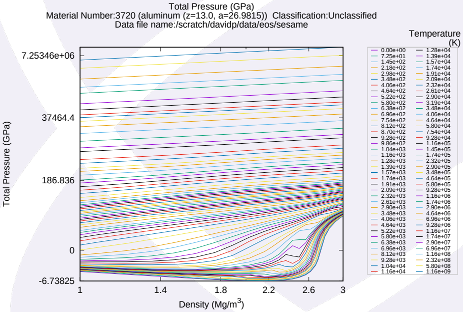
Plot SESAME data using gnuplot and other various SESAME queries/operations

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'
```



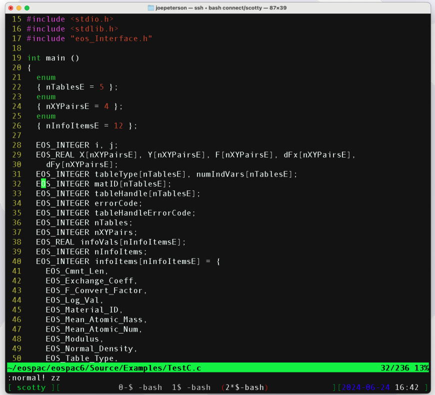


Sopes

- 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)



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)

joepeterson — ssh • bash connect/scotty — 87×39

```
};
EOS_CHAR *tableTypeLabel[nTablesE] = {
  "EOS_Pt_DT",
  "EOS_Dv_T",
  "EOS_Ogb",
  "EOS_Comment",
  "EOS_Info"
};
EOS_CHAR errorMessage[EOS_MaxErrMsgLen];

EOS_INTEGER one = 1;

nTables = nTablesE;
nXYPairs = nXYPairsE;
nInfoItems = nInfoItemsE;

/*
 * EOS_Pt_DT, material 2140 works for Sesame table 301 (record type 1)
 * EOS_Dv_T, material 2140 works for Sesame table 401 (record type 2)
 * EOS_Ogb, material 12140 works for Sesame table 501 (record type 3)
 * EOS_Comment, material 2140 works for Sesame tables 101-199 (record type 4)
 * EOS_Info, material 2140 works for Sesame table 201 (record type 5)
 */
tableType[0] = EOS_Pt_DT;
tableType[1] = EOS_Dv_T;
tableType[2] = EOS_Ogb;
tableType[3] = EOS_Comment;
tableType[4] = EOS_Info;

numIndVars[0] = 2;
numIndVars[1] = 1;
numIndVars[2] = 0;
numIndVars[3] = 0;
numIndVars[4] = 0;

matID[0] = 2140;
~/eospac/eospac6/Source/Examples/TestC.c                                84/236 35%
[ scotty ][          0-$ -bash  1$ -bash  (2*$-bash)          ][2024-06-24 16:45 ]
```

Example of use found in EOSPAC6 in C

Los Alamos NATIONAL LABORATORY

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)

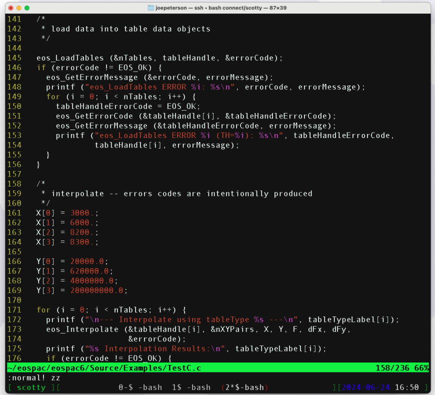
```
joepeterson — ssh • bash connect/scotty — 87×39
  matID[1] = 2140;
  matID[2] = 12140;
  matID[3] = 2140;
  matID[4] = 2140;

  errorCode = EOS_OK;
  for (i = 0; i < nTables; i++) {
    tableHandle[i] = 0;
  }

  /*
   * initialize table data objects
   */

  eos_CreateTables (&nTables, tableType, matID, tableHandle, &errorCode);
  if (errorCode != EOS_OK) {
    for (i = 0; i < nTables; i++) {
      tableHandleErrorCode = EOS_OK;
      eos_GetErrorCode (&tableHandle[i], &tableHandleErrorCode);
      eos_GetErrorMessage (&tableHandleErrorCode, errorMessage);
      printf ("eos_CreateTables ERROR %i: %s\n", tableHandleErrorCode,
              errorMessage);
    }
  }

  /*
   * set some options
   */

  for (i = 0; i < nTables; i++) {
    /* enable smoothing */
    eos_SetOption (&tableHandle[i], &EOS_SMOOTH, EOS_NullPtr, &errorCode);
    if (errorCode != EOS_OK) {
      eos_GetErrorMessage (&errorCode, errorMessage);
      printf ("eos_SetOption ERROR %i: %s\n", errorCode, errorMessage);
    }
~/eospac/eospac6/Source/Examples/TestC.c                                120/236 50%
:normal! zz
[ scotty ][          0-$ -bash  1$ -bash  (2*$-bash)          ][2024-06-24 16:47 ]
```

Example of use found in EOSPAC6 in C

Los Alamos NATIONAL LABORATORY

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)

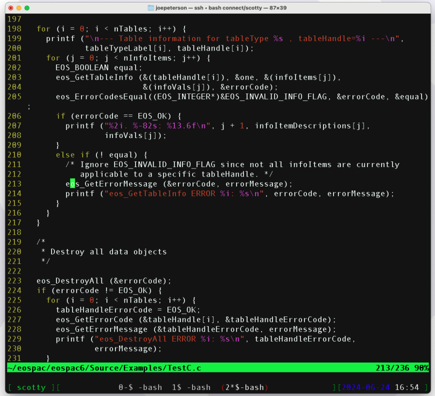



```c
/*
 * load data into table data objects
 */

eos_LoadTables (&nTables, tableHandle, &errorCode);
if (errorCode != EOS_OK) {
  eos_GetErrorMessage (&errorCode, errorMessage);
  printf ("eos_LoadTables ERROR %i: %s\n", errorCode, errorMessage);
  for (i = 0; i < nTables; i++) {
    tableHandleErrorCode = EOS_OK;
    eos_GetErrorCode (&tableHandle[i], &tableHandleErrorCode);
    eos_GetErrorMessage (&tableHandleErrorCode, errorMessage);
    printf ("eos_LoadTables ERROR %i (TH=%i): %s\n", tableHandleErrorCode,
            tableHandle[i], errorMessage);
  }
}

/*
 * interpolate -- errors codes are intentionally produced
 */
X[0] = 3000.;
X[1] = 6000.;
X[2] = 8200.;
X[3] = 8300.;

Y[0] = 20000.0;
Y[1] = 620000.0;
Y[2] = 4000000.0;
Y[3] = 200000000.0;

for (i = 0; i < nTables; i++) {
  printf ("\n--- Interpolate using tableType %s ---\n", tableTypeLabel[i]);
  eos_Interpolate (&tableHandle[i], &nXYPairs, X, Y, F, dFx, dFy,
                   &errorCode);
  printf ("%s Interpolation Results:\n", tableTypeLabel[i]);
  if (errorCode != EOS_OK) {
```

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)

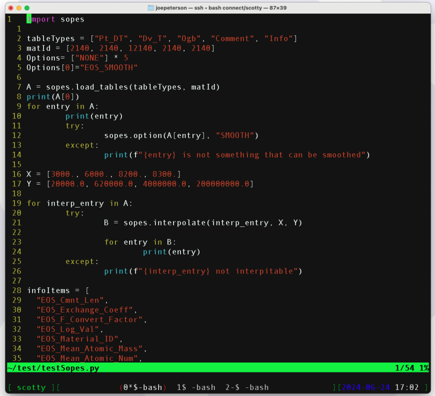


Example of use found in EOSPAC6 in C

Los Alamos NATIONAL LABORATORY

How can Sopes help? (Continued)

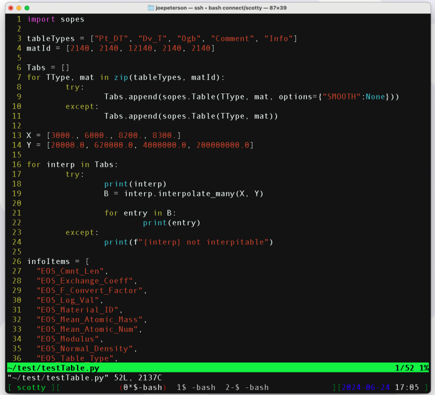
- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)



Equivalent Sopes example

How can Sopes help? (Continued)

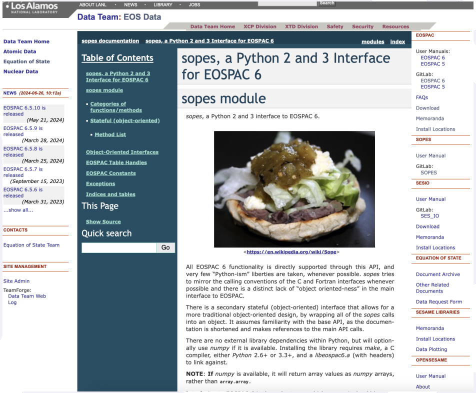
- If you know python, you don't necessarily need to learn a new language to use EOSPAC6
- Arguably, it provides a more intuitive interaction with EOSPAC6
- Possibility of interactively engaging with the Sesame files (Jupyter)



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



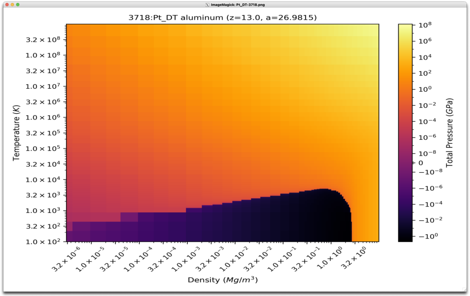
EOSplot

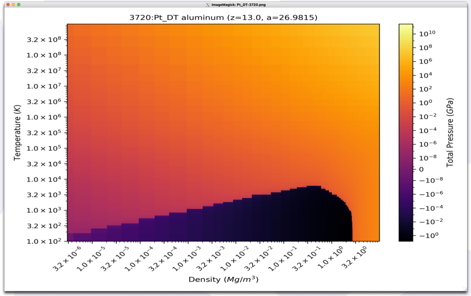
- Reinventing the wheel. Command line tools and scripting tool again. But for python

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()
```



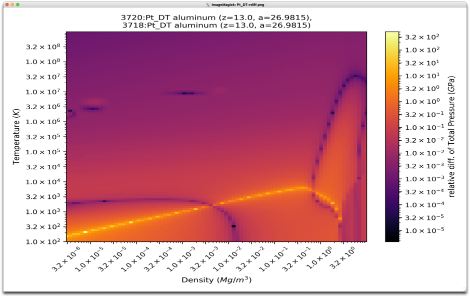


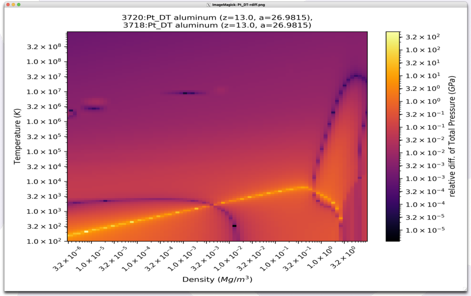
EOSplot Example

Command from terminal:

```
ml python
ml eospac
```

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. $P \approx -(\frac{\partial A}{\partial V})_T$
  - 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 griding 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 consistant with their relative phase-spacific 301 table?

validate.py: continued 3
under development

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