Fluent Tips and Tricks

R. Jeffrey Benko
Customer Support Manager

June 5, 2003

Outline

- Materials From Corporate UGM Training Seminar
  - Samir Rida, Suti Wirogo (Aero Team)
- General Tips and Tricks: From Pre to Post
  - For Mesh and Case Set-up
  - For Solving
  - For Post-Processing
  - For Customizing Fluent
- Appendices
  - Miscellaneous Slides from General Tips and Tricks
  - User Defined Function Tips
  - The Text User Interface: **FLUENT**'s Hidden Jewels
  - A Case Study: The Segregated Solver with High Pressure Differential.

General Tips and Tricks:

For Mesh and Case Set-up

Miscellaneous on mesh modifications

- How to repair left-handed faces in Fluent ?

  /grid/modify-zones/repair-face-handedness TUI

- How to repair problematic periodic boundary in Fluent ?

  /grid/modify-zones/repair-periodic TUI

- What is tfilter (now called utility)?

  A set of utilities used by Fluent/Gambit/Tgrid to perform mesh related modifications, such as:

  – Convert mesh to Fluent format (fe2ram)
  – Convert mesh (tconv)
  – Merge meshes (tmerge3d)

  Tfilter can be invoked manually. To find all the options:

  shell> utility

  shell> utility fe2ram

Miscellaneous on mesh modifications

- In recent release, tfilter has been renamed to utility
- How to convert 2D mesh into 3D surface mesh ?

  utility tconv –d2 sample2d.msh 3d-surface.msh

  The z-coordinate is assigned as zero in the 3d surface mesh
- How to convert 3D surface mesh into 2d mesh ?

  utility tconv –d3 3d-surface.msh sample2d.msh

  The z-coordinate is ignored in the 3d surface mesh
- How to merge two meshes ?

  utility tmerge3d -cl -p ./mesh1.msh ./mesh2.msh combine.msh
- How to convert CFX mesh to Fluent mesh format ?

  utility fe2ram -cl -tCFX -dN -cl -group cfx.geo fl.cas

Creation of non-conformal interface

- Intersected mesh is created using the two interface meshes and then used to ‘replace’ the original interface meshes
- To improve accuracy and robustness:
  - Maintain similar face element sizes at both interfaces
  - Maintain good face mesh skewness at both interfaces
- To create non-conformal interface which crosses periodic boundary, need to set the following Scheme command first:

  (rpsetvar 'nonconformal/allow-interface-at-periodic-boundary 0)
- To view non-overlapping portions of the interface surfaces, use the TUI Command: /display/zone-grid
- Creation of intersected mesh is delicate, if fails, try to modify a default by using either of the following Scheme commands:
  - (rpsetvar 'nonconformal/cell-faces 0)
  - (rpsetvar 'nonconformal/cell-faces 2)

How to copy setup and solution to a similar case ?

- Avoid lengthy manual re-setup and solution initialization
- To copy the case setup:
  - From the first case, write a bc setup file:

    /file/write-bc TUI
  - Load the second mesh and read the bc file

    /file/read-bc TUI
  - Restrictions:
    - Only zones with the same names (not same ID) as used in the first case will be setup
    - Does not include mesh scaling (need to do scaling separately)
    - Surface information not copied

      Use journal file to create the surfaces to avoid recreation

How to copy setup and solution to a similar case ?

- To reuse the previous solution:
  - From the first case, write an interpolation file:

    File – Interpolate… – Write Data GUI
  - Read the interpolation file to the second case:

    File – Interpolate… – Read and Interpolate GUI

    Can selectively apply to individual fluid zone(s)
- Tips:
  - Simple closest distance interpolation is used (no gradient or geometry based interpolation)
  - Use small under-relaxation factors initially to avoid large jump in the solution

How to store user defined materials for reuse ?

- If user uses certain materials often then can create a material file that can be reused and thus avoid redefinition each time
- A material file can be created from a bc file:
  - Start a fluent session and define the user material in the material panel
  - Write the b.c. file

    /file/write-bc
  - Edit the b.c. file:
    - Keep the first and last line
    - Keep the material line

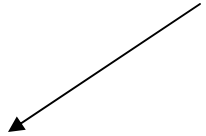
      (materials ((water-xxx fluid (chemical-formula . #f) ...
    - Delete all other lines
  - The next time the user would like to use this material file, then read the material file as if reading the b.c. file.

How to store user defined materials for reuse ?

- An alternative is to add entries to the file propdb.scm which is located in *~fluent6.x/cortex/lib*

```
(air
 fluid
 (chemical-formula . #f)
 (density (constant . 1.225))
 (specific-heat (constant . 1006.43))
 (thermal-conductivity (constant . 0.0242))
 (viscosity (constant . 1.7894e-05) (sutherland 1.7894e-05 273.11 110.56) (power-law 1.7894e-05 273.11 0.666))
 (molecular-weight (constant . 28.966))
 (lennard-jones-length (constant . 3.711))
 (lennard-jones-energy (constant . 78.6))
 (thermal-accom-coefficient (constant . 0.9137))
 (velocity-accom-coefficient (constant . 0.9137))
)
```

Example entry on propdb.scm



Custom Field Functions

- Custom field functions are GUI based, it is not possible to create them through journal files
- If a set of existing custom field functions are to be used for different cases:
  - Write the existing functions as a scheme file using:
    - Define – Custom Field Functions – Manage – Save GUI
    - file/write-field-functions TUI
  - Read the scheme file to the new case using:
    - Define – Custom Field Functions – Manage – Load GUI
    - file/read-field-functions TUI

General Tips and Tricks:

For Solving

Setting uneven parallel load
(OTS Solution 487)

- It is possible to partition the mesh in an uneven distribution to accommodate differences in the CPU speeds and RAM
- Suppose the 3 machines (m0, m1, m2) are to be used in a parallel job with 10%, 20%, and 70% load distribution respectively
  - Load the case and data file into serial Fluent
  - Using TUI, execute the command:

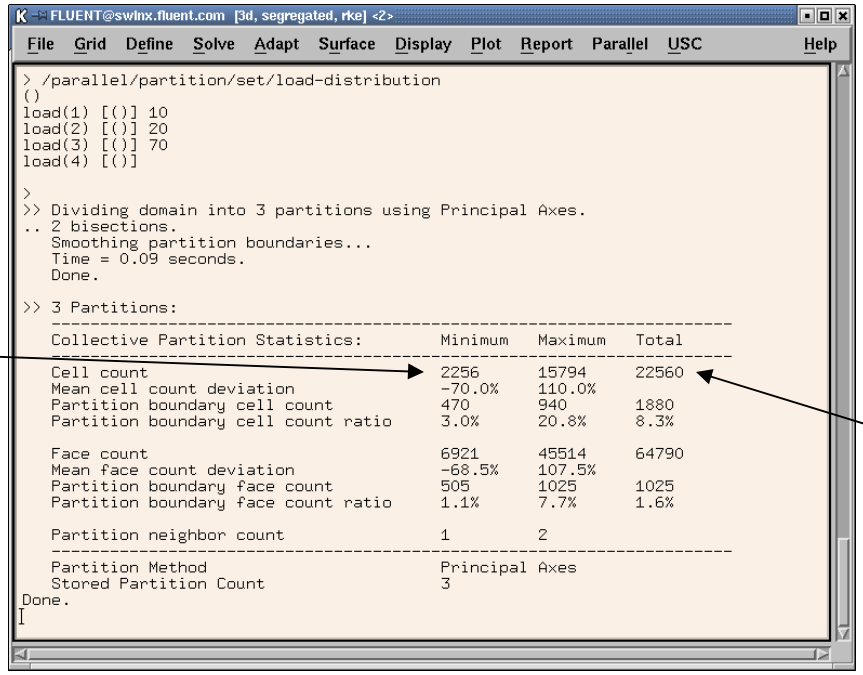
    /parallel/partition/set/load-distribution

    and enter the percentage load of each partition (must add to 100%)
  - Partition the mesh for 3 partitions
    - Can use any partitioning scheme

Setting uneven parallel load

Total mesh cell count is 22560.

10% of cell count is 2256.



70% of cell count is 15792.

Miscellaneous on parallel IO (1)

- In parallel, the mesh is first read into the host process and then distributed to the compute nodes
- If reading case into parallel solver takes unusually long time, do the following:
  - Put the host and compute-node-0 on the same machine
  - Put the case and data files on a disk on the machine running the host and compute-node-0 processes
  - Try to execute the following Scheme commands before reading the case file
    - (%free-parallel-io-buffers)
    - (%limit-parallel-io-buffer-size 0)
    - (%allocate-parallel-io-buffers 10000000)
    - (%query-parallel-io-buffers)

Miscellaneous on parallel IO (2)

- Merge as many zones as possible
- If the machine running the host process has enough memory, execute the following Scheme command:

  (rpsetvar 'parallel/fast-io? #t)

- When loading a large mesh into parallel Fluent and the process hangs, execute the following Scheme command:

  (rpsetvar 'parallel/case-read-use-pack? #f)

  The above will disable buffer packing and will change the way entities are packed into messages during the build process. More but smaller messages will be created by disabling.

How to determine CPU time for serial solver ?
(OTS Solution 334)

- For parallel run, can use the built in parallel timer via the GUI:
  - Parallel – Timer – Reset — before iteration
  - Parallel – Timer- Usage — after completing iterations
- For serial solver, can use the following TUI command that is executed before and after the iteration period of interest:

  (solver-cpu-time)
  - The difference is the elapsed CPU time in seconds for the iteration period completed
  - If used in the parallel session, the difference represents the sum of the CPU times of all the compute nodes

Miscellaneous on batch processing (1)

- Commands to run in batch mode:
  - fluent 3d -g < batch.jou >& out.trn & for C-shell
  - fluent 3d -g < batch.jou > out.trn 2>&1 & for Bourne/Korne-shell
  - fluent 3d -g -i batch.jou -o out.trn for windows
- Use –cnf=*host_file* to automatically load frequently used machines for parallel run
- To receive mail notification upon the completion of a batch job, add the following line at the end of the batch journal:

  !mail email-address < message.txt

  where message.txt is a text file located inside the working directory and contains the message to be sent

Miscellaneous on batch processing (2)

- To enter a comment inside a journal file:

  ; This line is commented

- To execute a shell command inside a journal file:

  ! Shell-command-to-be-executed

- To generate animation files during a batch processing, add the following option to the start up command of the batch job:

  -driver null

- Is it possible to read a journal file from another journal file ?

  Strictly speaking, the answer is “No”. Nested journal files are not allowed in Fluent. Scheme can be used as a workaround. Please contact your support engineer if you need this functionality.

Miscellaneous on batch processing (3)

- How to start a batch job at a user specified time ?
  - Create the batch journal file as usual i.e. batch.jou
  - Open and create a file containing a single line which specify the unix shell command to execute the batch job. For example:

    fluent 3d -g < batch.jou > out.trn 2>&1 & for Bourne/Korne-shell
  - Make this file executable

    unix> chmod a+x *filename*
  - Suppose that the batch job is to be executed at 2003 June 03 09:25:00 then, enter the following command in the unix prompt:

    unix> at -m -t 200306030925.00 –f *filename*
  - When the job finish, an email will be sent to you
  - The at command is also available on windows

Miscellaneous on batch processing (4)

- How to execute several batch processes sequentially ?
  - In Unix/Linux environments, if a shell command is not ended by an ampersand (&) then the OS will wait until the completion of the execution of that line before going to the next line. This is not the case with Windows.
  - Suppose there are two directories corresponding to two cases where each has its own corresponding journal file. Then see the following shell scripts:

Unix/Linux

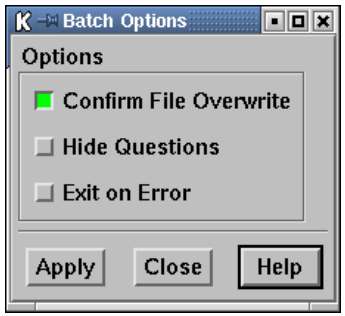
```
cd ./dir1
fluent 3d -psmpi -t2 -g < batch1.jou
cd ../dir2
fluent 3d -psmpi -t2 -g < batch2.jou
```

Windows

```
call cd .\dir1
call fluent 3d -wait -g -i batch1.jou
call cd ..\dir2
call fluent 3d -wait -g -i batch2.jou
```

Miscellaneous on batch processing (5)

- Useful options for batch processing have been implemented in Fluent 6.1



  - Confirm file overwrite is a must for batch processing
  - Option to turn off question dialog boxes
  - Exit when an error occurs during journal execution
  - Through GUI File → Batch Options or TUI: /file/set-batch-options
  - Equivalent Scheme command to turn on exit on error is:

    (set! *cx-exit-on-error* #t)

    Can be appended inside the journal file or .fluent file

Check pointing (1)

(OTS Solution 291)

- It is possible to tell the Fluent process to stop the iteration and write the case/data files without interacting with the GUI
- This procedure is useful to stop a batch process or when the GUI process has crashed, and can be used as long as Fluent is still in the iteration loop
- In Unix/Linux environment, the owner of the process can create a kill file inside the /tmp directory:

```
unix>  cd /tmp
unix>  touch exit-fluent          exit Fluent after writing files
                or
unix>  touch check-fluent         continue iteration after writing files
```

If either exit-fluent or check-fluent found in /tmp, Fluent will finish the current iteration, write the files, and then delete the exit-fluent or check-fluent file

Check pointing (2)

- The files will be written to the same directory where the original input files were read and will have the same names but with appended current iteration number
- To write the data file only, execute the following Scheme command before iterating:

  (rpsetvar 'checkpoint/write-case? #f)

  By default, the case file will be written
- Make sure that sufficient disk space is available
  - Check-pointing code calls the same file I/O routines used by the GUI or TUI and will produce the same error messages if disk space is insufficient
  - In such case, Fluent will not return to the iteration loop

Check pointing (3)

- The touch command will produce file of zero length and is also available in Windows
- For Windows, the check point files need to be created at:

  c:\temp\check-fluent.txt

  c:\temp\exit-fluent.txt

Check pointing (4)

- If the machine has several Fluent sessions running, named check pointing can be used to selectively stop a specific Fluent process
- A specific check point name can be added to the first line of the batch journal file, as shown below:

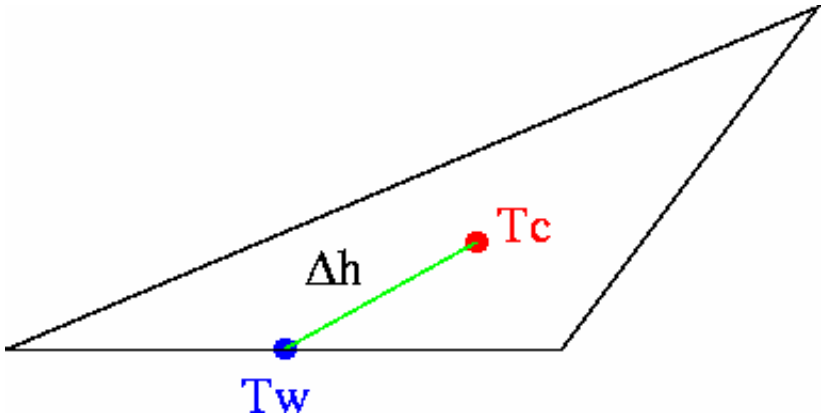
  (set! checkpoint/*exit-filename* "/*tmp*/*exit-flu-job-1*")
  file/read-case-data sample.cas
  solve/iterate 1000
  file/write-case-data junk2

- To stop this particular job, use the following command inside the /tmp directory:

  unix> touch exit-flu-job-1

Alternative Wall Formulation (1)

- /solve/set/expert/use alternate formulation for wall temperature? [no]
  - FLUENT models the flux at the wall as follows:



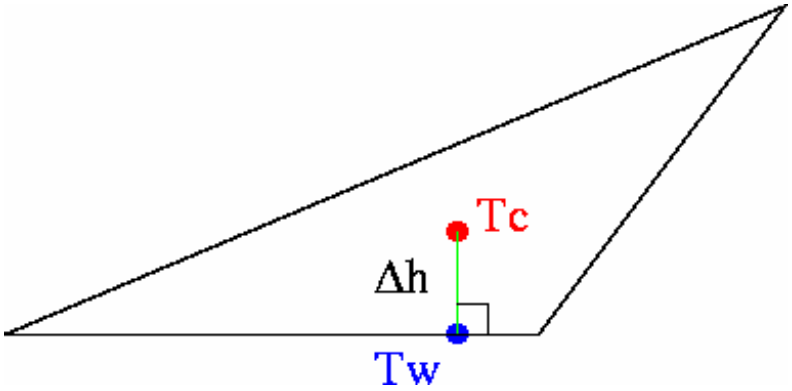
$$q = k\nabla T \cdot \vec{n}$$

$$q = k\frac{(T_w - T_c)}{\Delta h} + f(\Delta T)$$

  - The term $f(\Delta T)$ includes second order terms that need to be determined, which involves approximations.

Alternative Wall Formulation (2)

- /solve/set/expert/use alternate formulation for wall temperature? [yes]
  - Assume face center is defined such as the vector between cell center and face center is perpendicular to the wall.



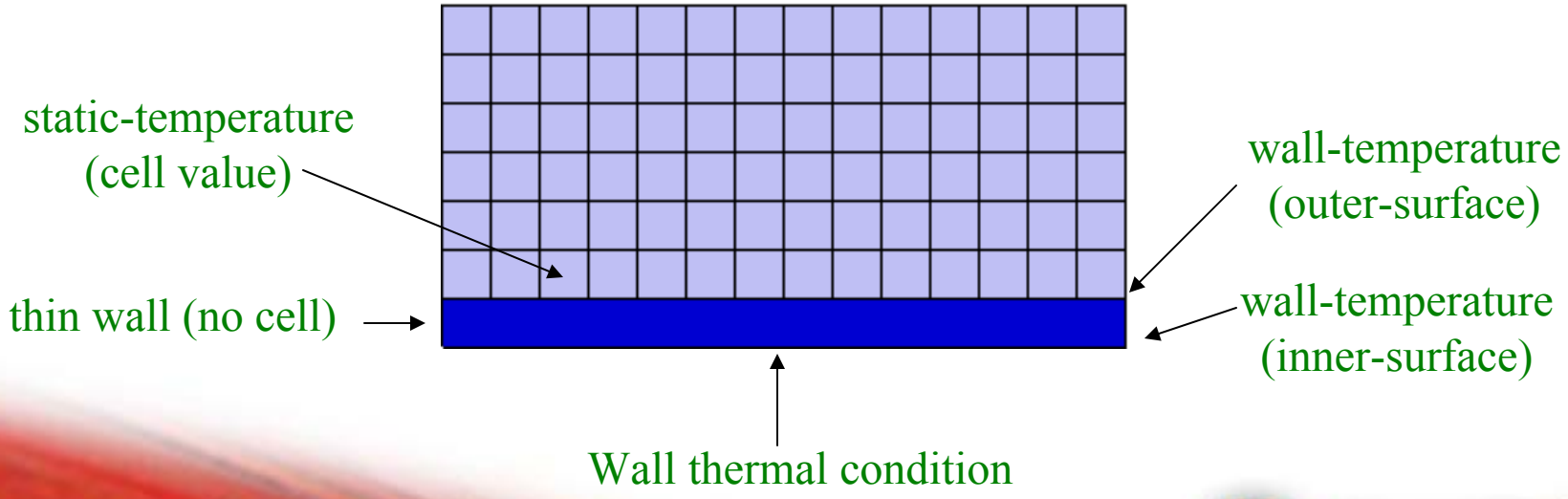
$$q = k\nabla T \cdot \vec{n}$$

$$q = k\frac{(T_w - T_c)}{\Delta h}$$

  - The term $f(\Delta T)$ vanishes and Δh changes.
  - This option doesn't impact the results if the wall cells are not skewed.

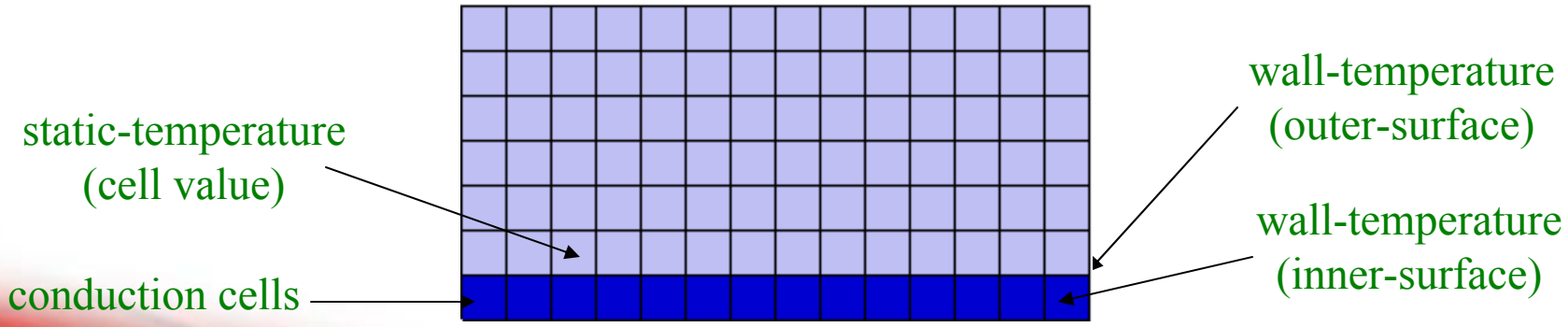
Temperature definitions for thin wall model

- Thin wall model applies normal conduction only (no in-plane conduction) and no actual cells are created
- Wall thermal boundary condition is applied at the outer layer



Temperature definitions for shell conduction zone

- A thin wall conduction with both normal and in-plane conductions
- Actual conduction cells are created but can not be displayed and can only be limitedly accessed by UDFs
- Solid properties of the conduction zones must be constant and can not be specified as temperature dependent
- Case must be partitioned in the serial mode before loading into parallel (conduction zones need to be encapsulated)



Misc. Text Commands

- /monitors/force/clear-all-monitors
  - Fast way to discard the internal and external file data associated with the force monitors.
- /set/flow-warnings?
  - Specify whether or not to print warning messages when reversed flow occurs at inlets and outlets, and when mass flow inlets develop supersonic regions. By default, flow warnings are printed.

General Tips and Tricks:

For Post-Processing

Pressure force calculation

- Force due to pressure is computed using the following equation in Fluent:

$$Fp = \text{Sum}\left(p_g - p_{ref} \right) * \text{Area}$$

where p_g is the gauge pressure and p_{ref} is the reference pressure specified in the reference value panel

- For closed surface (closed body), it does not matter what the value of p_{ref} is since the total pressure force due to p_{ref} is zero
- For open surface, p_{ref} must be set to the negative of the operating pressure so that the equation becomes:

$$Fp = \text{Sum}\left(p_g + p_{op} \right) * \text{Area}$$
$$= \text{Sum}\left(p_{abs} * \text{Area} \right)$$

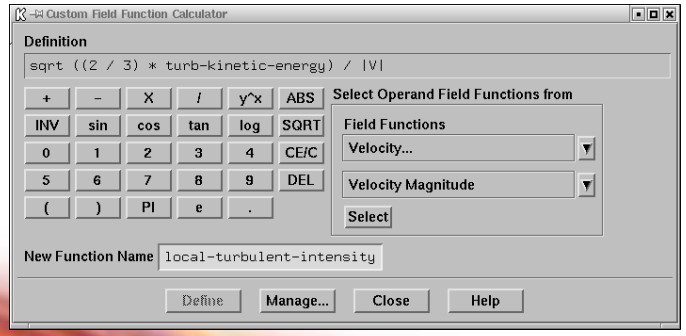
Turbulent intensity

- The definition used by Fluent when computing the turbulence intensity is:

  $$TI = \sqrt{(2/3)k}/V_{ref}$$

  where Vref is specified inside the reference value panel
- It is more instructive to use the local velocity magnitude for turbulence intensity calculation

  $$TI = \sqrt{(2/3)K}/V_{mag\text{-}local}$$
- Can use a custom field function to define modified turbulent intensity



How to create an iso-surface that passes through a selected zone?

- Create a surface that passed through the entire domain

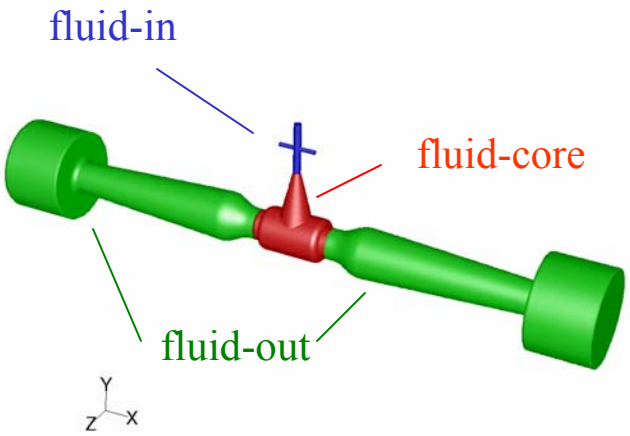
  Surface → Iso-Surface…

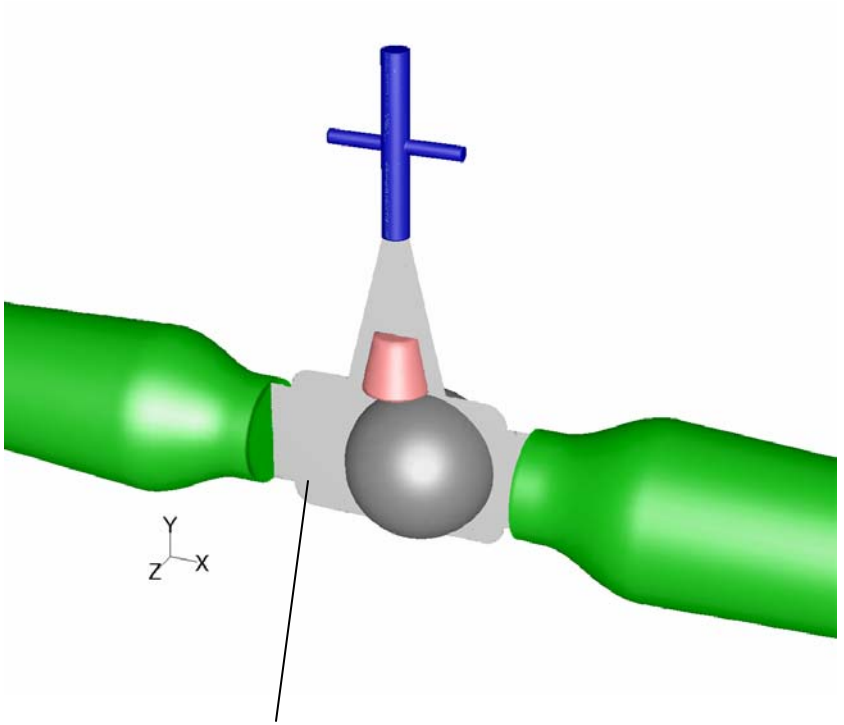
- From the boundary condition panel, determine the Cell Zone ID (same as the Cell Zone Index) where the iso-surface is to be retained. Lets say the Cell Zone ID is 7.
- Clip the iso-surface using to the desired ID:

  Surface → Iso-Clip…

  - Under "Clip To Values Of" select "Cell Info" and "Cell Zone Index"
  - Under "Clip Surface" select the surface created in first step
  - Enter 6.9 and 7.1 for "Min" and "Max", respectively
  - Specify the name and click on "Clip"

Example

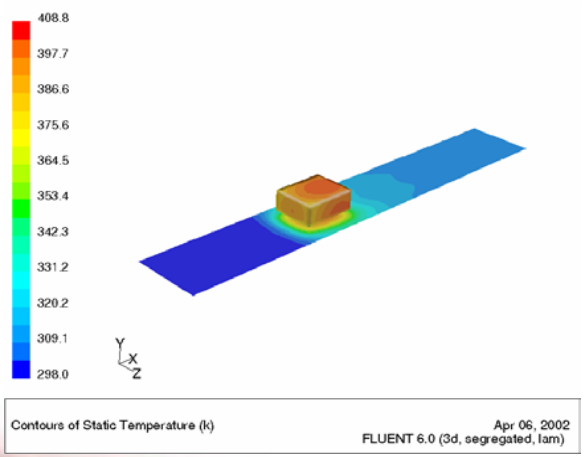


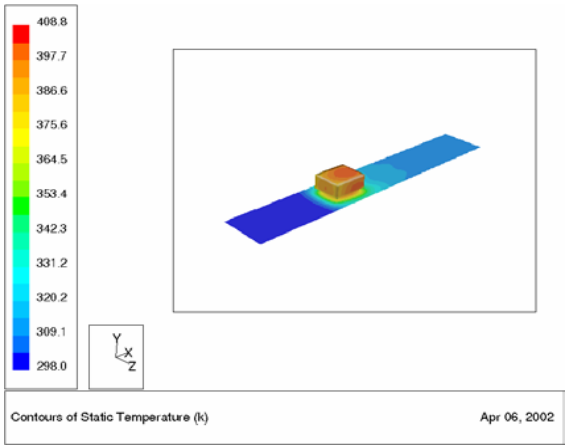


iso-clip passing through
fluid-core only

Altering the Display Window

- /display/set/windows/… [example]
  - Add borders to the scale, the axis, the image.
  - Eliminate the application title
  - etc





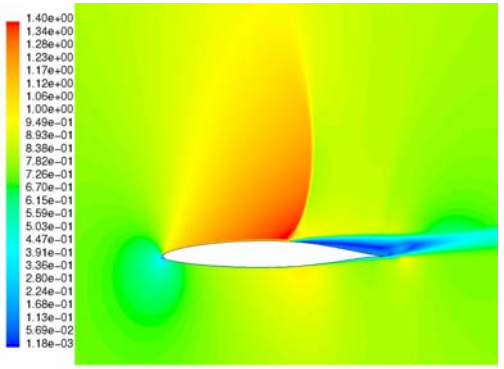
How to specify floating format for contour scale ?

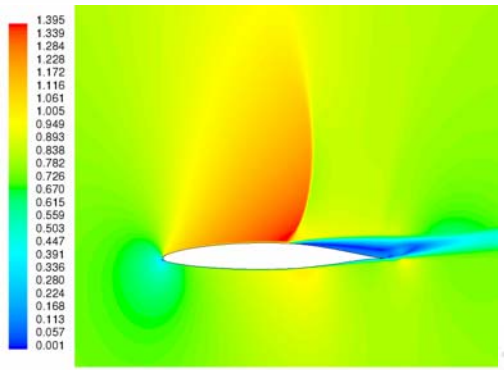
- The floating format for the contour scale is sometimes preferred since it is easier to read
- Use the following TUI command:

  /display/set/windows/scale/format "%f"

  – This will give the default 6 significant digits after the decimal point

  – To get 3 significant digits after the decimal point, use: "%0.3f"





General Tips and Tricks:

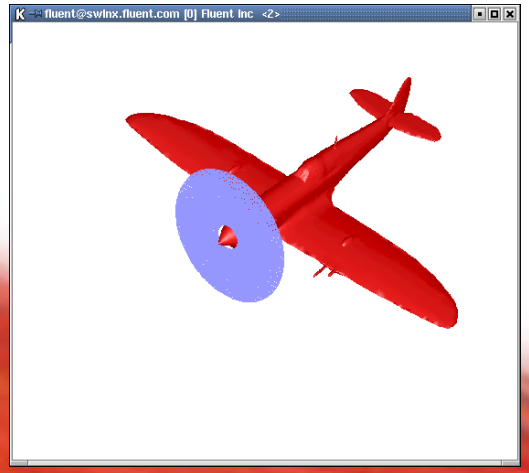
For Customizing Fluent

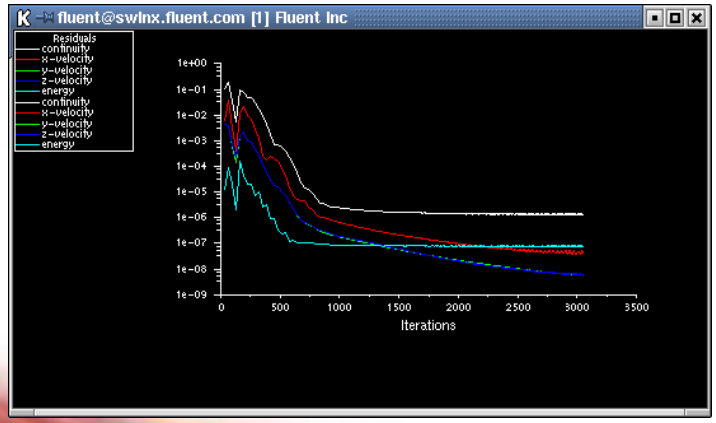
How to save graphics/GUI layout ?

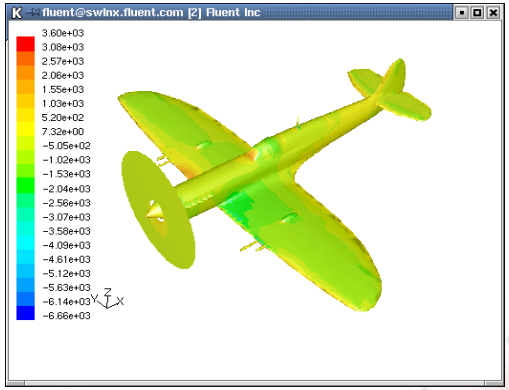
- To save custom window layout, can use the GUI command:

  File – Save Layout

- Sizes and locations of currently defined windows (0,1,2,...) as well as other windows for iteration panel, display panel, etc that have been opened will be saved in .cxlayout located at the same location as the .fluent file







Scheme

- What is Scheme ?

  A high level, functional, and interpreted based language used by Fluent interface

- How to load a Scheme file ?
  - File – Read – Scheme… GUI
  - file/read-macros TUI
  - (load "*filename.scm*") TUI

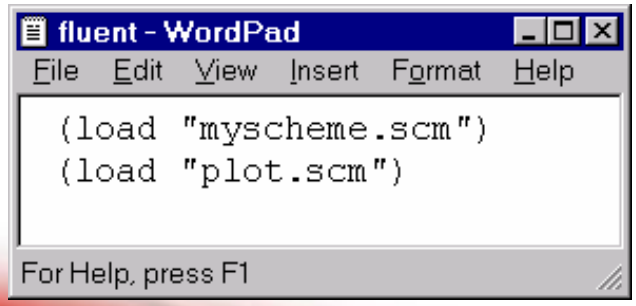
  The opening and closing double quotes are part of the command.

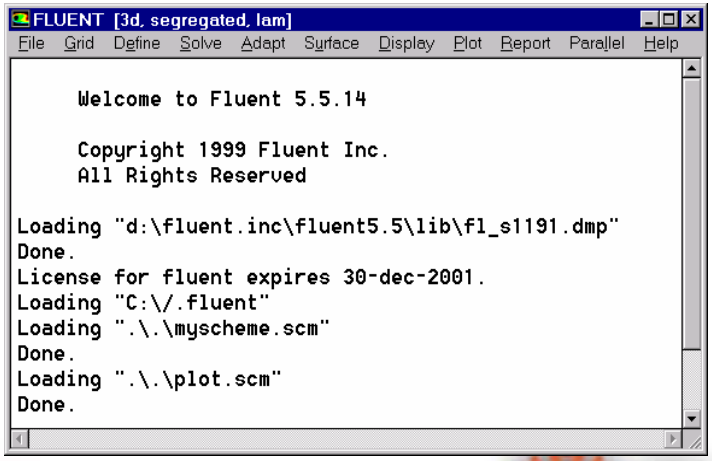
  Scheme file is not part of the case file and must be reloaded every time Fluent is restarted

- A Scheme command is executed using TUI only, e.g.
  - (rpsetvar 'flow-time 0.0) reset the flow time to 0.0
  - (display " Welcome to scheme…") print a welcome message at the fluent console

The .fluent file

- When FLUENT opens, it auto executes the .fluent file in:
  - home directory of unix users
  - c:\ directory of NT machines
- A .fluent file can contain any number of scheme file names to load
  - Note: unless full path is specified for each scheme file in .fluent file, FLUENT tries to locate the scheme files from the working directory

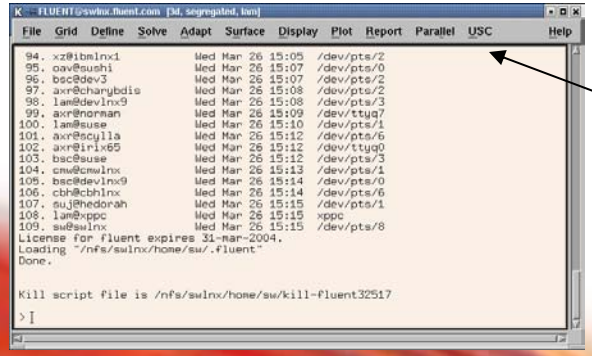




The .fluent file

- The Scheme commands can also be appended inside the .fluent file itself

```
;; ------------------------------------------------------------------------------;;
;; Code to create USC panel
(let ((menu (cx-add-menu "USC" #\U)))
(cx-add-item menu "User Services Center" #\O #f and (lambda ()
        (system "netscape http://www.fluentusers.com/ &"))))
;; Various useful stuff
(set! *cx-exit-on-error* #f)
;; ------------------------------------------------------------------------------;;
```



USC panel created by

How to modify Fluent defined variables ?

- Fluent defines a set of variables that control the model and interface setup
- Examples of these variables are:
  - flow-warnings? boolean
  - mg/grid-levels integer
  - smooth-mesh/niter
  - pressure/relax real
  - flow-time
  - physical-time-step
  - matching-tolerance
- List of the available variables is not made public but user can contact support engineer for a specific variable

How to modify Fluent defined variables ?

- These variables can be queried or changed via Scheme or UDF
- To query or change via Scheme:
  - (rpgetvar 'flow-warnings?)
  - (rpgetvar 'flow-time)
  - (rpsetvar 'pressure/relax 0.5)

  Note that there is no need to specify variable type in scheme
- To query or change via UDF:
  - RP_Set_Boolean("dpm/track-in-absolute-frame?",TRUE);
  - RP_Set_Real("flow-time",0.7);
  - RP_Set_Integer("time-step",3);

  Use RP_Get_*type* to query values

How to customize the Fluent startup setup ?

- Fluent imposes standard defaults on solver/model setup during startup of a new case, for example:
  - Printing but not plotting residuals
  - Under relaxation default values
  - Standard KE turbulence model
  - and many more …
- User can customize some of these defaults to avoid repetitive manual change every time fluent is started
- The scheme file given in the next slide shows an example which can be modified and then appended to the .fluent file

How to customize the Fluent startup setup ?

Append to the .fluent file

```
(let ((old-rc client-read-case))
  (set! client-read-case
        (lambda args
          (apply old-rc args)
          (if (cx-gui?)
              (begin
                ;; Do your customization here
                (rpsetvar 'residuals/plot? #t)
                (rpsetvar 'residuals/settings '((continuity #t 0 #f 0.001) (x-velocity #t 0 #f 0.001) (y-velocity #t 0 #f 0.001)
                      (z-velocity #t 0 #f 0.001) (energy #t 0 #f 1e-06) (k #t 0 #f 0.001) (epsilon #t 0 #f 0.001)))
                (rpsetvar 'mom/relax 0.4)
                (rpsetvar 'pressure/relax 0.5)
                (rpsetvar 'ke-realizability-on? #t)
                (rpsetvar 'realizable-epsilon? #t)
                (cxsetvar 'vector/style  "arrow")
                ;; You can add more settings here
                )))))
```

Turning off convergence check

Use ke-realizable as default

Use arrow instead of harpoon for vector head

How to use TUI to patch a variable ?

(OTS Solution 366 & 681)

- The Scheme file *tui-patch.scm* (given in next slide) provides user with the functionality to patch a variable on a given list of zones when running in batch mode
- The procedure consists of:
  - Load the Scheme file using TUI command:

    (load "*tui-patch.scm*")
  - Execute the TUI command:

    (patch (arg-patch variable-name desired-value list-of-zone-names))

  Example:

  (patch (arg-patch 'x-velocity 20.3 '(fluid-1 fluid-2 fluid-3)))

  (patch (arg-patch 'temperature 545.8 '(fluid-1 fluid-2 solid-4)))

  Note the key string x-velocity and temperature used by Fluent to identify field name and all values are in SI

How to use TUI to patch a variable ?

tui-patch.scm

```
(define thr-list
(lambda (name-list) (map thread-name->id name-list)))
(define cb-p
(let ((pa-var-list))
(set! pa-var-list (%inquire-patch-variable-names))
(lambda (varn) (assq varn pa-var-list))))
(define n-index
(lambda (varn index) (list-ref (cb-p varn) index)))
(define arg-patch
(let ((reg-list))
(set! reg-list '())
(lambda (varn xt name-list) (list (n-index varn 2) (n-index varn 3) (thr-list name-list) reg-list xt))))
```

Which interior zone(s) belong to a fluid zone ?

(OTS Solution 470)

- Can be done by visual inspection or changing the interior zone to wall and check the adjacent fluid zone(s)
- The Scheme function *find-zone.scm* (given in next slide) will list all the interior zones that are immersed within the user specified fluid zone
- The procedure consists of:
  - Load the Scheme file using TUI command:

    (load "*find-zone.scm*")
  - Execute the TUI command:

    (imme-info 'fluid-1)

Which interior zone(s) belong to a fluid zone ?

find-zone.scm

```
(define (imme-info fluid-zone-name)
(let ((id-fz (thread-name->id fluid-zone-name))
      (int-list (map thread-id (get-threads-of-type 'interior))))
(for-each (lambda (id)
(let ((zz (inquire-adjacent-threads id))(id1)(id2))
(set! id1 (car zz))
(set! id2 (cadr zz))
(if (eqv? id1 id2) (if (eqv? id1 id-fz)
(format "\n Interior zone \"~a\" is immersed in fluid zone  \"~a\" \n"
        (thread-id->name id) fluid-zone-name) ))))) int-list))
(format "-----------------------------------------------------------"))
```

How to define customized color map ?

- It is possible to read a custom color map or write an existing one
- The procedure consists of:
  - Load the Scheme file:

    (load "*rw-colormap.scm*")
  - To read a new color map:

    /file/read-colormap
  - To write an existing color map:

    /file/write-colormap

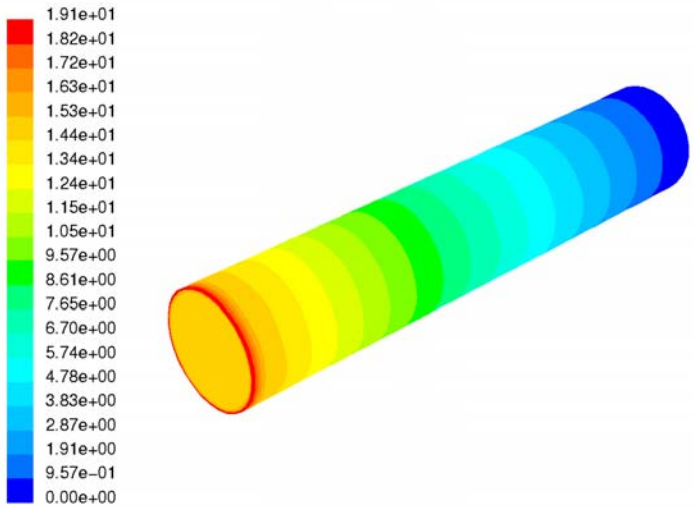
Example: thermacam colormap

Scale (0-1)

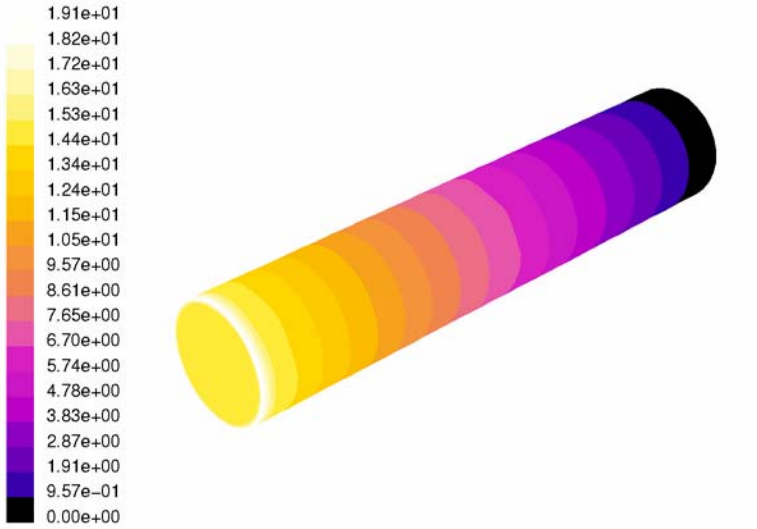
	R	G	B
("thermacam"			
(0.0	0.000	0.000	0.000)
(0.083	0.055	0.000	0.467)
(0.167	0.306	0.000	0.592)
(0.250	0.545	0.000	0.616)
(0.333	0.725	0.016	0.584)
(0.417	0.824	0.114	0.455)
(0.500	0.898	0.267	0.098)
(0.583	0.945	0.404	0.012)
(0.667	0.973	0.545	0.000)
(0.750	0.996	0.702	0.000)
(0.833	0.996	0.847	0.047)
(0.917	1.000	0.945	0.455)
(1.000	1.000	1.000	0.976)
)			

How to define customized color map ?

Example: Static pressure variation for a simple duct



default Fluent color map



therma-cam color map

How to define customized color map ?

rw_colormap.scm

```
(define (write-cmap fn)
  (let ((port (open-output-file (cx-expand-filename fn)))
        (cmap (cxgetvar 'def-cmap)))
    (write (cons cmap (cx-get-cmap cmap)) port)
    (newline port)
    (close-output-port port)))

(define (read-cmap fn)
  (if (file-exists? fn)
      (let ((cmap (read (open-input-file (cx-expand-filename fn)))))
        (cx-add-cmap (car cmap) (cons (length (cdr cmap)) (cdr cmap)))
        (cxsetvar 'def-cmap (car cmap)))
      (cx-error-dialog
       (format #f "Macro file ~s not found." fn))))

(define (ti-write-cmap)
  (let ((fn (read-filename "colormap filename" "cmap.scm")))
    (if (ok-to-overwrite? fn)
        (write-cmap fn))))
```

(1)

```
(define (ti-read-cmap)
  (read-cmap (read-filename "colormap filename" "cmap.scm")))

(ti-menu-insert-item!
   file-menu
   (make-menu-item "read-colormap" #t ti-read-cmap
    "Read a colormap from a file."))

(ti-menu-insert-item!
   file-menu
   (make-menu-item "write-colormap" #t ti-write-cmap
    "Write a colormap to a file."))
```

(2)

One Miscellaneous Scheme Tip

- Given a Zone ID, how do you find its zone name?
  - Can be useful to know which zone is being mentioned in an error message.

  In the TUI: (thread-id→name ID_NUMBER)

Summary

- General Tips and Tricks: From Pre to Post
  - For Mesh and Case Set-up
  - For Solving
  - For Post-Processing
  - For Customizing Fluent
- Appendices
  - Miscellaneous Slides from General Tips and Tricks
  - User Defined Function Tips
  - The Text User Interface: **FLUENT**'s Hidden Jewels
  - A Case Study: The Segregated Solver with High Pressure Differential.
- Check out the User Services Center (USC) for solutions in the Online Technical Support (OTS) Area.

Appendices

- Miscellaneous Slides from General Tips and Tricks
- User Defined Function Tips
- The Text User Interface: **FLUENT**'s Hidden Jewels
- A Case Study: The Segregated Solver with High Pressure Differential.

Part I : Miscellaneous Slides from General Tips and Tricks

How to address the heap memory limitation?

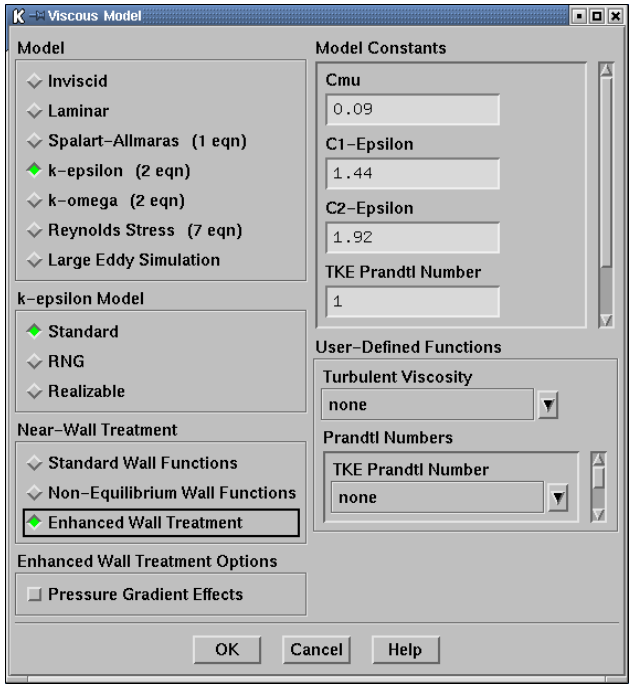
- In certain instances on Unix and Linux platforms, the heap memory limit can be reached when:
  - Creating a large number of surfaces
  - Defining a large number of DPM injections
- To increase the heap memory allocation, start Fluent using the "-h10000000" option
  - This will allocate 10000000 storage
  - The default is 500000
- For the DPM case, deleting all individual injections and replacing them by several injections of type "file" will alleviate the problem

How to modify the Schmidt number?

- In V5, Sc_t is "hardwired" to the Pr_t
  - Sc_t could not be changed independently of Pr_t
  - For adiabatic PDF simulations, Sc_t could not be changed at all
- In V6.0, updated Sc_t definition provides greater user flexibility
  - "PDF Schmidt number"
    - Used in mixture fraction based calculations (default is 0.85)
  - "Turbulent Schmidt number"
    - Used in species transport calculations (default is 0.7)
  - Either may be updated in the Define/Models/Viscous GUI panel

How to read V5 case with two-layer-zonal model in V6?

- The two-layer-zonal model is no longer available in Fluent 6.
- When reading a Fluent 5 case with 2-layer-zonal model, Fluent 6 will chose the enhanced wall treatment option



How to define custom turbulent thermal conductivity?

- The UDF DEFINE_PROPERTY allows the user to modify only the laminar thermal conductivity. For turbulent flows, FLUENT computes a turbulent thermal conductivity based on turbulent viscosity and add it to the (UDF-defined) laminar conductivity

```
#define   C_K_T(c,t)   (C_MU_T(c,t) * C_CP(c,t) / M_keprt)
#define   C_K_EFF(c,t)   (C_K_L(c,t) + C_K_T(c,t))
```

- If the turbulent Prandtl number "M_keprt" is set to a high value in the define/models/viscous panel, C_K_T will become infinitely small. The customized the turbulent heat conductivity can be entrered through DEFINE_PROPERTY UDF for C_K_L

Note: Setting the turbulent Prandtl number to a very high value will eliminate the turbulence production due to buoyancy effects

How to compute the gradient of a scalar ?

- Many field variables in Fluent have their gradients stored internally which can then be accessed via UDF for post-processing or other purposes
- For variables where the gradients are not stored, can use the following procedure:
  - Load case/data file, define one UDS (UDS-0) and one UDM (UDM-0)
  - Modify, load, and compile the UDF given in the next slide
  - Execute the assign_scalar on-demand UDF

    This will assign the variable (whose gradient is to be calculated) to the UDS-0
  - Using TUI, execute 0 iteration

    /solve/iterate 0

    This will compute all the coefficients and gradients but won't solve the
  - Execute the store_gradient on-demand UDF
  - This will assign the computed gradient to UDM-0

How to compute the gradient of a scalar ?

```
#include "udf.h"
#define zone_ID 2

DEFINE_ON_DEMAN(assign_scalar)
{
  Thread  *tc;
  Domain *domain;
  cell_t     c;
  face_t    f;

  domain = Get_Domain(zone_ID);

  /* Fill UDF with the scalar */
  thread_loop_c(tc,domain)
    {
      begin_c_loop(c,tc)
        C_UDSI(c,tc,0) = C_P(c,tc);
      end_c_loop(c,tc)
    }
}
```

modify this

(1)

```
DEFINE_ON_DEMAND(store_gradient)

{
  Thread  *tc;
  Domain *domain;
  cell_t     c;

  domain = Get_Domain(zone_ID);

  /* Fill UDM with magnitude of gradient */
  thread_loop_c(tc,domain)
    {
      begin_c_loop(c,tc)
        C_UDMI(c,tc,0) = NV_MAG(C_UDSI_C(c,tc));
      end_c_loop(c,tc)
    }
}
```

(2)

Part II : User Defined Function Tips

Miscellaneous on UDF (1)

- Define Adjust UDF is called at the beginning of every iteration even for unsteady solver
- For unsteady runs, there are instances where it is desirable for the Adjust UDF to be called only at the beginning of the first iteration of every time-step
- Can use the macro *first-iteration*, example:

```
DEFINE_ADJUST(myadjust,domain)
{

 if (first_iteration)
  {
   /* Do procedures to be executed only at the
        first
      iteration of every timestep */
  }
}
```

Miscellaneous on UDF (2)

- To get the current iteration number, use:

```
int current_iteration_number = (nres==0)?(1):((int)count2[nres-1]);
```

- To check if a face zone is of type wall, use:

```
if (THREAD_TYPE(tf) = = THREAD_F_WALL)
```

Similarly for other face zone types, can use:

- THREAD_F_INTERIOR
- THREAD_F_SYMMETRIC
- THREAD_F_PERIODIC
- THREAD_F_MFINLET
- THREAD_F_PINLET
- THREAD_F_POUTLET
- THREAD_F_PFAR
- THREAD_F_VINLET
- THREAD_F_OUTFLOW

Miscellaneous on UDF (3)

- To toggle between 2d and 3d, use the following compiler directives:

```
#if RP_2D
    { … codes for 2D …}
#else
    { … codes for 3D … }
#endif
```

- To toggle between 2d and 2d-axisymmetric, use:

```
#if RP_2D
    if (rp_axi)
        { … codes for axisymmetric …}
    else
        { … codes for 2D planar … }
#endif
```

Miscellaneous on UDF (4)

- To toggle between segregated and coupled solvers, use:

```
if (rp_seg)
  { … codes for segregated solver …}
else
  { … codes for segregated solver …}
```

- To toggle between inviscid and viscous, use:

```
if (rp_visc)
  { … codes for viscous …}
else
  { … codes for inviscid …}
```

- To exclude boundary face when looping through the faces:

```
if (!BOUNDARY_FACE_THREAD_P(t))
```

Miscellaneous on UDF (5)

- How to access solution residuals using UDF ?

```
#include "udf.h"DEFINE_ADJUST(my_adjust,domain)
{
  int nw;
  real scaled_res;
  FILE *fp;

  fp = fopen("output_test","a");
  fprintf(fp,"\n");
  fprintf(fp, "current iter = %d\n",count2[nres-1]);

  for (nw=0; nw<MAX_EQNS; ++nw)
    {
     if (strlen(DOMAIN_EQN_LABEL(domain,nw))>0)
         {
           scaled_res = DOMAIN_RES(domain,nw)[nres-1]/
                        DOMAIN_RES_SCALE(domain,nw)[nres-1];
           fprintf(fp,"%s equation: %g\n",DOMAIN_EQN_LABEL(domain,nw),scaled_res);
         }
    }

    fclose(fp);
}
```

```
current iter = 1
continuity equation: 1
x-velocity equation: 1.01033e-08
y-velocity equation: 1.01114e-08
z-velocity equation: 0.000495387
energy equation: 5.91456e-08
k equation: 0.77859
epsilon equation: 0.7963
```

Miscellaneous on UDF (6)

- How to loop through all nodes of a given face zone only once ?

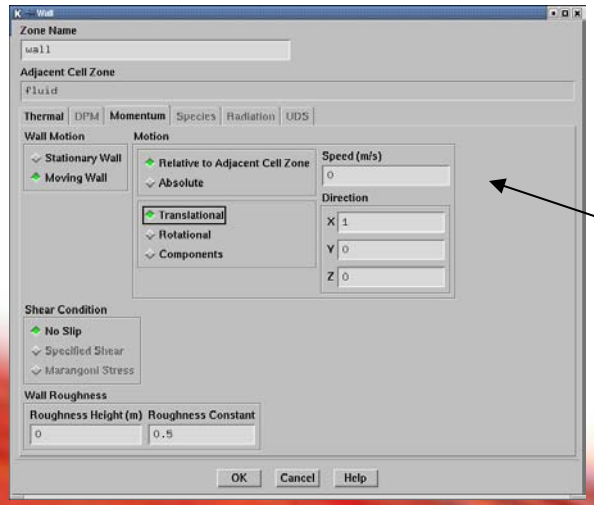
```
begin_f_loop(f,tf)
   f_node_loop(f,tf,n)  SET_V_FLAGS(F_NODE(f tf,n),TAG_FLAG);
end_f_loop(f,tf)
```

Each node of the corresponding face zone will be visited only once

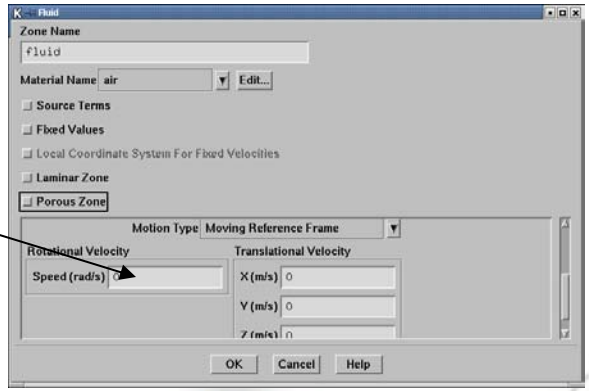
```
begin_f_loop(f,tf)
  {
    f_node_loop(f,tf,n)
      {
        if (NODE_FLAG (v = F_NODE(f,tf,n),TAG_FLAG))
          {
            CLEAR_V_FLAGS (v, TAG_FLAG);
             /* Your code */
          }
      }
  }
end_f_loop(f,tf)
```

Miscellaneous on UDF (7)

- Certain fields inside the Fluent model panel do not have built-in UDF hookups, thus it is difficult to specify user specified time/iteration varying values
- Examples are the wall and fluid speeds
- Is there any workaround ?



No UDF hookup



Miscellaneous on UDF (8)

- Can write a DEFINE_ADJUST UDF and directly access the macro of the variable of interest (contact support engineer for variable names)

```
THREAD_VAR(tf).wall.origin[0]
THREAD_VAR(tf).wall.origin[1]
THREAD_VAR(tf).wall.origin[2]

THREAD_VAR(tf).wall.axis[0]
THREAD_VAR(tf).wall.axis[1]
THREAD_VAR(tf).wall.axis[2]

THREAD_VAR(tf).wall.translate_mag
THREAD_VAR(tf).wall.translate_dir[0]
THREAD_VAR(tf).wall.translate_dir[1]
THREAD_VAR(tf).wall.translate_dir[2]

THREAD_VAR(tf).wall.omega

THREAD_VAR(tf).wall.u
THREAD_VAR(tf).wall.v
THREAD_VAR(tf).wall.w
```

```
THREAD_VAR(tc).fluid.origin[0]
THREAD_VAR(tc).fluid.origin[1]
THREAD_VAR(tc).fluid.origin[2]

THREAD_VAR(tc).fluid.axis[0]
THREAD_VAR(tc).fluid.axis[1]
THREAD_VAR(tc).fluid.axis[2]

THREAD_VAR(t1).fluid.velocity[0]
THREAD_VAR(t1).fluid.velocity[1]
THREAD_VAR(t1).fluid.velocity[2]

THREAD_VAR(tc).fluid.omega
```

Miscellaneous on UDF (9)

- Example UDF to ramp up the fluid rotational speed for MRF to obtain improved convergence

```
#include "udf.h"

#define Niter 25
#define omg_mn 1
#define omg_mx 100

DEFINE_ADJUST(myadjust,domain)
{
  int    zoneID;
  int    iter = (nres==0)?(1):((int)count2[nres-1]);
  float  omega;
  Thread *tc;
  FILE   *fp;
```

(1)

```
  fp = fopen("output_check","a");

  zoneID = 3;
  tc = Lookup_Thread(domain,zoneID);

  if ( (iter%Niter)==0 )
    {
      omega = omg_mn + (iter/Niter)*10;
      if ( omega > omg_mx ) omega = omg_mx;
      THREAD_VAR(tc).fluid.omega = omega;
      fprintf(fp,"At iter=%d - update omega to %f\n",
                 iter,omega);
      fclose(fp);
    }
}
```

(2)

Part III : Fluent Text User Interface

FLUENT's Hidden Jewels

Text User Interface

- The text menu system provides a hierarchical interface to the program's underlying procedural interface.
- The user can easily manipulate its operation with standard text-based tools: input can be saved in files, modified with text editors, and read back in to be executed.
- Text menu system is tightly integrated with the scheme extension language, it can easily be programmed to provide sophisticated control and customized functionality.

Text User Interface

- The general menu consists of

adapt/	grid/	surface/
display/	plot/	view/
define/	report/	exit
file/	solve/	

- Certain commands are available only through the TUI. They will be outlined and described hereinafter.

define/models/dpm

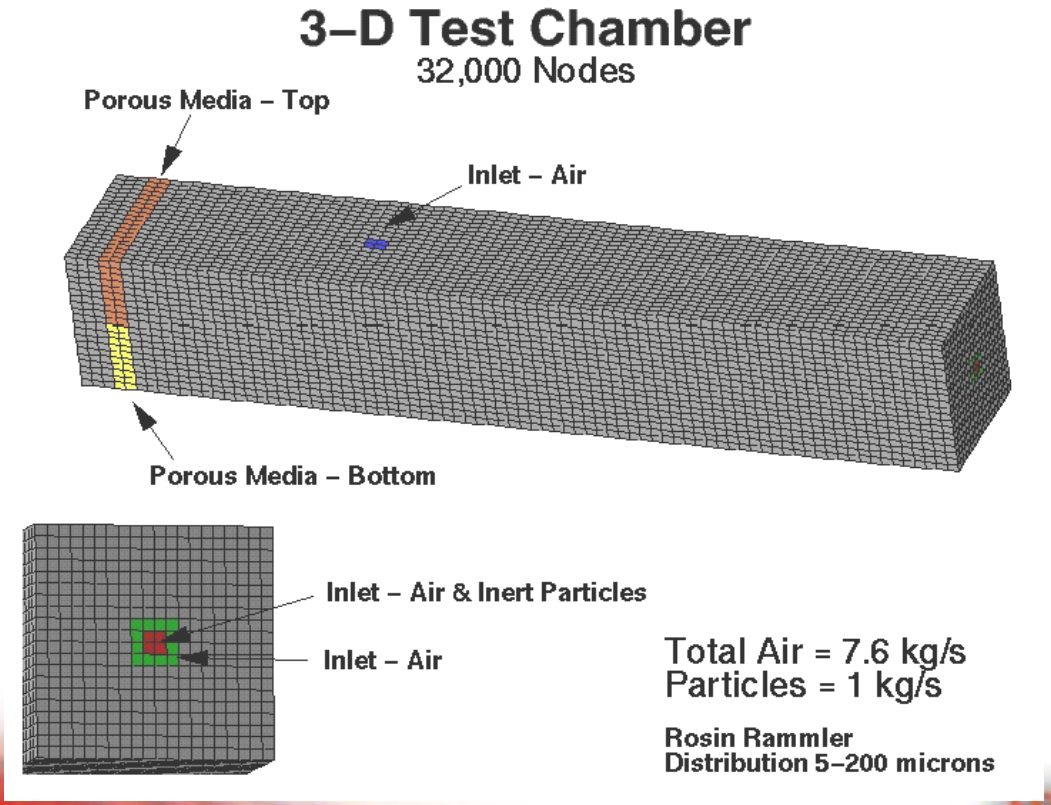
- clear-particles-from-domain
  - Remove all particles currently in the domain.
- coupled-heat-mass-update
  - By default, the solution of the particle heat and mass equations are solved in a segregated manner. If you enable this option, FLUENT will solve this pair of equations using a stiff, coupled ODE solver with error tolerance control. The increased accuracy, however, comes at the expense of increased computational expense.
  - This option gives a more accurate temperature and mass content for the particles when having a strong coupling between both equations. It doesn't affect nor accelerate the coupling of the particle source terms to the fluid equations.

define/models/dpm

- implicit-momentum-coupling
  - Use an implicit treatment for the DPM momentum source terms (else explicit).
  - In certain circumstances, the implicit source term linearization model doesn't seem to respond very quickly to major changes in the flow field (turning an air inlet off or adding / removing porous media). In these cases, FLUENT behaves better when the explicit source term formulation is used
  - Example: The geometry is a 2x2x12 meter chamber. Air and particles are introduced through inlets on one end of the chamber. An additional air inlet is located on the top of the chamber at a distance of approximately 8m.

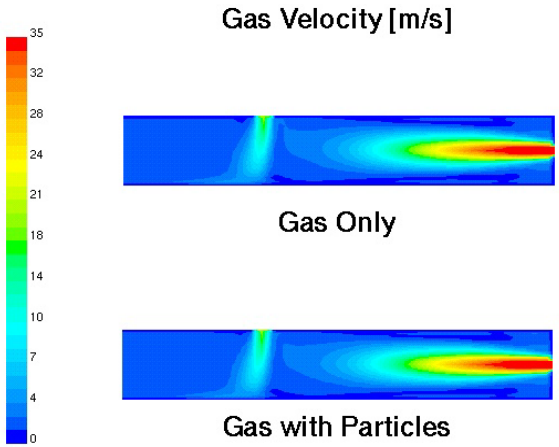
define/models/dpm

- implicit-momentum-coupling



define/models/dpm

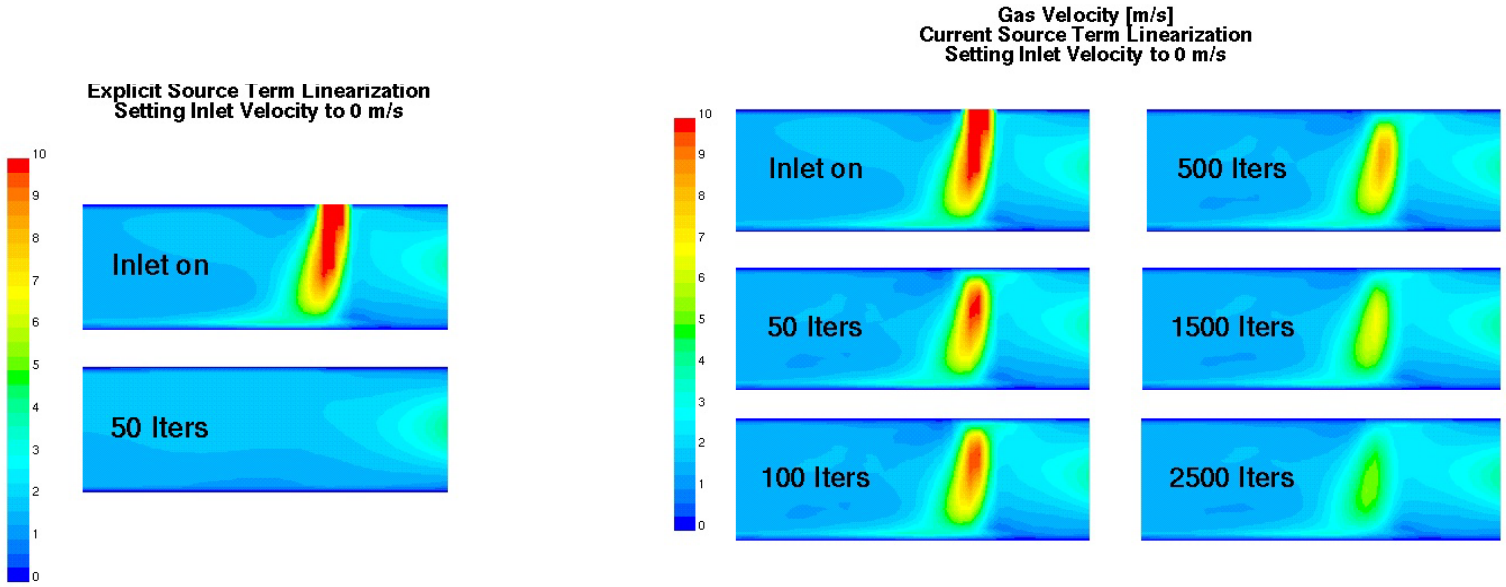
- implicit-momentum-coupling



  - The predicted gas velocity magnitude on the chamber mid-plane is shown. The main jet is introduced on the right side and persists approximately halfway down the chamber. A second smaller jet can be seen entering from the top approximately 2/3 of the way down the chamber.

define/models/dpm

- implicit-momentum-coupling



  - Top inlet was turned off. With explicit formulation, after 50 iteration there is little evidence that an air jet was injected from chamber roof. With implicit formulation it takes more time.

define/models/dpm

- velocity-gradient-correction
  - During the particle tracking step the velocity of the surrounding medium changes. In FLUENT 6 this has been taken into account for the particle trajectory equation.
  - FLUENT predicts the trajectory of a discrete phase particle (or droplet or bubble) by integrating the force balance on the particle, which is written in a Lagrangian reference frame.
  - With the trajectory itself predicted by:

$$\frac{du_p}{dt} = F_D\left(u - u_p\right) + g_i\left(\rho_p - \rho\right)/\,\rho_p + F_i$$

$$\frac{dx}{dt} = u_p$$

define/models/dpm

- velocity-gradient-correction
  - The trajectory equation is rewritten in simplified form as:

$$du_p / dt = \left(u - u_p\right) / \tau_p$$

    τ is the particle relaxation time.
  - As opposed to Fluent 5, FLUENT 6 uses a trapezoidal scheme to integrate this equation

$$\frac{u_p^{n+1} - u_p^n}{\Delta t} = \frac{1}{\tau}\left(u^* - u_p^{n+1}\right) + \ldots$$

$$u^* = \frac{1}{2}\left(u^n + u^{n+1}\right) \qquad u^{n+1} = u^n + \Delta t\, u_p^n \cdot \nabla u^n$$

  - The trapezoidal scheme approach is the default with velocity-gradient-correction switched on.

define/models/dpm

- Track-in-absolute-frame
  - When multiple reference frames are used in conjunction with the discrete phase model, the display of particle tracks will not, by default, be meaningful. Similarly, coupled discrete-phase calculations are not meaningful.
  - An alternative approach for particle tracking and coupled discrete-phase calculations with multiple reference frames is to track particles based on absolute velocity instead of relative velocity. To make this change, this command should be used. However, tracking particles based on absolute velocity may result in incorrect particle-wall interaction.

define/models/dpm

- Track-in-absolute-frame
  - The particle injection velocities (specified in the Injection panel) are defined relative to the frame of reference in which the particles are tracked. By default, the injection velocities are specified relative to the local reference frame. If this option is enabled, the injection velocities will be specified relative to the absolute frame.

define/models/energy?

- include diffusion at inlets?
  - The net transport of energy at inlets consists of both the convection and diffusion components. The convection component is fixed by the inlet temperature specified by the user. The diffusion component, however, depends on the gradient of the computed temperature field. Thus the diffusion component (and therefore the net inlet transport) is not specified a priori.
  - In some cases, the user may wish to specify the net inlet transport of energy rather than the inlet temperature. By default, FLUENT includes the diffusion flux of energy at inlets. To turn off inlet diffusion, answer [no] to include diffusion at inlets?
  - Available only for the segregated solver.

define/models/species

- inlet-diffusion?
  - Similar to the energy equation, the net transport of species at inlets consists of both the convection and diffusion components. The convection component is fixed by the inlet species concentration specified by the user. The diffusion component, however, depends on the gradient of the computed species concentration field.
  - If the user wants to include only the convective transport of species through the inlets of the domain. This can be done by disabling inlet species diffusion. By default, FLUENT includes the diffusion flux of species at inlets.
  - Available only with the segregated solver.

define/models/viscous/

- sa-damping? [yes]
  - The full Spalart-Allmaras model is a low-Reynolds-number model. Damping functions have been built into the model in order to properly reduce the turbulent viscosity in the viscous sublayer.

$$\rho\frac{D\tilde{\nu}}{Dt} = \rho c_{b1}\tilde{S}\tilde{\nu} + \frac{1}{\sigma_{\tilde{\nu}}}\left[\frac{\partial}{\partial x_j}\left\{(\mu+\rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right\} + c_{b2}\rho\left(\frac{\partial\tilde{\nu}}{\partial x_j}\right)^2\right] - c_{w1}\rho f_w\left(\frac{\tilde{\nu}}{d}\right)^2$$

$$\mu_t \equiv \rho\tilde{\nu}f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi \equiv \frac{\tilde{\nu}}{\nu}$$

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}$$

damping functions

$$f_w = g\left[\frac{1+c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}, \quad g = r + c_{w2}\left(r^6 - r\right), \quad r \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$$

distance from wall

Wall boundary condition : $\tilde{\nu} = 0$

define/models/viscous/

- sa-damping? [no]
  - There are some situations in which the damping functions will adversely affect the accuracy of the model when it is used with coarse meshes. A high-Reynolds-number form of the model has been included.

$$f_{v1} = 1, \quad f_{v2} = 0, \quad f_w = unchanged$$

  - Wall Boundary condition :

$$\tilde{\nu} = u_\tau \kappa d$$

  - It is assumed that the mesh is too coarse to resolve the laminar sublayer, and the wall shear-stress is obtained from the law-of-the-wall:

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \ln\left(E \frac{\rho y u_\tau}{\mu} \right)$$

define/models/viscous/

- turbulence-expert/
  - Menu for expert options related to turbulence.
- turbulence-expert/low-re-ke?
  - Extend the k-ε model to low Reynolds numbers and allow it to be applied very close to the wall. Modifications involve adding damping functions to the source terms in the transport equation for ε and to the expression for turbulent viscosity.
  - While the damping functions allow the equations to be used through the turbulent boundary layer, including the viscous sub-layer, they are typically of an ad hoc nature and therefore cannot be relied upon to give consistently good results in all types of flows.

define/models/viscous/

- turbulence-expert/low-re-ke? [yes]
  - Standard k-ε model modified by damping functions: f_μ, f_1, f_2

ε–transport equation

$$\rho \frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{\varepsilon}{k}\left(f_1 C_{1\varepsilon}\mu_t S^2 - \rho f_2 C_{2\varepsilon}\varepsilon\right)$$

turbulent viscosity

$$\mu_t = \rho f_\mu C_\mu \frac{k^2}{\varepsilon}$$

  - Damping functions are written in terms of Reynolds numbers:

$$Re_t = \frac{\rho k^2}{\mu\varepsilon} \quad ; \quad Re_y = \frac{\rho\sqrt{k}\,y}{\mu} \quad ; \quad Re_\varepsilon = \frac{\rho(\mu\varepsilon/\rho)^{1/4} y}{\mu}$$

define/models/viscous/

- turbulence-expert/low-re-ke-index
  - Select which low-Reynolds-number k-ε model is to be used among the six models available.

low-re-ke-index : 0 ➔ Abid

$$f_\mu = \tanh(0.008 Re_y)\left(1 + 4 Re_t^{-3/4}\right)$$

$$f_1 = 1$$

$$f_2 = \left[1 - \frac{2}{9}\exp\left(-\frac{Re_t^{\,2}}{36}\right)\right]\left[1 - \exp\left(\frac{Re_y}{12}\right)\right]$$

low-re-ke-index : 1 ➔ Lam-Bremhorst

$$f_\mu = \left[1 - \exp(-0.0165 Re_y)\right]^2 (1 + 20.5 / Re_t)$$

$$f_1 = 1 + (0.05 / f_\mu)^3$$

$$f_2 = 1 - \exp(-Re_t^2)$$

define/models/viscous/

- turbulence-expert/low-re-ke-index

low-re-ke-index : 2 ➔ Launder-Sharma

$$f_\mu = \exp\left[\frac{-3.4}{(1+\mathrm{Re}_t/50)^2}\right]$$

$$f_1 = 1$$

$$f_2 = 1 - 0.3\exp(-Re_t^2)$$

low-re-ke-index : 3 ➔ Yang-Shih

$$f_\mu = \sqrt{1-\exp(-1.5\times10^{-4}\,\mathrm{Re}_y - 5\times10^{-7}\,\mathrm{Re}_y^3 - 10^{-10}\,\mathrm{Re}_y^5)}\left(1+\mathrm{Re}_t^{-2}\right)$$

$$f_1 = 0.95 + 0.05\frac{G_k}{\rho\varepsilon}$$

$$f_2 = 1/\left(1+\mathrm{Re}_t^{-2}\right)$$

define/models/viscous/

- turbulence-expert/low-re-ke-index

low-re-ke-index : 4 ➔ Abe-Kondoh-Nagano

$$f_\mu = \left[1-\exp(-\mathrm{Re}_e/14)\right]^2\left(1+5\,\mathrm{Re}_t^{-0.75}\right)\exp\left[-2.5\times10^{-5}\,\mathrm{Re}_t^2\right]$$
$$f_1 = 1$$
$$f_2 = \left[1-\exp(-\mathrm{Re}_e/3.1)\right]^2\left[1-0.3\exp(-\mathrm{Re}_t^2/42.25)\right]$$

low-re-ke-index : 5 ➔ Chang-Hsieh-Chen

$$f_\mu = \left[1-\exp(-0.0215\,\mathrm{Re}_y)\right]^2\left(1+31.66\,\mathrm{Re}_t^{-1.125}\right)$$
$$f_1 = 1$$
$$f_2 = \left[1-0.01\exp(-\mathrm{Re}_t^2)\right]^2\left[1-0.0631\exp(-0.0631\,\mathrm{Re}_y)\right]$$

define/models/viscous/

- turbulence-expert/low-re-ke-index
  - Low-Reynolds model k-ε equations solved on fine mesh (required) right to the wall. The mesh resolution requirements are similar to those of the two-layer model.
  - y+ at the wall-adjacent cell should be of 1. Most ideally, One must have at least 10 cells within the viscosity-affected near-wall region

define/models/viscous/

- turbulence-expert/kato-launder-model?
  - This model accounts for vorticity magnitude in the production term of turbulent kinetic energy. It results in a reduced level of production close to stagnation points (e.g. impinging jets).

- turbulence-expert/kw-vorticity-base-production?
  - Production of turbulent kinetic energy is function of vorticity magnitude instead of strain rate magnitude. The main purpose is to prevent over production of turbulence close to stagnation point.
  - Damping functions are written in terms of Reynolds numbers:

define/models/viscous/

- rke-cmu-rotation-term?
  - The use of realizable k-ε model with multiple reference frame (MRF) model is not recommended.

$$C_\mu = \frac{1}{A_0 + A_s \frac{U^* k}{\varepsilon}} \qquad U^* \equiv \sqrt{S_{ij} S_{ij} + \tilde{\Omega}_{ij} \tilde{\Omega}_{ij}}$$

Absolute

$$\tilde{\Omega}_{ij} = \Omega_{ij} - 2\varepsilon_{ijk}\omega_k$$

Relative

$$\Omega_{ij} = \overline{\Omega}_{ij} - \varepsilon_{ijk}\omega_k$$

  - Where $\overline{\Omega}_{ij}$ is the mean rate-of-rotation tensor viewed in a rotating reference frame with the angular velocity ω_k .
  - When using this expression of C_μ with the MRFmodel, there is a jump in the turbulent viscosity ratio at the interface between the stationary and rotating fluid zones because of the ad hoc term $-2\varepsilon_{ijk}\omega_k$.
  - This option is provided for users who want to experiment with this model and modify the definition of C_μ by switching off the term $-2\varepsilon_{ijk}\omega_k$.

define/boundary-conditions/

- Modify-zone/extrude-face-zone-…
  - Ability to extrude a boundary face zone and extend the solution domain without having to exit the solver. A typical application of the extrusion capability is to extend the solution domain when recirculating flow is impinging on a flow outlet.
  - Current extrusion capability creates prismatic or hexahedral layers based on the shape of the face and normal vectors to the face zone's nodes.
  - New fluid zone is created
  - Implemented only in 3D.

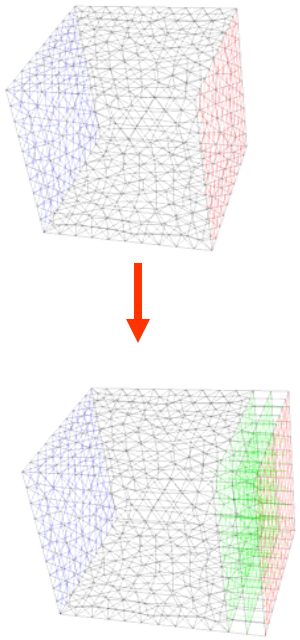
define/boundary-conditions/

- Two options available
  - Modify-zone/extrude-face-zone-delta
    - Extrude a face thread by specifying a list of displacements (in SI units).
  - Modify-zone/extrude-face-zone-para
    - Extrude a face thread by specifying a total distance (in SI units) and a list of parametric locations between 0 and 1 (e.g. 0., 0.2, 0.4, 0.8, 1.0).

define/boundary-conditions/

- modify-zone/extrude-face-zone-delta [example]
  - A cube (10x10x10) is extruded twice at the outlet by a distance of 1.

```
/define/boundary-conditions/modify-zones> extrude-face-zone-delta
Distance delta(1) [()] 1
Distance delta(2) [()] 1
Distance delta(3) [()]
Extrude face zone? [yes]
Moved original zone (outlet) to interior-7
Created new prism cell zone fluid-10
Created new prism cap zone pressure-outlet-11
Created new prism side zone wall-12
Created new prism interior zone interior-13
```



define/grid/

- modify-zone/make-periodic
  - Attempt to establish periodic/shadow zone connectivity.
- modify-zone/repair-periodic
  - Modify the grid to enforce a rotational angle or translation distance for periodic boundaries.
  - For translationally periodic boundaries, the command computes an average translation distance and adjusts the node coordinates on the shadow face zone to match this distance.
  - For rotationally periodic boundaries, the command prompts for an angle and adjusts the node coordinates on the shadow face zone using this angle and the defined rotational axis for the cell zone.

define/grid/

- modify-zone/slit-periodic
  - Slit periodic zone into two symmetry zones.
- modify-zone/slit-face-zone
  - Slit two-sided wall into two connected wall zones.

define/boundary-conditions/

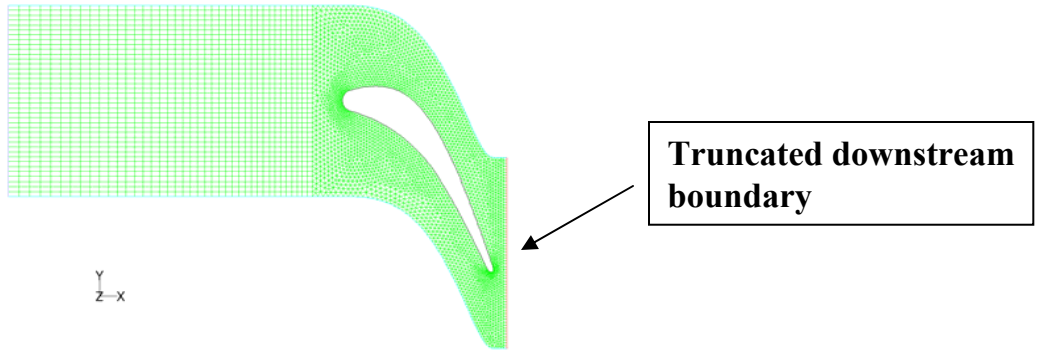
- non-reflecting-bc/
  - Enter the non-reflecting boundary conditions (NRBC) menu.
  - Available for pressure inlet/outlet BCs and mixing plane.
  - Implementation with a number of limitations
    - SS, coupled explicit solutions only
    - NRBC and NRMP faces require structured meshes
    - NRMP requires equal numbers of divisions equally spaced in the "spanwise" direction for each plane
    - Quad/hex layer adjacent to NRBC/MP highly recommended for 2D, required for 3D
    - To choose "non-reflecting" requires ALL pressure boundaries and mixing planes to be non-reflecting
    - ALL must meet requirements for NR option to be available

define/boundary-conditions/

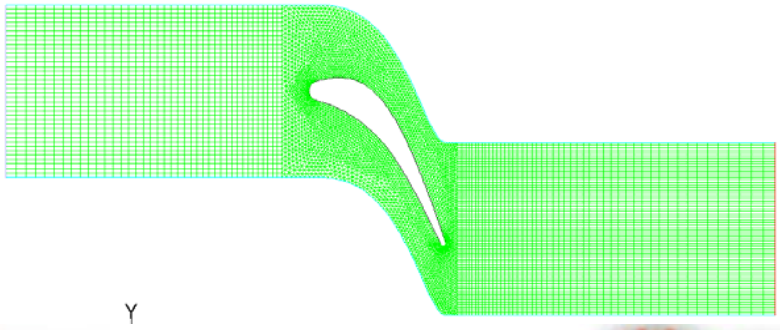
- non-reflecting-bc/

2D Turbine Vane

Short Domain



Long Domain

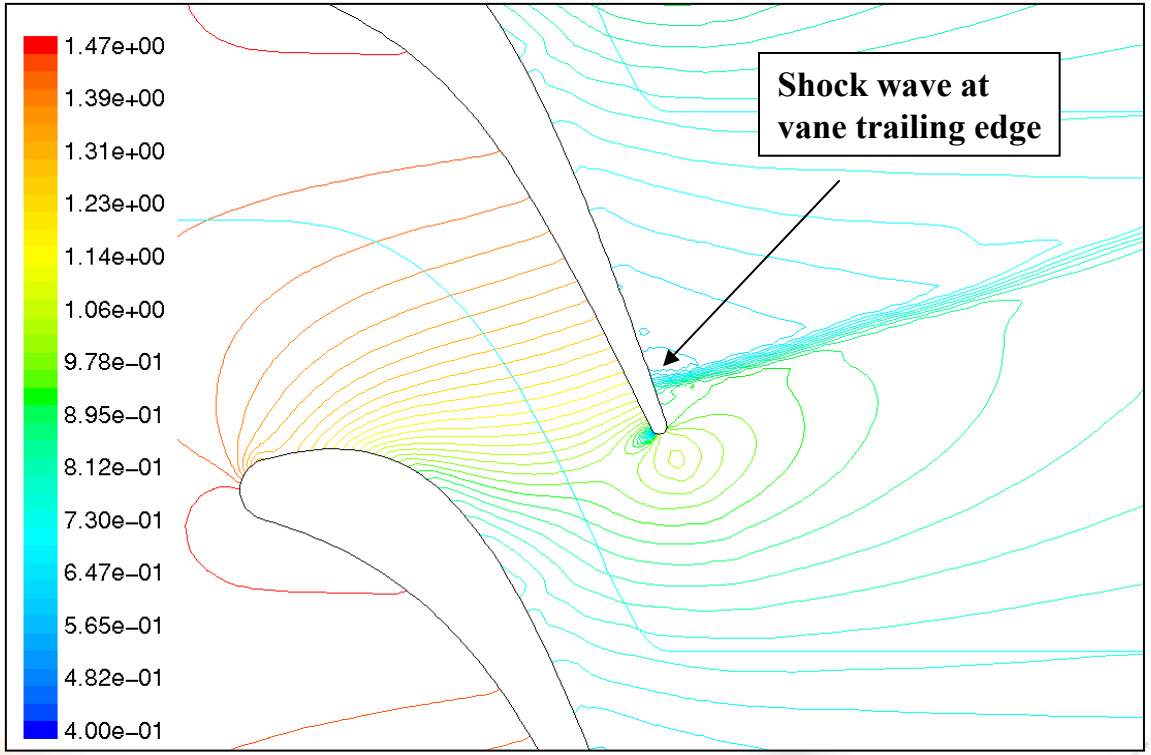


define/boundary-conditions/

- non-reflecting-bc/

Long Domain
Constant Pressure BC

Contours of Static Pressure (atm)

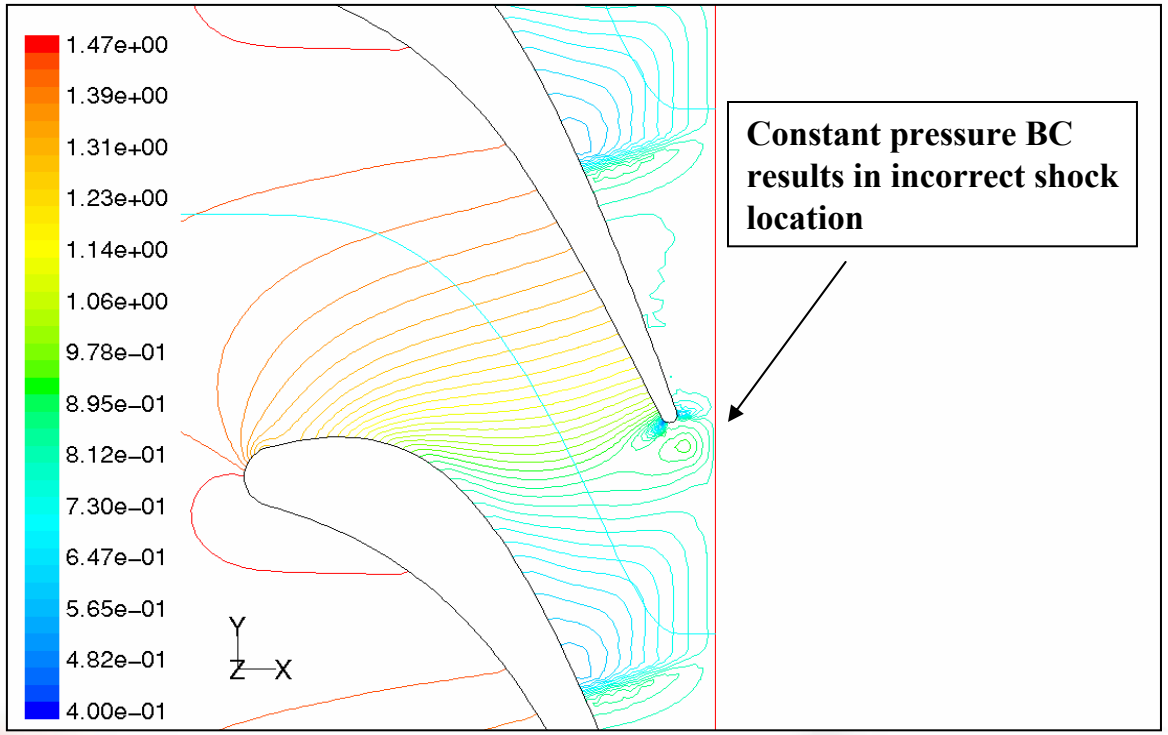


define/boundary-conditions/

- non-reflecting-bc/

Short Domain
Constant Pressure BC

Contours of Static Pressure (atm)

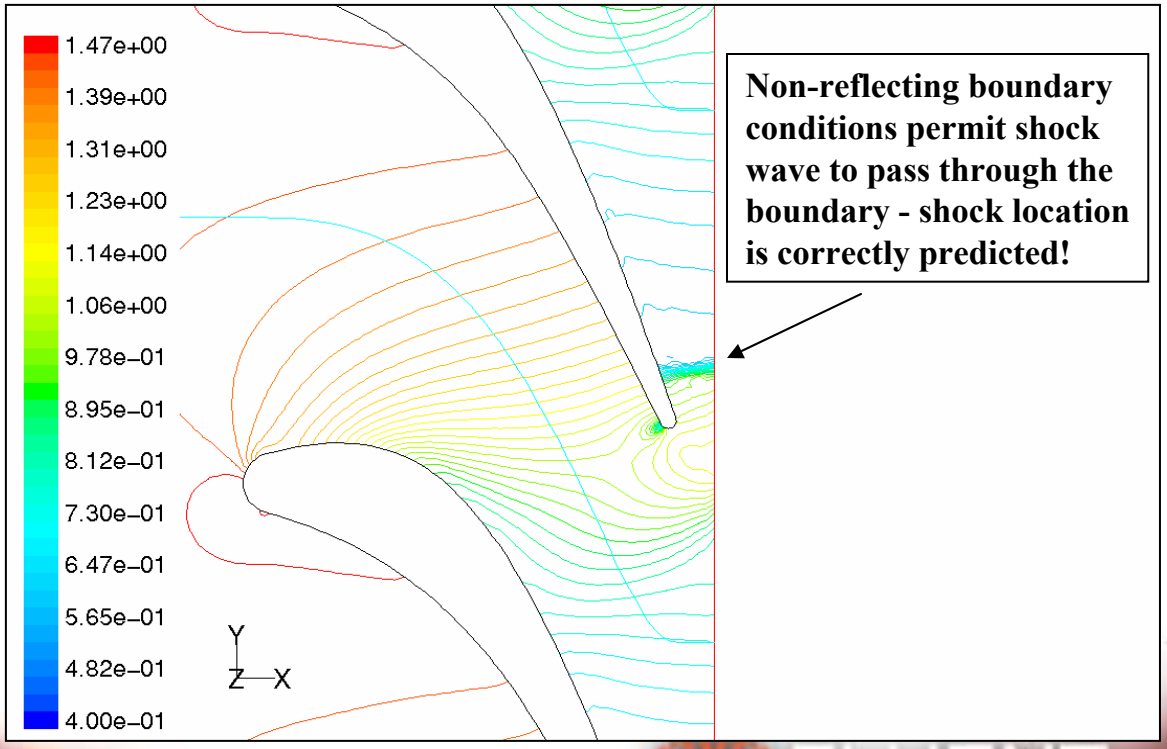


define/boundary-conditions/

- non-reflecting-bc/

Short Domain
Non-Reflecting BC

Contours of Static Pressure (atm)



define/boundary-conditions/

- mixing-planes/set/fix-pressure-level
  - For certain turbomachinery configurations, such as a torque converter, there is no fixed-pressure boundary when the mixing plane model is used.
  - The mixing plane model is usually used to model the three interfaces that connect the components of the torque converter. In this configuration, the pressure is no longer fixed. As a result, the pressure may float unbounded, making it difficult to obtain a converged solution.
  - FLUENT offers an option for fixing the pressure level and adjust the gauge pressure field after each iteration by subtracting from it the pressure value in the cell closest to the Reference Pressure Location in the Operating Conditions panel.
  - Only available for incompressible flows using the segregated solver

define/boundary-conditions/

- mixing-plane/set/conserve-swirl
  - Conservation of swirl is important for applications such as torque converters. Once the option is turned on, the user can ask the solver to report information about the swirl conservation during the calculation.
  - If verbosity? is turned on, FLUENT will report for every iteration:
    - The zone ID for the zone on which the swirl conservation is active
    - The upstream and downstream swirl integration per zone area,
    - The ratio of upstream to downstream swirl integration before and after the correction.

define/user-defined/

- real-gas/
  - The real gas model uses the National Institute of Standards and Technology (NIST) Thermodynamic and Transport Properties of Refrigerants and Refrigerant Mixtures Database Version 6.0 (REFPROP v6.0) to evaluate thermodynamic and transport properties.
  - Limitations of the real gas model
    - Used only with the coupled solvers.
    - All fluid property information in the Materials panel is ignored by FLUENT. User cannot override the REFPROP database for a real gas material by modifying parameters in the Materials panel.
    - All fluid zones must contain the real gas.
    - The real gas model does not support mixtures and/or saturated fluids.

define/user-defined/

- real-gas/
  - Activating the real gas model is a two-step process. First enable the real gas model, and then select a material from the REFPROP database.

```
> define/user-defined/real-gas
    use real gas? [no]  yes

CO2.fld    R123.fld   R142b.fld   R236fa.fld   butane.fld
R11.fld    R124.fld   R143a.fld   R245ca.fld   ethane.fld
R113.fld   R125.fld   R152a.fld   R245fa.fld   isobutan.fld
R114.fld   R13.fld    R22.fld     R32.fld      propane.fld
R115.fld   R134a.fld  R227ea.fld  R41.fld      propylen.fld
R116.fld   R14.fld    R23.fld     RC318.fld
R12.fld    R141b.fld  R236ea.fld  ammonia.fld

   select real-gas data file [""] "R125.fld"
```

```
/usr/local/Fluent.Inc/fluent6.0/realgas/lib/R125.fld

Opening "/usr/local/Fluent.Inc/fluent6.0/realgas/
ultra/librealgas.so"...
Setting material "air" to a real-gas...
matl name: "R125"
              : "pentafluoroethane"
              : "354-33-6"
Mol Wt      : 120.022
Critical properties:
  Temperature : 339.33 (K)
  Pressure    : 3.629e+06 (Pa)
  Density     : 4.75996 (mol/L) 571.3 (kg/m^3)
```

define/boundary-conditions/

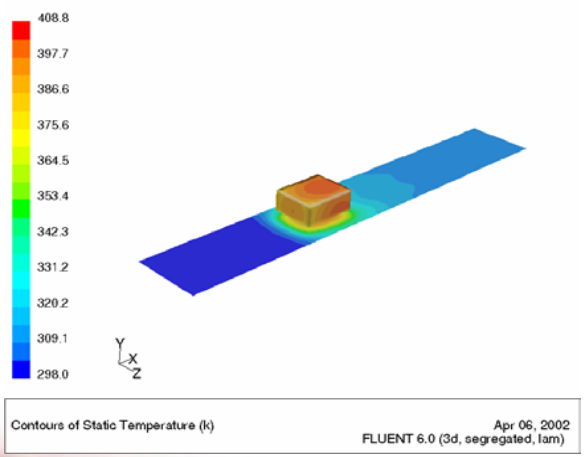
- mixing-plane/set/conserve-total-enthalpy
  - Global parameters such as efficiency are directly related to the change in total enthalpy across a blade row or stage.
  - Ensuring conservation of total enthalpy involves adjusting the downstream total temperature profile such that the integrated total enthalpy matches the upstream integrated total enthalpy.
  - If verbosity? is turned on, FLUENT will report for every iteration:
    - The zone ID for the zone on which the enthalpy conservation is active
    - The upstream and downstream heat fluxes,
    - The ratio of upstream to downstream heat fluxes.

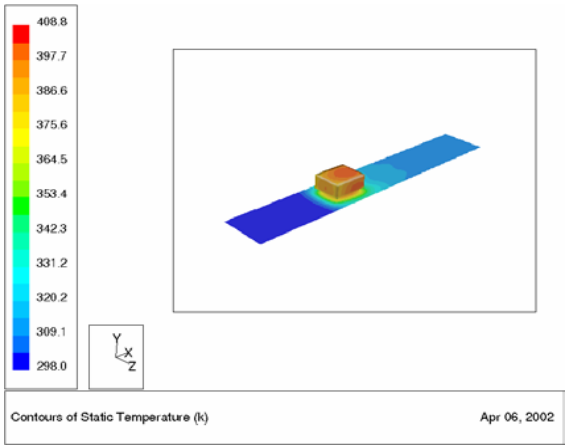
display/set/

- rendering-options/auto-spin?
  - Enable mouse view rotations to continue to spin the display after the button is released. To stop the spinning, click on the display window again.
  - It is recommended to use this option with the double buffering option turned on. Double buffering dramatically reduces screen flicker during graphics updates.
- reset-graphics
  - Reset the graphics system and kill all the display windows.
- windows/
  - Contains commands that allow user to customize the relative positions of sub-windows inside the active graphics window.

display/set/

- windows/… [example]
  - Add borders to the scale, the axis, the image.
  - Eliminate the application title
  - etc





file/

- write-bc
  - Save all currently-defined boundary conditions to a file, FLUENT will use the same format as the “zone”" section of the case file.
- read-bc
  - Read boundary conditions from a file and apply them to the corresponding zones in the model. FLUENT will set the boundary conditions in the current model by comparing the zone name associated with each set of conditions in the file with the zone names in the model.

    Note : If user wants to apply a set of conditions to multiple zones with similar names, the boundary-condition file saved with the write-bc can be edited to include “wildcards” (*) within the zone names (e.g. to apply a particular set of conditions to wall-12, wall-15, and wall-17 in the current model, edit the boundary-condition file and set the zone name associated with the desired conditions to wall-*).

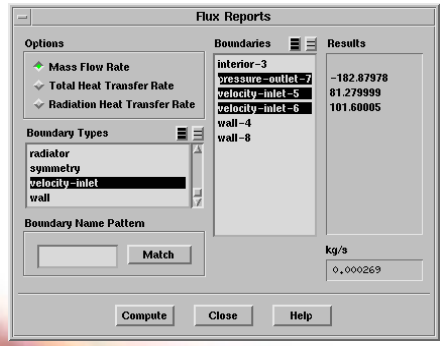
file/

- reread-grid
  - Given a case file with a particular grid, the user can use this option to merge a new grid with the existing boundary conditions, material properties, solution parameters, etc.
  - Useful situation: a better mesh is generated for the same problem and the user doesn't want to reenter all of the boundary conditions, properties, and parameters.
  - The new grid ***must*** have the same zone structure as the original grid.

report/

- species-mass-flow
  - Print list of species mass flow rate at inlets and outlets.
- uds-flow
  - Print list of user-defined scalar flow rate at boundaries.

Note: These options are more accurate than surface integrals at boundary zones because they do not involve cell interpolation. They are analogous to the report →fluxes in the GUI for mass and heat transfer rates.



solve/

- monitors/force/clear-all-monitors
  - Fast way to discard the internal and external file data associated with the force monitors.
- set/flow-warnings?
  - Specify whether or not to print warning messages when reversed flow occurs at inlets and outlets, and when mass flow inlets develop supersonic regions. By default, flow warnings are printed.
- set/numerics/implicit body force treatment?
  - For problems with dominant body forces, this option allows for a linearization of body force source terms.

solve/

- set/expert/use conservative form of energy equation?
  - FLUENT solves the energy equation in its conservative form

$$\frac{\partial}{\partial t}(\rho E)+\nabla\cdot\left(\vec{v}(\rho E+p)\right)=\nabla\cdot\left(k_{eff}\nabla T-\sum_{j}h_{j}\vec{J}_{j}+\left(\overline{\overline{\tau}}_{eff}\cdot\vec{v}\right)\right)+S_{h}$$

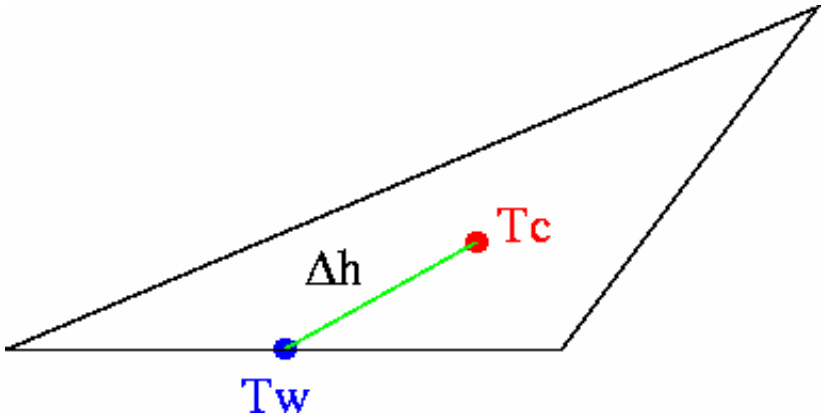
  - Which is consistent with the finite volume conservative form of governing equations.

$$\frac{\partial}{\partial t}(\rho\phi)+\nabla\cdot\left(\rho\vec{v}\phi\right)=\nabla\cdot\left(D_{eff}\nabla\phi\right)+S_{\phi}$$

  - If the user wants to solve the energy equation in non-conservative form this option should be switched off (not recommended)

solve/

- set/expert/use alternate formulation for wall temperature? [no]
  - FLUENT models the flux at the wall as follows:



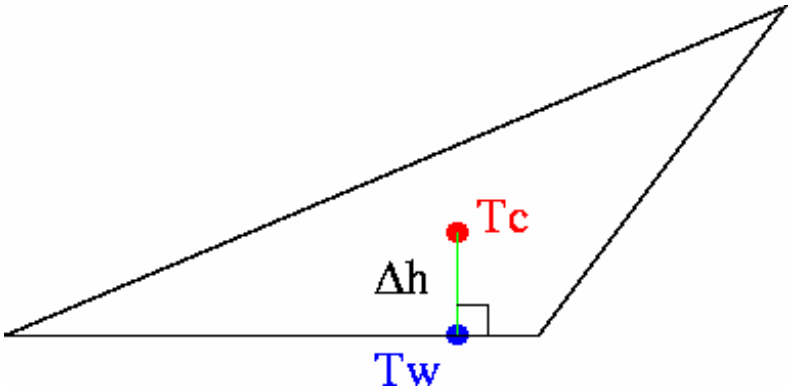
$$q = k\nabla T \cdot \vec{n}$$

$$q = k\frac{(T_w - T_c)}{\Delta h} + f(\Delta T)$$

  - The term $f(\Delta T)$ includes second order terms that need to be determined, which involves approximations.

solve/

- set/expert/use alternate formulation for wall temperature? [yes]
  - Assume face center is defined such as the vector between cell center and face center is perpendicular to the wall.



$$q = k\nabla T \cdot \vec{n}$$

$$q = k\frac{(T_w - T_c)}{\Delta h}$$

  - The term $f(\Delta T)$ vanishes and Δh changes.
  - This option doesn't impact the results if the wall cells are not skewed.

solve/

- set/relaxation-method?
  - By default FLUENT uses the Gauss-Seidel relaxation method ("gauss-seidel").
  - With Gauss-Seidel, each cell uses the updated value of the neighboring cells. This yields differences between the serial and parallel solvers.
  - If the parallel solver partitions are visited in an order that is different form the way the serial solver sweeps the domain, the results can differ from one iteration to another but will ultimately converge to the same solution.
  - If the Jacobi approach is used, the parallel and serial solvers will give the exact same solution at every iteration because the Jacobi approach always uses values from the previous iteration ("jacobi").
  - This option is used for testing purposes. It is recommended to use Gauss-Seidel method.

Part IV : Case Study

The Segregated Solver with High Pressure Differential.

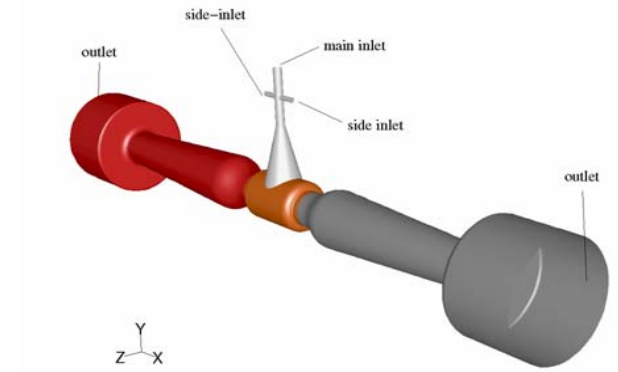
How to use the segregated solver when there are large pressure variations between the inlets and outlets ?

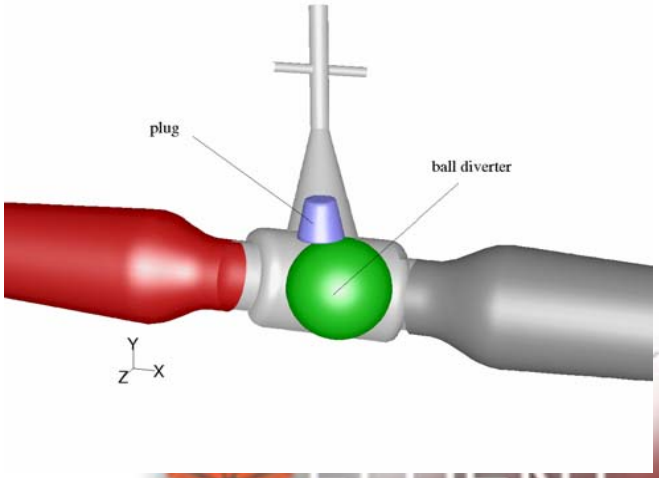
A case study for a ball diverter controlled valve

Valve controlled by ball diverter

Description

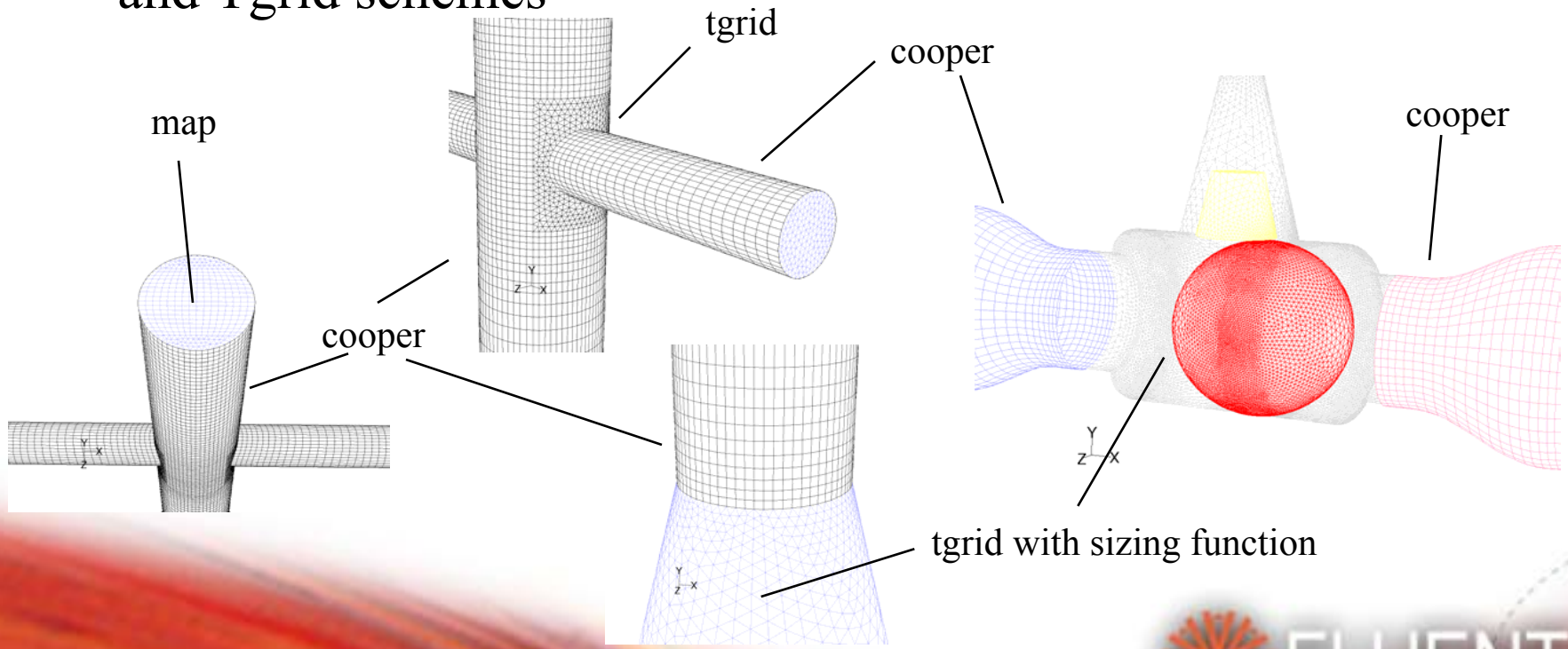
- Flow to the left and right channels is controlled by the position of the ball diverter
- Very high pressure at the inlets (~500 psi) and low pressures at the outlets (~0.01 psi)
- Large pressure variation within the domain





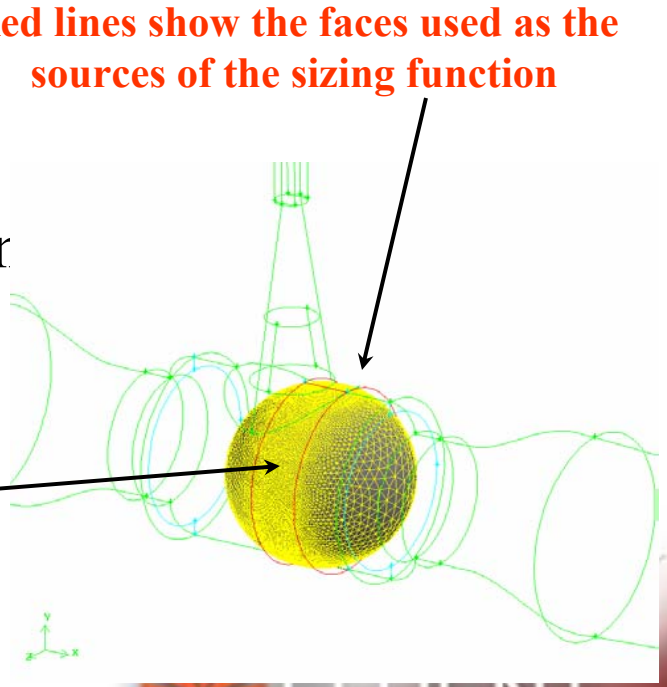
Valve controlled by ball diverter
Mesh Issue

- Hybrid mesh was created in Gambit using the map, cooper, and Tgrid schemes



Valve controlled by ball diverter
Mesh Issue

- Mesh count is 1.09M with low skewness
  - Cell equi-angle skew: [0, 0.84]
  - Cell equi-volume skew: [0, 0.79]
- High quality tetrahedral elements with appropriate clustering of cells are created ir Gambit using the sizing function tools



Valve controlled by ball diverter
Setup Parameters

- Inlet boundaries
  - Main inlet is specified as pressure-inlet type
    - Total pressure is 500 psi and total temperature is 2000 K
    - Flow is subsonic, set the static pressure to 499.5 psi
  - Side inlets are specified as Massflow-inlet type
    - Massflow rates are 7e-5 kg/s and 2e-5 kg/s
      - Correspond to velocities of 33.4 m/s and 9.8 m/s respectively
    - Total temperature is 2000 K
    - Flow is subsonic, set the static pressure to 499.5 psi
- Outlet boundaries
  - Both outlet boundaries are specified as pressure-outlet type
    - Static pressure is 0.01 psi
    - Backflow total temperature is 2000K

Valve controlled by ball diverter
Solution Procedure

- Case is very difficult to start and to converge due to large pressure variations in the domain
- Convergence difficulties are attributed to:
  - Difficult to specify good initial conditions
  - Patching is also difficult since there may be compression/expansion structures inside the domain
  - Difficult to setup appropriate solution limits due to large variations in pressure
- Will lowering URFs work ?
  - Using very low URFs at the beginning may work but it may be impractical since very low values may be needed

Valve controlled by ball diverter
Solution Procedure

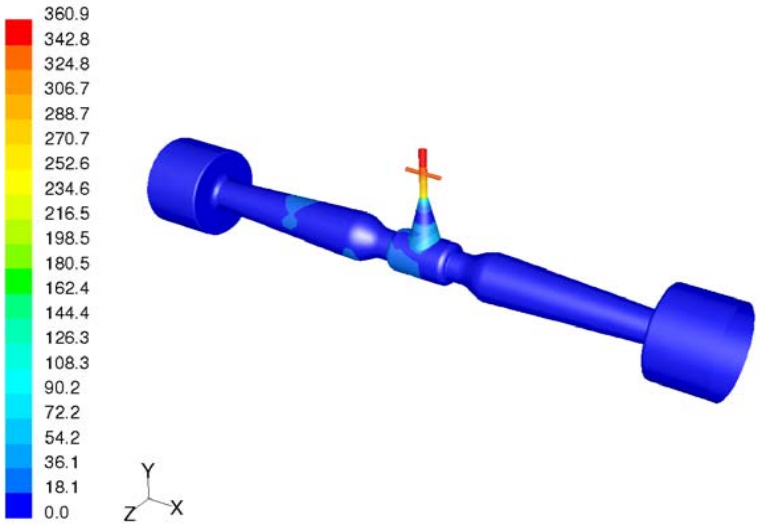
- Set the static pressure at the outlets to values slightly lower than the inlet values
  - Set appropriate solution limits which reflects the modified condition
- Initialize using the inlet condition but set velocities to zero
- Iterate until sufficient first order convergence
  - No backflow at the main inlet
- Gradually lower the outlet static pressure values and iterate until sufficient convergence at each step
  - Adjust the solution limits to reflect the updated outlet pressure values
- Save the case/data files as necessary

Valve controlled by ball diverter
Miscellaneous

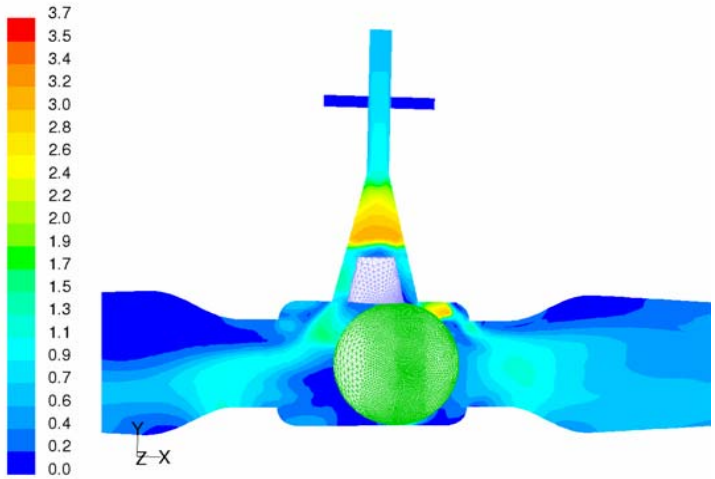
- Other tips:
  - Change to second order only after sufficient convergence using first order with the desired pressure values at the outlets
  - Density discretization needs to be changed to second-order to resolve shocks accurately, if any
  - Specify Massflow inlet instead of pressure-inlet at the main inlet
    - Prevent backflow at the inlet during the initial stages of the iteration
  - Use the PISO algorithm to handle the velocity-pressure coupling
    - Will reduce the pressure oscillations between the high pressure inlets and low pressure outlets

Valve controlled by ball diverter

Results



Static pressure



Mach number

Valve controlled by ball diverter
Practical Issues

- Stable solution procedure requires setting a high static pressure values at the outlets and then gradually reduces the values in steps until the desired value is achieved
- This manual adjustment of the static pressure values at the outlet boundaries can be time consuming and an automated procedure is desirable
- Once a good experience of how fast to reduce and how many to iterate at each step is obtained, the knowledge can be used to write a UDF to automate the whole process

Valve controlled by ball diverter
Practical Issues

- The UDF requires the following inputs:
  - Number of iterations to be performed before each update (iterP)
  - Change of pressure value at each update (deltaP)
  - Impose limit on the minimum pressure to prevent oscillations (Plimit)
- The sample UDF given in the next slide works only in serial and has not been parallelized yet.

Valve controlled by ball diverter

```
#include "udf.h"

#define PI       3.14159265
#define Factor 1.0

#define zoneID 3
#define iterP  50
#define deltaP 5000.0
#define Plimit 1000.0

int   iter=0;
float Pupdate=3446083.0;

DEFINE_ADJUST(Padjust,domain)
{
  face_t    f;
  Thread *tf;
  float     sumA,sumP,phif,Amag,Pavg,
            A[ND_ND];
```

(1)

```
  tf = Lookup_Thread(domain,zoneID);

  sumA = 0.0;
  sumP = 0.0;

  begin_f_loop(f,tf)
    {
      phif = F_P(f,tf);

      F_AREA(A,f,tf);
      Amag = Factor*NV_MAG(A);

      sumA += Amag;
      sumP += phif*Amag;
    }
  end_f_loop(f,tf)

  Pavg = sumP/sumA;
```

(2)

Valve controlled by ball diverter

```
  iter += 1;

  Message("Current pressure is %e\n",Pavg);

  if ( (iter%iterP)==0 )
    {
      Pupdate = Pavg - deltaP;
      if ( Pupdate<Plimit ) Pupdate = Plimit;
      Message("Update pressure to %f\n",Pupdate);
    }
}

DEFINE_PROFILE(prsbc,tf,pos)
{
  face_t f;

  begin_f_loop(f,tf)
    {F_PROFILE(f,tf,pos) = Pupdate;}
  end_f_loop(f,tf)
}
```

(3)