



CHAM

RhinoCFD® Quick Start Guide

Powered by PHOENICS





Contents

Introduction	2
Why CFD in CAD	2
Outline of Use	2
Installation of the RhinoCFD Plugin	3
Summary of RhinoCFD Options	5
Cylindrical Polar cases	7
Error Messages	13
Appendix.....	15
Handling of 2D Objects in Polar cases	15



Introduction

PHOENICS is a general purpose Computer Fluid Dynamics (CFD) code for the most types of flow simulation. A sub-set of PHOENICS is now available as the **RhinoCFD** plug-in for Rhino3D. This document describes how to use the CFD plug in within the Rhino CAD environment.

This document is intended to get you using RhinoCFD quickly; more comprehensive documents are provided for reference as needed. Many are available from the POLIS documentation of PHOENICS (online at www.cham.co.uk/PHOENICS/d_polis/polis.htm).

Why CFD in CAD

Many engineering and architectural designs rely on calculations of stress and of fluid dynamic forces, heat transfer and so on. Stress is commonly calculated with one of the many FE codes; fluid dynamics calculations are best handled by a specialist code; this is the realm of CFD and the purpose of RhinoCFD – powered by PHOENICS, a well-established software package used in a wide range of industries. With CAD as the preferred method for new design it is clearly beneficial to be able to undertake the CFD calculation and access its results whilst remaining within a familiar CAD environment.

Outline of Use

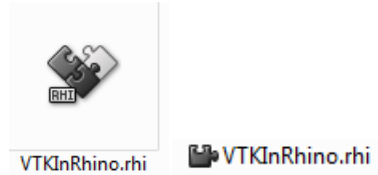
The process for using CFD in CAD is as follows:

1. Prepare the design of your device, building, etc (or load a pre-existing CAD design).
2. Specify a volume of space including the region of interest (the “domain”) for the CFD calculation.
3. Add boundary conditions, inlets & outlets.
4. Refine default computational-mesh settings and numerical settings.
5. Run the CFD code and check its convergence.
6. Load and interrogate the results.



Installation of the RhinoCFD Plugin

You will be supplied with four Rhino Installer files (extension “.rhi”) for use with the 64 bit version of Rhino. If your PC has Rhino installed correctly then the icon will be:

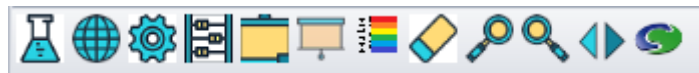


Above: Large and small icons for a Rhino Installer.

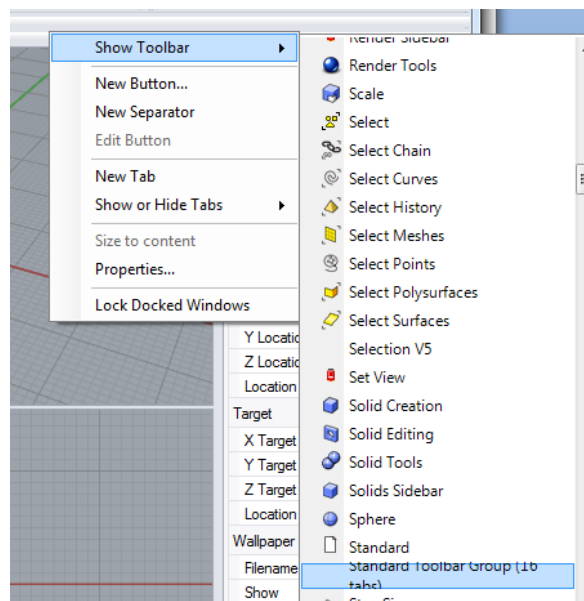
Double clicking these icons will install the RhinoCFD CFD plugin into a directory on your computer. You need to double click on all four plugins:

- RhinoSatex: Linking Rhino objects to the CFD properties and running the solver;
- VTKInRhino: Reading and displaying the results
- ChamDatImportRhino: Imports older RhinoCFD geometry files.
- Cfd4rhino.rui: Contains the toolbars

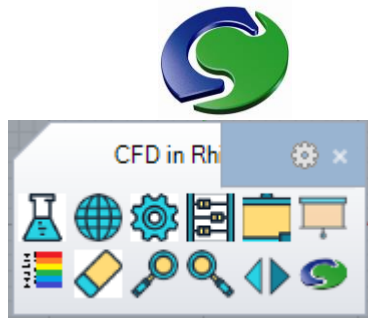
Once installed restart Rhino (64 bit) and you should see a new toolbar:



If these icons are not visible for some reason try right clicking in the toolbar region to present this menu and select Show Toolbar->Standard Toolbar. If you are having difficulty with finding the RhinoCFD tab contact the technical support team.



If the RhinoCFD toolbar still cannot be found try using “Show Toolbar-> RhinoCFD” – the



floating RhinoCFD tab will appear then drag the tab (click where 'CFD in Rhino' is displayed) up to the standard toolbar where it will dock

Next return to the 'First Example Case' section in this document to test the system is functioning correctly.

***NOTES: If you are concerned about what is in the RHI files, they are actually ZIP files; change the extension of the RHI file to ZIP and they will open within your preferred ZIP program to show the files that will be installed. Most of the files are either DLLs or RHP (Rhino Plugin files) plus a RUI (Rhino User Interface file) called RhinoSatexe.rui. The RUI file defines a tool bar tab with some icons and their actions.

The actual installation directory will depend on whether you install RhinoCFD for All Users or for you alone (the choice may be limited by your administrative rights). Typically the directory (when installed as User xxx) is:

C:\Users\xxx\AppData\Roaming\McNeel\Rhinoceros\5.0\Plug-ins\ZZZ

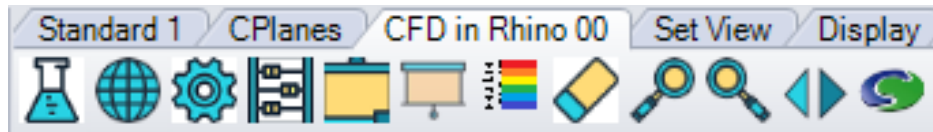
Subdirectory ZZZ is a rather complex name like "RhinoSatexe (376ce15a-ac8d-4c58-ae2c-913728cd2930)" – a unique identifier for the build of plugin installed. This directory can be used to delete RhinoCFD manually if needs be. Three plugin directories will be created – one called RhinoSatexe, VTKInRhino and ChamDatImportRhino (each with its UUID at the end). These plugins provide the elements stated previously.

The subdirectories contain large numbers of DLL files which are called from the .rhp file (eg RhinoSatexe.rhp depends on the various dlls in the same directory). The .rhp file is the plugin which Rhino links to.



Summary of RhinoCFD Options

The RhinoCFD toolbar.



Buttons 1-4 = setting up the case and running. Buttons 5-11 = display of results.

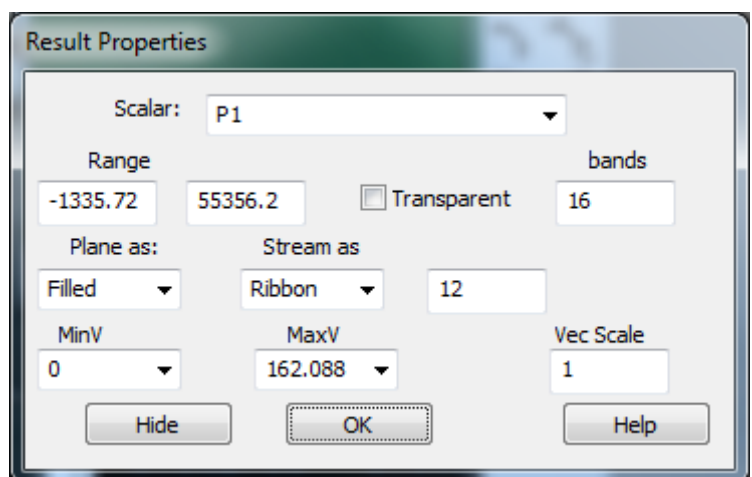
The buttons from left to right are:

- i) (Left click) Create domain (if there is no domain); (Right click) edit grid.
- ii) (Left) Display Grid in X, Y, Z plane; (Right) hide the grid if it is displayed.
- iii) Object properties (of selection); (Right) – remove PHOENICS properties from case.
- iv) (Left) – run the case; (Right) reload Q1 – e.g. if modified manually outside Rhino3D.
- v) (Left) display the Result Properties dialog (Right) contour the surface of an object (to be selected).
- vi) (Left) Toggle key to show colour corresponding to result values (Right) same.
- vii) (Left) hides current results probe (hiding the contours for example); (Right) – reshown all probes and their contours, etc.
- viii) (Left) load the VTK file for the latest results; (Right) clear results allowing geometry modification and more simulations to be run.
- ix) (Left) create a new contour or vector cutting plane; (Right) create a streamline probe.
- x) (Left) Create an iso-surface probe.
- xi) (Left) load the next time step result (if it exists); (Right) load previous time step result. This has a Submenu to progress 10 steps forward, backward, or animate the time dependent solution.

The Result Properties dialog can be thought of as four separate rows of controls.

The top option shows which scalar is to be drawn (P1 here). The drop down menu offers other variables solved for by RhinoCFD (U1, V1, W1, KE, EP and so on).


The second row specifies the range of the scalar to be converted to colour in the display and the number of colour bands. The minimum and maximum values are reset when the scalar is changed. The Transparent tick box hides display of values outside the scalar range (used to concentrate attention on specific values).





The third row controls the type of display (e.g. filled contours, lines, vectors etc) and the type of streamlines plus the number of streamlines generated by a probe (12 in this example).

The fourth row selects the smallest size for vectors to display (.1 means that a vector of infinitesimal length would be displayed as if it were 0.1 of the maximum vector length; 0 makes the smallest vectors nearly invisible) and a general vector scale – 2 makes the vectors twice as large as default; .5 half size.

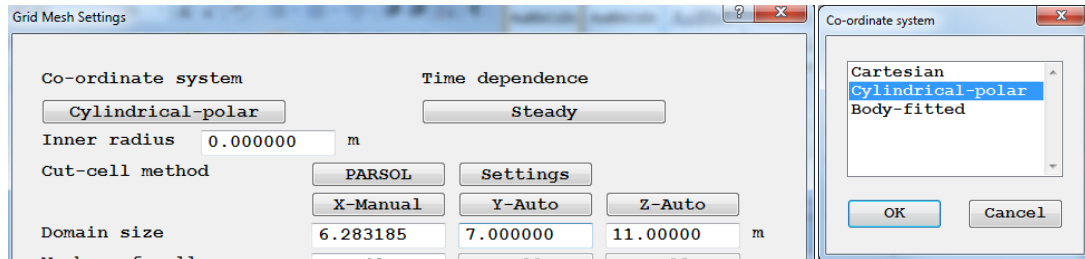
If a time dependent case is performed then the next/previous time step can be chosen using the  icon. All display values will remain as in the current display (particularly the range of scalar). [Note: sometimes changing the time displayed will change the range of scalar drastically with some loss of contours.]




Cylindrical Polar cases

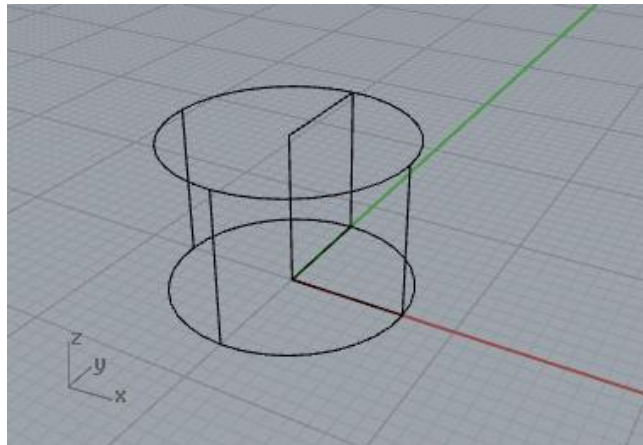
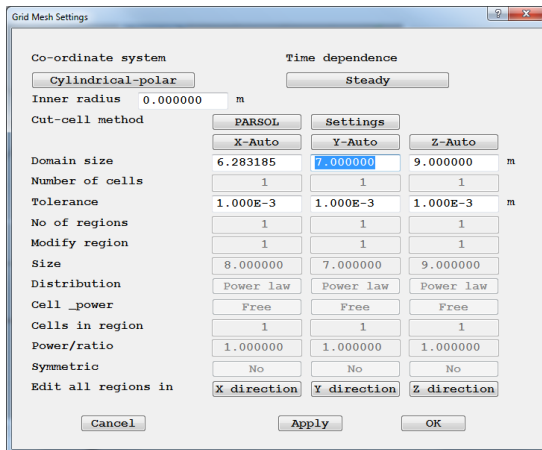
Some problems are better handled with a polar grid than with the default Cartesian grid – for example, engine cylinders, flows in tubes. The choice of grid is selected from the *Grid Edit* page

(right click on ) and change Cartesian to Cylindrical Polar in the Co-ordinate option menu.

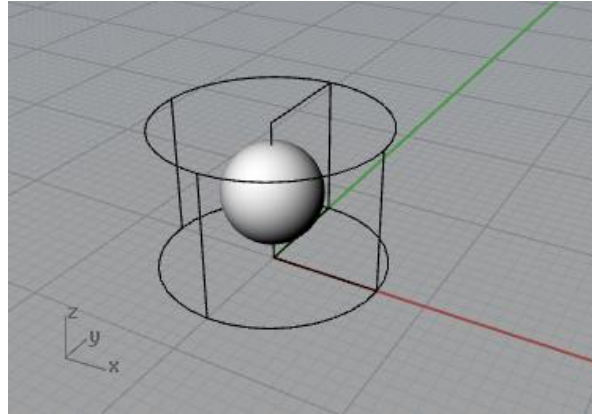


The example shown below starts a new case and shows how to generate suitable objects for polar inlets within Rhino3D. Start Rhino3D with a new case, and select the *Edit Domain* option (right click

on the flask ) . This will create a domain, change type to Polar, set its size to angle 6.28315 (type in any larger number such as 8 and RhinoCFD will limit the value to 2π). Set the radius to 7 and the height to 9 (the values used are arbitrary but are used to show how to edit the polar domain size).

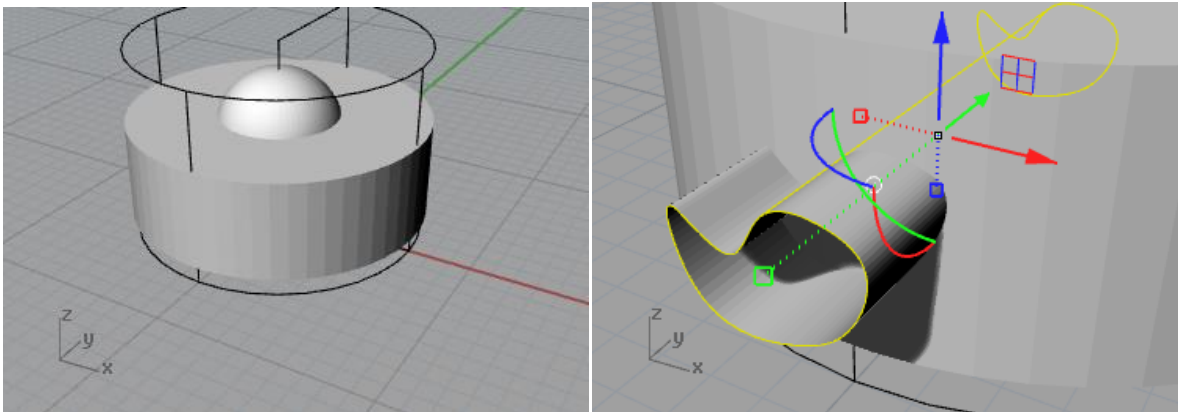


This shows the domain as a cylinder. Any blockage can now be inserted – such as a sphere, cylinder etc.

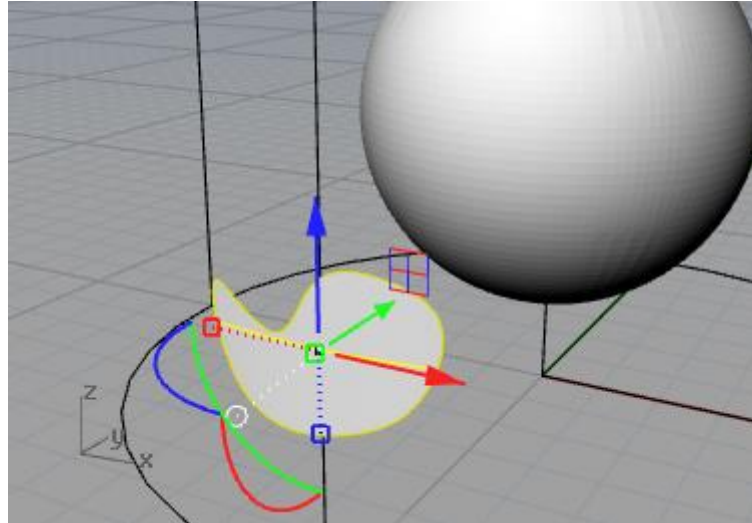


Adding inlets in the Z direction (at the bottom and top of the domain) is identical to the Cartesian case – no new techniques are required although the Z extent of the inlet/outlet should be very small indeed. This section will describe adding a radial inlet (on the maximum Radius of the domain). Creating the geometry can be done in a number of ways but the simplest (in my opinion) is as follows:

Create a cylinder at the size of the domain extending sufficiently far in the Z direction and create a clipping object (an extruded curve) – I have made a unusual curve; circular or square or hexagonal trimming objects can be created instead).

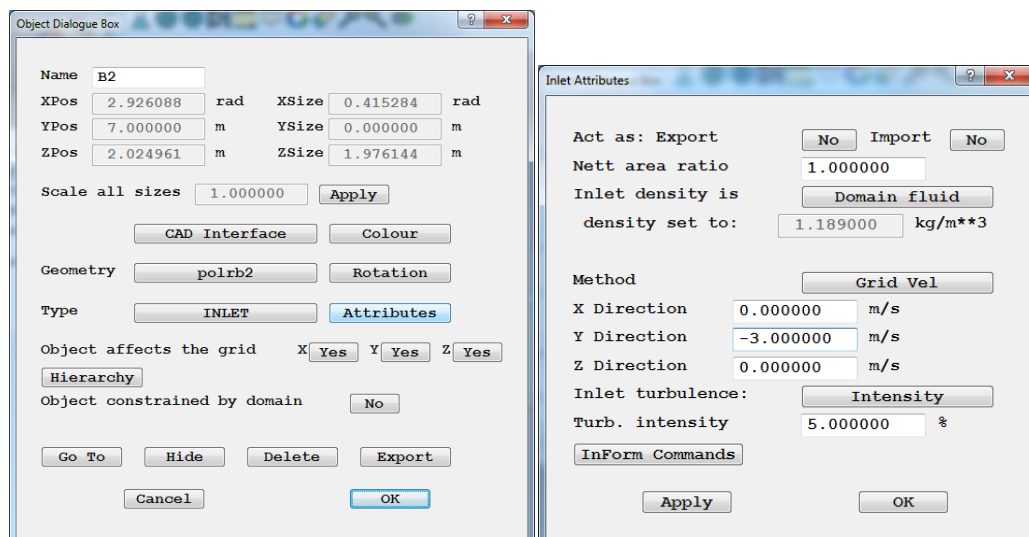


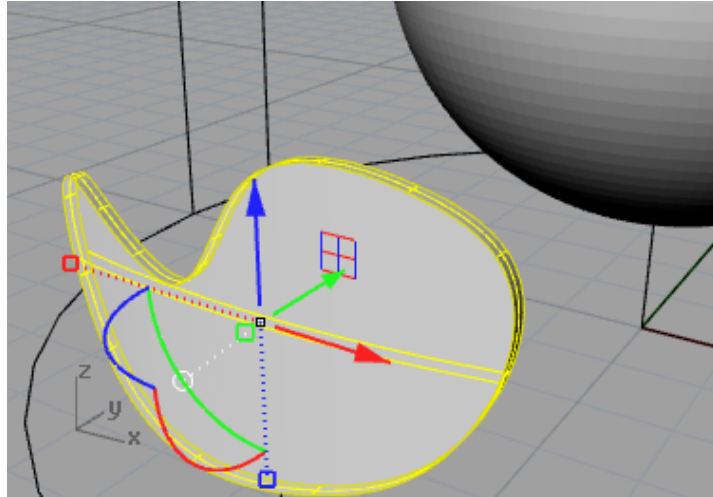
The trimming curve is show in the second figure. Use the *Trim* command (type Trim!) with the tri curve selected; then follow the prompts to pick the cylinder to be trimmed, and press Enter to finish the trim. Delete the trim curve and a section of the cylinder created is left to define the inlet/outlet:



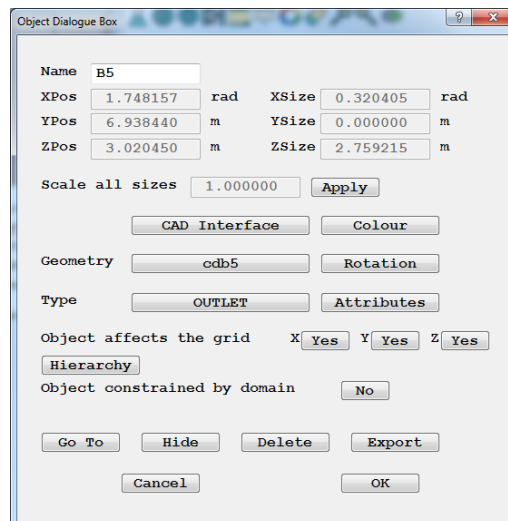
However, this inlet must be 3 dimensional; so select the trimmed face and use 'ExtrueSrf' to extrude it. The pick must be done in the window which most closely represents the direction for extrusion (*for Rhino experts: extrusion is made in the direction normal to the Construction plane which can be defined from the CPlanes tab*). You will need to experiment with this extrusion tool to get the shape extruded along the radius.

Set the properties of this object to be an inlet, with inlet velocity in the Y (radial) direction -3





Use a similar technique to generate the outlet object, set the properties of the outlet object to be an outlet.



Change the grid settings for Angle (X) to 'X-Manual' and adjust the grid properties to 36 angular grids (10 degree spacing).



Grid Mesh Settings

Co-ordinate system: **Cylindrical-polar**

Time dependence: **Steady**

Inner radius: 0.000000 m

Cut-cell method: **PARSOL** **Settings**

Domain size: 6.283185 7.000000 9.000000 m

Number of cells: 36 20 19

Tolerance: 1.000E-3 1.000E-3 1.000E-3 m

No of regions: 5 2 7

Modify region: 1 1 1

Size: 1.349676 3.078959 1.513504

Distribution: **Power law** Power law Power law

Cell_power: **Free** Set Set

Cells in region: 7 9 3

Power/ratio: 1.000000 1.000000 1.000000

Symmetric: **No** No No

Edit all regions in: **X direction** Y direction Z direction

Cancel Apply OK

X direction settings

Global settings:-

Domain size: 6.283185 m Number of cells: 36

Auto grid settings:- **OFF** Set defaults

Min cell factor: 0.005000 Init. cell factor: 0.050000 Fractions

Max size ratio: 1.500000 Expansion power: 1.200000


Form: **Geom Prog** Boundary - Low Off High Off

Region settings:- (Currently 5 regions)

Free all regions: **Free all** Tolerance: 1.000E-3 m

Reg	End posn	Cells	Distribtn	Power	Symmetric	Cell_power
1	1.349676	7	Power law	1.000000	No	Free
2	1.746580	3	Power law	1.000000	No	Free
3	2.926088	7	Power law	1.000000	No	Free
4	3.341371	3	Power law	1.000000	No	Free
5	6.283185	16	Power law	1.000000	No	Free

Cancel Apply OK

Set the case to repeat properties at theta = 0 (cyclic boundaries) from the *Sources* page of the main menu (accessed by selecting the domain and pick object properties with ):

Domain Settings

Geometry Models Properties Initialisation Help Top menu

Sources Numerics GROUND Output

Gravitational forces: **OFF**

Cyclic boundary conditions: **ALL SLABS ON**

Use Earth-Generated wall functions: EGWF **T**

Coeff. for auto wall functions: **LOG-LAW**

Global wall roughness: 0.000000 m

U values are: **Velocity**

Rotating coordinate system: **OFF**

Potential flow: **OFF**

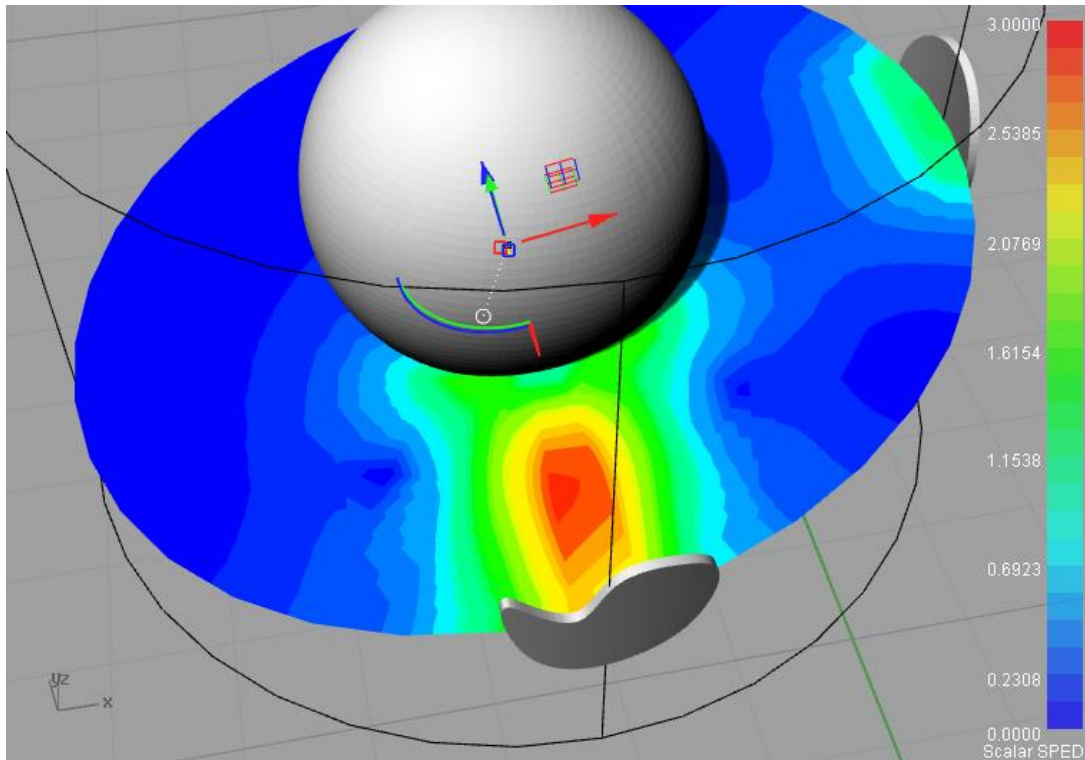
Coriolis Force: **OFF**

Advanced settings: **PIL** Edit InForm 13

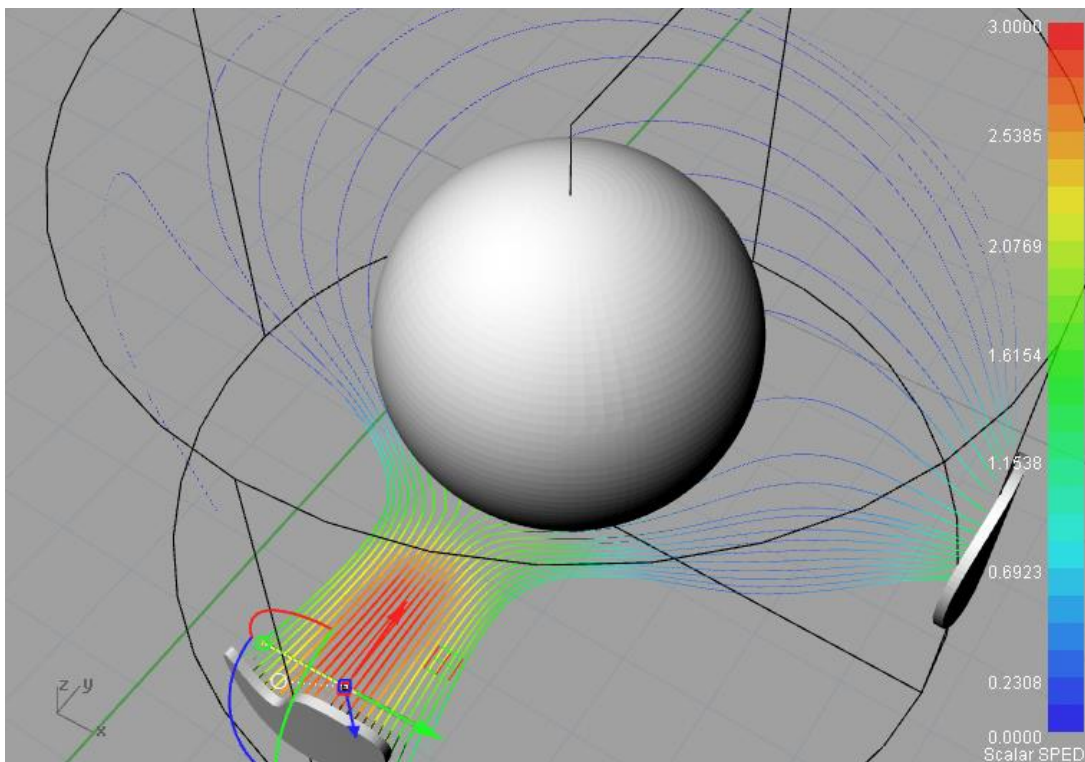
PIL Command:



Run the case and load the results.



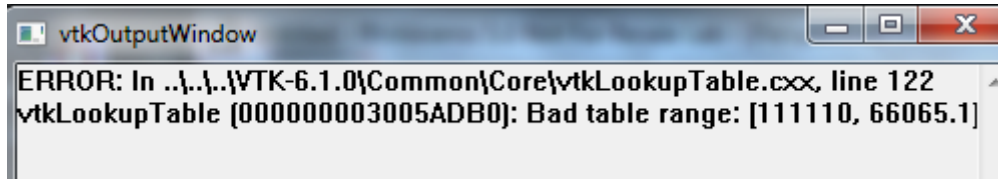
Adjust the cutting plane to show inlet and outlet, add streamlines:





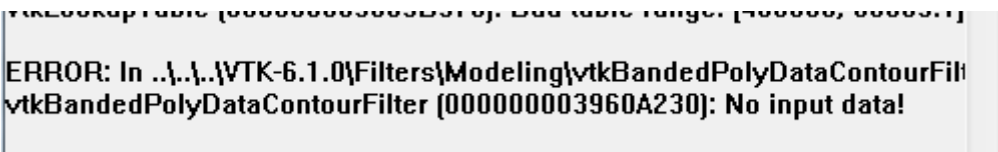
Error Messages

When viewing results a VTK warning window can appear occasionally; VTK is a third party visualisation suite and is giving information useful to debugging. For example:



```
ERROR: In ..\..\VTK-6.1.0\Common\Core\vtkLookupTable.cxx, line 122
vtkLookupTable (000000003005ADB0): Bad table range: [111110, 66065.1]
```

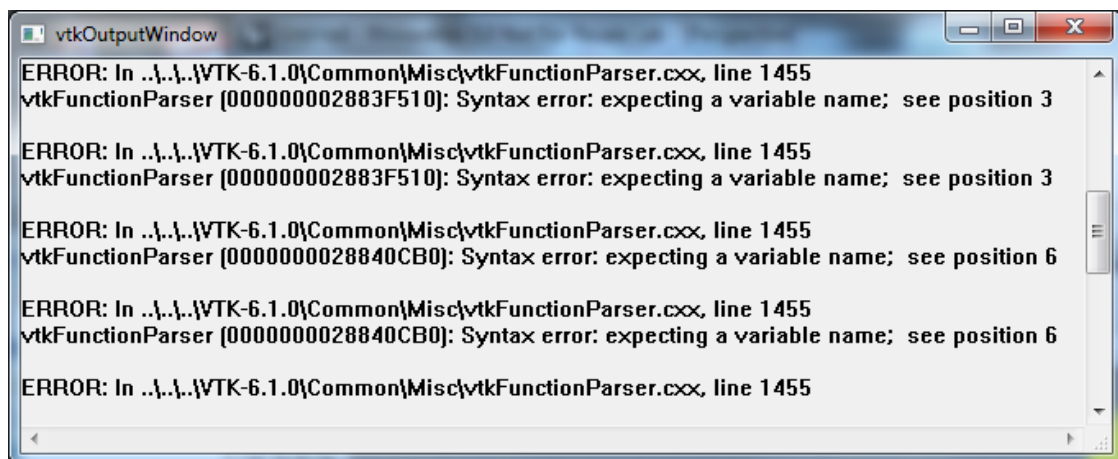
Means that you have set a range of scalars where the maximum scalar is less than the minimum value so no contours can be plotted. Adjust min and max in the Result Properties window.



```
ERROR: In ..\..\VTK-6.1.0\Filters\Modeling\vtkBandedPolyDataContourFilter.cxx, line 145
vtkBandedPolyDataContourFilter (000000003960A230): No input data!
```

'No Input Data' means that you have requested results at a position (cutting plane, iso-surface or streamline source) where there are no solution values. Move the relevant probe back into the domain.

When the results are being displayed there may be a warning about the formula for the current scalar, particularly while entering a new formula.



```
ERROR: In ..\..\VTK-6.1.0\Common\Misc\vtkFunctionParser.cxx, line 1455
vtkFunctionParser (000000002883F510): Syntax error: expecting a variable name; see position 3

ERROR: In ..\..\VTK-6.1.0\Common\Misc\vtkFunctionParser.cxx, line 1455
vtkFunctionParser (000000002883F510): Syntax error: expecting a variable name; see position 3

ERROR: In ..\..\VTK-6.1.0\Common\Misc\vtkFunctionParser.cxx, line 1455
vtkFunctionParser (0000000028840CB0): Syntax error: expecting a variable name; see position 6

ERROR: In ..\..\VTK-6.1.0\Common\Misc\vtkFunctionParser.cxx, line 1455
vtkFunctionParser (0000000028840CB0): Syntax error: expecting a variable name; see position 6

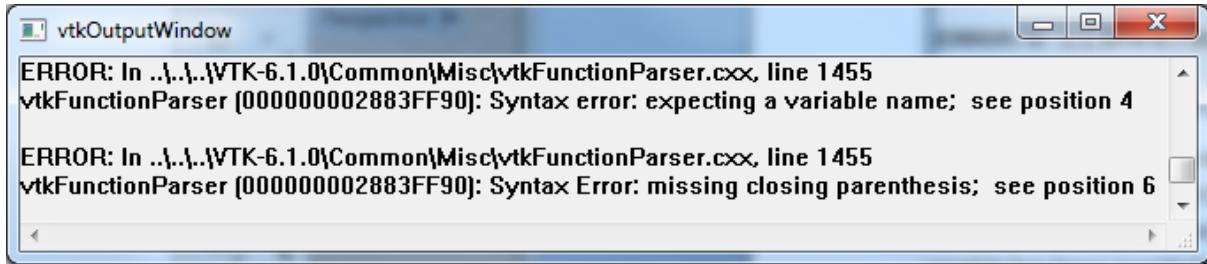
ERROR: In ..\..\VTK-6.1.0\Common\Misc\vtkFunctionParser.cxx, line 1455
vtkFunctionParser (0000000028840CB0): Syntax error: expecting a variable name; see position 6
```

This message identifies an error in the scalar formula such as unmatched parenthesis, reference to a scalar which is not present in the results or an unknown function (functions such as sin, cos, tan, sinh, sqrt are always in lower case).

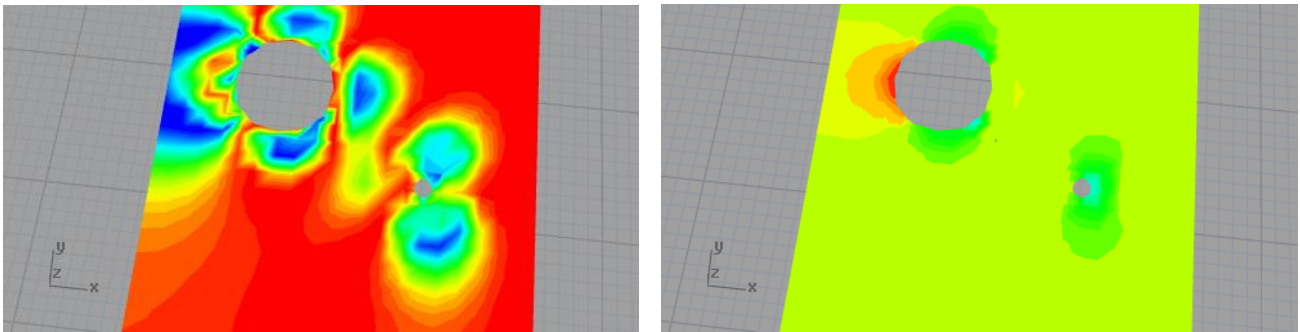
The errors will cease once the complete name of a function or variable has been inserted; however beware of the sqrt function which will display an enormous number of set of warnings if the value to be square rooted is negative – perhaps encase the contents with an abs()



function such as `"sqrt(abs(P1))"` or ensure that you are using an always positive value (`"sqrt(KE)"` where KE is always positive - Kinetic energy).



Was generated when typing in the function `"cos(P1)"` into the scalar field. The first warning was at position 0 while `"cos("` was inserted; once the cos and bracket is open the error moves to position 4 where a value or a scalar is expected (P1 of course). Once P1 is complete a closing bracket is expected at character 6, and after entering the `)"` the novel function displays.



`Cos(P1)` displayed and P1 contours. This shows that any function can be created for the displayed scalar derived from the existing scalars.

It is not suggested that cosine of pressure is a useful display (particularly as pressure has physical units and cos is relevant to a non-dimensional angular measure such as radians).

An example of a useful function would be `'P1+0.5*DEN1*(U1^2+V1^2+W1^2)'` to evaluate the total pressure.



Appendix

Handling of 2D Objects in Polar cases

PHOENICS allows the solution of the CFD equations in a cylindrical polar grid. Polar coordinates are useful for cases with string cylindrical symmetry such as flow in tubes, car engine cylinders and so on. Some cases can be reduced to 2D geometry when there is no swirl component, dramatically speeding the solution or allowing an extremely fine grid.

In principle, any 3D shape can be used in such cylindrical cases (using normal 3D geometry files). However there are special polar objects which have a number of useful properties; usually they are the 'polcube' shapes where the polar object fills a constant range of angle, radius and Z coordinate – making a segment of a pie shape (including cylinders and washers - circles with a hole). It is not possible to provide many such shapes as they do not preserve shape when moved – for example, if there were a circular object at constant radius then stretching it in the angle direction would change the shape to elliptical (wrapped around a cylinder). This restricts the shapes of inlets and outlets available to users of the Polar option – the library cases only show rectangular inlets on the curved sides of the polar grid, although any shape inlet can be used on the flat ends of the polar grid. As in the Cartesian cases the actual thickness of the object defined in the data file must be non-zero.

This appendix defines how new shapes can be created in CAD for use with Polar case inlets, outlets, plates and openings. In future other 2D objects may be defined. All Professional CAD programs have Boolean geometry options, so it is quite easy to create suitable 2D shapes by a number of means such as:

- subtracting a cylinder from a general shape to create the inner curved surface and making the union of the result with a larger cylinder to create the outer curved surface, or
- drawing a general curve on the surface of the cylinder then extruding the surface.

Both methods will result in a 3D shape with surface which fits the cylindrical grid. However to avoid accuracy problems in CAD we specify these 2D objects as 3D objects, typically spanning the boundary of the domain. At the "Z" ends of the polar grids this is trivial and uses the same algorithm as Cartesian to produce the 2D shape required by satellite. On the constant radius sides, the object should span the maximum radius (or inner radius) of the domain and conform closely to the cylindrical surface. On the constant angle surfaces the 2D object should span the minimum or maximum theta angle of the domain grid by a small angle.

For objects spanning the $R=\max$ (and $R=\min$) boundary the position of the object will be at R (max or min) and its range of theta set to the range of theta of all the vertices in the object; for PHOENICS the geometry file generated will be called POL<name> where name is the object name.

Concentration Heat And Momentum Limited [CHAM]

Bakery House, 40 High Street, Wimbledon Village, London SW19 5AU, UK

Tel: +44 (0)20 8947 7651 Fax: +44 (0)20 8879 3497 Email: phoenics@cham.co.uk

Web: www.cham.co.uk