Debakey pump - objective

Invert the SSMUM layer
(to obtain volume elements with the same sign for Jacobian)
linf: ne 20000, nn 10200
lxyz
lien
add1 (inf): ne 20000, nn 9800
add2 (xyz):
add3 (ien):
element 1 (no inversion in makelayer) (elag=1)

layer1inner (imp)

layer1outer (str)

e1

e2

e1

layer1inner (ienl1(3,ne))

ien(1,ne) = layer1inner (ienl1(3,ne))

1 = face

rng(1,ne) = irngisnew = 14 (equals irngis=10)
element 1 (after inversion in imp makelayer) (elag=1)

layer1inner (imp)

layer1outer (str)

eqn(1,ne) = layer1inner(ien1(3,ne))

1 = face

mg(1,ne) = irngisnew =14 (equals irngis=10)
element 2 (no inversion in makelayer) (elag = 1)

layer1inner (imp)

layer1outer (str)

irng(1,ne) = irngsnew = 13 (equals irngs=9)

if ie > nel-2*nr: rng(3,ne) = irng2 = 8 (housing)
element 2 (inversion in imp makelayer) (elag = 1)

layer1inner (imp)

layer1outer (str)

irng(1,ne) = irngsnew = 13 (equals irngs=9)

if ie>nel-2*nr: rng(3,ne) = irng2 = 8 (housing)
layer1inner (imp)

(element 2)

(elag = 1)

layer1outer (str)
element 3

elag = 1

layer1inner (imp)
layer1outer (str)
Debakey pump - comolayer (element generation)

ne = 0

do ie=1,nel

if (elag1(ie).eq.1) then

    ne = ne + 1
    ien(1,ne) = layer1outer(ienl1(1,ie)) + nni
    ien(2,ne) = layer1outer(ienl1(2,ie)) + nni
    ien(3,ne) = layer1outer(ienl1(3,ie)) + nni
    ien(4,ne) = layer1inner(ienl1(2,ie))

    if (ie.gt.nel-2*nr) rng(3,ne) = irng2
    rng(1,ne) = irngsnew

    ne = ne + 1

...
irng(1,ne) = irngsnew = 13 (equals irng=9)
if ie>nel-2*nr: rng(3,ne) = irng2 = 8 (housing)
Debakey pump - comolayer (invert)

invert = .true.
if (invert) then
doi=1,ne
in1 = ien(1,ie)
in2 = ien(2,ie)
in3 = ien(3,ie)
ien(1,ie) = in3
ien(2,ie) = in2
ien(3,ie) = in1
...

element 2  
(elag = 1)

irng(1,ne) = irngsnew = 13 (equals irngs=9)
if ie>nel-2*nr: rng(3,ne) = irng2 = 8 (housing)

iss(1,ne) = ien(2,ne)
iss(4,ne) = -1

if invert:
1 ⇔ 3
rng(2,ne) = rng(3,ne)
no change in iss

1 = face  1 = iss
Debakey pump - comolayer (invert)

\[
\text{invtmp2} = \text{rng}(2, \text{ie}) \\
\text{invtmp3} = \text{rng}(3, \text{ie}) \\
\text{rng}(2, \text{ie}) = \text{invtmp3} \quad \text{rng}(3, \text{ie}) = \text{invtmp2}
\]

(iss(3,\text{ie}).gt.0) \text{iss}(3, \text{ie}) = \text{in3}

if (iss(1,\text{ie}).gt.0) \text{iss}(1, \text{ie}) = \text{in2}
end do
end if

invert = .false.
For correct inversion of elements:

   do not invert elements between str and imp
   but invert elements between impeller and diffuser
element 2
(elag = 1)

layer1inner (imp)
layer1outer (str)

irng(1,ne) = irngsnew = 13 (equals irngs=9)
if ie>nel-2*nr: rng(3,ne) = irng2 = 8 (housing)

iss(1,ne) = ien(2,ne)
iss(4,ne) = -1

if invert:
  1 ⇔ 3
  rng(2,ne) = rng(3,ne)
  no change in iss

ien(1,ne) = layer1outer(ienl1(1,ne))+nni

1 = face  1 = iss
element 1
(\text{elag}=1)

layer1inner (imp)

layer1outer (str)

\text{mg}(1,\text{ne}) = \text{irngisnew} = 14 \text{ (equals irngis}=10)

no change in rmg
no change in iss

if invert:

\text{1} \iff \text{3}

\text{1} = \text{face} \quad \text{1} = \text{iss}
Scripts for MicroMed Debakey Pump

Scripting for Debakey:

glyph script Parameterize_Debakey
script1_makefolder
meshview
script2_makelayer
script3_DMG script3_prepare_for_gridgen
glyph script ME2STL
glyph script fmtetra_XXX
script4_comolayer script4_altern_comolayer
script5_mprm
script6_export

scripts from jubl:
script7_makefolder

to come: scripts_for_xns (runs on jubl, analysis, storage)
Scripts for MicroMed Debakey Pump

some abbreviations used in the following:

STR = straightener    CON = connector
IMP = impeller       DOM = domain
DIF = diffuser
XXX = any of the above
Scripts for MicroMed DeBakey Pump

Gridgen base model
(Str_2.1new.dba/gg, Dif.dba/gg, and Imp.dba/gg)
database constrained CONS and DOMS
included some improvements (e.g. unstructured DOMS
based on triangulation of structured DOMS)
Scripts for MicroMed Debakey Pump

Gridgen sizeable model (Script: Parameterize_Debakey)

objective:
generate meshes (creates coarse, medium, and fine meshes)

realization:
TCL script with a total of 4800 lines, thereof
• 1800 lines for DIF,
• 1200 lines for IMP,
• 1600 lines for STR

variables for computation of grid points on CONS
Scripts for MicroMed Debakey Pump

features:
documentation / easy adaptability:

#introduce scaling factor
set DIF_SCALING_FACTOR $DifScale

#correcting_factor of 0.95 to make elements smaller
set NEWDIF0_05 [expr {($DIF_SCALING_FACTOR)*0.05*0.95}]

#define No of GPs on CONS

gg::conDim [list 
    $_CN(50) \ 
    $_CN(52) \ 
] -spacing $NEWDIF0_05

modification of script after mesh analysis (meshview, DMG, Gridgen ...)
Scripts for MicroMed De Bakey Pump

features:

fast creation

### give CN(62) same distribution as CN(58)
set _CN(62) [lindex [gg::conGetAll] 61]

gg::conBeginSpacing $_CN(62) -sub 1 0

gg::conEndSpacing $_CN(62) -sub 1 0

set _CN(58) [lindex [gg::conGetAll] 57]

gg::conDim $_CN(62) -copy $_CN(58)

# make structured-unstructured mesh

...
Scripts for MicroMed Debakey Pump

Gridgen sizeable model (Script: Parameterize_Debakey)
Usage: glyph/execute PARAMETERIZED_DEBAKEY.glf

GUI interface written in TK language

independant scaling factors for STR, DIF, IMP
Scripts for MicroMed Debakey Pump

script: Parameterize_Debakey

modifies sequentially DIF, IMP, and STR
(except for their scaling factor meshes are identical)

export files for all 3 components:
• XXX20070504_0.85.gg (name includes creation date and scaling factor)
• XXXmod.grd (exported gridpoints in UCD format)

required time: 7 minutes for a 6 million element mesh
Shell scripts

**script1_makefolder:**

Usage: from Parameterized_Deakey.Y.YY: script1_makefolder Y.YY

- makes a project folder with sub-folders.
- moves/copies all relevant data to folders
- creates XXXmod.mod.e (format required by meshview)
Scripts for MicroMed DeBakey Pump

Meshview
(only step which is not yet automated)
- invert orientation of triangles
- define boundary groups
Scripts for MicroMed Debakey Pump

script2_makelayer

Usage: (from Parameterized_Debakey.X.XX): script2_makelayer

• compiles and runs makelayer for STR, IMP, and DIF
• makelayer makes closing layers
Scripts for MicroMed De Bakey Pump

script3_DMG

Usage: on hydra from folder Parameterized_DeBakey.Y.YY: script3_DMG Y.YY

- launches 3 jobs on hydra
- filling closed volumes of STR, IMP, and DIF with tetrahedra
- saves diagnostic about mesh quality
  (diagnostic points to worst element ==> mesh improvement)

```
NUMBER OF REALLY BAD ELEMENTS IS 0!!!
Element quality diagnostics:
  Volume max 2.115426e-04 : min 6.407650e-09 : avg 1.751653e-05
  AR max 1.006531e+01 : min 1.000680e+00 : avg 1.429853e+00
  Min edge length is 3.925548e-03 : -1.144197e-02 2.983156e-02 1.408846e+00

Worst element centered about 2.357936e-01 9.998700e-03 1.346699e+00
Remember, the best-worst aspect ratio possible with this
surface grid is around 3.71 times optimal.

DIAGNOSTICS:
Intype 1 called 558140 times.
Intype 2 called 15 times.
Intype 3 called 0 times.
Case 1 called 5533772 times.
Case 2 called 3647945 times.
Case 3 called 235 times.

FYI: nn = 558155 and ne = 3234886
```
Scripts for MicroMed Debakey Pump

**script4_comolayer**

Usage: (from Parameterized_Debakey.X.XX):

- sets symbolic links to all required files
- runs comolayer
- comolayer combines layers of STR, IMP, and DIF to form one mesh with SSMUM layers
- converts complete mesh back into .grd-format for Gridgen visualization
- makes out.case, out.geo files for Ensight visualization
- compiles and runs test_length_radius_debakey.F for dimensional check
Scripts for MicroMed Debakey Pump

```plaintext
script3_prepare_for_gridgen

Usage: from folder Parameterized_DeBakey.Y.YY): script3_prepare_for_gridgen

f90 ME2STL.F  (original version was slightly modified by trimming rng numbers)
ME2STL  XXXclosed.mod  outputME2STL  ZZ  0.1 (split angle ZZ=
40 for Str, 70 for Dif, 90 for Imp) this converts .mod into stl format

for each component individually:
run glyph_script ME2STL.glf in gridgen (imports the .grd-file as STL)
run glyph_script formtetra_XXX.glf
(forms tetras, sets BNDRY-conditions via AS/W
RNG numbers have to be given in accordance with definitions in fv2mixd.F - here added BNDRY-group 10 as NBC0,
important: use own defined fv2mixd not the one under local/bin/
exports grid named Xxxtetra.grd)

====> we receive a .grd file, which is saved under Xxxtetra.grd
```
**Scripts for MicroMed Debakey Pump**

`script4_comolayer_altern` or `script4_comolayer_altern_invert`

Usage: (from Parameterized_Debakey.X.XX): script4_altern_comolayer

- option invert will invert also volume elements of 3 components to become negative
- sets symbolic links to all required files
- runs comolayer
- comolayer combines layers of STR, IMP, and DIF to form one mesh with SSMUM layers
- converts complete mesh back into .grd-format for Gridgen visualization
- makes out.case, out.geo files for Ensight visualization
- compiles and runs `test_length_radius_debakey.F` for dimensional check
Scripts for MicroMed Debakey Pump

re-import of complete model into gridgen

• visualization and use of block analysis functions for improvements

Size of tetra-hedra at DIF-side should be smaller in relation to those on STR-side
Scripts for MicroMed Debekey Pump
Scripts for MicroMed Debakey Pump

script5_mprm
Usage: on hydra from folder Parameterized_Debakey.Y.YY): script5_mprm

- launches several jobs on hydra to create different mprm-files

required time for generation of mprm-files:
- 0008 = 2 minutes
- 0016 and 0032 = 3 minutes
- 0064, 0128, and 0256 = 4 minutes
- 0512 = 5 minutes
Scripts for MicroMed Debakey Pump

partitions visualized in pager:

8 processors

512 processors
Scripts for MicroMed Debakey Pump

**script6_export**

Usage: from hydra from folder Parameterized_Debakey_YY): script6_export Y.YY

- makes an export folder with links to all relevant data required for running XNS  *(here a problem with data transfer was observed: in some cases mrng was not fully copied)*

- copies folder to JUMP
Scripts for MicroMed Debakey Pump

summary

script advantages:

• meshes of the same type can be generated in desired sizes
• good traceability
• improvements can be implemented on a continuous basis
• fast mesh generation

<table>
<thead>
<tr>
<th>Gridgen script</th>
<th>7 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>script1_makefolder</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>meshview</td>
<td>5 minutes</td>
</tr>
<tr>
<td>script2_makelayer</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>script3_DMG</td>
<td>9 minutes</td>
</tr>
<tr>
<td>script4_comolayer</td>
<td>4 minutes</td>
</tr>
<tr>
<td>script5_mprm</td>
<td>5 minutes</td>
</tr>
<tr>
<td>script6_export</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>(w/t copying to JUMP)</td>
<td></td>
</tr>
<tr>
<td>total computation time required:</td>
<td>~ 30 min</td>
</tr>
<tr>
<td>total user time required</td>
<td>&lt; 10 min</td>
</tr>
</tbody>
</table>
JUBL_Scripts for MicroMed Debakey Pump

script7_makefolder
Usage: change ne if required
from jubl working directory: /work/hac094: ./Scripts_Debakey_jubl/script7_makefolder X.XX

makes a folder structure in working directory on jubl
Debakey - diagnostic functions

some diagnostics gave reasonable results, e.g. min angle, max angle, ...

other diagnostic functions gave strange results, e.g.:

-aspect ratio (ratio of radii of circumscribing sphere to 3x inscribing sphere of tetrahedron) gave values of up to 8000,

-skewness (equi-vol) = (Optimal cell volume - Actual cell volume)/Optimal cell volume

-some values (120,000 in case with gridgen modeling) were above 1 which is a contradiction

Gridgen analysts informed that points of tetrahedra in fv-uns file are stored in the opposite sequence than specified by Intelligent Light.
gridgen analysis of the complete pump without meshview reveals 120,000 elements with negative volume - these are the elements in the SSMUM layer ==> when running xns checkx finds different signs for Jacobian determinant

analysis of pump constructed via DMG shows all volumes negative ==> as all elements are negative xns runs without error
Debakey - volume of elements

negative volumes explain also the unexpected values for skewness equi-vol.

The formula used for calculating aspect ratio is not documented (circumscribing circle radius can according to Heron's formula be computed as a function of triangle area (abc/4F) )

Makelayer and Comolayer both offer the option to invert elements via an option in makelayer.in

Default is: Dif inv, Str #inv, impstr inv, impstrdif #inv, comolayer #inv

so far I have tried out some other combinations (without a change)

next step will be to not invert all elements but only each second one

==> this is supposed to give SSMUM layer elements with different sign of the Jacobian
Helmholtz Couette Device

- do simulations in parallel with Helmholtz
- first step: simulate flow field
- second step: estimate hemolysis (different hemolysis models)
- compare to experimental results
- use advantages of scripting
Helmholtz Couette device

parabolic inflow of blood

region to be modeled

outflow of blood
(no seating fluid as since recent modification)

to close the bottom and top:
introduce solid wall with radial velocity as a function of radius and impermeable to axial flow

gap width = 0.09mm