

Workshop Appendix A: Scripting and Batch Processing

16.0 Release

A visualization of fluid dynamics showing blue, wavy, semi-transparent surfaces that resemble smoke or liquid flow, set against a light yellow background.

Fluid Dynamics

A 3D model of a purple gear with a glowing white and purple center, surrounded by other faint gear shapes, representing structural mechanics.

Structural Mechanics

A series of concentric green and white circles, resembling a target or a cross-section of an electromagnetic field, set against a light yellow background.

Electromagnetics

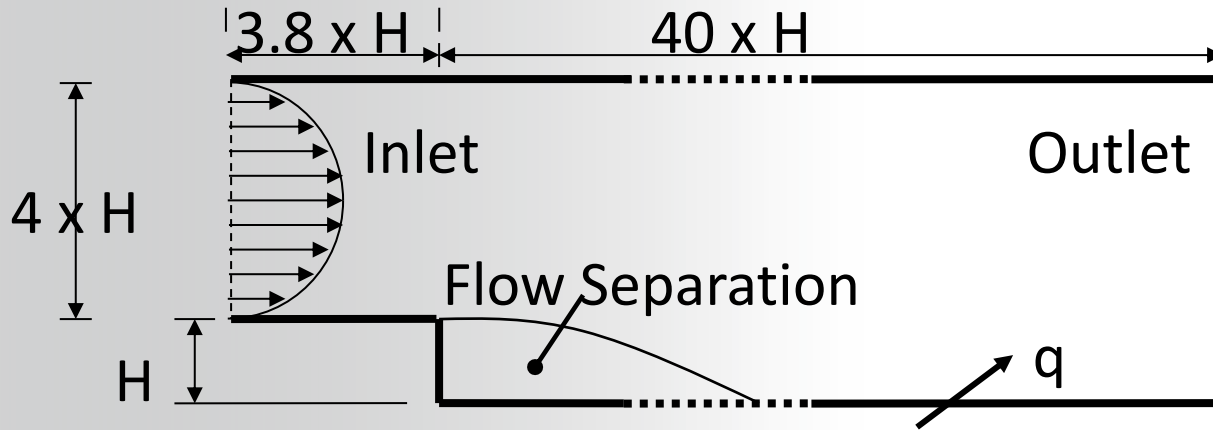
A 3D arrangement of teal and black rectangular blocks, some stacked and some floating, with light rays emanating from them, representing systems and multiphysics.

Systems and Multiphysics

Introduction to ANSYS CFX

Introduction

This workshop models flow over a backwards facing step with heat transfer through the lower wall. The quantities of interest are the Skin Friction Coefficient and the Stanton Number on the lower wall. The choice of turbulence model can influence these results. So you will use session files and scripts to run three simulations, each with a different turbulence model, and then compare the results.

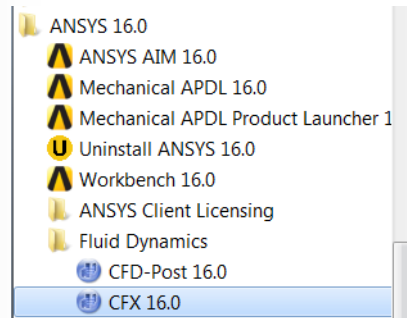


Overview of Steps

A CCL file containing the initial physics definition is provided. You will use this to create a definition file. The same definition file will be used to run all three simulations with additional CCL passed to the solver to alter the turbulence model. You will write a short script to run the three simulations.

You will create a *CFD-Post* session file to generate post-processing output.

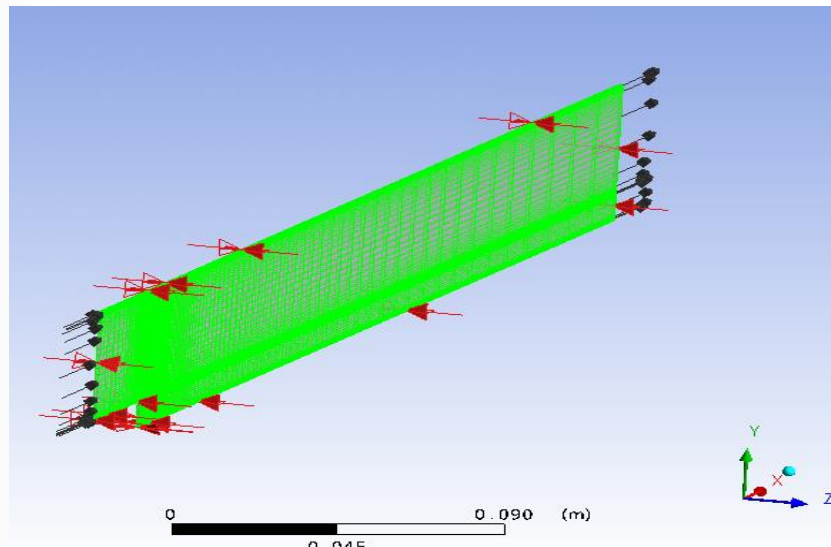
This workshop requires you to work outside *ANSYS Workbench*, using the standalone *ANSYS CFX* launcher.



First Simulation (k-epsilon)

- In the *CFX-Launcher* browse to your *Working Directory* & start *CFX-Pre*
- Create a new case
 - Import the mesh, *backstep.gtm* (the type is *CFX-Mesh*)
 - Import the CCL file, *ke.ccl*

This CCL file was generated by setting up the simulation in *CFX-Pre* and then exporting the CCL, *File > Export CCL*



Examining the Setup/Writing Solver File

Take a minute to look at the simulation set-up

- 1D Interpolation *User Functions* have been used to define inlet velocity and turbulence profiles based on experimental data
- The mesh is 1 element thick with symmetry boundaries on the X-Y planes. This simplifies the simulation to 2D
- There is a boundary named *HeatedWall* through which a constant *Heat Flux* is applied
- The k-epsilon turbulence model is used

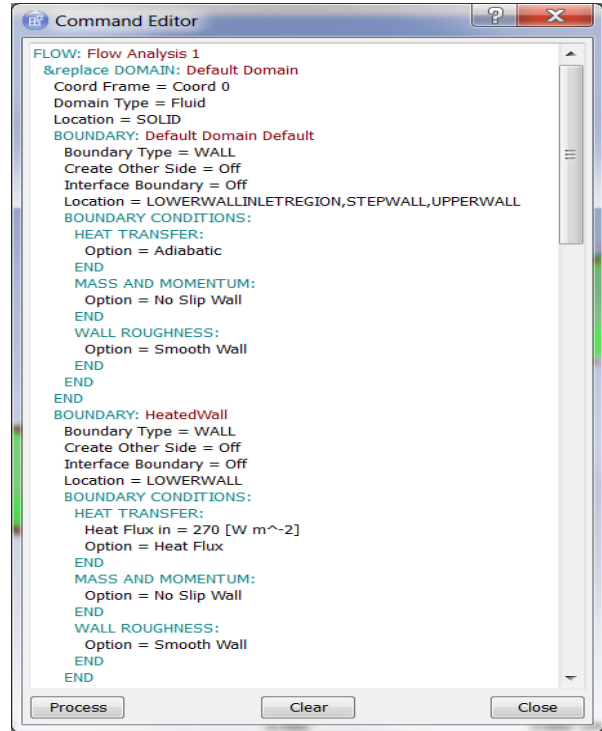
The second and third simulations will use the SST and the k-omega turbulence models

- Write Solver Input File to *ke.def*

Preparing CCL Files

Next you will prepare CCL files for different turbulence models using the existing CCL as a template. One way to extract the existing CCL is through the *Command Editor* in *CFX-Pre*.

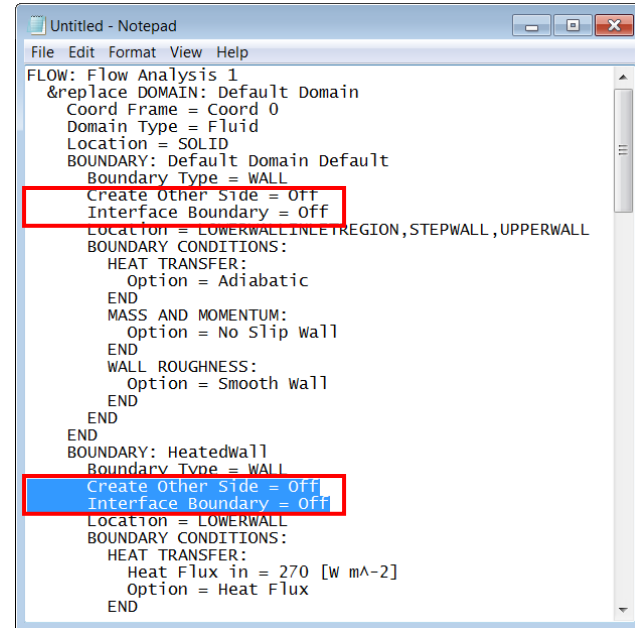
- Open a new text file in Notepad
- In *CFX-Pre* right-click on *Default Domain* in the *Outline* tree and select *Edit in Command Editor*
- Copy and paste all the text from the *Command Editor* to your text file



```
Command Editor
FLOW: Flow Analysis 1
&replace DOMAIN: Default Domain
Coord Frame = Coord 0
Domain Type = Fluid
Location = SOLID
BOUNDARY: Default Domain Default
Boundary Type = WALL
Create Other Side = Off
Interface Boundary = Off
Location = LOWERWALLINLETREGION,STEPWALL,UPPERWALL
BOUNDARY CONDITIONS:
HEAT TRANSFER:
Option = Adiabatic
END
MASS AND MOMENTUM:
Option = No Slip Wall
END
WALL ROUGHNESS:
Option = Smooth Wall
END
END
BOUNDARY: HeatedWall
Boundary Type = WALL
Create Other Side = Off
Interface Boundary = Off
Location = LOWERWALL
BOUNDARY CONDITIONS:
HEAT TRANSFER:
Heat Flux in = 270 [W m^-2]
Option = Heat Flux
END
MASS AND MOMENTUM:
Option = No Slip Wall
END
WALL ROUGHNESS:
Option = Smooth Wall
END
END
Process Clear Close
```

Preparing CCL Files (SST)

- For the Wall boundaries, *Default Domain Default* and *HeatedWall*, delete the lines:
 - ‘Create Other Side = Off’
 - ‘Interface Boundary = Off’
- Save the text file in your working directory, naming it SST.ccl



```
Untitled - Notepad
File Edit Format View Help
FLOW: Flow Analysis 1
&replace DOMAIN: Default Domain
Coord Frame = Coord 0
Domain Type = Fluid
Location = SOLID
BOUNDARY: Default Domain Default
Boundary Type = WALL
Create Other Side = Off
Interface Boundary = Off
Location = LOWERWALL,INLETREGION,STEPWALL,UPPERWALL
BOUNDARY CONDITIONS:
HEAT TRANSFER:
Option = Adiabatic
END
MASS AND MOMENTUM:
Option = No Slip wall
END
WALL ROUGHNESS:
Option = Smooth Wall
END
END
BOUNDARY: HeatedWall
Boundary Type = WALL
Create Other Side = Off
Interface Boundary = Off
Location = LOWERWALL
BOUNDARY CONDITIONS:
HEAT TRANSFER:
Heat Flux in = 270 [W m^2]
Option = Heat Flux
END
```

Preparing CCL Files (SST)

- **Edit the 'TURBULENCE MODEL' Option and the 'TURBULENT WALL FUNCTIONS' Option located at the bottom of the file as shown:**

```
TURBULENCE MODEL:  
  Option = k epsilon  
END  
TURBULENT WALL FUNCTIONS:  
  Option = Scalable  
END
```



```
TURBULENCE MODEL:  
  Option = SST  
END  
TURBULENT WALL FUNCTIONS:  
  Option = Automatic  
END
```

- **Save the changes to SST.ccl**

If you do not know the correct CCL syntax, make changes in the *CFX-Pre* GUI and then edit the object in the *Command Editor* to view the syntax.

Preparing CCL Files (k-omega)

- For the k-omega model, edit the **TURBULENCE MODEL Option:**

```
TURBULENCE MODEL:
```

```
Option = SST
```

```
END
```

```
TURBULENT WALL FUNCTIONS:
```

```
Option = Automatic
```

```
END
```

```
TURBULENCE MODEL:
```

```
Option = k omega
```

```
END
```

```
TURBULENT WALL FUNCTIONS:
```

```
Option = Automatic
```

```
END
```



- Save the file as **komega.ccl**

The subdirectory called `scripts` contains copies of `komega.ccl` and `SST.ccl`. You can use these files if necessary.

Do not copy and paste from Powerpoint as there could be hidden characters in the formatting.

Create a Solver Script

The next step is to create a script that will run all the simulations in the solver. Some options for the scripting language are Perl, a Windows batch script (.bat) or a UNIX shell script (.sh). You will use Perl, which:

- can be run on Windows and UNIX/Linux platforms
- is built into your CFX installation and is integrated into CCL

Learning some basic Perl will allow you to add advanced features to CCL. You will see an example of this when post-processing this workshop.

Create a Solver Script

- **Open a new text file in Notepad and save it as 'runsolver.pl'**
- **Enter the following text**

```
#!/ perl -w

print "Running the k-epsilon simulation\n";
system "cfx5solve -def ke.def";

print "Running the SST simulation\n";
system "cfx5solve -def ke.def -ccl SST.ccl -cont-from-file ke_001.res -name SST";

print "Running the k-omega simulation\n";
system "cfx5solve -def ke.def -ccl komega.ccl -cont-from-file ke_001.res -name komega";
```

- **Save the changes to runsolver.pl**

The file is also provided in the scripts subdirectory with the workshop.

Notes on the Perl Script

The following provides a brief explanation of the syntax used in the Perl script:

- The first two lines provide information on how Perl should interpret the script.
- The details are not necessary here, but you can start all your Perl scripts with these two lines
- # is the comment character
- 'system' executes the command in quotes
- Each statement should finish with the ; character

Notes on the Perl Script

The Perl script runs the solver three times. The first time the k-epsilon simulation is run using the definition file, ke.def. The second and third time the following additional arguments are provided:

`-ccl <file>.ccl:`

This passes the CCL file to the solver. If a parameter is defined more than once, the last CCL to be processed takes precedence

`-cont-from-file <file>:`

Uses the named results to initialize the run

`-name <name>:`

This sets the name of the .out and .res files output by the solver.

Running the Script

- In the *CFX-Launcher* check that the **Working Directory** is the directory containing the files you have been working with.
- Select *Tools > Command Line*
- Type *perl runsolver.pl* and press Enter

While the simulations are running you can track their progress by opening the *CFX-Solver Manager*, selecting *File > Monitor Run in Progress* and selecting the appropriate *_00x.dir* directory.

Starting the Command Line from the *CFX-Launcher* is recommended:

- The current directory is set to the *CFX-Launcher Working Directory*
- Several CFX environment variables are set, including paths to commands such as *cfx5solve*

Session Files in CFD-Post

Once the runs have finished you will create session files in CFD-Post. These record actions performed so that they can be replayed later.

- **Start *CFD-Post* from the *CFX-Launcher* but do not load results**
- **Create a new CFD-Post session file, *Session > New Session***
 - Save the session file to post.cse
- **Select *Session > Start Recording***
- **Select *File > Load Results***
 - Load the results file ke_001.res
- **Select *Session > Stop Recording***

The session file now contains CCL to load the results file into *CFD-Post*. You will use this later to load the other files in batch mode

Session Files in CFD-Post


- Create a vector plot of *Velocity* on *Sym1*
- Examine *Temperature* on *Sym1* with a *User Specified Range* of 293[K] to 1500[K]
 - Note the hot pocket of temperature in the recirculation zone
- Select *Session > Start Recording* to begin recording commands again
- Select *Insert > Location > Polyline*, and accept the default name *Polyline 1*
- Choose *Boundary Intersection* as the *Method* and in the *Boundary List* pick *HeatedWall*
- For *Intersect With* select *Sym1* and then click *Apply*

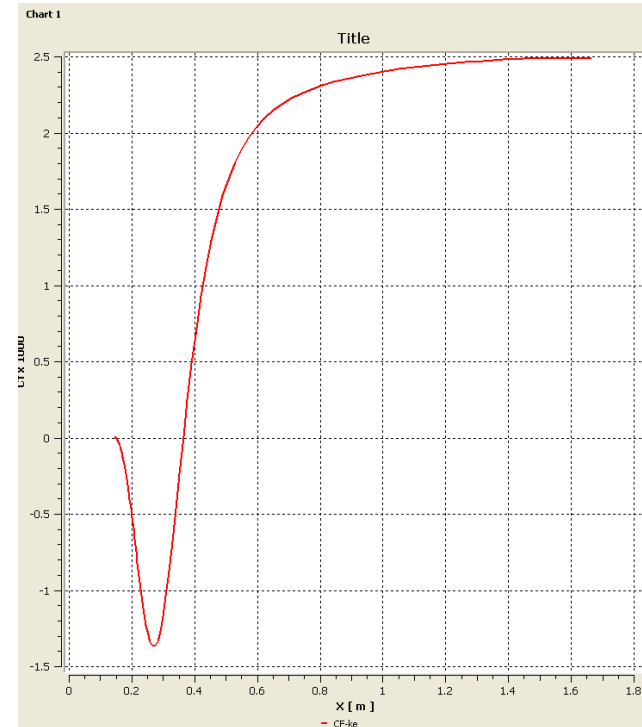
Create new Variables

Next create new variables for the Skin Friction Coefficient (Cf) and the Stanton Number (St). Both will be multiplied by 1000 for a more sensible scale. You will then chart these variables along the polyline.

- ***Insert > Variable***
- **Set the *Name* to 'Cf x 1000'**
- **Make sure that *Method* = *Expression* and in the *Expression* box enter:**
$$1000 * \text{Wall Shear X} / (0.5 * \text{Density} * (\text{massFlowAve(Velocity)}@In^2))$$
- **Click *Apply* (Do not enable *Calculate Global Range*)**
- **Create a variable called 'St x 1000' equal to $1000 * \text{Wall Heat Transfer Coefficient} / (\text{massFlowAve(Velocity)}@In * \text{Density} * \text{Specific Heat Capacity at Constant Pressure})$**

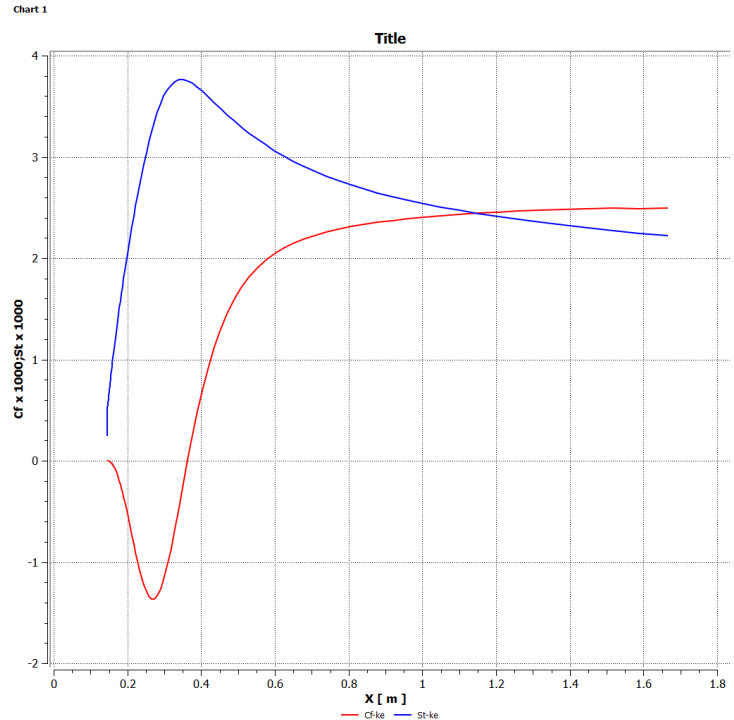
Create a Chart

- ***Insert > Chart***, accept default name ***Chart 1***
- Switch to the ***Data Series*** tab
- Set the ***Name*** to ***Cf-ke***
- Set the ***Location*** to ***Polyline 1***
- Toggle ***Custom Data Selection***
- Set the ***X Axis Variable*** to ***X***
- Set the ***Y Axis Variable*** to ***'Cf x 1000'***
- Click ***Apply***
- Click the ***New*** button 
- Set the ***Name*** to ***'St-ke'***



Create a Chart (contd)

- Set the *Location* to *Polyline 1*
- Toggle *Custom Data Selection*
- Set *X Axis Variable* to *X*
- Set *Y Axis Variable* to *St x 1000*. *Apply*
- Select the *Export* button
- Set the name to 'ChartKE.csv'. *Save*
- Select *File > Close*. Do not save the state
- Select *Session > Stop Recording*



Saving the Session File

The session file now contains commands that:

- Open a results file
- Create a polyline and charts
- Export data
- Close the results file

The next step is to edit the session file to for running in batch mode. A discussion of all the commands appears at the end of the workshop.

If you have problems, you can look at the post.cse file provided in the 'scripts' directory

Editing the Session File

- Open your *Session* file *post.cse* in a text editor.
- Insert/edit the text highlighted in red. This will loop over all three sets of results.
- Delete `'force_reload=true'`

```
COMMAND FILE:
  CFX Post Version = 16.0
END
!foreach $res ('ke_001.res','SST_001.res','komega_001.res')
!{
! print "Processing $res\n";
! @temp = split('_001.res',$res);
! $type = $temp[0];

DATA READER:
  Clear All Objects = false
.
.
.
END
DATA READER:
  Domains to Load=
END
> load filename=$res ,force_reload=true
```

Editing the Session File

- **Continue to make the following changes. You will need to scroll down to find these areas. This will set the Line Name for the Charts based on the results file being processed**

```
CHART SERIES:Series 1
```

```
Chart Line Custom Data Selection = On
```

```
Chart Line Filename =
```

```
Chart Series Type = Regular
```

```
Chart X Variable = X
```

```
Chart Y Variable = Cfx1000
```

```
Histogram Y Axis Weighting = None
```

```
Location = /POLYLINE:Polyline 1
```

```
Series Name = Cf-$type
```

```
CHART LINE:Chart Line 1
```

```
Auto Chart Line Colour = On
```

```
Case Name =
```

```
.
```

```
.
```

```
.
```

```
Chart X Variable = X
```

```
Chart Y Variable = Cf x 1000
```

```
Line Name = Cf-$type
```

Editing the Session File

CHART SERIES:Series 2

```
Chart Line Custom Data Selection = On
Chart Line Filename =
Chart Series Type = Regular
Chart X Variable = X
Chart Y Variable = Stx1000
Histogram Y Axis Weighting = None
Location = /POLYLINE:Polyline 1
Series Name = St-$type
```

CHART LINE:Chart Line 2

```
Auto Chart Line Colour = On
Chart Line Colour = 1.0, 0.0, 0.0
Chart Line Style = Automatic
Chart Line Visibility = On
Chart Symbol Colour = 0.0, 1.0, 0.0
Chart Symbol Style = None
Fill Area = On
Fill Area Options = Automatic
Is Valid = True
Line Name = St-$type
```

Editing the Session File

You will add commands to evaluate print out some quantities, namely:

- The highest temperature in the domain
 - The average skin friction coefficient on the polyline
 - The average Stanton number on the polyline
- **Insert the text below after the END statement for the Chart, near the bottom of the file, before the start of the EXPORT object:**

```
END
```

```
! $maxtemp = maxVal("Temperature", "Default Domain");
```

```
! $saveCf = lengthAve("Cf x 1000", "Polyline 1");
```

```
! $saveSt = lengthAve("St x 1000", "Polyline 1");
```

```
! printf("For $type model, Highest Temp in domain is %.0f, average Cf is %.2f, average St is  
%.2f\n", $maxtemp, $saveCf, $saveSt);
```

```
EXPORT :
```


Editing the Session File

- **Make the following changes to set a filename for each exported csv file and close the foreach loop:**

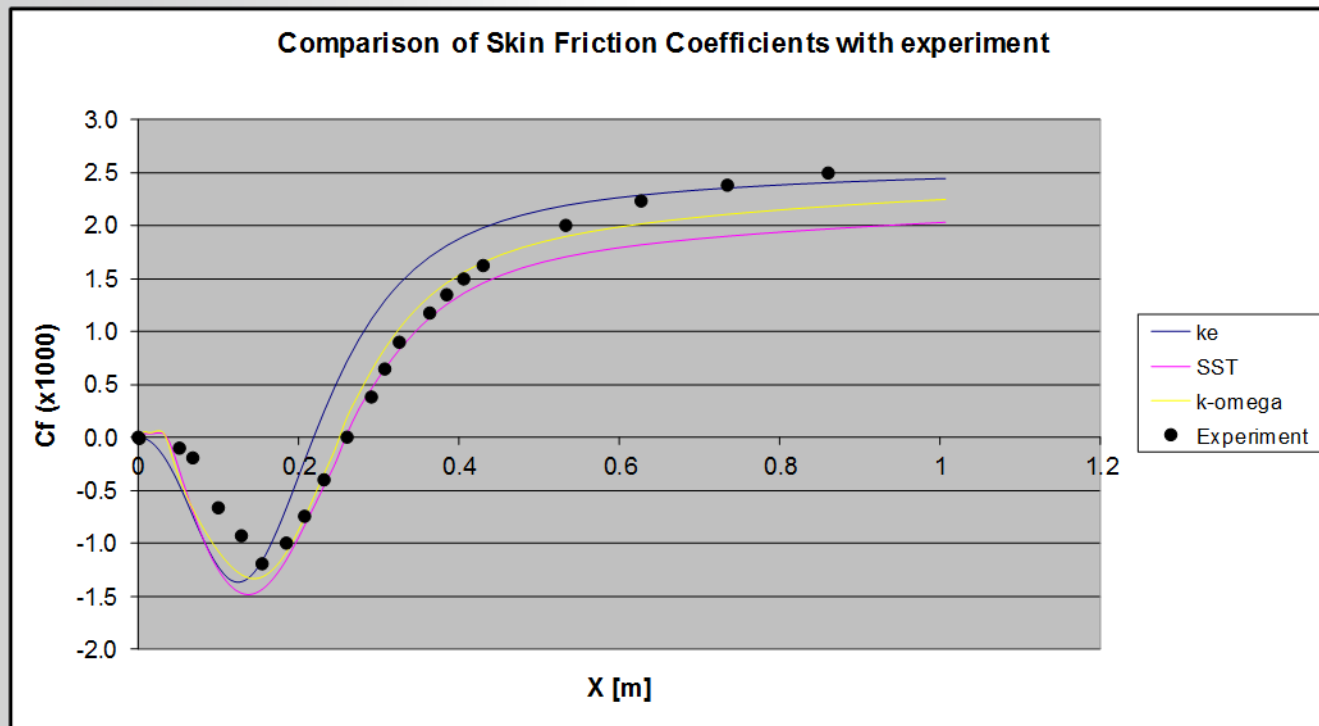
```
! $exfile = "Chart".$type.".csv";  
EXPORT:  
  Export File = $exfile  
  Export Chart Name = Chart 1  
  Overwrite = On  
END  
>export chart  
> close  
!}
```

- **The session file is now ready. Save the file in your text editor**

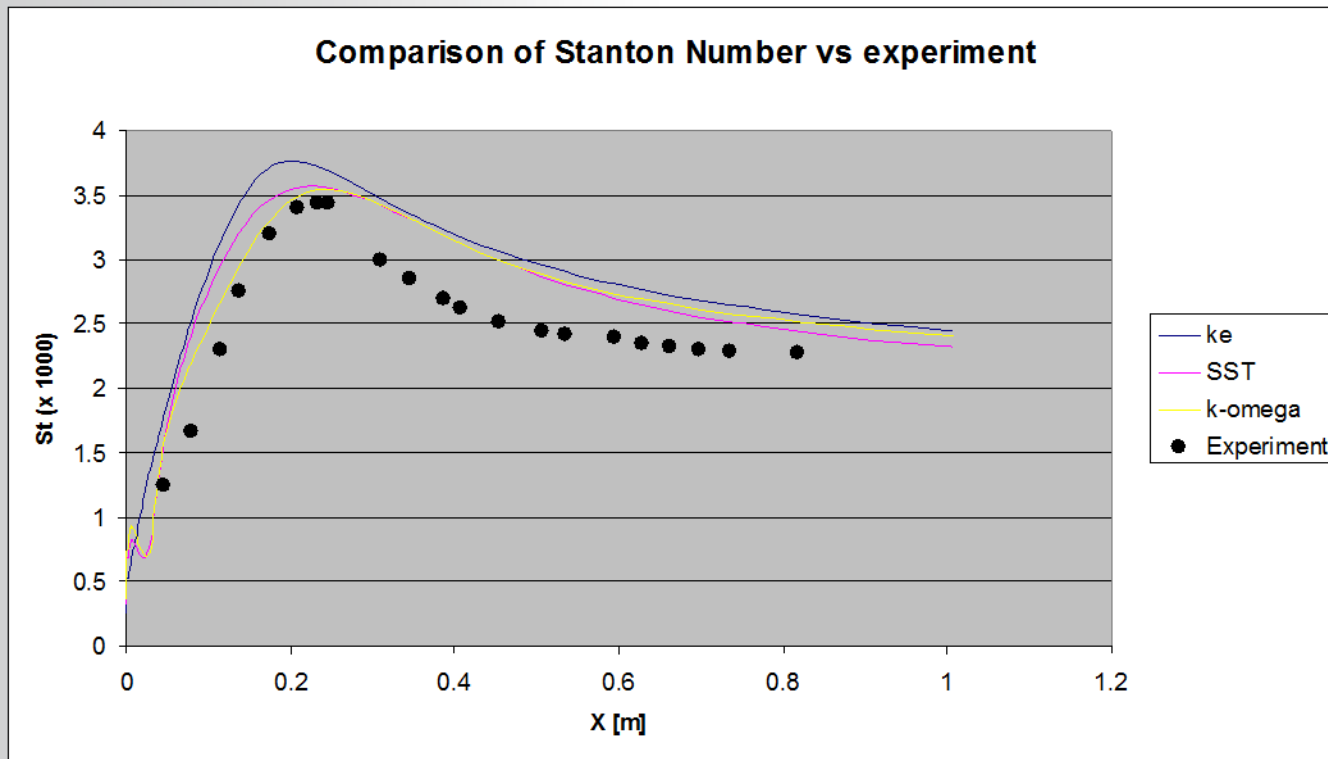
Running the Session File

- **At the Command Line type *cfx5post -batch post.cse* and press *Enter***
 - **The output will print out:**
 - **results file being processed**
 - **maximum temperature, average skin friction coefficient and Stanton number.**
 - **Three csv files containing the exported chart data will also be written.**
- **The csv data files can be imported into a spreadsheeting package and the results charted there.**
 - **An sample Excel file is provided. It includes experimental results.**

Comparing Data



Comparing Data



Data Discussion

The results show reasonable agreement with the skin friction and Stanton number data. Note that the SST and k-omega model give a more accurate downstream re-attachment location (where the skin friction coefficient is zero). It has been found that a finer mesh will produce results closer to the experimental data.

You can download the validation paper that was used as the basis for this workshop from the ANSYS Customer Portal.

Discussion of Commands

A number of Perl commands and CFX Power Syntax commands were added to the Session file. These command are outlined here. For a more complete understanding refer to a Perl manual and in the Help Documentation at CFX // Reference Guide // Power Syntax in ANSYS CFX.

foreach \$res(...)

This evaluates each object inside the brackets, assigns the current value inside the bracket to \$res, and then processes all the commands inside the curly brackets {.....}. Hence, \$res changes for each loop

@temp = split(...);

This creates an array called “temp” by splitting the filename into parts, separated by the pattern ‘_001.res’.

\$type = \$temp[0];

We now use the first element in the “temp” array as our type name. We have now extracted the first part of the results file name (e.g. ke, SST or komega)

Discussion of Commands

>load filename = \$res

This is a CCL action that gets the value of \$res and loads it.

Line Name = Cf-\$type

We change the line name so it is appended with either ke, SST, or komega, depending on the loop we are currently in

! \$maxtemp = maxVal("Temperature", "Default Domain");

This is a power syntax function that obtains the max value of a variable at a location (in this case over the entire domain) and stores the value in a variable (in this case \$maxtemp). There are many more of these functions available. See the Power Syntax documentation for details.