CRETIN

Session 2

Howard Scott

March 12, 2018

LLNL-PRES-747809

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Session topics

- Regions and meshes
  - scalings
  - parameter studies
- Getting data into the code
  - sources, histories, profiles
  - xfiles
- Getting data out of the code
  - edits
  - dumps

There are usually multiple methods to specify values or get results

Regions and meshes are defined by nodes

- The node index *ir* runs from 1 to N
  - N = total # of specifications (e.g. density, temperature, position, …)
  - All material properties for *ir* are co-located
  - 2nd index in edit specifications

- Meshes are defined by a set of nodes and positions
  - 0D: positions not required
  - 1D: N = kmax
  - 2D: N = kmax x lmax
  - 3D: N = kmax x lmax x mmax

- Regions are composed of a set of nodes with the same materials
  - Usually contiguous in (real space) and (logical space)
  - Common materials, e.g. list of elements
  - Common properties, e.g. (initial) temperature

Cretin is (mostly) a node-centered logical mesh code

Regional commands apply to nodes in the current region

- **element** *iz ytot [iso1 iso2]* [**lte**]
- **material** *rho ab zb z2b*
- **rho** *rho*
- **ne** *ne_value*
- **nehot** *value1 value2 value3 …*
- **tehot** *value1 value2 value3 …*
- **bfield** *bfield [bx by bz]*
- **opacity file** *filename*
- **vacuum**

If nodes are not specified for a property or value, the specification applies to the current region

Dimensionality and geometry commands

- Dimensionality is implicit in the region and mesh commands

  region k1 k2 … ← 1-dimensional
  rlin k1 k2 …

  regionkl k1 k2 l1 l2 … ← 2-dimensional
  quad k1 k2 l1 l2 …

  regionklm k1 k2 l1 l2 m1 m2 … ← 3-dimensional
  hex k1 k2 l1 l2 m1 m2 …

- Geometry is specified separately, but must be consistent with dimensionality

  geometry slab / cylinder / sphere ← 1-dimensional
  geometry xy / rz ← 2-dimensional
  geometry xyz ← 3-dimensional
  geometry none ← 1-, 2-, or 3-dimensional

Positions and geometry are required <u>only</u> if some requested physics needs it

1D region / mesh / geometry example

```
atoms iz=1   hydrogenic h
atoms iz=18 dca_18k     ar
```

"iz=18": assign element index
"dca_18k": designate atomic datafile

```
region 1 N Te Te Tr
  element   1 1.
  element 18 1.e-2     !  Argon number fraction
  rho           RHO     !  total mass density
```

Define 1D region + initial temperatures

Specify relative number densities
plus total mass density

```
geometry sphere
```

Define 1D geometry

Mesh options:

Define 1D mesh

```
rlin     1 N  R0 R1                 !  equal spacing between R0 R1
rlog     1 N  R0 R1  RATIO          !  equal ratio spacing
rgeom 1 N  R0 R1  DRMIN  +/-1       !  equal ratio with specified minimum spacing in +/- direction
r        r0  r1  r2  ….             !  specify individual positions
```

1D parameter scan with scaling

```
atoms iz=18 dca_18k     ar

region 1 N  1. 1. 0.
  element 18 1.e20    !  Argon number density

alias  Tmin     100.
alias  Tmax   2000.
alias  N          20

rlin     1 N   Tmin  Tmax

scale  temperature   1 N   0.  1. 0.  1.  1.
                           a0 a1 r1 dr1 b1
```

Set initial temperatures to 1. (or 0.)

Set positions to desired temperatures

Scale temperatures by factor

$$f = a_0 + a_1 \left(\frac{r - r_1}{dr_1}\right)^{b_1}$$

Also available:

Scalings in other variables, more complicated scaling functions

2D (scalekl) and 3D (scaleklm) options

These options are handy for simple scans or spatial profiles

2D parameter scan with scaling

```
regionkl  1 NTE 1 NRHO  10.
  element 1 1.
  rho         1.

geometry xy

x    1.0e0  1.25e0  1.6e0  2.0e0  2.5e0  3.2e0  4.0e0  5.0e0  6.4e0  8.0e0
x    1.0e1  1.25e1  1.6e1  2.0e1  2.5e1  3.2e1  4.0e1  5.0e1  6.4e1  8.0e1
x    1.0e2  1.25e2  1.6e2  2.0e2  2.5e2  3.2e2  4.0e2  5.0e2  6.4e2  8.0e2
x    1.0e3  1.25e3  1.6e3  2.0e3  2.5e3  3.2e3  4.0e3  5.0e3  6.4e3  8.0e3
x    1.0e4

ylog  1 NRHO  RHOMIN RHOMAX

product_mesh

scalekl temperature        1 NTE  1 NRHO  0.  1.  0. 1. 1. 0.  0. 0. 0. 0.
scalekl density            1 NTE  1 NRHO  0.  1.  0. 0. 0. 0.  0. 1. 1. 0.
```

Define x values

Define y values

Define mesh as cross-product of x, y values

This provides an opacity file density / temperature grid

Other methods for specifying conditions

- Sources specify values for a region or set of nodes

  source te **value constant** TE0 *multiplier [ireg/k1 k2 l1 l2 m1 m2]*
  source laser **4x rate history** 1 *multiplier [ireg/k1 k2 l1 l2 m1 m2]*
  source ni *ix iz z [a] [multiplier] [ireg]*

- Histories specify time-dependent values

  history 1 *vmult tmult* ! flat-topped pulse
  tv 0.0 1.
  tv 1.0 1.
  tv 1.0 0.

- Profiles function similar to scalings, but for time rather than space

  tprofile *[ireg] type time1 time2* a_0 a_1 t_1 dt_1 b_1 c_1 a_2 t_2 dt_2 b_2 c_2 …

Including data from other files

```
alias  SYM   al
alias  Z      13
alias  ADF   dca_13

#include ../nlte.gen

   atoms   ADF   SYM

   reg 1

   tstart 0.
   tquit  1.

   switch 29    5          ! use timesteps from history 1
   switch 90    1          ! use xfile for initialization

   source te value history 1 1.0 1 1
   source ti  value history 1 1.0 1 1

xfile AlTD.xf              ← regions + initial conditions

#include AlTD.dat          ← time history
```

AlTD.dat:

```
history 1
  tv   0.00E+00  4.40E+00
  tv   1.00E-15  1.47E+01
  tv   4.45E-15  2.15E+01
  tv   7.90E-15  2.56E+01
  tv   1.14E-14  2.98E+01
  tv   1.48E-14  3.37E+01
  tv   1.82E-14  3.71E+01
  tv   2.17E-14  4.01E+01
  tv   2.51E-14  4.27E+01
  tv   2.86E-14  4.50E+01
  tv   3.20E-14  4.73E+01
  tv   3.55E-14  4.95E+01
  tv   3.89E-14  5.16E+01
  tv   4.24E-14  5.37E+01
  tv   4.58E-14  5.58E+01
  tv   4.93E-14  5.79E+01
  tv   5.27E-14  6.00E+01
  tv   5.62E-14  6.21E+01
  tv   5.96E-14  6.42E+01
  tv   6.31E-14  6.63E+01
  tv   6.65E-14  6.84E+01
  tv   7.00E-14  7.04E+01
  tv   7.34E-14  7.25E+01
  tv   7.69E-14  7.46E+01
  tv   8.03E-14  7.66E+01
  tv   8.38E-14  7.85E+01
  tv   8.72E-14  8.05E+01
  tv   9.07E-14  8.23E+01
  tv   9.41E-14  8.42E+01
  tv   9.76E-14  8.59E+01
         . . .
```

Xfiles are the most general input method

- Xfiles can contain space- and time-dependent values for variables
  r, **v**, T_e, T_i, T_r, ρ, n_e, n_i, J_ν, f_e, **B**, …
- Data in an xfile will overwrite other values, except each n_i requires
  source ni *ix iz z [a] [multiplier] [ireg]*
  to connect the xfile data to an atomic model
- Multiple xfiles may be used in a generator
- Xfile structure:
  Header containing frequency mesh, (Z,A), regions, …
  Problem time:
  [mesh values for variable 1]
  [mesh values for variable 2]
  . . .

Xfile example – parameter scan in (T_e , n_e)

```
logmesh 9 3

problem

te
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.

ti
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.

tr
    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.

ne
  1.e14 1.e14 1.e14 1.e14 1.e14 1.e14 1.e14 1.e14 1.e14
  1.e18 1.e18 1.e18 1.e18 1.e18 1.e18 1.e18 1.e18 1.e18
  1.e22 1.e22 1.e22 1.e22 1.e22 1.e22 1.e22 1.e22 1.e22

x2d
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.
   10.   20.   50.  100.  200.  500. 1000. 2000. 5000.

y2d
  1.e14 1.e14 1.e14 1.e14 1.e14 1.e14 1.e14 1.e14 1.e14
  1.e18 1.e18 1.e18 1.e18 1.e18 1.e18 1.e18 1.e18 1.e18
  1.e22 1.e22 1.e22 1.e22 1.e22 1.e22 1.e22 1.e22 1.e22

done
```

← kmax lmax

← time segment (defaulted to time = 0.)

← Mesh positions set to (T_e, n_e)

This is equivalent to:

```
x  10. 20. 50. 100. 200. …
y  1.e14 1.e18 1.e22
product_mesh
scalekl  te …
scalekl  ti …
scalekl  ne …
```

Yorick scripts build xfiles from rad-hydro output

- Scripts are available for HYDRA, ARES, CALE, … in /usr/apps/cretin/yorick
- Can process restart, opacity, or visualization dumps from HYDRA
- ARES must be instructed to make a Yorick dump
- Example:

  yorick -i hydra2cr.i
  > hydra2cr, "hydr", "hyfile"
  Constructing xfile …/hyfile.xfa
  processing file hydr00200 time = 4.000e-10
  . . .

- Important notes:

  Zone-centered rad-hydro meshes → node-centered Cretin mesh
  User may need to identify axes to script
  User must identify non-standard variable names (ARES)

This method is used for postprocessing rad-hydro runs

Output options - Edits

Edits are composed of
"plot" command + (optional) title
(optional) edit specifications
1 – 3 xvars : name + 4 indices + (optional) multiplier
1 – n yvars : name + 4 indices + (optional) multiplier

Indices correspond to a "dimension"
1st: element / transition
2nd: node (spatial or logical)
3rd: frequency / isosequence
4th: direction / level

- Snapshots are edits which do not have time / cycle as an xvar
  - *xvar* possibilities are listed in the manual
  - snapshots are produced as the run progresses
  - snapshot intervals controlled by switch(30) and param(40)
- Time edits are produced only when the run finishes
  - time edits include results from every timestep
- Edits may be integrated / averaged / maximum / minimum
  over time / space / mass / frequency
- Spectral simulations are specialized (and expensive) edits
  - separate controls [switch(72), param(91)] allow better time-integrations

Output options – edit examples

```
plot "YISOFRAC vs ISO"
  xvar  iso          1
  yvar  yisofrac    1 1:-1

plot "TE vs TIME"
  xvar  time
  yvar  tev         0 1:N:DN

plot "ZBAR vs R"
  xvar  r
  yvar  zbar

plot "ZBAR(1) vs TE"
  xvar  x2d          0  1
  yvar  zbar         1   1:LMAX

plot "TIME-INTEGRATED SPECTRUM"
time-integrated snapshot
  xvar  sp_energy
  yvar  jsparea    0 -1

plot "INTEGRATED ENERGY vs TIME"
time-integrated
sp-integrated
  xvar  time
  yvar  jsparea    0 -1
```

Charge state distribution snapshot for all nodes
first:last syntax for index
-1 in 2nd index → last node

Time plot of temperature
first:last:stride syntax for index

Snapshot on 2D mesh
"xvar r" expands to "xvar x2d" + "xvar y2d"

Snapshot along K-line on 2D mesh
"xvar x2d" requires L index
yvar requires compatible L index

Time-integrated snapshot
appears at each edit interval
w/o "snapshot" would appear only at conclusion of run

Time- and energy-integrated
"xvar time" → saved for each cycle

Homework

- Calculate the average charge state <Z> and radiative power loss for Kr as a function of temperature T_e for
  - fixed electron density: n_e = 10^{20} cm^{-3} or 10^{14} cm^{-3}
  - radiation field: $J_\nu = 0$

  Notes:
  - temperature range sufficient to cover <Z> from ~neutral to K-shell
  - radiative power loss = frequency-integrated emissivity
    edit variable: cemistot
  - check for sufficient extent / resolution in temperature / frequency