Writing Benchmark Models with a Computer Program

Jeffrey A. Favorite
Radiation Transport Applications Group (XTD-RTA)
Los Alamos National Laboratory

Nuclear Criticality Safety Program Technical Program Review
Riverhead, NY
February 20–22, 2024

Introduction

- The goal is to automate as much as possible.

- The program is a *tool* for a specific evaluation.

- I am not presenting or promoting a generic computer program for all benchmark evaluations!

- I've been doing this since my first evaluation in 2012.

- There was a recent talk along similar lines:

A. Hauck, K. Stolte, et al., "Generating Models of the Flattop Critical Assembly for Benchmark Experiments with Python," *Transactions of the American Nuclear Society*, **127**, 678–681 (2022);
ANS Winter Meeting, Phoenix, AZ.

- The purposes of this talk:
    1. Encourage efficiency!
    2. Find out how you do this.

- Caveat: No tool or algorithm is foolproof or solves all problems.

Making a Model—Part I

- Set dimensions
- Set masses

- Calculate volumes
- Calculate part mass densities
- Calculate material and cell atom densities

If this only had to be done once, automation would not be necessary.

```
call set_dims_dwg(maxparts,maxdims,maxangs,dims,angs)
if(i_Mos_dim == 1)then
  call set_dims_Mos(maxparts,maxdims,maxangs,dims,angs)
end if
if(i_meas_dim == 1)then
  call set_dims_meas(maxparts,maxdims,maxangs,dims,angs)
end if
if(i_ta18_dim == 1)then
  call set_dims_ta18(maxparts,maxdims,maxangs,dims,angs)
end if
dims(1:maxparts,1:maxdims)=dims(1:maxparts,1:maxdims)*in2cm
call calc_volume(maxparts,maxdims,maxangs,i_meas_dim,pname,dims,angs,volume)
call set_mass(maxparts,i_ta18_dim,pname,mass)
call calc_dens(maxparts,pname,volume,mass,dens)
⋮
call calc_atdens(iuo,1,denmat,den_calc)
```

```
! Ring 101 (128Y1719423)
  dims(R101,1)=7.000d0 ! outer diameter
  dims(R101,2)=3.500d0 ! inner diameter
  dims(R101,3)=1.125d0 ! height
  dims(R101,4)=5.000d0 ! recess (top) diamete
  dims(R101,5)=0.125d0 ! recess (top) depth
  dims(R101,6)=0.875d0 ! hole diameter
  dims(R101,7)=2.625d0 ! hole location
  dims(R101,8)=0.875d0 ! slot width
  dims(R101,9)=0.156d0 ! slot depth, from OD
  dims(R101,10)=0.875d0 ! groove (for pads) c
  dims(R101,11)=0.100d0 ! groove (for pads) c
  dims(R101,12)=2.437d0 ! location of groove
  angs(R101,1)=64.d0 ! angle subtended by gro
  angs(R101,2)=28.d0 ! angle subtended by nei
```

```
! Measurement Worksheets (notes for Jul-11-22)
mass(R101)=8381.7d0
mass(R102)=8131.4d0
mass(R103)=8022.6d0
mass(R104)=6880.7d0
mass(R105)=8165.5d0
mass(R106)=9282.4d0
```

- Write the cell densities with the cells.

Making a Model—Part II

- Set surface locations and dimensions.
- Write surfaces and their parameters.

- For this one I made a list of parts in the stack:

```
c
       list_refl(1:max_refl_cells)=0
       list_core(1:max_core_cells)=0
       list_out(1:max_out_cells)=0
c parts in reflector and Planet
       n=0
       n=n+1 ; list_refl(n)=Tilt_Weigh_Table
       n=n+1 ; list_refl(n)=Platen_Adapter_Plate
       n=n+1 ; list_refl(n)=Platen_Heat_Sink
       n=n+1 ; list_refl(n)=Bottom_Reflector
       n=n+1 ; list_refl(n)=SR_Base
       select case (ic)
         case (1)
           n=n+1 ; list_refl(n)=SR_Middle_030_1
           n=n+1 ; list_refl(n)=SR_Middle_030_2
           n=n+1 ; list_refl(n)=SR_Middle_030_3
           n=n+1 ; list_refl(n)=SR_Flat_Upper_030
           n=n+1 ; list_refl(n)=Core_Support_Plate
           n=n+1 ; list_refl(n)=SR_Flat_Lower_030
           n=n+1 ; list_refl(n)=SR_Middle_030_4
           n=n+1 ; list_refl(n)=SR_Middle_030_5
           n=n+1 ; list_refl(n)=SR_Trans_030
           n=n+1 ; list_refl(n)=TR_1000_1
           n=n+1 ; list_refl(n)=TR_1000_2
           n=n+1 ; list_refl(n)=TR_1000_3
           n=n+1 ; list_refl(n)=RTD_slot_r
```

```
c block 2, surfaces
       else if(nb.eq.2)then
         write(iuo,'("")') ! blank
         nsurf=0
         zzz1=-zbase
c reflector surfaces
         do n=1,nrefl
c          write(*,'(2i6)')n,list_refl(n)
           if(list_refl(n).lt.0)then
             cycle
           else if(list_refl(n).eq.Tilt_Weigh_Table)then
             write(iuo,'("c ",a)')trim(pname(list_refl(n)))
             nsurf=nsurf+1
             zzz2=zzz1+dimsp(list_refl(n),3)
             xxx=dimsp(list_refl(n),1)/2.d0
             yyy=dimsp(list_refl(n),2)/2.d0
             call write_surf(nsurf,iuo)
             write(iuo,'("rpp",5x,0p6f11.6)')
     1        -xxx,xxx,-yyy,yyy,zzz1,zzz2
             if(ic.eq.1)then
               core_stack(4,1)=zzz2
               refl_stack(1,1)=zzz2
             else if(ic.eq.2)then
               core_stack(10,1)=zzz2
               refl_stack(3,1)=zzz2
             end if
             call write3(1,iu3,pname(list_refl(n)),zzz1,zzz2)
             xxx=dimsp(list_refl(n),11)/2.d0
             yyy=dimsp(list_refl(n),12)/2.d0
             rrr=dimsp(list_refl(n),13)/2.d0
             do i=1,4
               nsurf=nsurf+1
               call write_surf(nsurf,iuo)
               write(iuo,'("c/z",5x,a,0pf9.6,a,f9.6,f11.6)')
     1          c4(1,i),xxx,c4(2,i),yyy,rrr
             end do ! i
             zzz1=zzz2
```

Auxiliary Files

- Need a base or "skeleton" input file.
  + It is easier for a code to alter something than to create from scratch.
  + Probably the base input should have cells defined, but the densities will be written each time.
  + Different surface arrangements can be accommodated with if/then structures.

- Material files Ex. SS 303: gets converted to:

```
8.
C        0.075
Si       1.00
P        0.10
S        0.30
Cr       18.00
Mn       1.00
Fe       70.225
Ni       9.00
Mo       0.30
```

```
c SS 303 (8.0 g/cc)
c begin at.dens., wt.dens.  8.829956206E-02     8.000000000000
c round at.dens., wt.dens.  8.829894685E-02     7.999944324017
m1      nlib=00c
         6012   2.9752E-04        6013   3.3091E-06
        14028   1.5821E-03       14029   8.0107E-05
        14030   5.3176E-05
        15031   1.5554E-04
        16032   4.2830E-04       16033   3.3806E-06
        16034   1.8977E-05       16036   9.0150E-08
        24050   7.2465E-04       24052   1.3974E-02
        24053   1.5844E-03       24054   3.9443E-04
        25055   8.7693E-04
        26054   3.5744E-03       26056   5.5567E-02
        26057   1.2723E-03       26058   1.6963E-04
        28058   5.0438E-03       28060   1.9283E-03
        28061   8.3485E-05       28062   2.6523E-04
        28064   6.7232E-05
        42092   2.2358E-05       42094   1.3936E-05
        42095   2.3985E-05       42096   2.5130E-05
        42097   1.4388E-05       42098   3.6354E-05
        42100   1.4509E-05
```

(• Materials can be defined within the code.)

My System is Based on Parts and Materials

```
! part indices
   IISP=1
   UISP=IISP+1
   SafetyBlock=UISP+1
   R101=SafetyBlock+1
   R102=R101+1
   R103=R102+1
   R104=R103+1
   R105=R104+1
   R106=R105+1
   CR1=R106+1
   CR2=CR1+1
   BR=CR2+1
   Spindle=BR+1
   AlignPin=Spindle+1
   SafetyBase=AlignPin+1
   Pads=SafetyBase+1
   BRing=Pads+1
   SubCvrPl=BRing+1
   MountPl=SubCvrPl+1
   Leg=MountPl+1
   Clamp=Leg+1
   BellyBand=Clamp+1
   NutsBolts=BellyBand+1
   CrCvr=NutsBolts+1
   CntSh=CrCvr+1
   MntPl2=CntSh+1
   Room=MntPl2+1
   Air=Room+1
```

```
pname(IISP)="Inter. Inner Subassembly Plate"
pname(UISP)="Upper Inner Subassembly Plate"
pname(SafetyBlock)="Safety Block"
pname(R101)="Ring 101"
pname(R102)="Ring 102"
pname(R103)="Ring 103"
pname(R104)="Ring 104"
pname(R105)="Ring 105"
pname(R106)="Ring 106"
pname(CR1)="Control Rod 1"
pname(CR2)="Control Rod 2"
pname(BR)="Burst Rod"
pname(Spindle)="Spindle"
pname(AlignPin)="Alignment Pin"
pname(SafetyBase)="Safety Block Base"
pname(Pads)="Support Pads"
pname(BRing)="Bearing Ring"
pname(SubCvrPl)="Subassembly Cover Plate"
pname(MountPl)="Mounting Plate"
pname(Leg)="Clamp Support (Leg)"
pname(Clamp)="Clamp"
pname(BellyBand)="Belly Band"
pname(NutsBolts)="Nuts and Bolts"
pname(CrCvr)="Core Cover"
pname(CntSh)="Contamination Shield"
pname(MntPl2)="Mounting Plate 2"
pname(Room)="Room"
pname(Air)="Air"
```

```
material indices
  SS303=1
  SAE4340=SS303+1
  Vasco=SAE4340+1
  Al6061=Vasco+1
  Lex=Al6061+1
  AISI1019=Lex+1
  mR101=AISI1019+1
  mR102=mR101+1
  mR103=mR102+1
  mR104=mR103+1
  mR105=mR104+1
  mR106=mR105+1
  mSafetyBlock=mR106+1
  mIISP=mSafetyBlock+1
  mUISP=mIISP+1
  mCR1=mUISP+1
  mCR2=mCR1+1
  mBR=mCR2+1
  mRoom=98
  mAir=99
```

```
!
mat(0:maxparts)=-1
mat(0)=0
mat(IISP)=mIISP
mat(UISP)=mUISP
mat(SafetyBlock)=mSafetyBlock
mat(R101)=mR101
mat(R102)=mR102
mat(R103)=mR103
mat(R104)=mR104
mat(R105)=mR105
mat(R106)=mR106
mat(CR1)=mCR1
mat(CR2)=mCR2
mat(BR)=mBR
mat(Spindle)=SS303
mat(AlignPin)=SS303
mat(SafetyBase)=SS303
mat(Pads)=SAE4340
mat(BRing)=SAE4340
mat(SubCvrPl)=SAE4340
mat(MountPl)=Al6061
mat(Leg)=SS303
mat(Clamp)=Vasco
mat(BellyBand)=SS303
mat(NutsBolts)=SS303
mat(CrCvr)=Al6061
mat(CntSh)=Lex
mat(MntPl2)=SS303 ! check this
mat(Room)=mRoom
mat(Air)=mAir
denmat(SS303)=8.d0
denmat(SAE4340)=7.85d0
denmat(Vasco)=8.d0
denmat(Al6061)=2.70d0 ! m_Al6061T6 i
if(i_Mos_den == 0)denmat(Lex)=1.20d0
denmat(AISI1019)=7.87d0
denmat(mAir)=1.052d-3
```

Geometry Perturbations

- This is where this system really pays off!

- I have used a separate code for each type of part (e.g., fuel, reflector, structure) or evaluation subsection.

```
 parameter(nparts=4)

 data parts/
1 "Plate_","StudT_","StudB_","Spacer_"/
 data ndims/
1  2,       2,       2,       3       /
```

Will go on to calculate volumes and densities using perturbed dimensions

- Volumes are not always analytic.

```
        do ifl=1,nparts
          do id=1,ndims(ifl)
            do ip1=1,nperts
              nseed=(ifl+22)*100+id*10+ip1
              write(sss,'(i4.4)')nseed
c reset everything
              dimsp(1:maxparts,1:maxdims)=dims(1:maxparts,1:maxdims)
              volume_p(1:maxparts)=volume(1:maxparts)
              den_p(1:maxparts)=den(1:maxparts)
              devn=0.d0
c
c    1 "Plate_","StudT_","StudB_","Spacer_"/
c    1  2,       2,       2,       3       /
c make perturbations. dimensions are in cm here.
              select case(ifl)
              case (1)
              if(id.eq.1)then
                dimsc="lw_"
                devn=0.03d0 ! inches
                dr=rp(ip1)*devn*in2cm
                dimsp(Compression_Plate,1)=dims(Compression_Plate,1)+dr
                dimsp(Compression_Plate,2)=dims(Compression_Plate,2)+dr
              else if(id.eq.2)then
                dimsc="th_"
                devn=0.030d0 ! inches
                dr=rp(ip1)*devn*in2cm
                dimsp(Compression_Plate,3)=dims(Compression_Plate,3)+dr
              end if
c
              case(2)
              if(id.eq.1)then
                dimsc="ht_"
                devn=0.03d0 ! inches
                dr=rp(ip1)*devn*in2cm
                dimsp(Compression_Stud_Top,3)=
     1           dims(Compression_Stud_Top,3)+dr
              else if(id.eq.2)then
                dimsc="dd_"
                devn=0.03d0 ! inches
                dr=rp(ip1)*devn*in2cm
                dimsp(Compression_Stud_Top,4)=
     1           dims(Compression_Stud_Top,4)+dr
              end if
```

```
      dims(Mod_Inn_600_01:Mod_Inn_600_18,3:4)=
     1 dims(Mod_Inn_600_01:Mod_Inn_600_18,3:4)
     2 *(1.d0+alphat(CPVC_600)*devn) ! z, outer diameter
c
      dims(ZPPR_Sleeve,1:4)=dims(ZPPR_Sleeve,1:4)
     2 *(1.d0+alphat(Cladding)*devn) ! x, y, z, thickness
      dims(ZPPR_Plug,1)=dims(ZPPR_Plug,1)
     2 *(1.d0+alphat(SS304L)*devn) ! x
      dims(ZPPR_Spring,1:3)=dims(ZPPR_Spring,1:3)
     2 *(1.d0+alphat(Carbon_Steel)*devn) ! x, y, z
c warning, using one value for all.
      dims(ZPPR_PANN_A:ZPPR_PANN_S,1:3)=
     1 dims(ZPPR_PANN_A:ZPPR_PANN_S,1:3)
     2 *(1.d0+alphat(PANN_A)*devn) ! x, y, z
c
```

Thermal Expansion Calculations Are Easy

- The change Δx in dimension x due to a change in temperature ΔT is $\Delta x = x' - x_0 = \Delta T\,\alpha_T\,x_0$
  - Or $x' = x_0\,(1 + \alpha_T\,\Delta T)$

```
      dims(Mod_Inn_600_01:Mod_Inn_600_18,3:4)=
     1 dims(Mod_Inn_600_01:Mod_Inn_600_18,3:4)
     2 *(1.d0+alphat(CPVC_600)*devn) ! z, outer diameter
c
      dims(ZPPR_Sleeve,1:4)=dims(ZPPR_Sleeve,1:4)
     2 *(1.d0+alphat(Cladding)*devn) ! x, y, z, thickness
      dims(ZPPR_Plug,1)=dims(ZPPR_Plug,1)
     2 *(1.d0+alphat(SS304L)*devn) ! x
      dims(ZPPR_Spring,1:3)=dims(ZPPR_Spring,1:3)
     2 *(1.d0+alphat(Carbon_Steel)*devn) ! x, y, z
c warning, using one value for all.
      dims(ZPPR_PANN_A:ZPPR_PANN_S,1:3)=
     1 dims(ZPPR_PANN_A:ZPPR_PANN_S,1:3)
     2 *(1.d0+alphat(PANN_A)*devn) ! x, y, z
c
```

x' (the left-hand side `dims(ZPPR_PANN_A:ZPPR_PANN_S,1:3)=`), x_0 (`dims(ZPPR_PANN_A:ZPPR_PANN_S,1:3)`), α_T (`alphat`), ΔT (`devn`)

File Naming Conventions

- Perturbed input file name:

```
              fn=trim(ofile(ic))//trim(ext)
 1             //"_"//trim(parts(ifl))//trim(dimsc)//pt(ip1)
```

- `ofile` is for the case

```
  data ofile/"c1", "c2", "c3"/
```

- `ext` is an “extension” for a version number
- `parts` is the part names

```
  data parts/
 1 "Plate_","StudT_","StudB_","Spacer_"/
```

- `dimsc` is the perturbed dimension
- `pt` is m for negative perturbations and p for positive perturbations
- The code also writes the job submission scripts.

```
ro.compile% ls *.o
     c1_15j_Plate_lw_m.o        c3_15j_Plate_lw_m.o
     c1_15j_Plate_lw_p.o        c3_15j_Plate_lw_p.o
     c1_15j_Plate_th_m.o        c3_15j_Plate_th_m.o
     c1_15j_Plate_th_p.o        c3_15j_Plate_th_p.o
     c1_15j_Spacer_ht_m.o       c3_15j_Spacer_ht_m.o
     c1_15j_Spacer_ht_p.o       c3_15j_Spacer_ht_p.o
     c1_15j_Spacer_id_m.o       c3_15j_Spacer_id_m.o
     c1_15j_Spacer_id_p.o       c3_15j_Spacer_id_p.o
     c1_15j_Spacer_od_m.o       c3_15j_Spacer_od_m.o
     c1_15j_Spacer_od_p.o       c3_15j_Spacer_od_p.o
     c1_15j_StudB_dd_m.o        c3_15j_StudB_dd_m.o
     c1_15j_StudB_dd_p.o        c3_15j_StudB_dd_p.o
     c1_15j_StudB_ht_m.o        c3_15j_StudB_ht_m.o
     c1_15j_StudB_ht_p.o        c3_15j_StudB_ht_p.o
     c1_15j_StudT_dd_m.o        c3_15j_StudT_dd_m.o
     c1_15j_StudT_dd_p.o        c3_15j_StudT_dd_p.o
     c1_15j_StudT_ht_m.o        c3_15j_StudT_ht_m.o
     c1_15j_StudT_ht_p.o        c3_15j_StudT_ht_p.o
     c2_15j_Plate_lw_m.o        c3_15k_Plate_lw_m.o
     c2_15j_Plate_lw_p.o        c3_15k_Plate_lw_p.o
     c2_15j_Plate_th_m.o        c3_15k_Plate_th_m.o
     c2_15j_Plate_th_p.o        c3_15k_Plate_th_p.o
     c2_15j_Spacer_ht_m.o       c3_15k_Spacer_ht_m.o
     c2_15j_Spacer_ht_p.o       c3_15k_Spacer_ht_p.o
     c2_15j_Spacer_id_m.o       c3_15k_Spacer_id_m.o
     c2_15j_Spacer_id_p.o       c3_15k_Spacer_id_p.o
     c2_15j_Spacer_od_m.o       c3_15k_Spacer_od_m.o
     c2_15j_Spacer_od_p.o       c3_15k_Spacer_od_p.o
     c2_15j_StudB_dd_m.o        c3_15k_StudB_dd_m.o
     c2_15j_StudB_dd_p.o        c3_15k_StudB_dd_p.o
     c2_15j_StudB_ht_m.o        c3_15k_StudB_ht_m.o
     c2_15j_StudB_ht_p.o        c3_15k_StudB_ht_p.o
     c2_15j_StudT_dd_m.o        c3_15k_StudT_dd_m.o
     c2_15j_StudT_dd_p.o        c3_15k_StudT_dd_p.o
     c2_15j_StudT_ht_m.o        c3_15k_StudT_ht_m.o
     c2_15j_StudT_ht_p.o        c3_15k_StudT_ht_p.o
```

Going Further…

- Simplifying the model takes a long time, and I end up doing it more than once.
- So I use a program for that too:

```
c
      exclude(1:99,1:maxnucl)=0 ! default is keep all nuclides
      avgmatdens(1:ncases,1:99)=-1.d0
      dens_BR=1.2622d-01 ! Bottom Reflector unpert. density
      dens_SRB=1.2632d-01 ! Side Reflector, Base unpert. density
      dens_TR(1:ncases)=0.d0
c
      nsimp=22
c 1. Remove air.
c 2. Remove fill gas.
c 3. Center fuel, moderators, and absorbers.
c 4. Remove nylon screws and holes; preserve density.
c 5. Remove thermocouple probes, fill slots; preserve density.
c 6. Restore tray mass; combine trays and frames; use average density.
      if(nsimp.ge.6)then
        avgmatdens(1,23)=5.9692d-02
        avgmatdens(1,24)=avgmatdens(1,23)
        avgmatdens(2,23)=5.9711d-02
        avgmatdens(2,25)=avgmatdens(2,23)
        avgmatdens(3,23)=5.9788d-02
        avgmatdens(3,24)=avgmatdens(3,23)
      end if
c 7. Remove protruding part of RTD plugs; preserve density.
c 8. Remove slip nuts and compression studs, top, above compression spacer; preserve density.
c 9. Remove holes from tilt-weigh table and top plate; preserve density.
c 10. Remove knobs from compression studs, fill holes in platen adapter plate and core support plate; preserve density.
c 11. Eliminate hex nuts and washers.
c 12. Square rounded corners in upper adapter plate and top plate; preserve density.
c 13. Square rounded corners platen heat sink, bottom reflector, and side reflector, base; preserve density.
c 14. Restore mass of platen heat sink, bottom reflector, and side reflector, base.
      if(nsimp.ge.14)then
        avgmatdens(1:3,26)=5.9894d-02
        dens_BR=1.2619d-01
        dens_SRB=1.2630d-01
      end if
```

Conclusions

- I recommend that we all stop cutting and pasting from spreadsheets.
  + Automate the things you do repeatedly.
- I believe that because of my system, the model is less likely to have errors than Section 3.
  + Our institutional bias against the model in favor of Section 3 is outdated.
  + For my evaluations, I give you permission to use the model instead of Section 3.
- How do *you* automate?
  + Talk to me; email me: favorite@lanl.gov
- Acknowledgment:

This work was supported by the DOE Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.