

ANSYS Fluent Adjoint Solver



ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
<http://www.ansys.com>
(T) 724-746-3304
(F) 724-514-9494

Release 15.0
November 2013

ANSYS, Inc. is certified to ISO 9001:2008.
--

Copyright and Trademark Information

© 2013 SAS IP, Inc. All rights reserved. Unauthorized use, distribution or duplication is prohibited.

ANSYS, ANSYS Workbench, Ansoft, AUTODYN, EKM, Engineering Knowledge Manager, CFX, FLUENT, HFSS and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. is certified to ISO 9001:2008.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the [legal information](#) in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.

Table of Contents

Using This Manual	v
1. The Contents of This Manual	v
2. The Contents of the Fluent Manuals	v
3. Typographical Conventions	vi
4. Mathematical Conventions	vii
5. Technical Support	viii
1. Introduction to the Adjoint Solver	1
1.1. Overview	1
1.1.1. General Observables	3
1.1.2. General Operations	5
1.2. Discrete Versus Continuous Adjoint Solver	6
1.3. Discrete Adjoint Solver Overview	7
1.4. Adjoint Solver Stabilization	10
1.5. Solution-Based Adaption	11
1.6. Using The Data To Improve A Design	11
1.7. Smoothing and Mesh Morphing	12
2. Using the Adjoint Solver Module	15
2.1. Installing the Adjoint Solver Module	15
2.2. Loading the Adjoint Solver Module	16
2.3. Understanding the Scope of the Functionality	16
2.4. Defining Observables	17
2.4.1. Managing Adjoint Observables	19
2.4.2. Creating New Observables	22
2.4.3. Renaming Adjoint Observables	23
2.5. Using the Adjoint Solution Methods Dialog Box	23
2.6. Using the Adjoint Solution Controls Dialog Box	24
2.6.1. Stabilized Scheme Settings	26
2.6.1.1. Modal Stabilization Scheme	26
2.6.1.2. Spatial Stabilization Scheme	29
2.7. Working with Adjoint Residual Monitors	31
2.8. Running the Adjoint Calculation	31
2.9. Postprocessing of Adjoint Solutions	32
2.9.1. Field Data	32
2.9.2. Scalar Data	37
2.10. Modifying the Geometry	38
2.10.1. Shape Modification	38
2.10.2. Defining the Control Volume for Control-Volume Deformation	40
2.11. Using the Adjoint Solver Module's Text User Interface	41
3. Tutorial: 2D Laminar Flow Past a Cylinder	43
3.1. Introduction	43
3.2. Setup and Solution	43
3.2.1. Preparation	44
3.2.2. Step 1: Loading the Adjoint Solver Add-on	46
3.2.3. Step 2: Solver Setup and Adjoint Solution Convergence	47
3.2.4. Step 3: Postprocessing	53
3.2.4.1. Boundary Condition Sensitivity	54
3.2.4.2. Sources of Mass and Momentum	54
3.2.4.3. Shape Sensitivity	57
3.2.5. Step 4: Shape Modification	59
3.2.6. Step 5: Extending the Observable to a Ratio	63

3.3. Summary 70

Index 73

Using This Manual

This preface is divided into the following sections:

1. [The Contents of This Manual](#)
2. [The Contents of the Fluent Manuals](#)
3. [Typographical Conventions](#)
4. [Mathematical Conventions](#)
5. [Technical Support](#)

1. The Contents of This Manual

The ANSYS Fluent Adjoint Solver Module Manual provides information about using the adjoint solver with ANSYS Fluent. An adjoint solver is a specialized tool that extends the scope of the analysis provided by a conventional flow solver by providing detailed sensitivity data for the performance of a fluid system. This manual is divided into the following chapters:

- [Introduction to the Adjoint Solver \(p. 1\)](#)
- [Using the Adjoint Solver Module \(p. 15\)](#)
- [Tutorial: 2D Laminar Flow Past a Cylinder \(p. 43\)](#)

2. The Contents of the Fluent Manuals

The manuals listed below form the Fluent product documentation set. They include descriptions of the procedures, commands, and theoretical details needed to use Fluent products.

- [Fluent Getting Started Guide](#) contains general information about getting started with using Fluent and provides details about starting, running, and exiting the program.
- [Fluent Migration Manual](#) contains information about transitioning from the previous release of Fluent, including details about new features, solution changes, and text command list changes.
- [Fluent User's Guide](#) contains detailed information about running a simulation using the solution mode of Fluent, including information about the user interface, reading and writing files, defining boundary conditions, setting up physical models, calculating a solution, and analyzing your results.
- [ANSYS Fluent Meshing User's Guide](#) contains detailed information about creating 3D meshes using the meshing mode of Fluent.
- [Fluent in Workbench User's Guide](#) contains information about getting started with and using Fluent within the Workbench environment.
- [Fluent Theory Guide](#) contains reference information for how the physical models are implemented in Fluent.
- [Fluent UDF Manual](#) contains information about writing and using user-defined functions (UDFs).
- [Fluent Tutorial Guide](#) contains a number of examples of various flow problems with detailed instructions, commentary, and postprocessing of results.
- [ANSYS Fluent Meshing Tutorials](#) contains a number of examples of general mesh-generation techniques used in ANSYS Fluent Meshing.

Tutorials for release 15.0 are available on the ANSYS Customer Portal. To access tutorials and their input files on the ANSYS Customer Portal, go to <http://support.ansys.com/training>.

- [Fluent Text Command List](#) contains a brief description of each of the commands in Fluent's solution mode text interface.
- [ANSYS Fluent Meshing Text Command List](#) contains a brief description of each of the commands in Fluent's meshing mode text interface.
- [Fluent Adjoint Solver Module Manual](#) contains information about the background and usage of Fluent's Adjoint Solver Module that allows you to obtain detailed sensitivity data for the performance of a fluid system.
- [Fluent Battery Module Manual](#) contains information about the background and usage of Fluent's Battery Module that allows you to analyze the behavior of electric batteries.
- [Fluent Continuous Fiber Module Manual](#) contains information about the background and usage of Fluent's Continuous Fiber Module that allows you to analyze the behavior of fiber flow, fiber properties, and coupling between fibers and the surrounding fluid due to the strong interaction that exists between the fibers and the surrounding gas.
- [Fluent Fuel Cell Modules Manual](#) contains information about the background and the usage of two separate add-on fuel cell models for Fluent that allow you to model polymer electrolyte membrane fuel cells (PEMFC), solid oxide fuel cells (SOFC), and electrolysis with Fluent.
- [Fluent Magnetohydrodynamics \(MHD\) Module Manual](#) contains information about the background and usage of Fluent's Magnetohydrodynamics (MHD) Module that allows you to analyze the behavior of electrically conducting fluid flow under the influence of constant (DC) or oscillating (AC) electromagnetic fields.
- [Fluent Population Balance Module Manual](#) contains information about the background and usage of Fluent's Population Balance Module that allows you to analyze multiphase flows involving size distributions where particle population (as well as momentum, mass, and energy) require a balance equation.
- [Fluent as a Server User's Guide](#) contains information about the usage of Fluent as a Server which allows you to connect to a Fluent session and issue commands from a remote client application.
- [Running Fluent Under LSF](#) contains information about using Fluent with Platform Computing's LSF software, a distributed computing resource management tool.
- [Running Fluent Under PBS Professional](#) contains information about using Fluent with Altair PBS Professional, an open workload management tool for local and distributed environments.
- [Running Fluent Under SGE](#) contains information about using Fluent with Sun Grid Engine (SGE) software, a distributed computing resource management tool.

3. Typographical Conventions

Several typographical conventions are used in this manual's text to facilitate your learning process.

- Different type styles are used to indicate graphical user interface menu items and text interface menu items (for example, **Iso-Surface** dialog box, `surface/iso-surface` command).

- The text interface type style is also used when illustrating exactly what appears on the screen to distinguish it from the narrative text. In this context, user inputs are typically shown in boldface.
- A mini flow chart is used to guide you through the navigation pane, which leads you to a specific task page or dialog box. For example,

 **Models** →  **Multiphase** → **Edit...**

indicates that **Models** is selected in the navigation pane, which then opens the corresponding task page. In the **Models** task page, **Multiphase** is selected from the list. Clicking the **Edit...** button opens the **Multiphase** dialog box.

Also, a mini flow chart is used to indicate the menu selections that lead you to a specific command or dialog box. For example,

Define → **Injections...**

indicates that the **Injections...** menu item can be selected from the **Define** pull-down menu, and

`display` → `mesh`

indicates that the `mesh` command is available in the `display` text menu.

In this manual, mini flow charts usually precede a description of a dialog box or command, or a screen illustration showing how to use the dialog box or command. They allow you to look up information about a command or dialog box and quickly determine how to access it without having to search the preceding material.

- The menu selections that will lead you to a particular dialog box or task page are also indicated (usually within a paragraph) using a "/". For example, **Define/Materials...** tells you to choose the **Materials...** menu item from the **Define** pull-down menu.

4. Mathematical Conventions

- Where possible, vector quantities are displayed with a raised arrow (e.g., \vec{a} , \vec{A}). Boldfaced characters are reserved for vectors and matrices as they apply to linear algebra (e.g., the identity matrix, **I**).
- The operator ∇ , referred to as grad, nabla, or del, represents the partial derivative of a quantity with respect to all directions in the chosen coordinate system. In Cartesian coordinates, ∇ is defined to be

$$\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \quad (1)$$

∇ appears in several ways:

- The gradient of a scalar quantity is the vector whose components are the partial derivatives; for example,

$$\nabla p = \frac{\partial p}{\partial x} \vec{i} + \frac{\partial p}{\partial y} \vec{j} + \frac{\partial p}{\partial z} \vec{k} \quad (2)$$

- The gradient of a vector quantity is a second-order tensor; for example, in Cartesian coordinates,

$$\nabla (\vec{v}) = \left(\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) (v_x \vec{i} + v_y \vec{j} + v_z \vec{k}) \quad (3)$$

This tensor is usually written as

$$\begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix} \quad (4)$$

- The divergence of a vector quantity, which is the inner product between ∇ and a vector; for example,

$$\nabla \cdot \vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \quad (5)$$

- The operator $\nabla \cdot \nabla$, which is usually written as ∇^2 and is known as the Laplacian; for example,

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \quad (6)$$

$\nabla^2 T$ is different from the expression $(\nabla T)^2$, which is defined as

$$(\nabla T)^2 = \left(\frac{\partial T}{\partial x} \right)^2 + \left(\frac{\partial T}{\partial y} \right)^2 + \left(\frac{\partial T}{\partial z} \right)^2 \quad (7)$$

- An exception to the use of ∇ is found in the discussion of Reynolds stresses in [Turbulence in the Fluent Theory Guide](#), where convention dictates the use of Cartesian tensor notation. In this chapter, you will also find that some velocity vector components are written as u , v , and w instead of the conventional v with directional subscripts.

5. Technical Support

If you encounter difficulties while using ANSYS Fluent, please first refer to the section(s) of the manual containing information on the commands you are trying to use or the type of problem you are trying to solve. The product documentation is available from the online help, or from the ANSYS Customer Portal. To access documentation files on the ANSYS Customer Portal, go to <http://support.ansys.com/documentation>.

If you encounter an error, please write down the exact error message that appeared and note as much information as you can about what you were doing in ANSYS Fluent.

Technical Support for ANSYS, Inc. products is provided either by ANSYS, Inc. directly or by one of our certified ANSYS Support Providers. Please check with the ANSYS Support Coordinator (ASC) at your

company to determine who provides support for your company, or go to www.ansys.com and select **Contact ANSYS > Contacts and Locations**.

If your support is provided by ANSYS, Inc. directly, Technical Support can be accessed quickly and efficiently from the ANSYS Customer Portal, which is available from the ANSYS Website (www.ansys.com) under **Support > Customer Portal**. The direct URL is: support.ansys.com.

One of the many useful features of the Customer Portal is the Knowledge Resources Search, which can be found on the Home page of the Customer Portal.

Systems and installation Knowledge Resources are easily accessible via the Customer Portal by using the following keywords in the search box: *Systems/Installation*. These Knowledge Resources provide solutions and guidance on how to resolve installation and licensing issues quickly.

NORTH AMERICA

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Toll-Free Telephone: 1.800.711.7199

Fax: 1.724.514.5096

Support for University customers is provided only through the ANSYS Customer Portal.

GERMANY

ANSYS Mechanical Products

Telephone: +49 (0) 8092 7005-55 (CADFEM)

Email: support@cadfem.de

All ANSYS Products

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

National Toll-Free Telephone:

German language: 0800 181 8499

English language: 0800 181 1565

Austria: 0800 297 835

Switzerland: 0800 546 318

International Telephone:

German language: +49 6151 152 9981

English language: +49 6151 152 9982

Email: support-germany@ansys.com

UNITED KINGDOM

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Telephone: Please have your Customer or Contact ID ready.

UK: 0800 048 0462

Republic of Ireland: 1800 065 6642

Outside UK: +44 1235 420130

Email: support-uk@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

JAPAN

CFX , ICEM CFD and Mechanical Products

Telephone: +81-3-5324-8333

Fax: +81-3-5324-7308

Email:

CFX: japan-cfx-support@ansys.com;

Mechanical: japan-ansys-support@ansys.com

Fluent Products

Telephone: +81-3-5324-7305

Email:

Fluent: japan-fluent-support@ansys.com;

Polyflow: japan-polyflow-support@ansys.com;

FfC: japan-ffc-support@ansys.com;

FloWizard: japan-flowizard-support@ansys.com

Icepak

Telephone: +81-3-5324-7444

Email: japan-icepak-support@ansys.com

Licensing and Installation

Email: japan-license-support@ansys.com

INDIA

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Telephone: +91 1 800 209 3475 (toll free) or +91 20 6654 3000 (toll)

Fax: +91 80 6772 2600

Email:

FEA products: feasup-india@ansys.com;

CFD products: cfdsup-india@ansys.com;

Ansoft products: ansoftsup-india@ansys.com;

Installation: installation-india@ansys.com

FRANCE

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Toll-Free Telephone: +33 (0) 800 919 225 **Toll Number:** +33 (0) 170 489 087

Email: support-france@ansys.com

BELGIUM**All ANSYS Products**

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Telephone: +32 (0) 10 45 28 61

Email: support-belgium@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

SWEDEN**All ANSYS Products**

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Telephone: +44 (0) 870 142 0300

Email: support-sweden@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

SPAIN and PORTUGAL**All ANSYS Products**

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Telephone: +34 900 933 407 (Spain), +351 800 880 513 (Portugal)

Email: support-spain@ansys.com, support-portugal@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

ITALY**All ANSYS Products**

Web: Go to the ANSYS Customer Portal (<http://support.ansys.com>) and select the appropriate option.

Telephone: +39 02 89013378

Email: support-italy@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

Chapter 1: Introduction to the Adjoint Solver

This chapter provides background for the ANSYS Fluent adjoint solver.

- 1.1. Overview
- 1.2. Discrete Versus Continuous Adjoint Solver
- 1.3. Discrete Adjoint Solver Overview
- 1.4. Adjoint Solver Stabilization
- 1.5. Solution-Based Adaption
- 1.6. Using The Data To Improve A Design
- 1.7. Smoothing and Mesh Morphing

In addition, see [Using the Adjoint Solver Module \(p. 15\)](#) and [Tutorial: 2D Laminar Flow Past a Cylinder \(p. 43\)](#) for more information about using the adjoint solver.

1.1. Overview

An adjoint solver is a specialized tool that extends the scope of the analysis provided by a conventional flow solver by providing detailed sensitivity data for the performance of a fluid system.

In order to perform a simulation using the ANSYS Fluent standard flow solvers, a user supplies system geometry in the form of a computational mesh, specifies material properties and physics models, and configures boundary conditions of various types. The conventional flow solver, once converged, provides a detailed data set that describes the flow state governed by the flow physics that are being modelled. Various post-processing steps can be taken to assess the performance of the system.

If a change is made to any of the data that defines the problem, then the results of the calculation can change. The degree to which the solution changes depends on how sensitive the flow is to the particular parameter that is being adjusted. Indeed, the derivative of the solution data with respect to that parameter quantifies this sensitivity to first order. Determining these derivatives is the domain of *sensitivity analysis*.

There is a large collection of derivative data that can be computed for a fluid system, given the extensive set of input data that is required, and the extensive flow data that is produced. The matrix of derivatives of output data with respect to input data can be vast. Depending upon the goal of the analysis only a portion of this derivative data may be needed for engineering analysis and decision-making.

The adjoint solver accomplishes the remarkable feat of calculating the derivative of a single engineering observation with respect to a very large number of input parameters simultaneously via a *single computation*. The engineering observation could be a measure of the system performance, such as the lift or drag on an airfoil, or the total pressure drop through a system. Most importantly, the derivatives with respect to the geometric shape of the system are found.

Understanding such sensitivities in a fluid system can provide extremely valuable engineering insight. A system that is highly sensitive may exhibit strong variability in performance due to small variations in manufacturing or variations in the environment in which it is operating. Alternatively, high-sensitivity may be leveraged for fluid control, with a small actuator being able to induce strong variations in behavior. Yet another perspective is that sensitivity of a performance measure implies that the device in

question is not fully optimized and there is still room for improvement—assuming that constraints do not preclude further gains.

The sensitivities of a fluid system provided by an adjoint solver satisfy a central need in gradient-based shape optimization. This makes an adjoint solver a unique and powerful engineering tool for design optimization.

Adjoint data can also play a role in improving solver numerics. Regions of high sensitivity are indicative of areas in the flow where discretization errors can potentially have a strong effect. This information can be used to guide how best to refine a mesh to improve flow solution accuracy.

The process of computing an adjoint solution resembles that for a standard flow calculation in many respects. The adjoint solver solution advancement method is specified, residual monitors configured, and the solver is initialized and run through a sequence of iterations to convergence. One notable difference is that a scalar-valued observation is selected as being of interest prior to starting the adjoint calculation.

Once the adjoint solution is converged the derivative of the observable with respect to the position of each and every point on the surface of the geometry is available, and the sensitivity of the observation to specific boundary condition settings can be found. This remarkable feature of adjoint solutions has been known for hundreds of years, but only in the last 25 years has the significance for computational physics analysis been recognized widely.

The power of this methodology is highlighted when an alternative for assembling the same information is considered. Imagine a sequence of flow calculations in which each point on an airfoil surface is moved in turn a set small distance in the surface-normal direction, and the flow and drag recomputed. If there are N points on the surface, then N flow calculations are required to build the data set. Considering that the same data is provided by a *single* adjoint computation, the adjoint approach has an enormous advantage. Remember that for even modest 3D flow computations there may be many thousands of coordinates or more on a surface.

Once the adjoint is computed it can be used to guide intelligent design modifications to a system. After all, the adjoint sensitivity data provides a map across the entire surface of the geometry of the effect of moving the surface. Design modifications can be most effective if made in regions of high sensitivity since small changes will have a large effect upon the engineering quantity of interest. This principle of making changes to a system in proportion to the local sensitivity is the foundation for the simple gradient algorithm for design optimization.

Once a candidate change in shape or other boundary condition has been selected, the effect of that change can be estimated using the computed derivative data. This amounts to a first order extrapolation using a Taylor series expansion around the baseline flow state. Clearly if a modification is chosen that is large enough that nonlinear effects become important then the accuracy of the predicted change cannot be guaranteed.

The derivatives for shape sensitivity are represented by a vector field defined on the nodes of the computational mesh—both interior and boundary nodes. In practice this vector field can be noisy and is observed to vary rapidly between adjacent nodes, especially for large, turbulent flow problems. Such rapid variations, if used as guidance for shape modification, will lead to geometric surfaces that have severe wrinkles. Such undesirable behavior is avoided if the shape sensitivity field is smoothed appropriately.

There are several ways in which smoothing can be accomplished. Since deformation of the boundary and interior mesh is needed, it is fortunate that mesh morphing technology based on Bernstein polynomials serves a dual purpose. Not only does it provide a valuable smoothing effect on the shape sensit-

ivity field, but it also provides a convenient and stable method for deforming the boundary and interior computational mesh.

In this chapter the different methods that can be used to compute adjoint sensitivity data are described, together with the specific strategy adopted for the current adjoint solver. The mechanism for encoding the observable of interest is described, together with the definition of shape-sensitivity data. The basic principles of the mesh deformation method are also described.

1.1.1. General Observables

Several observables are available and serve as the foundation for specifying the quantity that is of interest for the computation. Basic quantities such as forces and moments can be defined and each is given a name. The following observables are available:

- Force: the aerodynamic force in a specified direction on one or more walls.
- Moment of force: the aerodynamic moment about a specified moment center and moment axis. The integration is made over a specified collection of wall zones. Note that in 2D, the moment axis is normal to the plane of the flow.
- Swirl: the moment of the mass flow relative to an axis defined by a point, \underline{r}_c , and a direction \underline{d} . The two options are:

- Swirl integral

$$\int_V \rho(\underline{r} \times \underline{u}) \cdot \underline{d} dV \quad (1.1)$$

where V denotes the volume over which the integration is made, and \underline{r} denotes the relative position to the point \underline{r}_c .

- Normalized swirl integral

$$\frac{\int_V \rho(\underline{r} \times \underline{u}) \cdot \underline{d} dV}{\int_V \rho(\underline{r} \times (\underline{d} \times \underline{r})) \cdot \underline{d} dV} \quad (1.2)$$

Note

These integrals can be used to construct quantities such as tumble ratio that are of significance for internal combustion engine analysis.

- pressure drop between an inlet (or group of inlets) and an outlet (or group of outlets)
- fixed value: a simple fixed value can be specified and used in the assembly of the observables. This is convenient when some scaling or normalization of the observable is desired. It must be noted that this value is treated strictly as a constant for the purposes of any derivative calculations.
- surface integral: a variety of surface integrals can be constructed for a specified field variable on a set of user-selected surfaces:

- Facet sum: a simple sum of the field value on each facet of the computational mesh

$$\sum_f \varphi_f \quad (1.3)$$

- Facet average: the facet sum is divided by the total number of facets, N_f

$$\bar{\varphi} = \frac{\sum_f \varphi_f}{N_f} \quad (1.4)$$

- Facet variance: the sum of the squares of the deviations from the facet average

$$\frac{\sum_f (\varphi_f - \bar{\varphi})^2}{N_f} \quad (1.5)$$

- Integral: the sum of the field value on each face multiplied by the face area

$$\sum_f |A_f| \varphi_f \quad (1.6)$$

- Area-weighted average: the integral divided by the total face area

$$\frac{\sum_f |A_f| \varphi_f}{\sum_f |A_f|} \quad (1.7)$$

- Area-weighted variance: the sum of the squares of the deviations from the area-weighted average, divided by the total face area

$$\frac{\sum_f |A_f| (\varphi_f - \bar{\varphi})^2}{\sum_f |A_f|} \quad (1.8)$$

- Mass-weighted integral: the sum of the field value on each face, weighted by the magnitude of the local mass flow through the face.

$$\sum_f |\rho \underline{u}_f \cdot \underline{A}_f| \varphi_f \quad (1.9)$$

- Mass-weighted average integral: the mass-weighted integral divided by the sum of the magnitudes of the local mass flow rates through the faces on which the integral is defined.

$$\frac{\sum_f |\rho \underline{u}_f \cdot \underline{A}_f| \varphi_f}{\sum_f |\rho \underline{u}_f \cdot \underline{A}_f|} \quad (1.10)$$

- Mass-weighted variance: the mass-weighted integral of the square of the deviation from the mass-weighted average, divided by the sum of the magnitudes of the local mass flow rates through the faces on which the integral is defined.

$$\frac{\sum_f |\rho \underline{u}_f \cdot \underline{A}_f| (\phi_f - \bar{\phi})^2}{\sum_f |\rho \underline{u}_f \cdot \underline{A}_f|} \quad (1.11)$$

- Flow-rate weighted: the rate at which the field is convected through the defined surfaces.

$$\sum_f (\rho \underline{u}_f \cdot \underline{A}_f) \phi_f \quad (1.12)$$

- The field variables that can be used are:
 - Pressure
 - Total pressure
 - Mass flow per unit area
 - Temperature (when adjoint energy is enabled)
 - Heat flux (when adjoint energy is enabled)

1.1.2. General Operations

Several operations are available to combine observables in various ways, if needed, to make a wide variety of compound observables. The following operations are available:

- ratio of two quantities
- product of two quantities
- linear combination of N quantities $q_0 \dots q_{N-1}$, with constant coefficients $a_0 \dots a_{N-1}$ raised to various powers $v_0 \dots v_{N-1}$, plus a constant offset c .

$$c + \sum_{i=0}^{N-1} a_i q_i^{v_i} \quad (1.13)$$

- arithmetic average of N quantities $q_0 \dots q_{N-1}$

$$\bar{q} = \frac{\sum_{i=0}^{N-1} q_i}{N} \quad (1.14)$$

- mean variance of N quantities

$$\frac{\sum_{i=0}^{N-1} (q_i - \bar{q})^2}{N} \quad (1.15)$$

- unary operation (sin, cosine, etc.) on observable value q :

$$- 1 / q$$

- $|q|$
- \sqrt{q}
- $\sin q$
- $\cos q$
- $\tan q$
- $\sin^{-1} q$
- $\cos^{-1} q$
- $\tan^{-1} q$
- $\ln q$
- $\log_{10} q$

1.2. Discrete Versus Continuous Adjoint Solver

When an adjoint solver is developed there is a key design decision that is made regarding the implementation, namely whether to use a *continuous* or a *discrete* approach. Both approaches are intended to compute the same sensitivity data—however, the approaches are remarkably different.

While a user of the adjoint solver may experience no particular difference in workflow for the two separate solver types, it is important to be aware that there are two primary classes of adjoint solver and the approach chosen can have implications for the accuracy of the results.

A *continuous adjoint solver* relies heavily on mathematical properties of the partial-differential equations that define the physics of the problem. In this case those equations are the Navier-Stokes equations. With this approach an adjoint partial differential equation set is formulated explicitly and is accompanied by adjoint boundary conditions that are also derived mathematically. Only after this derivation is complete can the adjoint partial differential equations be discretized and solved, often with extensive re-use of existing solver machinery. This class of solver was implemented by ANSYS Fluent as a research effort.

Such a solver has the benefit that it is decoupled largely from the original flow solver. They share only the fact that they are based on the Navier-Stokes equations. The process of discretizing and solving the partial differential equations in each case could in principle be very different indeed. While this flexibility may be appealing it can also be the downfall of the approach. Inconsistencies in modelling, discretization and solution approaches can pollute the sensitivity information significantly, especially for problems with wall functions and complex engineering configurations such as those of interest to ANSYS Fluent users.

The continuous adjoint approach can be effective for some classes of problem. However, until there is a significant advance in handling some key challenges, ANSYS Inc. has concluded that it is an unsuitable approach for meeting the needs of our broad client base for the classes of problem of interest to them.

A *discrete adjoint solver* is based not on the form of the partial differential equations governing the flow, but the particular discretized form of the equations used in the flow solver itself. The sensitivity of the

discretized equations forms the basis for the sensitivity calculation. In this approach the adjoint solver is much more tightly tied to the specific implementation of the original flow solver. This has been observed to yield sensitivity data that provides valuable engineering guidance for the classes of problem of interest to ANSYS Fluent users, including problems with wall functions.

For the above reasons, the discrete adjoint approach has been adopted for the ANSYS Fluent adjoint solver.

1.3. Discrete Adjoint Solver Overview

As discussed in the previous section, ANSYS, Inc. has chosen to adopt a discrete adjoint approach to solving the adjoint problem since it is believed to provide the most useful engineering sensitivity data for the classes of problems of interest to ANSYS clients.

An adjoint method can be used to compute the derivative of an observation of interest for the fluid system with respect to all the user-specified parameters, *with any changes that arise in the flow variables themselves eliminated*. There are three key ingredients to consider when developing the method:

1. All of the user-specified inputs:

- All values set by a user in the boundary condition panels for each boundary in the problem.
- The computational mesh. More specifically the locations of the nodes of the mesh and how they define the edges, faces, and ultimately the cells used in the finite-volume computation. This includes both interior as well as boundary nodes.
- Material properties.
- Model parameters such as model coefficients for turbulence models.

Note that the settings that define the problem are being distinguished from settings that define how the solution advancement is to be performed to converge the problem. Only the former are of interest here. For the sake of clarity, let us denote the vector of all of the values in the list by \underline{c} . These are considered to be the control variables for the problem, that is, the variables that a user can set explicitly that affect the solution.

It is worth noting that the topological definition of the mesh is fixed—it is not considered to be a control variable here. The effect of collapsing cells or remeshing on the flow solution is not addressable without further development in view of the discrete changes that are implied. This topic is beyond the scope of the current document.

2. The governing equations for the fluid system:

The main effort in a flow computation is in the determination of the flow state, namely the velocity, pressure, density and possibly other fluid-related variables. For a cell-centered finite-volume scheme, the flow state is defined at the cell centroids by a vector of real values. In the simplest case these values are the pressure and flow velocity components. Let the vector of the variables in the ν th cell be denoted here by \underline{q}^ν .

At convergence the flow variables satisfy

$$\mathcal{R}_i^\mu(\underline{q}^0, \underline{q}^1, \dots, \underline{q}^{M-1}; \underline{c}) = 0, \quad \mu = 0, \dots, M-1, \quad i = 0, \dots, L-1 \quad (1.16)$$

where M is the number of cells in the problem, and there are L conditions on each cell. This expression is a compact way of denoting conservation of mass and momentum and other constraints.

3. The engineering observation of interest:

Let

$$\mathcal{J}(\underline{q}^0, \underline{q}^1, \dots, \underline{q}^{M-1}; \underline{c}) \quad (1.17)$$

denote a scalar of interest that depends both on the flow state and perhaps directly on the control variables. It is assumed that the observable is differentiable with respect to both the flow and the controls. The inclusion of the control variables here is essential since in many cases the mesh geometry is included directly in the evaluation of the observable. For example, the evaluation of the force on a boundary involves the wall normal and face areas, which change when mesh nodes are moved.

The goal is to determine the sensitivity of the observation with respect to the user-specified control variables. What makes defining this relationship more challenging is the fact that changing the user inputs changes the flow, which indirectly changes the engineering observation. The adjoint method has a specific role in managing this chain of influences by providing a mechanism for eliminating the specific changes that happen in the flow whenever the inputs change.

If a variation δc_j is introduced into the control variables then a linearization of the governing equations (Equation 1.16 (p. 7)) shows that the variations in the flow state δq_j^v must satisfy

$$\frac{\partial R_i^\mu}{\partial q_j^v} \delta q_j^v = - \frac{\partial R_i^\mu}{\partial c_j} \bigg|_q \delta c_j, \quad \mu=0, \dots, M-1, \quad i=0, \dots, L-1. \quad (1.18)$$

where there is an implied summation over j and v , and $\big|_q$ denotes that the flow solution is held constant while the derivative is taken.

Meanwhile, if both the control variables and the flow state change, then the observation will change:

$$\delta \mathcal{J} = \frac{\partial \mathcal{J}}{\partial q_j^v} \delta q_j^v + \frac{\partial \mathcal{J}}{\partial c_j} \bigg|_q \delta c_j. \quad (1.19)$$

The particular way in which the flow responds to the changes in the control variables can be computed using (Equation 1.18 (p. 8)) only after specific changes, δc_j , have been chosen. It is prohibitive to consider solving (Equation 1.18 (p. 8)) for more than a handful of prescribed changes δc_j because of the excessive computing time that would be needed. However, when redesigning the shape of parts of a system there may be pressure to explore a large number of candidate modifications. This conflict is reconciled by eliminating the variations of the flow solution from the expression (Equation 1.19 (p. 8)) and producing an explicit relationship between changes in the control variables and the observation of interest.

This is accomplished by taking a weighted linear combination of the linearized governing equations (Equation 1.18 (p. 8)) in a very particular way. A set of adjoint variables \tilde{q}^μ is introduced with a one-to-one correspondence with the governing equations (Equation 1.16 (p. 7)). This results in a relationship

$$\left[\tilde{q}_i^\mu \frac{\partial R_i^\mu}{\partial q_j^v} \right] \delta q_j^v = - \tilde{q}_i^\mu \frac{\partial R_i^\mu}{\partial c_j} \bigg|_q \delta c_j. \quad (1.20)$$

The term in square brackets on the left is now matched to the coefficient for the variation in the flow in (Equation 1.19 (p. 8)) in order to define values for the adjoint variables:

$$\frac{\partial R_i^\mu}{\partial q_j^v} \tilde{q}_i^\mu = \frac{\partial \mathcal{J}}{\partial q_j^v}. \quad (1.21)$$

These are the discrete adjoint equations, and the solution of this system that is the primary goal for the adjoint solver. It is important to recognize that these equations have not been derived. They are defined in this way with a specific goal in mind—the elimination from (Equation 1.19 (p. 8)) of the perturbations to the flow field as shown below:

$$\begin{aligned} \delta \mathcal{J} &= \frac{\partial \mathcal{J}}{\partial q_j^v} \delta q_j^v + \frac{\partial \mathcal{J}}{\partial c_j} \bigg|_q \delta c_j \\ &= \tilde{q}_i^\mu \frac{\partial R_i^\mu}{\partial q_j^v} \delta q_j^v + \frac{\partial \mathcal{J}}{\partial c_j} \bigg|_q \delta c_j \\ &= \left\{ \frac{\partial \mathcal{J}}{\partial c_j} \bigg|_q - \tilde{q}_i^\mu \frac{\partial R_i^\mu}{\partial c_j} \bigg|_q \right\} \delta c_j \end{aligned} \quad (1.22)$$

Note the use of Equation 1.20 (p. 9) and Equation 1.21 (p. 9) in the derivation of Equation 1.22 (p. 9). The flow perturbation has now been eliminated from the expression, yielding a direct relation between the control variables and the observable of interest.

There are several important observations to be made about (Equation 1.21 (p. 9)):

- The dimension of the problem to be solved is the same as the original flow problem, although the adjoint problem is linear.
- While the adjoint solution may be considered strictly as a vector of numeric values, experience with the continuous adjoint provides guidance on how the adjoint solution can be interpreted. The vector of weights associated with the components of the residual of the momentum equation in each cell is denoted the *adjoint velocity*. The adjoint value associated with the residual of the continuity equation is termed the *adjoint pressure*.
- The right hand side is defined purely on the basis of the observable that is of interest.
- The matrix on the left hand side is the *transpose* of the Jacobian of the governing system of equations (Equation 1.16 (p. 7)). This seemingly innocent transposition has a very dramatic impact on how the adjoint system is solved. This will be discussed below.
- The adjoint equations are defined by the current state of the flow, and the specific physics that is employed in the modeling. Each adjoint solution is specific to the flow state.

At first glance it appears that solving the adjoint problem may be straightforward. After all it simply involves setting-up and solving a linear (albeit large) system of equations. In practice both steps can represent a significant challenge, especially when the problem is large.

The evaluation of the residuals of the flow equations is an integral part of the pre-existing ANSYS Fluent flow solvers. However, it is necessary to compute the Jacobian of the system $\partial R_i^\mu / \partial q_j^\nu$ and then transpose it, or at the very least to be able to make a matrix-free transpose matrix-vector product with the adjoint solution. There are several technical approaches to accomplishing this task whose description goes beyond the scope of this document. Suffice to say that it is not trivial to encode this functionality, but that it has been done successfully here.

For the present implementation, a pre-conditioned iterative scheme, based on pseudo-time marching, is adopted to solve the adjoint system. The equations that define the advancement process can be written as

$$\left\{ \frac{P_{i,j}^{\mu,\nu}}{CFL \Delta t} + L_{ij}^{\mu,\nu} \right\} \Delta \tilde{q}_i^\nu = \frac{\partial \mathcal{J}}{\partial q_j^\mu} - \frac{\partial R_i^\nu}{\partial q_j^\mu} \tilde{q}_i^\nu \quad (1.23)$$

where Δt is a local time step size based on the local flow conditions, CFL is a user-specified CFL number. The matrix $L_{ij}^{\mu,\nu}$ is a simplified form of the system Jacobian that is amenable to solution using the AMG linear solver that is the workhorse of the conventional flow solver. The preconditioning matrix $P_{i,j}^{\mu,\nu}$ is a diagonal matrix. Artificial compressibility is introduced on the adjoint continuity equation to aid in the relaxation of the adjoint pressure field.

The correction $\Delta \tilde{q}_i^\mu$ is under-relaxed and added to the adjoint solution \tilde{q}_i^μ and the process repeated until the right hand side is adequately small.

1.4. Adjoint Solver Stabilization

When applied to problems with large cell counts and complex geometry, adjoint solvers sometimes experience stability issues. These instabilities can be associated with small scale unsteadiness in the flow field and/or strong shear, and tend to be restricted to small parts of the flow domain. Despite the spatial localization of these instabilities, the linearity of the adjoint problem provides no intrinsic limit on their growth during solution advancement. Their presence, if not handled, can disrupt the entire adjoint calculation despite the problem occurring in sometimes just a few cells.

Two stabilized solution advancement schemes are available in Fluent in order to overcome these stability difficulties when larger cases are being solved. The stabilization schemes are designed to intervene only when the standard advancement scheme is experiencing instability.

Both schemes operate by identifying unstable growth patterns in the adjoint solution process and applying a more stable solution advancement strategy. The **spatial** scheme operates by identifying parts of the domain where unstable growth is occurring and applying a more direct and stable solution procedure in those regions. The **modal** scheme involves a process of identifying the particular details of the unstable growth patterns or modes. These patterns are localized in space and they are used to split the solution into parts that have stable and unstable characteristics when advanced. The stable part is advanced as usual, while the algorithm is designed now to compensate for the unstable part so that the overall calculation is stabilized. Some additional computational overhead is associated with both schemes, although the modal scheme is observed to demand less memory than the spatial scheme for typical problems.

The emergence of the unstable patterns can happen at any time during the adjoint calculation and the total number of unstable patterns is case-dependent. Having 10 to 20 unstable modes present would not be considered unusual. Large cases may have many more. As a general trend, the most rapidly-growing unstable patterns appear often within a few iterations with more slowly-growing modes appearing later in the calculation. The unstable patterns are handled as they appear.

For information about using the stabilization schemes, refer to [Stabilized Scheme Settings \(p. 26\)](#).

1.5. Solution-Based Adaption

An adjoint solution provides guidance on where best to adapt a computational mesh in order to resolve quantities of engineering interest.

Once the governing equations for the system ([Equation 1.16 \(p. 7\)](#)) have been converged there remains a discretization error, ε_i^μ , such that

$$\mathcal{R}_i^\mu(\underline{q}^0, \underline{q}^1, \dots, \underline{q}^{M-1}; \underline{c}) = \varepsilon_i^\mu, \quad \mu=0, \dots, M-1, \quad i=0, \dots, L-1. \quad (1.24)$$

While specific estimates for this discretization error may be tricky to define, it is often estimated to be $O(h^p)$, where h is the local grid size, and p is the order of the discretization scheme. That is, $p=1$ for a first-order scheme and $p=2$ for a second order scheme. Alternatively, ε_i^μ can be considered to be the residual associated with a solution that is not converged fully.

The correction to the flow field, δq_j^v , that compensates for this inhomogeneity is given by

$$\frac{\partial R_i^\mu}{\partial q_j^v} \delta q_j^v = -\varepsilon_i^\mu \quad (1.25)$$

from which it follows quickly that

$$\delta \mathcal{J} = \frac{\partial \mathcal{J}}{\partial q_j^v} \delta q_j^v = -\varepsilon_i^\mu \tilde{q}_i^\mu. \quad (1.26)$$

This simple expression provides an estimate of the effect of the presence of ε_i^μ on the observation, \mathcal{J} .

The presence of discretization errors, or lack of convergence, on the engineering quantity of interest is assessed by weighting the inhomogeneous term by the local adjoint solution. It is clear that even in regions of the domain where the residuals or discretization errors are small, an accompanying adjoint velocity or pressure that is large in magnitude implies that there may be a significant source of error in the observable. A finer mesh in regions where the adjoint is large will reduce the influence of discretization errors that may adversely affect the engineering result of interest. In practice, adapting cells which have large magnitude adjoint velocity and/or adjoint pressure will achieve this goal.

1.6. Using The Data To Improve A Design

Adjoint sensitivity data can be used to guide how to modify a system in order to improve the performance. The observable of interest can be made larger or smaller, depending upon the engineering goal.

A common strategy for deciding how to modify the system is based on the gradient algorithm. The underlying principle is quite simply that modifying a system in a manner to which it is most sensitive

maximizes the effect of the change. The change to a control variable is made in proportion to the sensitivity of the value of interest with respect to that control variable.

Denote the sensitivity of the cost with respect to shape by

$$\delta \mathcal{J} = \frac{\partial \mathcal{J}}{\partial x_j^n} \delta x_j^n \quad (1.27)$$

where x_j^n is the j th coordinate of the n th node in the mesh. Here x_j^n is a notation for the subset of the control variables c_j for the system that correspond to mesh node positions. Then an adjustment

$$\delta x_j^n = \lambda \frac{\partial \mathcal{J}}{\partial x_j^n} \quad (1.28)$$

will provide the maximum adjustment to \mathcal{J} for given L^2 norm of δx_j^n , where λ is an arbitrary scaling factor. Note that λ can be picked to be positive or negative depending upon whether \mathcal{J} is to be increased or decreased respectively. This is essentially a statement of the method of steepest descent.

Furthermore, the change is estimated to first order to be

$$\delta \mathcal{J} = \lambda \frac{\partial \mathcal{J}}{\partial x_j^n} \frac{\partial \mathcal{J}}{\partial x_j^n}. \quad (1.29)$$

For a sufficiently small adjustment, the change to the observation will strictly have the same sign as the scaling factor λ , provided the gradient is not identically zero.

In regions where the sensitivity is high, small adjustments to the shape will have a large effect on the observable. This corresponds directly with the idea of engineering robustness. If a configuration shows high sensitivities, then the performance will likely be subject to large performance variations if there are manufacturing inconsistencies. For a robust design, the goal is to have the sensitivity be tolerably small.

It is noted that in practical cases, the field $\partial \mathcal{J} / \partial x_j^n$ can be noisy. If the noisy field is used directly to modify a boundary shape using (Equation 1.28 (p. 12)), then the modified surface can have many inflections. This is not helpful for engineering design work. In the next section, the use of mesh morphing technology not only to smooth the sensitivity field, but also to provide smooth boundary and interior mesh deformation, is described.

1.7. Smoothing and Mesh Morphing

As has been noted, for typical engineering problems, the shape sensitivity field can have smoothness properties that are not adequate to define a shape modification. Mesh morphing technology is used here for both two- and three-dimensional systems in a two-fold role. The first role is as a smoother for the surface sensitivity field. The second role is to provide smooth distortions not only of the boundary mesh, but also the interior mesh. This approach is very appealing since it functions for arbitrary mesh cell types.

A rectangular control volume is picked that encloses the boundary, or part of the boundary, whose shape is to be modified. A regular array of control points is then distributed in the control volume.

The properties of Bernstein polynomials are then invoked to define a local coordinate system mapping. The standard coordinates, \underline{x}^k , of each boundary and interior node in the mesh, lying within the control volume, are defined by a local coordinate, (u^k, v^k) . In a control volume, with $\ell \times m$ control points, the linear relationship between the k th grid node position and the ij th control point location is

$$\underline{x}^k = \sum_{i,j=0}^{\ell,m} \xi^{ij} B_{i,\ell}(u_k) B_{j,m}(v_k) \quad (1.30)$$

where $B_{i,\ell}(u)$ is the i th Bernstein polynomial of degree ℓ ,

$$B_{i,\ell}(u) = \binom{\ell}{i} u^i (1-u)^{\ell-i} \quad (1.31)$$

and ξ^{ij} denotes the coordinate of the ij th control point.

If the regular array of control points is modified by moving one or more control points, the mapping provides a smooth repositioning of the grid nodes.

It has already been discussed in [Using The Data To Improve A Design \(p. 11\)](#) how the variation in the observation varies with mesh node locations:

$$\delta \mathcal{J} = \frac{\partial \mathcal{J}}{\partial x_j^n} \delta x_j^n \quad (1.32)$$

Using the mapping between the mesh nodes and control points, shows that

$$\delta \mathcal{J} = \sum_{i,j=0}^{\ell,m} \delta \xi^{ij} \cdot \tilde{W}^{ij} \quad (1.33)$$

where

$$\tilde{W}^{ij} = \sum_n \frac{\partial \mathcal{J}}{\partial x^n} B_{i,\ell}(u_n) B_{j,m}(v_n) \quad (1.34)$$

is the control point sensitivity field, and $\delta \xi^{ij}$ denotes the adjustment to the control point position.

Since there are typically fewer control points than grid nodes, the summation operation in ([Equation 1.34 \(p. 13\)](#)) has a smoothing effect on the sensitivity field.

A simple gradient algorithm based on choosing

$$\delta \xi^{ij} = \lambda \tilde{W}^{ij} \quad (1.35)$$

leads to a smooth boundary mesh deformation that improves the design, for λ sufficiently small. The added benefit is that a smooth volume mesh deformation is also defined.

Continuity of the mesh displacement derivatives at the control volume perimeter to order p can be preserved by explicitly setting the control point displacement field to be zero for control points in layers adjacent to the control volume boundary. If no control point displacement is permitted in layers 0 through p and $\ell-p$ through ℓ in the x -coordinate direction, and likewise in the y -direction, the properties of Bernstein polynomials guarantee the enforcement of the continuity condition.

If walls that are not permitted to move intersect the control volume, an additional set of constraints is placed on the control point gradient field. Any component of the field that gives rise to movement of fixed points is removed. The positions of nodes that are constrained not to move are not updated during the mesh deformation process. It must be noted that constraints of this type can affect the gradient to the extent that following the gradient is not guaranteed to lead to a favorable change in the observable.

Chapter 2: Using the Adjoint Solver Module

This chapter describes the process for loading the adjoint solver add-on, as well as setting up, running, and post-processing the adjoint solutions. Also, this chapter demonstrates the shape modification process that is guided by the adjoint solution.

The typical use of the adjoint solver involves the following steps:

1. Load or compute a conventional flow solution.
2. Load the adjoint solver add-on module.
3. Specify the observable of interest.
4. Set the adjoint solver controls.
5. Set the adjoint solver monitors and convergence criteria.
6. Initialize the adjoint solution and iterate to convergence.
7. Post-process the adjoint solution to extract the sensitivity of the observable with respect to boundary condition settings.
8. Post-process the adjoint solution to extract the sensitivity of the observable with respect to shape of the geometry.
9. Modify boundary shapes based on shape-sensitivity data and recompute the flow solution.

This chapter provides information about using the ANSYS Fluent adjoint solver in the following sections:

- [2.1. Installing the Adjoint Solver Module](#)
- [2.2. Loading the Adjoint Solver Module](#)
- [2.3. Understanding the Scope of the Functionality](#)
- [2.4. Defining Observables](#)
- [2.5. Using the Adjoint Solution Methods Dialog Box](#)
- [2.6. Using the Adjoint Solution Controls Dialog Box](#)
- [2.7. Working with Adjoint Residual Monitors](#)
- [2.8. Running the Adjoint Calculation](#)
- [2.9. Postprocessing of Adjoint Solutions](#)
- [2.10. Modifying the Geometry](#)
- [2.11. Using the Adjoint Solver Module's Text User Interface](#)

In addition, see [Tutorial: 2D Laminar Flow Past a Cylinder \(p. 43\)](#) for more details about using the adjoint solver.

2.1. Installing the Adjoint Solver Module

The adjoint solver model is provided as an add-on module with the standard ANSYS Fluent licensed software. The module is installed with the standard installation of ANSYS Fluent in a directory called `addons/adjoint` in your installation area. The adjoint solver module consists of a UDF library and a

pre-compiled scheme library, which need to be loaded and activated before calculations can be performed.

2.2. Loading the Adjoint Solver Module

The adjoint solver module is loaded into ANSYS Fluent through the text user interface (TUI). The module can be loaded only when a valid ANSYS Fluent case file has been set or read. The text command to load the module is

```
define → models → addon-module
```

A list of ANSYS Fluent add-on modules is displayed:

The adjoint solver module is loaded as follows:

```
> /define/models/addon-module
Fluent  Addon Modules:
  0. None
  1. MHD Model
  2. Fiber Model
  3. Fuel Cell and Electrolysis Model
  4. SOFC Model with Unresolved Electrolyte
  5. Population Balance Model
  6. Adjoint Solver
  7. Battery Module
Enter Module Number: [0] 6
```

Select the adjoint solver model by entering the module number 6. During the loading process a scheme library containing the graphical and text user interface, and a UDF library containing a set of user-defined functions (UDFs) are loaded into ANSYS Fluent. The **Adjoint** menu item will appear in the main menu.

2.3. Understanding the Scope of the Functionality

The current adjoint solver implementation provides basic adjoint solutions that accompany a conventionally-computed flow solution provided certain criteria are met. When the adjoint solver is initialized, or observables evaluated, and before iterations are performed, a series of checks is performed to determine the suitability of the existing flow solution for analysis with the adjoint solver. Two types of message may appear:

- *Warning message:* There are certain physics models and boundary condition types that are not explicitly modeled in the adjoint solver, but their absence does not disallow you from proceeding with the calculation. In this case, a warning message will be printed that explains the nature of the inconsistency.

```
Checking adjoint setup...
-- Warning: Model is active but not included in adjoint calculation: Cell zone conditions - Porous zone
-- Warning: Model is active but not included in adjoint calculation: P1 radiation model Done
```

The adjoint solver will still run in this case but the quality of the adjoint solution data can be expected to be poorer as a result of the inconsistency. This is because in such cases the unsupported settings will revert to corresponding supported settings for the adjoint solver (for example, a porous zone will be viewed as a non porous fluid zone, and second order discretization will be treated as first order). Though the calculation will proceed, it is important to note that the results produced should be considered on this basis. When reverting back to the fluid calculation, the original settings will be preserved and the modified settings are not migrated onto the original case. The warning indicates that the setting change will be made automatically and specifically for the adjoint solution.

- *Error message:* There are some model and boundary conditions that are incompatible with the computation of an adjoint solution with the current adjoint solver implementation. In this case, an explanatory error message will be printed in the console window.

```
Checking adjoint setup...
** Unable to proceed: Model is active but not compatible with adjoint calculation:
   Transition SST turbulence model
Done
```

The adjoint solver will not run in this case and changes must be made manually to the settings identified in the console before the solver can be run.

The adjoint solver is implemented on the following basis:

- The flow state is for a steady incompressible single-phase flow that is either laminar or turbulent lying in an inertial frame of reference.
- For turbulent flows a *frozen turbulence* assumption is made, in which the effect of changes to the state of the turbulence is not taken into account when computing sensitivities.
- For turbulent flows standard wall functions are employed on all walls.
- The adjoint solver uses methods that are first order accurate in space by default. If desired, you can select second order accurate methods (see [Using the Adjoint Solution Methods Dialog Box \(p. 23\)](#)).
- The boundary conditions are only of the following type
 - Wall
 - Velocity inlet
 - Pressure outlet
 - Symmetry
 - Rotational and translation periodic

It is important to note that these requirements are not strict conditions for the conventional flow solver, but rather modeling limitations for the adjoint solver.

2.4. Defining Observables

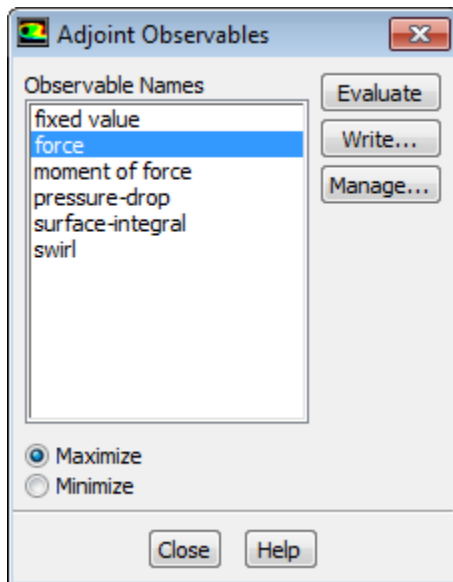
The definition of the observable is a key initial step that must be performed for any adjoint calculation. The observable that is defined is the quantity with respect to which derivatives are sought.

Important

The observable setup procedure was changed starting with Fluent 14.5, however cases that were created and run in earlier versions of Fluent can be still be imported and their characteristics will be respected.

To define an observable, click the **Observable...** option under the **Adjoint** menu to open the **Adjoint Observables** dialog box ([Figure 2.1: Adjoint Observables Dialog Box \(p. 18\)](#)).

Adjoint → Observable...

Figure 2.1: Adjoint Observables Dialog Box

You can use the **Adjoint Observables** dialog box to evaluate and manage your predefined observables. Clicking the **Evaluate** button computes the specified quantity (for example, the total pressure drop based on the selected inlets and outlets), and prints the result in the console window. Clicking the **Write...** button provides the option to write the result to a named file. Clicking the **Manage...** button displays the **Manage Adjoint Observables** dialog box (see [Managing Adjoint Observables \(p. 19\)](#)).

The **Maximize** and **Minimize** options are available to assist in postprocessing the adjoint observables. For example, if a drag observable type is selected, then you can indicate that you want to minimize drag. Likewise for a lift observable type, you can indicate that you want to maximize the lift. The sign on the post-processed fields is adjusted accordingly. A standard rule of thumb can be applied: if you want to improve the solution, then follow the direction of the sensitivity vectors. This can apply to shape sensitivity, boundary velocity sensitivity, optimal displacement fields, etc. Another standard rule of thumb can also be applied for post-processed scalars: if you want to improve the solution, then increase values where the sensitivity is positive and/or decrease values where the sensitivity is negative.

Note

The observables that appear in the **Observables Names** list may be shorter than the list appearing when managing the observables. This is because only observables that have complete definitions are presented for evaluation and use as the subject of an adjoint calculation.

Important

Only one observable can be used in any one adjoint calculation. In the **Adjoint Observables** dialog box, the observable that is currently highlighted is the observable that will be used in the adjoint calculation.

For more information, see the following sections:

[2.4.1. Managing Adjoint Observables](#)

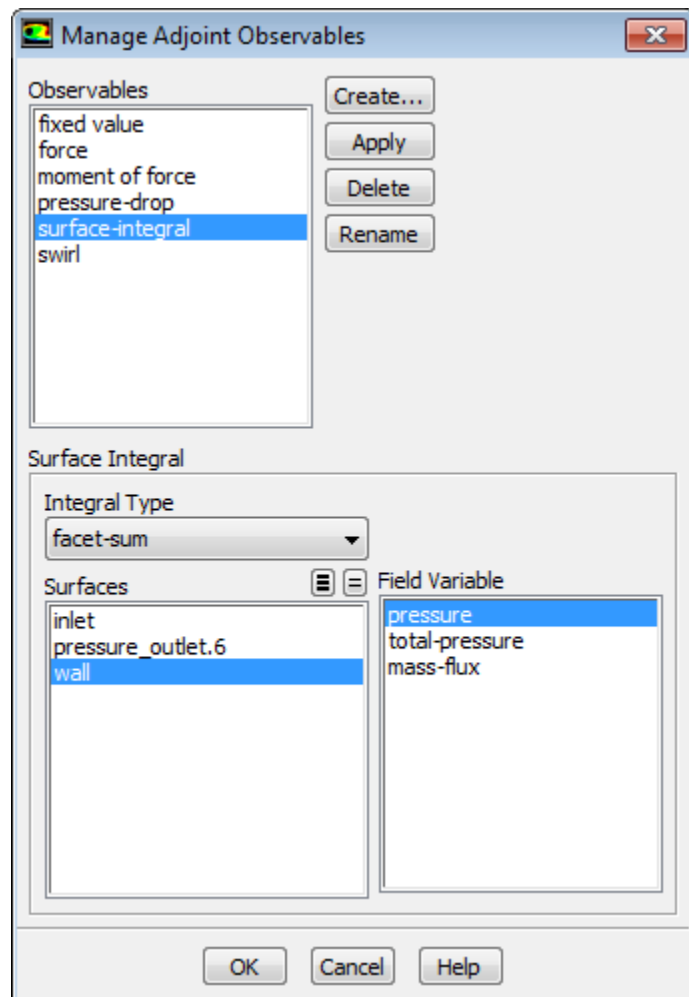
[2.4.2. Creating New Observables](#)

[2.4.3. Renaming Adjoint Observables](#)

2.4.1. Managing Adjoint Observables

You can manage existing observables and create and define new observables by clicking the **Manage...** button in the **Adjoint Observables** dialog box to display the **Manage Adjoint Observables** dialog box (Figure 2.2: Manage Adjoint Observables Dialog Box (Surface Integral) (p. 19)).

Figure 2.2: Manage Adjoint Observables Dialog Box (Surface Integral)



Clicking the **Create...** button displays the **Create New Observables** dialog box (see [Creating New Observables](#) (p. 22)) where you can create an observable of a specific type. Clicking the **Apply** button applies the current settings to the selected observable. Clicking the **Delete** button removes the selected observable from the **Observables** list. Clicking the **Rename** button displays the **Rename Observable** dialog box (see [Renaming Adjoint Observables](#) (p. 23)).

Once an observable is created, it appears in the **Observables** list. Once you select an observable from the **Observables** list, the **Manage Adjoint Observables** dialog box changes to expose various properties that can be assigned for the selected observable (for example, selecting the **pressure-drop** observable allows you to specify the participating inlets and outlets). Once the observable properties are defined, click **Apply** to apply the settings and proceed to define other observables, or click **OK** in the **Manage**

Adjoint Observables dialog box to apply the settings and close the dialog box. Available observables are described in [General Observables \(p. 3\)](#).

Note

It is permissible to leave observable definitions incomplete. The **Manage Adjoint Observables** dialog box can be considered as a workspace for the definition and manipulation of observables. Only those operations with complete definitions will appear in the **Adjoint Observables** dialog box. Operations with undefined fields, or operations that depend on themselves, will be excluded from the list of usable observables.

- To define a **force** observable:
 1. Select **force** in the **Observables** list.
 2. Select the walls that are to contribute to the force of interest in the **Wall Zones** list.
 3. Define the direction in which the force is to be computed by entering the components of this direction in the **X Component** and **Y Component** fields.
- To define the **moment of force** observable:
 1. Select **moment of force** in the **Observables** list.
 2. Select the **X, Y, and Z** (for 3D) components of the **Moment Center**.
 3. Select the **X, Y, and Z** (for 3D) components of the **Moment Axis**.
 4. Select a wall in the **Wall Zones** list.
- To define a **total pressure drop** observable:
 1. Select **total pressure drop** in the **Observables** list.
 2. Select the inlets and outlets between which the total pressure drop is to be computed using the **Inlets** and **Outlets** lists.
- To define an **arithmetic average** observable:
 1. Select **arithmetic average** in the **Observables** list.
 2. Select the number of **Components**.
 3. Select an observable from the corresponding **Observable** list.
- To define a **fixed value** observable, perform the following:
 1. Select **fixed value** in the **Observables** list.
 2. Select the **Value** of the **Constant**.
- To define the **linear combination** observable:
 1. Select **linear combination** in the **Observables** list.

2. Under **Linear Combination of powers**, select a **Constant** value.
 3. Select the number of **Components**.
 4. For each component, select a corresponding **Coefficient**, an **Observable**, and a **Power**.
- To define the **mean variance** observable:
 1. Select **mean variance** in the **Observables** list.
 2. Select the number of **Components**.
 3. Select the corresponding **Observable**.
 - To define the **product** observable:
 1. Select **product** in the **Observables** list.
 2. Select two observables from the corresponding drop-down list that will be used to compute their product.
 - To define the **ratio** observable:
 1. Select **ratio** in the **Observables** list.
 2. Select an observable from the **Numerator** drop-down list to represent the numerator of the ratio.
 3. Select an observable from the **Denominator** drop-down list to represent the denominator of the ratio.
 - To define the **surface integral** observable:
 1. Select **surface integral** in the **Observables** list.
 2. Select the type of integral from the **Integral Type** drop-down list. Available integral types are described in [General Observables \(p. 3\)](#).
 3. Select a **Surface** and the corresponding **Field Variable**.
 - To define the **swirl** observable:
 1. Select **swirl** in the **Observables** list.
 2. Select the **X**, **Y**, and **Z** (for 3D) components of the **Swirl Center**.
 3. Select the **X**, **Y**, and **Z** (for 3D) components of the **Swirl Axis**.
 4. Select a fluid in the **Fluid Zones** list.
 - To define an **unary operation** observable:
 1. Select **unary operation** in the **Observables** list.
 2. Select an operation from the drop-down list (see [General Observables \(p. 3\)](#) for a list of available operators).

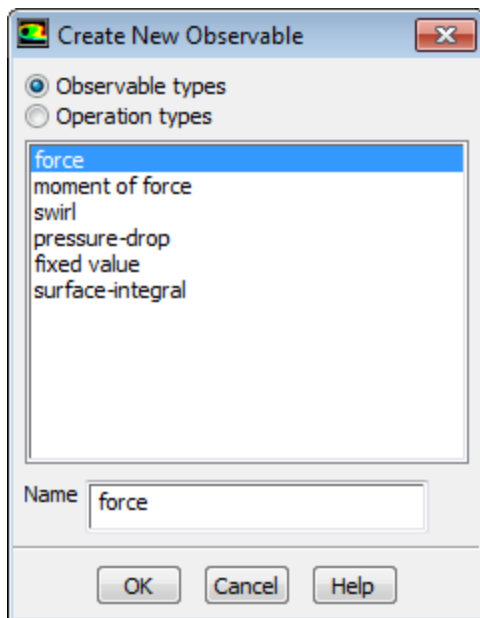
3. Select an observable from the corresponding drop-down list upon which the unary operation is to be applied.

2.4.2. Creating New Observables

You can create a new observable by clicking the **Create...** button in the **Manage Adjoint Observables** dialog box to display the **Create New Observable** dialog box (Figure 2.3: Create New Observable Dialog Box (Observable Types) (p. 22)) where you can choose from several observable types and operation types.

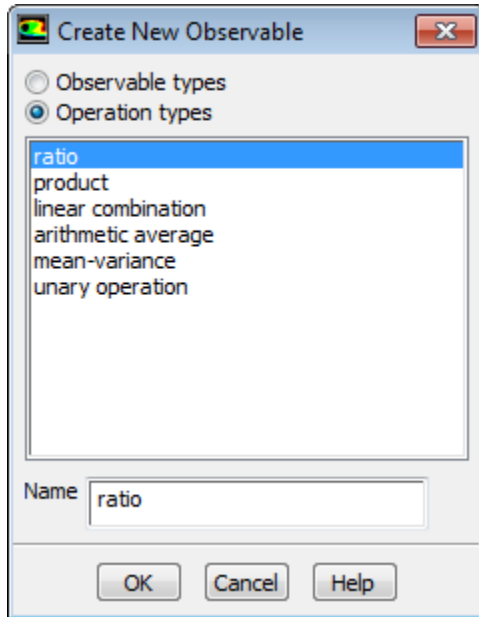
A variety of observables are available to specify the quantity that is of interest for the computation. Basic quantities such as forces and moments can be defined and each can be given a name. There are several types of observables that you can create, described in [General Observables](#) (p. 3).

Figure 2.3: Create New Observable Dialog Box (Observable Types)



Select **Observable types** and pick an observable from the list. Keep the default name, or use the **Name** field to designate a different name for the observable of interest. Click **OK** to create the new observable, or click **Cancel** to dismiss the dialog box without creating an observable.

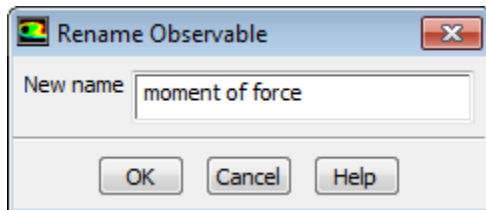
As with the observable types, there are several operations also available that allow you to combine observables in various ways, if needed, to make a wide variety of compound observables. There are several types of operations that you can apply to the created observables, described in [General Observables](#) (p. 3).

Figure 2.4: Create New Observable Dialog Box (Operation Types)

Select **Operation types** and choose an operation from the list. Keep the default name, or use the **Name** field to designate a different name for the observable of interest. Click **OK** to create the new observable, or click **Cancel** to dismiss the dialog box without creating an observable.

2.4.3. Renaming Adjoint Observables

You can apply a new name for a selected observable by clicking the **Rename** button in the **Manage Adjoint Observables** dialog box to display the **Rename Observable** dialog box (Figure 2.5: Rename Adjoint Observable Dialog Box (p. 23)).

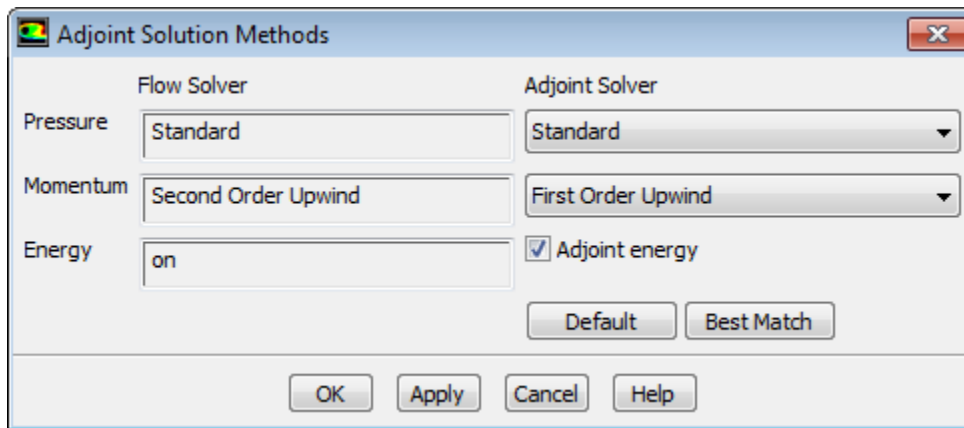
Figure 2.5: Rename Adjoint Observable Dialog Box

Enter a name in the **New name** field and click **OK** to apply a new name for the observable. Any observable that refers to an observable by its old name will automatically be updated to refer to the observable by its new name.

2.5. Using the Adjoint Solution Methods Dialog Box

You can specify the methods used for computing the adjoint solutions in the **Adjoint Solution Methods** dialog box. To open the **Adjoint Solution Methods** dialog box, select the **Methods...** option under the **Adjoint** menu.

Adjoint → **Methods...**

Figure 2.6: Adjoint Solution Methods Dialog Box

The **Adjoint Solution Methods** dialog box shows a side-by-side comparison of the schemes used for the flow solver and for the adjoint solver. Using the same scheme for the adjoint solution and the flow solution yields the most accurate discrete derivative calculation when the adjoint solution is converged. However, not all schemes used for the flow solver are supported for the adjoint solver. In these cases an alternate adjoint scheme must be used. This does not typically lead to severe deterioration of the adjoint results quality. Even if the same scheme is available for the adjoint solver this is not always practical because stability may be reduced with some schemes. Therefore, you can use the drop-down lists under **Adjoint Solver** to select alternate schemes as needed.

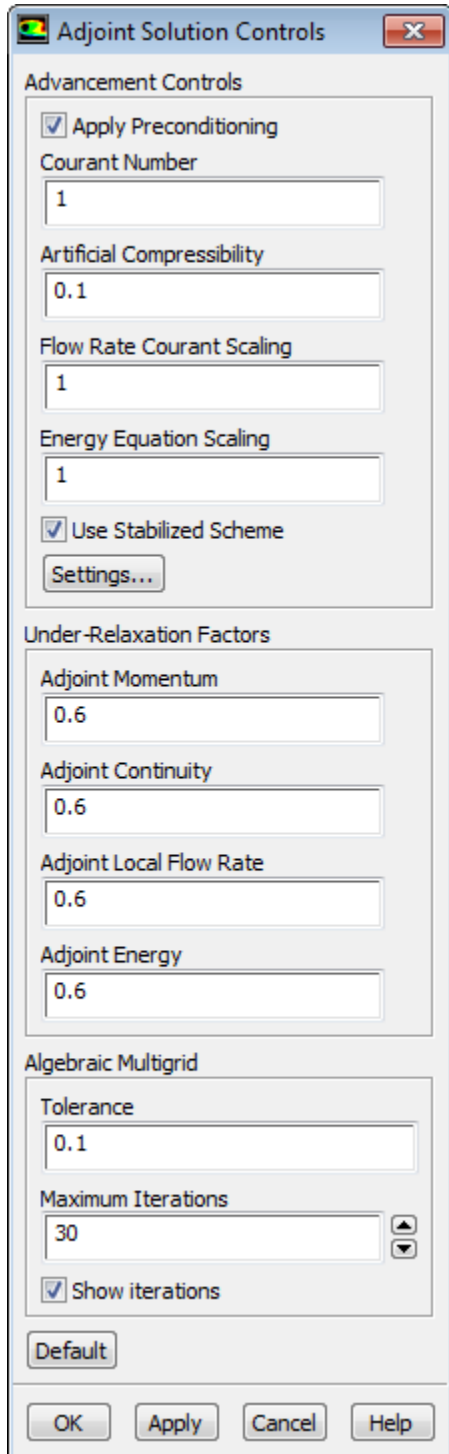
In some cases it may not be desirable to solve the energy adjoint even if energy is solved in the flow solver. For example, consider an incompressible flow where the specified observable does not involve thermal quantities. In this case the adjoint for the energy equation is identically zero, but its inclusion would add an unnecessary numerical burden. You can disable the **Adjoint energy** option to avoid solving the energy adjoint.

Clicking **Default** will configure the adjoint solver to use the default low-order schemes (**Standard** for pressure and **First Order Upwind** for momentum) which are chosen for their stability. Clicking **Best Match** will attempt to match the adjoint schemes to the flow schemes which should in general provide more accurate calculations at convergence. Where they cannot be matched, the default scheme will be used.

2.6. Using the Adjoint Solution Controls Dialog Box

To open the **Adjoint Solution Controls** dialog box, select the **Controls...** option under the **Adjoint** menu.

Adjoint → **Controls...**

Figure 2.7: Adjoint Solution Controls Dialog Box

The solution algorithm for the adjoint solver is similar to the coupled pressure-based solver that is available for conventional flow computations in ANSYS Fluent. The following steps are to be followed:

1. Choose whether or not to use solution preconditioning using the **Apply Preconditioning** option. This is needed for most cases involving turbulent flow. If **Apply Preconditioning** is activated, then
 - a. Select a **Courant Number** for the calculation. A higher number corresponds to a more aggressive advancement of the computation at the risk of instability. Disabling preconditioning corresponds to an infinite Courant number.

- b. Select a level for **Artificial Compressibility**. A non-zero value introduces artificial compressibility into the computation of the adjoint continuity equation. A value of 1.0 or less is reasonable.
 - c. Select values for **Flow Rate Courant Scaling** and, if you are solving the energy adjoint, **Energy Equation Scaling**. These should be values larger than zero, and the default values are 1. A smaller choice implies a less aggressive algorithm that encourages stability of the AMG linear solver.
2. Choose whether or not to use a stabilized solution advancement scheme using the **Use Stabilized Scheme** option. When selected, the adjacent **Settings...** button becomes active. The details of these settings are described in [Stabilized Scheme Settings](#) (p. 26). A stabilized solution scheme will be required to obtain adjoint solutions for problems at high Reynolds number in which there is strong shear and/or complex geometry.
3. Set the **Adjoint Momentum**, **Adjoint Pressure**, and **Adjoint Local Flow Rate** under-relaxation factors. If the adjoint energy equation is enabled in the **Adjoint Solution Methods** dialog box, then you may also specify an under-relaxation factor for **Adjoint Energy**. These values each lie in the range 0.0 to 1.0. A higher value leads to a more aggressive algorithm that is more likely to be less stable. A value of 1.0 for each can be used for some simple cases without difficulty.
4. If desired, the **Tolerance** for convergence, and **Maximum Iterations** for the inner AMG iterations can be adjusted under **Algebraic Multigrid**.
5. The **Show iterations** toggle button under the **AMG controls**, when activated, provides a more verbose iteration history in the main console window during iterations. The details of the inner iteration can be useful when deciding an appropriate **Courant Number**, **Artificial Compressibility**, and **Flow Rate Courant Scaling**. If many inner iterations are needed, or indeed the inner iterations diverge, this signals that a reduction in Courant number may be needed. Alternatively, an increase in **Artificial Compressibility** or a reduction in the **Flow Rate Courant Scaling** may be sufficient to establish a convergent AMG iteration process. Having a convergent AMG iteration process is essential to the convergence of the adjoint solver.
6. The **Default** button provides a set of default values for the various fields in the dialog box.

2.6.1. Stabilized Scheme Settings

If you choose to use one of the stabilization schemes ([Adjoint Solver Stabilization](#) (p. 10)) you will need to specify the scheme and settings in the **Stabilized Scheme Settings** dialog box (accessed by enabling **Use Stabilized Scheme** and clicking **Settings...** in the **Adjoint Solution Controls** dialog box).

The stabilization scheme settings are described in the following sections:

[2.6.1.1. Modal Stabilization Scheme](#)

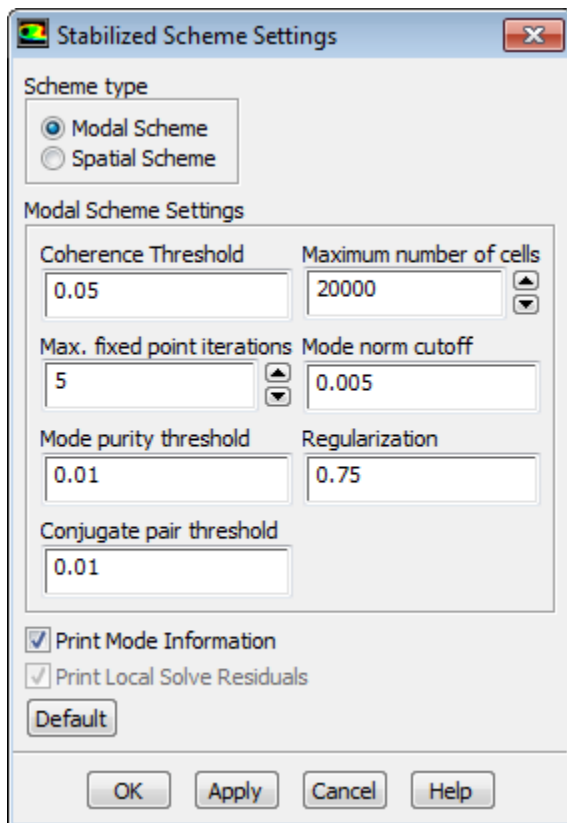
[2.6.1.2. Spatial Stabilization Scheme](#)

2.6.1.1. Modal Stabilization Scheme

The behavior of the modal scheme is as follows. As the calculation progresses the solution is monitored for unstable growing patterns or modes as the adjoint solution is corrected by the primary advancement algorithm. The first stage of the stabilization scheme involves a lightweight monitoring process to screen for any unstable patterns. If the calculation is progressing in a stable manner there is no indication even that the stabilization scheme is operating. If a candidate instability appears then it is monitored. In the event that unstable behavior that cannot be handled by the primary advancement scheme becomes evident a more intensive process is activated. Unstable patterns are refined using a fixed point iteration scheme so that they can be handled cleanly and advanced in a stable manner using an alternative to the primary advancement scheme.

The settings for the modal scheme are described below.

Figure 2.8: Stabilized Scheme Settings for Modal Scheme



Coherence Threshold

controls the preliminary screening for growth. It defines the threshold for relative variance of growth between iterations at which a more detailed search is activated. A smaller value will tend to delay the activation of the more detailed search for unstable growth at the risk of allowing unstable growth to continue for more iterations.

Max. Fixed Point Iterations

is the maximum number of iterations taken to refine the unstable pattern during the more detailed search.

Mode Purity Threshold

is the tolerance for an acceptable unstable pattern. A smaller value will lead to a more pure unstable mode being identified. This may occur at the expense of a larger number of fixed point iterations. If the purity threshold is not met after the maximum specified number of iterations then the unstable pattern is rejected.

Conjugate Pair Threshold

is the threshold for inclusion of a second unstable mode alongside a candidate mode, forming a conjugate pair. This corresponds to a pseudo-time-periodic instability, versus purely exponential growth. Such instabilities can occur in some cases. Increasing this value will increase the occurrence of a second mode being included.

Maximum Number of Cells

is the largest number of cells allowed for an unstable pattern that will be handled by the stabilization scheme. The problematic instabilities for an adjoint solver are often localized in space to a small number

of cells out of the entire flow domain. However, there can be transiently growing corrections to the adjoint solution that masquerade as unstable patterns. This cell limit suppresses such false patterns from consideration.

Mode Norm Cutoff

serves to identify the main body of an unstable pattern and exclude the peripheral spatial parts of the mode that are of small amplitude. If the norm cutoff were set to zero, all unstable modes would involve all cells in the domain. Therefore, reducing this value increases the number of cells that are included in an unstable mode.

Regularization

acts to regularize the advancement of the unstable mode components. The stiffness of the adjoint governing equations can give rise to difficulty in discerning the unstable pattern from nearby stable patterns. Increasing the regularization tends to overcome this difficulty and adds stability to the calculation at the expense of potentially slower convergence. Valid values are between 0 and 1.

Print Mode Information

enables printing of stabilization information to the console.

When using the modal stabilization scheme, the following behavior can be observed:

- When the primary advancement scheme is proceeding in a stable manner, the stabilization scheme will remain idle and no output will be generated.
- When a candidate unstable pattern appears it will be tracked and a message will appear with information about the detected mode:

```
Mode growth:      1.34390e+00  Coherence:      8.28546e-02
```

In the above example, the message indicates that an unstable pattern that grows by more than 34% in one iteration is present. The value of the Coherence corresponds to the variance in growth between iterations.

- When a candidate unstable pattern is identified that meets the coherence threshold, the secondary refinement stage is activated:

```
Mode growth:      1.35752e+00  Coherence:      3.95870e-02
Growing pattern cell count: 929
Isolating new mode
Iteration: 0  Growth rate:      1.42665e+00  Residual:      5.88196e-02
Iteration: 1  Growth rate:      1.41329e+00  Residual:      1.34600e-02
Iteration: 2  Growth rate:      1.41227e+00  Residual:      2.25348e-02
New growing mode with 597 cells
```

If the growing pattern cell count exceeds **Maximum Number of Cells** the refinement procedure is postponed. Likewise, if after refinement the number of cells involved exceeds the specified limit the mode is rejected. Each iteration corresponds to a refinement of the unstable pattern. The adjoint solution itself is not advanced during this process. The residual corresponds to the purity of the mode and must eventually fall below the **Mode Purity Threshold** for the mode to be accepted. A minimum of 3 iterations is taken so that the occurrence or absence of a conjugate pair can be verified.

- The presence of the sometimes violent unstable growth that can occur gives a jagged appearance to the adjoint residual plots. However, the underlying trend of the residual plots should correspond to a converging adjoint solution process.

The unstable patterns of solution advancement are specific to the flow solution, choice of discretization schemes, and solution advancement controls for the adjoint solver. If any of these values changes, then the unstable modal patterns must be recomputed so that the stabilization scheme will function to full

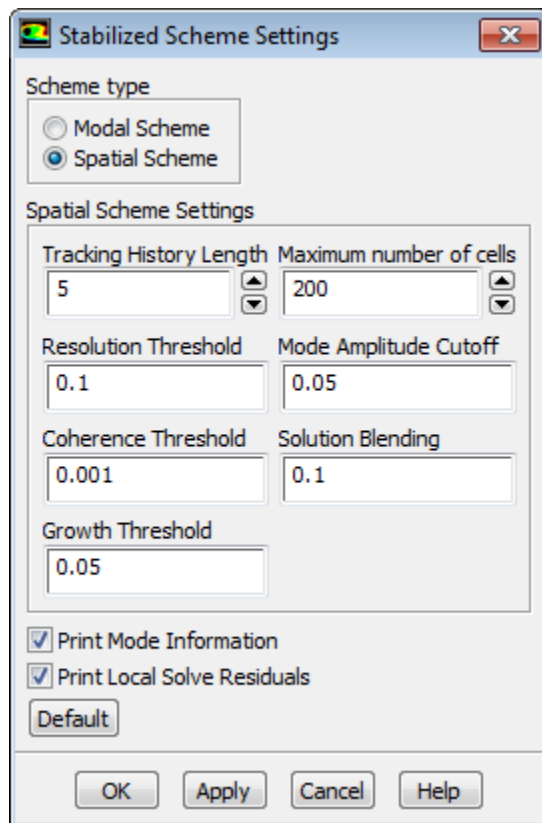
effect. In the event that any of these parameters is changed and the adjoint solution is restarted a mode recalibration process will be initiated. It should be noted that if there are many unstable modes that have been identified then this recalibration may take some time.

In contrast, the patterns of unstable growth are not specific to the observable that has been selected. Given that there is computational effort needed to identify the instabilities, a separate initialization process is available just for the unstable modal patterns. This allows the adjoint solution itself to be initialized for a new observable while retaining the knowledge of how to stabilize the solution advancement. A much smoother path to solution for the second and subsequent observables is then achieved.

2.6.1.2. Spatial Stabilization Scheme

The spatial scheme operates by identifying parts of the domain where unstable growth is occurring and applying a more stable scheme. The unstable mode tracking and handling is done in an automated manner. Several settings are exposed, but are recommended for expert users only. When the stabilized scheme is used and the **Settings...** button is selected, the following dialog box will appear ([Figure 2.9: Stabilized Scheme Settings for Spatial Scheme](#) (p. 29)):

Figure 2.9: Stabilized Scheme Settings for Spatial Scheme



The four items on the left side of the dialog box relate to unstable mode tracking and selection, while the items on the right provide options for the response once a mode is identified. The values have the following meaning:

Tracking History Length

defines the number of iterations that are employed in the unstable mode identification process. A range between 2 and 5 may be chosen. The longer the tracking history, the better the quality of the identification process.

Resolution Threshold

defines a threshold for the separation between modes based on their amplitudes. A value of 0.1 implies that the second most-dominant mode must be 0.1 of the amplitude of the primary mode before an unstable mode can be considered as being adequately resolved.

Coherence Threshold

defines a threshold that must be met for the closeness of the mode to pure exponential growth as the iterations progress. A smaller number means that a tighter tolerance must be met before an unstable mode is considered as having been identified.

Growth Threshold

defines the growth rate threshold for unstable modes that must be met before the algorithm responds to the presence of the mode. The growth threshold defines the number of iterations that would have to take place for the mode to grow by one order of magnitude. That is, a value of 0.01 signifies that a mode that would grow one order of magnitude or more in 100 iterations will be handled. A negative value for this threshold is not recommended as multiple stable modes may be tracked.

Maximum Number of Cells

defines a limit on the number of cells for which stabilization will be applied when an unstable mode is located.

Mode Amplitude Cutoff

defines the fraction of the peak mode amplitude beyond which cells will be included for stabilization.

Solution Blending

defines an under-relaxation factor for blending the stabilized part of the scheme with the standard advancement scheme. A default of 0.1 is chosen. The value should lie between 0 and 1.

Print Mode Information

(when enabled) leads to details of the unstable mode tracking process being printed in the console window. The following is an example of what is printed when a mode is resolved partially:

```
Mode present> Growth :      2.38166e-01  Coherence   :      1.27858e+00
Resolution Threshold :      1.00000e-01  Resolution  :      1.32611e-01
```

The following is an example of what is printed when a mode becomes more fully resolved:

```
Mode present> Growth :      1.35318e-01  Coherence   :      2.84133e-04
Resolution Threshold :      1.00000e-01  Resolution  :      2.75312e-04
Growth Threshold    :      5.60193e-02  Growth      :      1.35318e-01
Coherence Threshold :      1.00000e-02  Coherence   :      2.84133e-04
Local stabilization required: Instability detected in 5 cells
```

The final line only appears when a coherent and resolved instability has been identified successfully. Subsequently, the solution advancement algorithm automatically adjusts to eliminate this unstable behavior.

Print Local Solve Residuals

(when enabled) leads to details of the residuals for the local solution process being printed in the console window. An example of what is printed is given below:

Iter	Adjoint-continuity	Adjoint-xMom	Adjoint-yMom	Adjoint-Flowrate
....				
27	1.76334e-03	1.57503e-01	1.26984e-01	2.92618e-01
Local (%)	61.02095	57.41937	62.09185	99.99136

The percentage residual indicates the fraction of the total residual that is associated with the local solution process.

Default

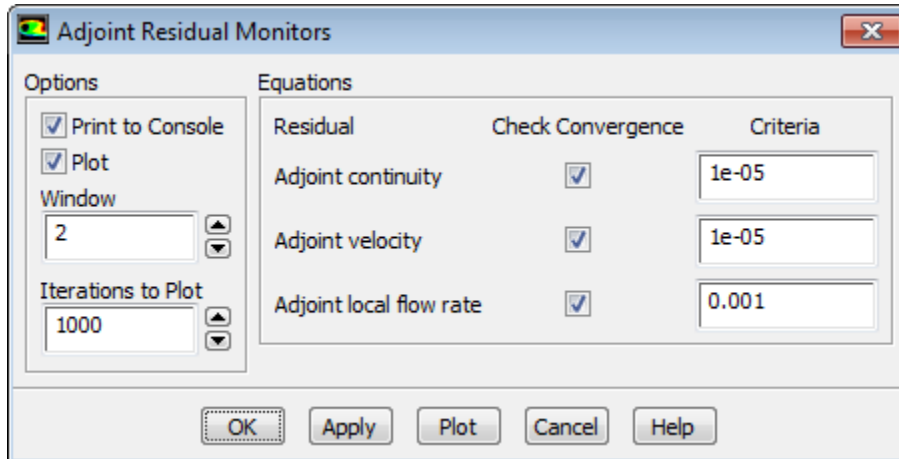
provides a set of default values for the various fields in the dialog box.

2.7. Working with Adjoint Residual Monitors

The progress of the iterations of the adjoint solver and the monitoring of the convergence is controlled in the **Adjoint Residual Monitors** dialog box. This dialog box is accessed using the **Monitors...** option under the **Adjoint** menu.

Adjoint → **Monitors...**

Figure 2.10: Adjoint Residual Monitors Dialog Box



The steps to define the monitor behavior are as follows:

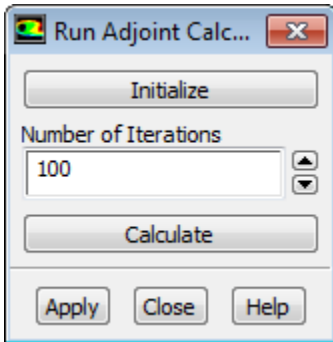
1. Decide whether or not the residuals are to be printed in the console by setting the **Print to Console** check box as desired.
2. Decide whether or not the residuals are to be plotted in the main window by setting the **Plot** check box as desired.
3. Set the id of the window in which the adjoint residuals are to appear in the **Window** field.
4. Set how many iterations are to be shown in the residual curves that are plotted in the **Iterations to Plot** field.
5. Enable and/or disable those values that are to be used as criteria for convergence, and set the **Criteria** in each case. The options are to test the residuals for the **Adjoint continuity** and/or **Adjoint velocity** and/or **Adjoint local flow rate** equations.

Click the **OK** or **Apply** buttons to confirm that the settings are acceptable. Clicking **Plot** will cause existing adjoint residuals to be plotted in the main graphics window.

2.8. Running the Adjoint Calculation

Initialization and execution of the adjoint solver is accomplished in the **Run Adjoint Calculation** dialog box. This dialog box is accessed using the **Run Calculation...** option under the **Adjoint** menu.

Adjoint → **Run Calculation...**

Figure 2.11: Run Adjoint Calculation Dialog Box

The function of this dialog box is as follows:

Initialize

sets the value of the adjoint velocity and pressure to zero everywhere in the problem domain.

Initialize Stabilization

clears all of the unstable modes that have been identified in calculations using the modal stabilization scheme, but leaves the adjoint solution unchanged. Only available when **Modal** is selected in the **Stabilized Scheme Settings** dialog box.

Calculate

advances the adjoint solver by the **Number of Iterations** specified in the adjacent field.

Depending on the monitor settings, the residuals may be printed in the console and/or plotted in the main graphics window.

2.9. Postprocessing of Adjoint Solutions

Adjoint solution data can be post-processed so that it provides both qualitative and quantitative views of the effect of many types of change that may be imposed on a system. While shape changes are often of particular interest there is a very rich data set, one as large as the original flow field, available for exploration. Post-processing tools are provided with the intention of permitting the adjoint data to be mined in such a way that it provides useful supporting information for an engineer who is making design decisions for a system, or has questions about the reliability of the original flow calculation.

Since the ANSYS Fluent adjoint solver is a discrete adjoint solver, the primitive adjoint solution data provides sensitivity to changes cell-wise for the computational mesh. Normalization of these results by the cell volume provides a *mesh-independent* view into the data set.

Information regarding postprocessing the adjoint solutions can be found in the following sections:

[2.9.1. Field Data](#)

[2.9.2. Scalar Data](#)

2.9.1. Field Data

Adjoint solution data can be post-processed using standard ANSYS Fluent post-processing tools including contours, vectors, xy-plots, histograms, and surface and volume integrals.

In the **Contours** dialog box, under **Sensitivities...**, you can find the following fields:

Magnitude of Sensitivity to Body Forces (Cell Values)

This field is the magnitude of the adjoint velocity primitive field. This field can be interpreted at the cell level as the sensitivity of the observable to a body force on the cell, or any factors that may change the momentum balance on the cell. As such it is a mesh-dependent quantity. This field is often observed to be large, for example, upstream of a body for which drag sensitivity is of interest, with the field diminishing in the upstream direction. This indicates the interference effect for an object positioned at various locations upstream of the object of interest.

Sensitivity to Body Force X-Component (Cell Values), Sensitivity to Body Force Y-Component (Cell Values), and Sensitivity to Body Force Z-Component (Cell Values)

These are the components of the adjoint velocity primitive field. These fields show the sensitivity of the observable to the various components of a body force that may be applied to a cell, or other factors that may change the momentum balance on the cell. As such, it is a mesh-dependent quantity.

Sensitivity to Mass Sources (Cell Values)

This field provides a plot of the primitive adjoint pressure field. This field can be interpreted at the cell level as the sensitivity of the observable with respect to any changes to the mass balance on the cell. As such, it is a mesh-dependent quantity.

Sensitivity to Energy Sources (Cell Values)

This field is available when the energy adjoint is solved and is the primitive adjoint temperature field. It can be interpreted at the cell level as the sensitivity of the observable with respect to changes in the thermal energy balance of the cell. As such, it is a mesh-dependent quantity.

Magnitude of Sensitivity to Body Forces

This field is the magnitude of the adjoint velocity primitive field normalized by cell volume. This field can be interpreted as the magnitude of the sensitivity of the observable to body force per unit volume. It can be used to identify regions in the domain where small changes to the momentum balance in the flow can have a large or small effect on the observable. This field is often observed to be large, for example, upstream of a body for which drag sensitivity is of interest, with the field diminishing in the upstream direction. This indicates the interference effect for an object positioned at various locations upstream of the object of interest. This field is normalized so that, in principle, it shows a mesh-independent quantity. In practice, this field has features that are often correlated with mesh structure. This is not unexpected. In fact, the adjoint solution field is providing very useful insight into regions of the flow domain where the observation of interest is potentially sensitive to particular features of the mesh and discretization process that affect the local momentum balance.

Sensitivity to Body Force X-Component, Sensitivity to Body Force Y-Component , and Sensitivity to Body Force Z-Component (in 3D) fields

These are the components of the adjoint velocity primitive field normalized by cell volume. This field can be interpreted as the magnitude of the sensitivity of the observable to components of a body force per unit volume. Consider a body force distribution, expressed as a force per unit volume. The integral of the vector product of that distribution with the components of this field gives a first-order estimate of the net effect of the body force on the observation.

Sensitivity to Mass Sources

This field is the primitive adjoint pressure field normalized by cell volume. This field can be interpreted as the sensitivity of the observable with respect to mass sources or sinks in the domain. Consider a mass source/sink distribution, expressed as mass flow rate per unit volume. The integral of that distribution, weighted by the local value of this field, gives the effect of the sources/sinks on the observation. This field is normalized so that, in principle, it shows a mesh-independent quantity. In

practice, this field has features that are often correlated with mesh structure. This is not unexpected. In fact the adjoint solution field is providing very useful insight into regions of the flow domain where the observation of interest is potentially sensitive to particular features of the mesh and discretization process that affect local mass balance. When plotted on a boundary, this field indicates the effect of the addition or removal of fluid from the domain upon the quantity of interest. It is important to note that in this scenario the effect of the momentum of the fluid that is added or removed is not taken into account. The boundary velocity sensitivity should be plotted if that effect is also of interest.

Sensitivity to Energy Sources

This field is available when the energy adjoint is solved and is the primitive adjoint temperature field normalized by cell volume. It can be interpreted at the cell level as the sensitivity of the observable with respect to thermal energy sources or sinks in the domain. This field is normalized so that, in principle, it is a mesh-independent quantity.

Sensitivity to Viscosity

This field shows the sensitivity of the quantity of interest to variations in the turbulent effective viscosity for a turbulent problem, or the laminar viscosity in a laminar case. The sensitivity is normalized by the cell volume to account for cell size variations in the mesh.

Shape Sensitivity Magnitude

This field is the magnitude of the sensitivity of the observable with respect to a deformation applied to the mesh (both boundary and interior mesh). When plotted on the surface of a body the locations where this quantity is large indicates where small changes to the surface shape can have a large effect on the observable of interest. If the shape sensitivity magnitude is small then the effect of shape changes in this region can be expected to have a small effect on the observable of interest. When viewing this field, it is often observed that the magnitude varies by many orders of magnitude. Contour plots will clearly draw attention to regions with the highest sensitivity (often sharp edges and corners). However, it should be remembered that a relatively small surface movement that is distributed over a large area can have a cumulative effect that is large.

Normal Shape Sensitivity

This field shows the normal component of the shape sensitivity. A positive value indicates an orientation directed into the domain, while a negative value indicates that the shape sensitivity is oriented outwards from the domain. This field eliminates the component of the vector shape sensitivity field that lies in the plane of the wall.

Normal Optimal Displacement

This field shows the normal component of the optimal displacement computed from the adjoint solution. This field is defined only for portions of walls lying within the control-volume specified for morphing. The size of the optimal displacement is determined by the scale factor that is chosen for the morphing. A positive value of displacement indicates that the surface will be displaced into the flow domain, whereas a negative value of displacement corresponds to wall movement outwards from the flow domain. This field eliminates the component of the optimal displacement vector that lies in the plane of the wall.

log₁₀(Shape Sensitivity Magnitude)

In view of the large range of values possible for the shape sensitivity magnitude a convenience function which plots \log_{10} of the magnitude is provided. This allows the importance of the surfaces in a domain to be ranked more easily based on how they affect the observation of interest when they are reshaped.

Shape Sensitivity X-Component, Shape Sensitivity Y-Component, and Shape Sensitivity Z-Component (in 3D) fields

These fields are the individual components of the sensitivity of the observable of interest with respect to the mesh node locations. It is plotted as cell data and is computed as the average of the nodal sensitivities for a given cell, divided by the cell volume. Note that for this discrete adjoint solver the sensitivity of the result with respect to node locations both on and off boundaries is computed. The normalization by cell volume indicates that the fields that are plotted are the weighting factors for a continuous spatial deformation field. (Note that the nodal sensitivity data itself is used when mesh morphing is performed, and predictions about the effect of shape changes are made.)

Sensitivity to Boundary X-Velocity, Sensitivity to Boundary Y-Velocity, and Sensitivity to Boundary Z-Velocity (in 3D) fields

These fields are defined on those boundaries where a user-specification of a boundary velocity is made for the original flow calculation. This includes no-slip walls. The field shows how sensitive the observable of interest is to changes in the boundary velocity at any point. It is interesting to note that even though the original boundary condition specification may be for a uniform velocity on the domain boundary, the effect of a non-uniform velocity perturbation is available. The effect of any specific boundary velocity change can be estimated as an integral of the vector product of the change to the velocity with the plotted sensitivity field. A plot of this quantity on a velocity inlet, for example, can be very useful for assessing whether or not the inlet is positioned too close to key parts of the system. That is, it addresses the question of whether or not the flow domain is too small to achieve a successful computation of the performance measure of interest. Viewing this field will also indicate whether or not the assumption of a uniform inflow is adequate.

Sensitivity to Boundary Pressure

This field is defined on boundaries where there is a user-specified pressure as part of a boundary condition, such as on a pressure outlet. The field shows the sensitivity of the observation of interest to variations in the boundary pressure across the flow boundary. It is interesting to note that even though the original boundary condition specification may be for a uniform pressure on the domain boundary, the effect of a non-uniform pressure perturbation is available. The effect of any specific boundary pressure change can be estimated as an integral of the product of the change to the pressure with the plotted sensitivity field. Viewing this field will also indicate whether or not the assumption of a uniform pressure is adequate for the simulation.

Sensitivity to Boundary Temperature

This field is available when the energy adjoint equation is solved and is defined on boundaries where a temperature boundary condition is applied. This includes walls, velocity inlets, and pressure outlets where a backflow temperature may be specified. The field shows the sensitivity of the observation of interest to variations in the boundary temperature across the boundaries. Note that even if the original boundary condition specification is for a uniform temperature on the boundary, the effect of a non-uniform temperature perturbation is available. The effect of any specific boundary temperature change can be estimated as an integral of the product of the change to the temperature with the plotted sensitivity field. This field can be used to indicate whether or not the assumption of a uniform temperature is adequate for the simulation.

Sensitivity to Boundary Heat Flux

This field is available when the adjoint energy equation is solved and is defined on walls where a heat flux boundary condition is imposed. The field shows the sensitivity of the observation of interest to variations in the boundary heat flux through the wall. Its properties are analogous to those of Sensitivity to Boundary Temperature.

Sensitivity to Flow Blockage

This field is provided as a convenient tool for identifying portions of the flow domain where the introduction of blockages or obstructions in the flow can affect the observation of interest. Consider a blockage in the flow that generates a reaction force on the flow that is proportional to the local flow speed, and acting in the opposite direction to the local flow: $F = -\alpha(r) v$ where α is a local coefficient for the reaction force. The local contribution of this force on the observation of interest is determined by the vector product of this force with the adjoint velocity field. The flow blockage field that is plotted is $-\tilde{v} \cdot v$, namely the negative of the vector product of the flow velocity and the adjoint velocity (Cell Value).

Adjoint Local Solution Marker

This field, intended for expert users, can be plotted to identify those portions of the flow domain where the stabilized adjoint solution advancement scheme is applied. It is preferable to plot this with the **Node Values** de-selected in the **Contours** dialog box. In this case, the **Adjoint Local Solution Marker** will take a value between 0 and 1. The **Mode Amplitude Cutoff** defined in the **Stabilized Scheme Settings** dialog box defines the lower bound for cells where the stabilized scheme is applied.

The surface shape-sensitivity fields contain the derivatives of the chosen observable with respect to the position of the bounding walls of the problem.

In the **Vectors** dialog box, under **Vectors of...**, you can find the following custom vector fields:

Sensitivity to Body Forces (Cell Values)

This vector field shows how sensitive the observable of interest is to the presence of body forces within the flow. A body force that locally has a component in the direction of the body-force sensitivity vector will lead to an increase in the observation of interest. The larger the sensitivity vector, the larger the effect that a body force can have on the observation of interest.

Sensitivity to Body Forces

This vector field shows how sensitive the observable of interest is to the presence of body forces within the flow. It is normalized by the local cell volume so that it is *mesh-independent* data. A body force that locally has a component in the direction of the body-force sensitivity vector will lead to an increase in the observation of interest. The larger the sensitivity vector the larger the effect that a body force can have on the observation of interest. The first-order effect of a body force distribution per unit volume can be computed as a volume integral of the vector product of the body force distribution with this field.

Sensitivity to Shape field defined on walls

This vector field shows the post-processed adjoint solution, independent of any morphing settings, that defines the sensitivity of the observable of interest to movement of the walls. Deformation of a wall where this vector is large can be expected to have a significant effect on the observable. In contrast, deformations of the wall in locations where this vector is small in magnitude, to first order, will have no effect on the observable.

Optimal Displacement

The optimal displacement field is defined within a fully-specified control volume for morphing once the control point sensitivities have been updated. This vector field shows the optimal displacement suggested by the adjoint solution. An increase or decrease in the observable will result depending upon whether the observable is to be maximized or minimized, as specified in the **Adjoint Observables** dialog box. Note that when plotting this vector field, if the **Auto Scale** option is deselected, and the vector **Scale** option is set to 1, then the vectors as shown display the absolute displacement that will occur.

Sensitivity to Boundary Velocity

This sensitivity field is defined on boundaries where velocities are specified as an input for the boundary condition. This includes no-slip, and slip walls. When plotted, this field illustrates where a change to the boundary velocity can affect the observation of interest. When plotted on a wall, it shows how the imposition of a non-zero velocity condition on the wall will affect the observation. This can include movement in the plane of the wall itself. When plotted, for example on a velocity inlet, this field illustrates how local adjustments to the velocity of the incoming flow affect the observation of interest. Integrating a potential change to the velocity field, weighted by this sensitivity field over the boundary provides a first order estimate of the effect of the change on the observation of interest.

Sensitivity to Boundary Mass Flux

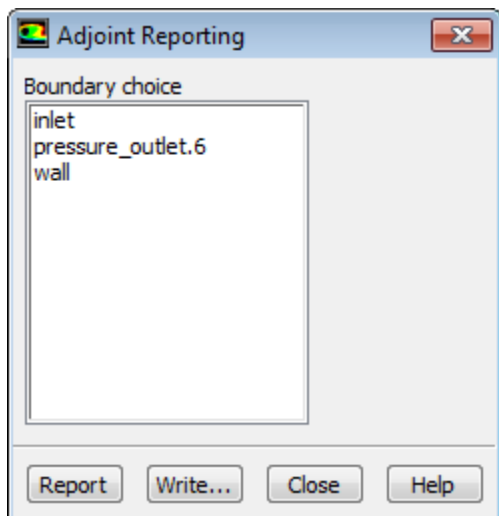
This sensitivity field that can be used to visualize the effect of adding or removing fluid via a wall-normal jet. This can be used, for example, to identify regions on walls where the introduction of suction could have a significant effect on the observable that has been specified. A plot of the **Sensitivity to Boundary Velocity** in general provides a richer view, but sometimes at the expense of the clarity of the visualization.

2.9.2. Scalar Data

The computed adjoint solution can provide sensitivity information for individual boundary condition settings through the **Adjoint Reporting** dialog box. This dialog box is accessed using the **Reporting...** option under the **Adjoint** menu.

Adjoint → **Reporting...**

Figure 2.12: Adjoint Reporting Dialog Box



Boundary condition sensitivity data is retrieved as follows:

- Select a single boundary of the problem in the **Boundary choice** selection box.
- Click **Report** to see the boundary condition settings and the sensitivity of the observable with respect to those settings. An example of a report is shown below:

```
Velocity inlet id 5
Velocity magnitude = 4.000000000e+01 (m/s),
Sensitivity = 5.429127866e+01 ((n)/(m/s))
```

In this case, the report shows that for every change of the boundary velocity by 1 m/s, a change in the predicted observable (in this case a force) of 54.2 N is expected. This prediction is based on a linear extrapolation.

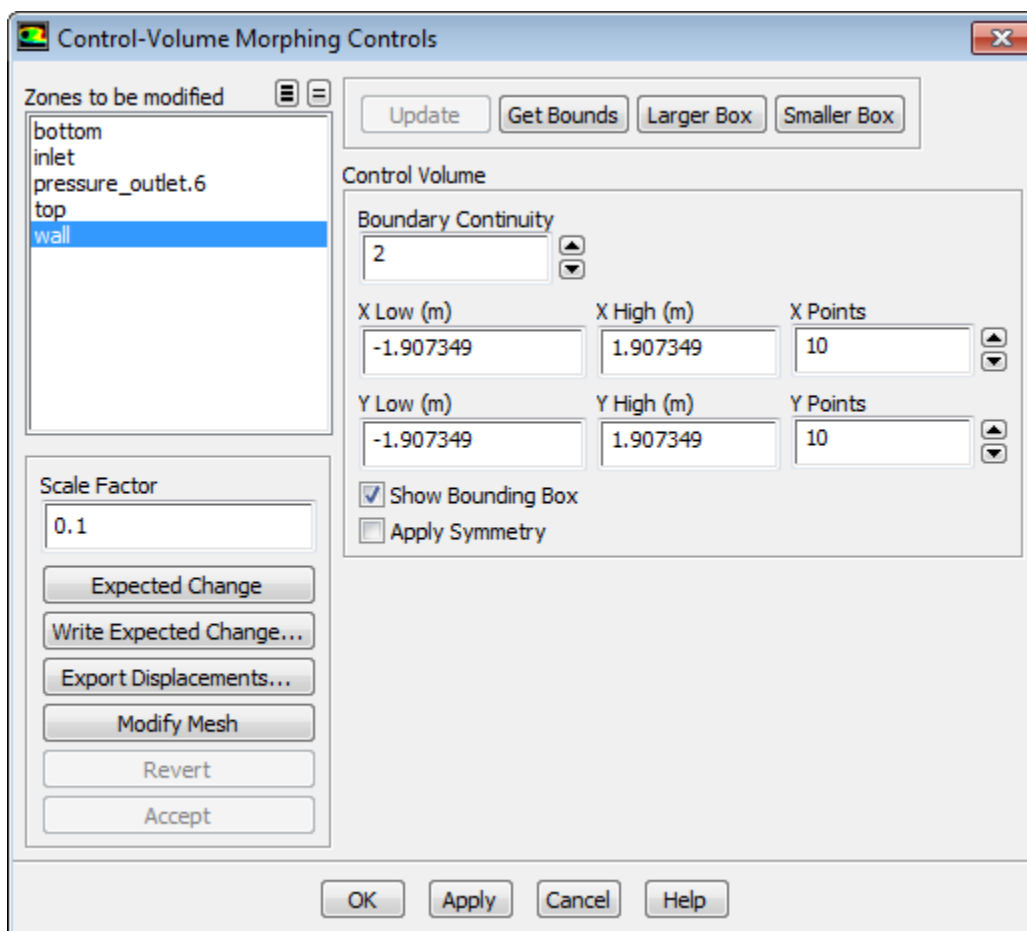
- Click **Write...** to write the report for the selected boundary to a file.

2.10. Modifying the Geometry

Tools to modify the boundary and interior mesh of the problem based on the adjoint solution are provided in the **Control-Volume Morphing Controls** dialog box. This dialog box is accessed using the **Control-Volume Morphing...** option under the **Adjoint** menu. These modifications are made using a simple gradient algorithm and allows an interactive optimization to be performed.

Adjoint → **Control—Volume Morphing...**

Figure 2.13: Control-Volume Morphing Controls Dialog Box



For more information, see the following sections:

[2.10.1. Shape Modification](#)

[2.10.2. Defining the Control Volume for Control-Volume Deformation](#)

2.10.1. Shape Modification

The procedure for deforming the boundary mesh for a system is as follows:

1. Select those boundaries that are free to be deformed in the **Zones to be modified** field. Boundaries that are not selected in this box that intersect the control volume will be constrained not to move. This is useful if there are walls which, for design reasons, must not move.
2. Define a control volume within which the boundary and interior mesh are to be deformed. The detailed steps to accomplish this task are given in [Defining the Control Volume for Control-Volume Deformation](#) (p. 40).
3. Pick a **Scale Factor** for the deformation. A scale factor of 1.0 leads to a maximum control point movement on the order of one sub-cell of the control volume. A **Scale Factor** of 1.0 is a reasonable default to preserve mesh quality during the morphing process.

A positive value for scale factor will lead to a change that is intended to minimize or maximize the value of the observable, depending upon the selection that has been made in the **Adjoint Observables** dialog box. As an example, consider a 2D airfoil with external air flow in the +X direction that has a lift force defined in the direction $X=0, Y=1$ and a specification that the lift should be maximized. With a scale factor of 1, the adjoint solution will give guidance on how to increase the lift. If a pressure drop is defined as the observable with a specification that it should be minimized, then, with a positive scale factor, the adjoint solution will give guidance on reducing the pressure drop.

4. Click **Update**. This is a key step and leads to the evaluation of the control-point sensitivities and surface deformation field. In the event that the control volume settings are modified, or the **Scale Factor** changes, then **Update** must be called again.
5. Click **Expected Change** to see a printout in the console window of the adjustment to the value of the observable of interest that is to be expected, based on the current settings.
6. Click **Write Expected Change...** to write to file the adjustment in the observable that is expected due to the change defined by the current settings.
7. Click **Export Displacements...** to write to file the optimal surface displacement field (overwriting any pre-existing file of the same name). An example of the format of the text file that is written (for three-dimensions) is as follows:

```
Line 1: n
Lines 2 to n+1: x y z dx dy dz
```

where n is the number of points that are written, (x, y, z) is the current coordinate of the point, and (dx, dy, dz) is the optimal displacement that is to be superimposed upon it. Any given set of node data will appear only once in the output listing.

8. The planned change to the geometry can be visualized by plotting the control-sensitivity vector field, or a surface contour plot of surface deformation.
9. Click **Modify Mesh** to deform the boundary and interior mesh using the shape sensitivity field.
10. The option to **Revert** the mesh back its previous state is available for as long as the dialog box remains open or until **Accept** is pushed.
11. Clicking **Accept** readies the morpher for a new control volume definition, and second or subsequent mesh deformation. This is useful when multiple parts of a configuration are being adjusted independently. *Note: Deformations based on the same adjoint solution must not have overlapping control volumes.*

Upon successful modification of the geometry of the system, the modified shape can be viewed using the usual commands for viewing the mesh in ANSYS Fluent.

2.10.2. Defining the Control Volume for Control-Volume Deformation

Note

For all numeric fields described below, a change to the value should be followed by a keyboard **Enter**. This serves as a signal that the control volume definition has changed and will require an **Update** of the control-point sensitivity.

The following procedure can be used to define the control volume:

1. Activate the **Show Bounding Box** toggle button to make a bounding box visible in the active graphics window.
2. The **Boundary Continuity** setting controls the order of continuity of the morphing process at the boundaries of the control volume. This is of particular relevance when the boundary of the control volume intersects a wall. The order of boundary continuity in this situation controls the smoothness of the transition from the unmoved wall lying outside the control volume to the deformed wall within. A value of 1 ensures first order continuity at the transition, in other words, a continuous first derivative of the morphing operation. A value of 2 ensures second order continuity, and so forth. This setting will not affect the transition between fixed and movable boundaries within the control volume, only mesh nodes in the transition zone from inside to outside the control volume.
3. After selecting the **Zones to be modified**, click the **Get Bounds** button under **Control Volume**. This will populate the **X Low**, **X High**, **Y Low**, and **Y High** fields with the bounding box for the selected walls. Click the **Larger Box** and **Smaller Box** buttons to adjust the control volume size as needed. The limits of the bounding box can be entered individually by hand if desired.
4. Select the number of control points **X Points** and **Y Points** in the x and y directions respectively.

The control points that are specified are distributed uniformly in each coordinate direction. The spacing between the control points defines the characteristic spatial scale for the morphing operation. The larger the number of control points, the smaller the spacing and hence, the smaller the spatial scale on which changes can be made. On the order of 20 to 40 control points for each coordinate direction is typical.

5. Decide whether or not it is necessary to **Apply Symmetry** to the sensitivity field. This option may be used when there is a symmetry plane in the problem and points in that symmetry plane are to be moved via morphing. The option is restricted to symmetry planes with normals in the Cartesian coordinate directions. To enforce symmetry, the control volume for morphing must first be defined (using **X Low**, **X High**, etc.) so that it is bisected by the symmetry plane. Secondly, the option for the symmetry plane must be selected based on which portion of the control volume lies within the physical flow domain. The options are:
 - If **Use x larger than mid-point** is chosen, then the mirror image control point sensitivity is installed as if a mirror plane lies at the x mid-point of the control volume. The control sensitivity for x larger than the mid-point is reflected in that mirror plane and installed in the image locations at x below the mid-point.
 - If **Use x smaller than mid-point** is chosen, a mirror plane lying at the x mid-point is used to reflect data from below the mid-point to above.
 - **Use y larger than midpoint** and **Use y smaller than midpoint** perform the same role for the y direction.

For example, consider a problem with a symmetry plane lying on $Y=0$ in which the mesh in the flow domain lying in the range $-5 < Y < 0$ is to be morphed while respecting the symmetry plane. The control volume is set with **Y Low** = -5, and **Y High** = 5 so that the symmetry plane bisects the control volume. **Apply Symmetry** is enabled and the selection **Use y smaller than midpoint** is used.

Once the mesh has been modified according to the control settings, the original flow calculation can be restarted and converged with the modified geometry. Upon successful convergence the observable value should now have changed by an amount defined when the **Expected Change** button was clicked prior to deforming the geometry.

2.11. Using the Adjoint Solver Module's Text User Interface

A text user interface (TUI) is provided for the adjoint solver. The root of the adjoint TUI is the **adjoint** keyword at the top-most level of the command tree for ANSYS Fluent. The tree structure is as shown below:

- `adjoint/observable`
 - `create` A new observable of the specified type and name is created and the definition is populated with default parameters.
 - `rename` An existing observable is renamed to a new specified name.
 - `delete` Removes a named observable.
 - `select` A named observable is selected as the one for which an adjoint solution is to be computed.
 - `specify` The parameters that define a named observable are configured.
 - `write` Evaluates the current value of the selected observable and writes the result to a file.
 - `evaluate` Evaluates the current value of the selected observable and prints the result to the console.
- `adjoint/controls`
 - `settings` Text menu to set parameters for the adjoint solver numerics.
 - `stabilization` Text menu to set parameters for the adjoint stabilization schemes.
- `adjoint/methods`
 - `settings` Text menu to set discretization methods for the adjoint solver.
 - `default-settings` Sets discretization methods for the adjoint solver to the defaults.
 - `best-match-settings` Sets discretization methods for the adjoint solver that best match with those for the flow solver.
- `adjoint/monitors`
 - `plot-residuals` Plots the adjoint residuals in the designated graphics window.
 - `settings` Text menu to configure the monitors and convergence criteria for the adjoint solver.
- `adjoint/run`

- `initialize` Initializes the adjoint solution field to zero everywhere.
- `iterate` Advances the adjoint solver by a specified number of iterations, or until the convergence criteria are met.
- `adjoint/reporting`
 - `report` Reports sensitivity data on a named flow boundary.
 - `write` Reports sensitivity data on a named flow boundary and writes it to a named file.
- `adjoint/morphing`
 - `control-point-update` Updates the shape sensitivity data at the control points.
 - `expected-change` Computes the expected change in the observable for the specified shape-modification and prints the result in the console.
 - `export-optimal-displacements` Computes the optimal surface displacement field in the observable for the specified shape-modification and writes the result to a named file (overwriting any pre-existing file of the same name).
 - `modify-mesh` Modifies the mesh based on the specified control volume, control-point sensitivities, and scale factor.
 - `specify-control-volume` Text menu to define the control volume for morphing.
 - `write-expected-change` Computes the expected change in the observable for the specified shape-modification and writes the result to a named file.

Chapter 3: Tutorial: 2D Laminar Flow Past a Cylinder

This tutorial is divided into the following sections:

- [3.1. Introduction](#)
- [3.2. Setup and Solution](#)
- [3.3. Summary](#)

3.1. Introduction

The adjoint solver is used to compute the sensitivity of quantities of interest in a fluid system with respect to the user-specified inputs, for an existing flow solution. Importantly, this also includes the sensitivity of the computed results with respect to the geometric shape of the system. This sensitivity information can then be used to guide systematic changes that result in predictable improvements in the system performance.

This tutorial provides an example of how to generate sensitivity data for flow past a circular cylinder, how to post-process the results, and to use the data to reduce drag by morphing the mesh. The tutorial makes use of a previously computed flow solution.

This simple flow configuration is used to illustrate the following functions of the adjoint solver:

- How to load the adjoint solver add-on
- How to select the observable of interest
- How to access the solver controls for advancing the adjoint solution
- How to set convergence criteria and plot and print residuals
- How to advance the adjoint solver
- How to post-process the results to extract sensitivity data
- How to modify the cylinder shape to reduce the drag

The particular goal is to investigate how to modify the shape of the cylinder so that the drag is reduced. This is followed by an additional section that extends the scope of the tutorial to consider lift by attempting to maximize the lift/drag ratio, a typical optimization parameter in the aerospace industry.

The configuration is a circular cylinder, bounded above and below by symmetry planes. The flow is laminar and incompressible with a Reynolds number of 40 based on the cylinder diameter. At this Reynolds number the flow is steady.

3.2. Setup and Solution

The following sections describe the setup steps for this tutorial:

- [3.2.1. Preparation](#)
- [3.2.2. Step 1: Loading the Adjoint Solver Add-on](#)

3.2.3. Step 2: Solver Setup and Adjoint Solution Convergence

3.2.4. Step 3: Postprocessing

3.2.5. Step 4: Shape Modification

3.2.6. Step 5: Extending the Observable to a Ratio

3.2.1. Preparation

1. Set up a working folder on the computer you will be using.
2. Go to the ANSYS Customer Portal, <https://support.ansys.com/training>.

Note

If you do not have a login, you can request one by clicking **Customer Registration** on the log in page.

3. Enter the name of this tutorial into the search bar.
4. Narrow the results by using the filter on the left side of the page.
 - a. Click **ANSYS Fluent** under **Product**.
 - b. Click **15.0** under **Version**.
5. Select this tutorial from the list.
6. Click **Files** to download the input and solution files.
7. Unzip `adjoint_cylinder_tutorial_R15.zip`.

The files `cylinder_tutorial.cas` and `cylinder_tutorial.dat` can be found in the `adjoint_cylinder_tutorial` folder created after unzipping the file.

8. Use Fluent Launcher to start the **2D** version of ANSYS Fluent.

For more information about Fluent Launcher, see [Starting ANSYS Fluent Using Fluent Launcher](#) in the separate Fluent [User's Guide](#).

9. Enable **Double Precision**.
10. Enable **Display Mesh After Reading**.
11. Load the converged case and data file for the cylinder geometry.

File → Read → Case & Data...

When prompted, browse to the location of the `cylinder_tutorial.cas` and select the file. The corresponding data file will automatically be loaded as well.

Note

Once you read in the mesh, it will be displayed in the embedded graphics windows, since you enabled the appropriate display option in Fluent Launcher.

The data file contains a previously-computed flow solution that will serve as the starting point for the adjoint calculation. Part of the mesh and the velocity field are shown below:

Figure 3.1: Mesh Close to the Cylinder Surface

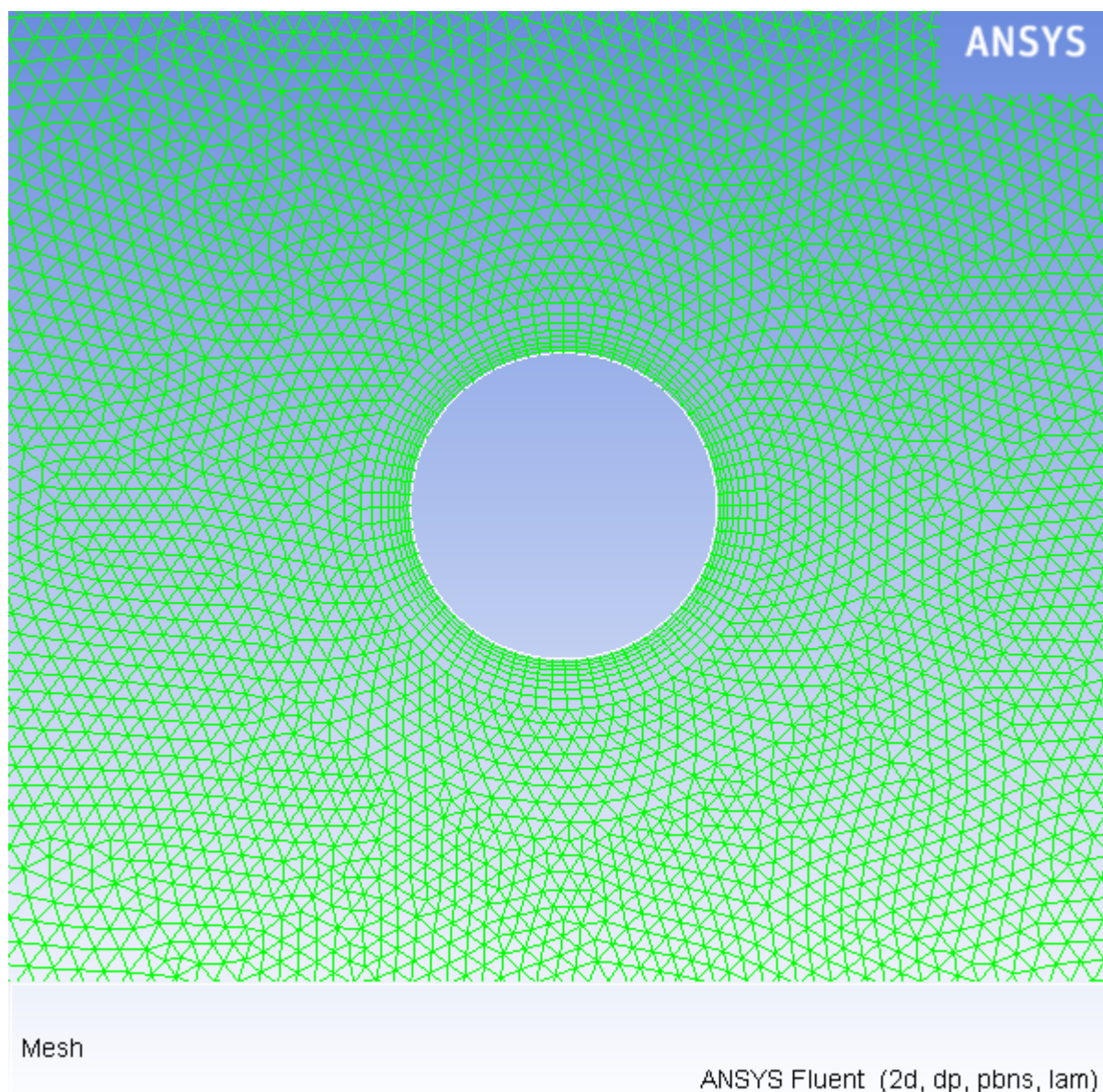
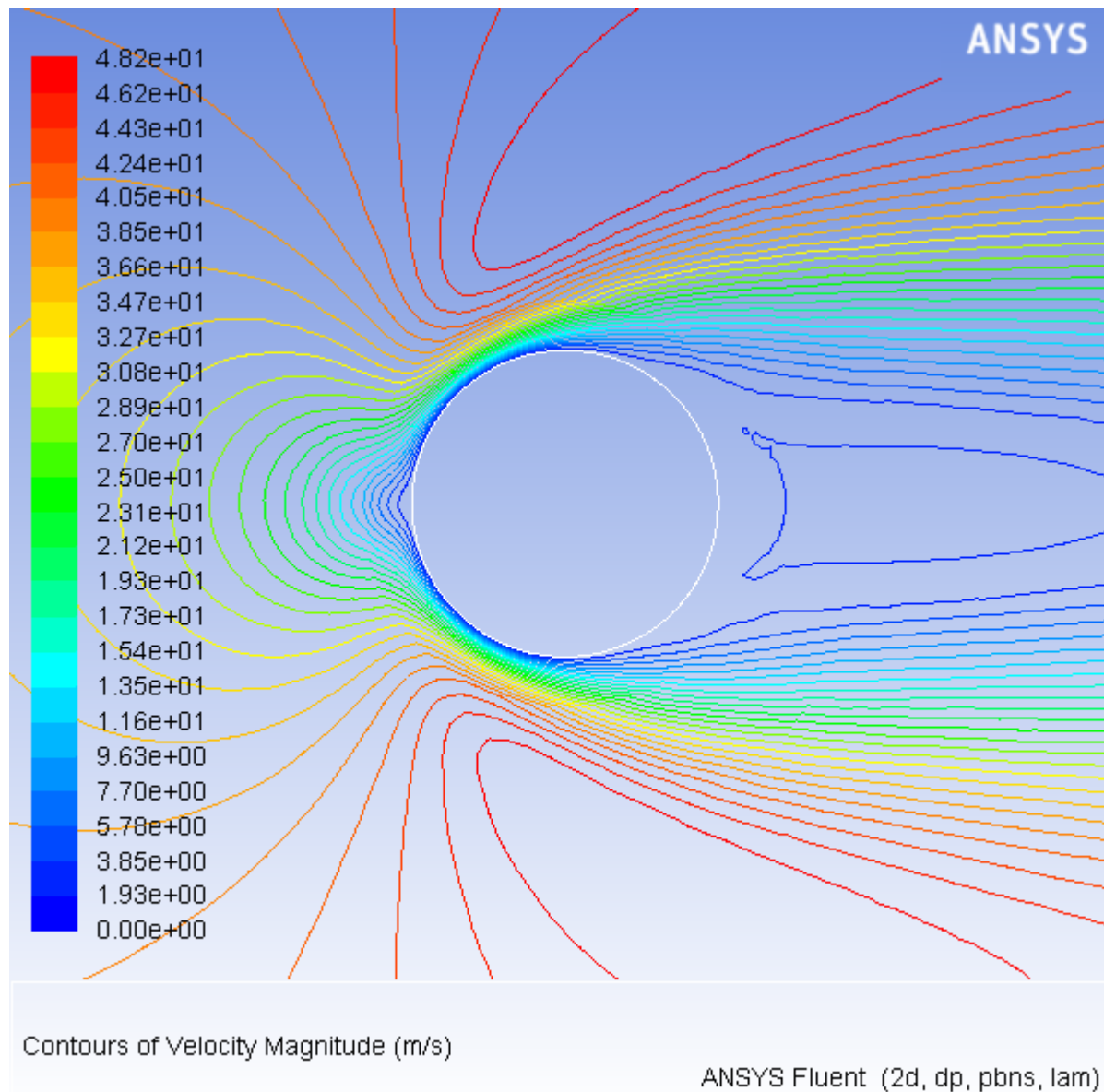


Figure 3.2: Contours of Velocity Magnitude

3.2.2. Step 1: Loading the Adjoint Solver Add-on

The adjoint solver add-on is loaded into ANSYS Fluent by typing the following into the console window:

```
>define/models/addon-module
```

to display the following:

```
Fluent Addon Modules:
0. none
1. MHD Model
2. Fiber Model
3. Fuel Cell and Electrolysis Model
4. SOFC Model with Unresolved Electrolyte
5. Population Balance Model
6. Adjoint Solver
7. Single-Potential Battery Model
8. Dual-Potential MSMD Battery Model
Enter Module Number: [0] 6
```

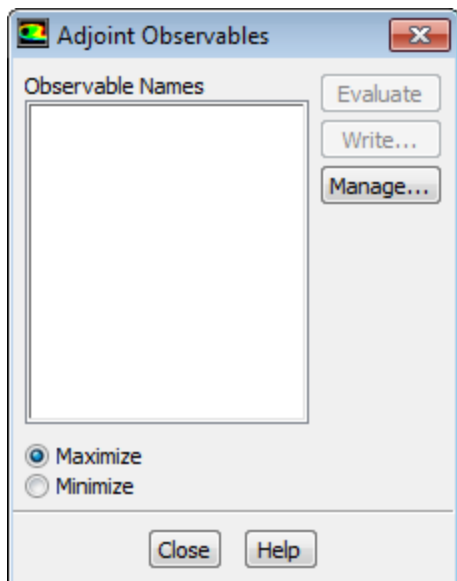
Enter 6 to load the adjoint solver itself. Once the adjoint solver is loaded, you will see the **Adjoint** menu within ANSYS Fluent.

3.2.3. Step 2: Solver Setup and Adjoint Solution Convergence

1. Begin setting up the adjoint solver by opening the **Adjoint Observable** dialog box. Here we will assign the observable as the drag force on the cylinder, and we will use the adjoint solution to assist in modifying the body in order to reduce drag.

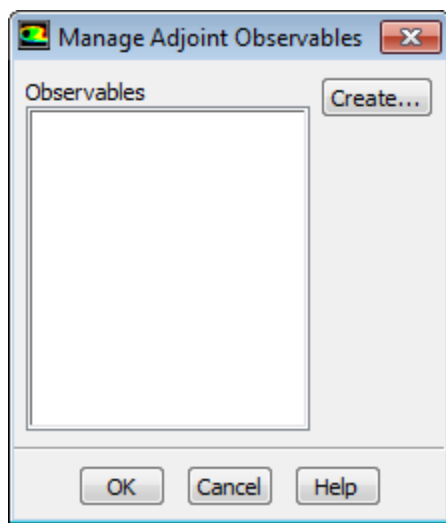
Adjoint → **Observable**

Figure 3.3: Adjoint Observables Dialog Box



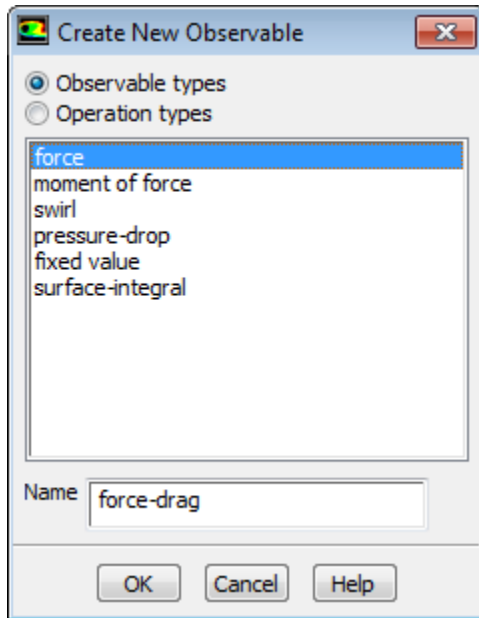
- a. Click the **Manage...** button in the **Adjoint Observables** dialog box, to open the **Manage Observable** dialog box.

Figure 3.4: Manage Adjoint Observables Dialog Box

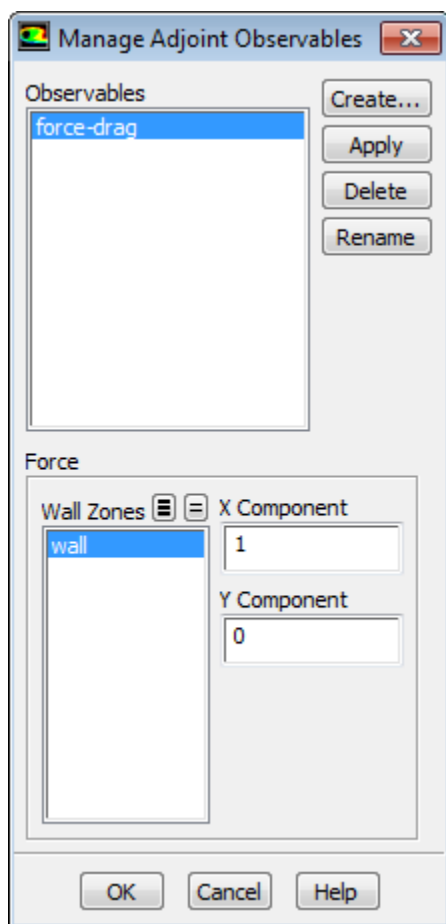


- b. Click the **Create...** button in the **Manage Observables** dialog box to open the **Create New Observable** dialog box.

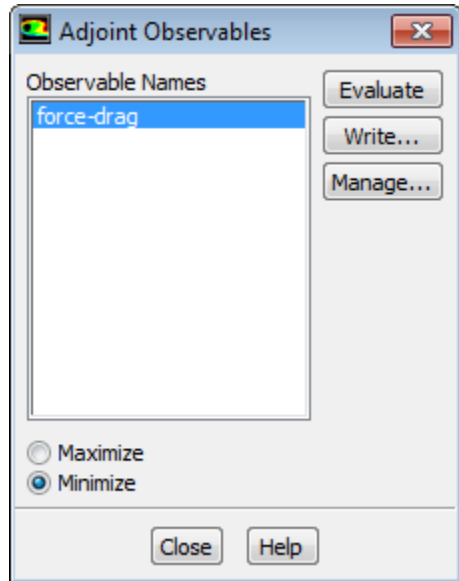
Figure 3.5: Create New Observable Dialog Box



- c. In the **Create New Observable** dialog box:
 - i. Select **force** in the **Observables types** list.
 - ii. Enter `force-drag` in the **Name** field.
 - iii. Click **OK** to close the dialog box.
- d. In the **Manage Adjoint Observable** dialog box ([Figure 3.6: Manage Observables Dialog Box \(p. 49\)](#)):

Figure 3.6: Manage Observables Dialog Box

- i. Select **force-drag** in the **Observables** list.
This will expand the dialog box to expose additional controls.
 - ii. Select **wall** under **Wall Zones** as being the wall on which the force is of interest - this is the cylinder wall. Set the value for the **X-Component** direction to 1, and keep the **Y-Component** direction set to 0. With these settings, the drag is identified as the observation of interest.
 - iii. Click **OK** to close the dialog box.
- e. In the **Adjoint Observables** dialog box (Figure 3.7: Adjoint Observables Dialog Box (p. 50)):

Figure 3.7: Adjoint Observables Dialog Box

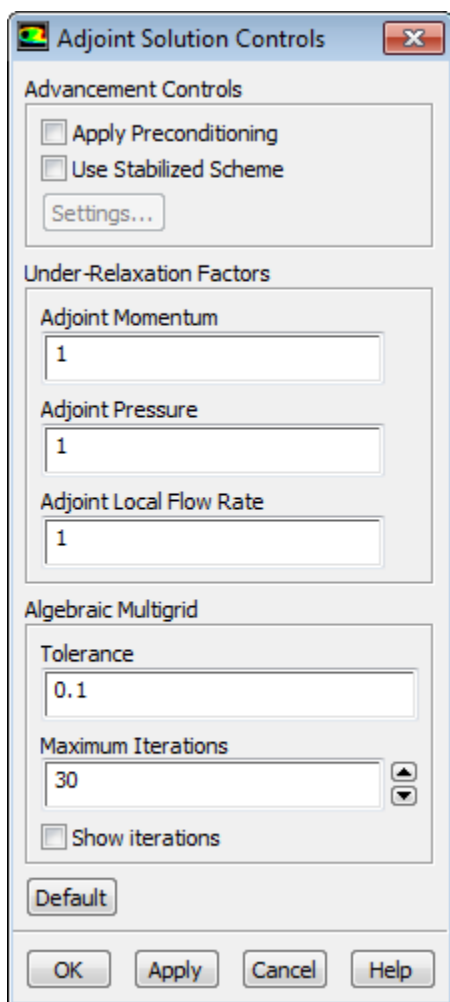
- i. Select **force-drag** in the list of **Observable Names**.
- ii. Click **Minimize** since we are trying to reduce the drag force.
- iii. Click **Evaluate** to print the force on the wall in the X -direction in the console

```
Observable name: force-drag  
Observable Value (n) = 1337.8475
```

This value is in SI units, with n denoting Newtons

- iv. Click **Close** to close the **Adjoint Observables** dialog box.
2. Verify the solution control settings by opening the **Adjoint Solution Controls** dialog box ([Figure 3.8: Adjoint Solution Controls Dialog Box \(p. 51\)](#)).

Adjoint → Controls

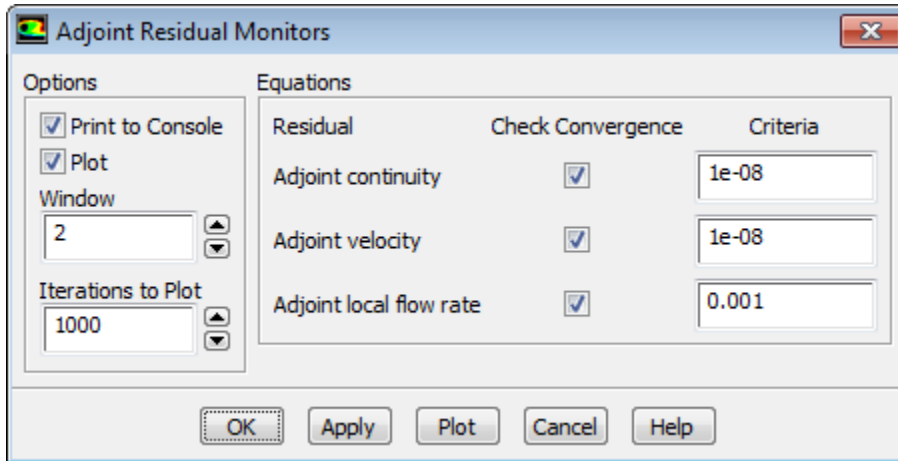
Figure 3.8: Adjoint Solution Controls Dialog Box

- a. Deselect the **Use Stabilized Scheme** option.

This corresponds to a setup without any preconditioning and all under-relaxation factors set to 1. This setup works for this simple laminar case, however less aggressive settings will be needed for larger turbulent cases.

- b. Click **OK** to close the dialog box.
3. Configure the adjoint solution monitors by opening the **Adjoint Residual Monitors** dialog box ([Figure 3.9: Adjoint Residual Monitors Dialog Box \(p. 52\)](#)).

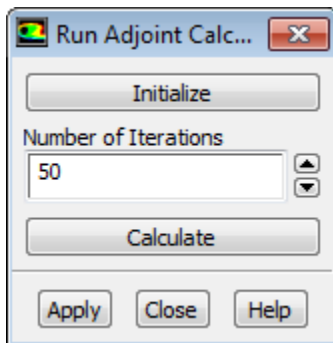
Adjoint → Monitors

Figure 3.9: Adjoint Residual Monitors Dialog Box

In the **Adjoint Residual Monitors** dialog box, you set the adjoint equations that will be checked for convergence as well as set the corresponding convergence criteria.

- a. Ensure that the **Print to Console** and **Plot** check boxes are selected.
 - b. Set the **Adjoint continuity** and **Adjoint velocity** convergence criteria to $1e-08$ and ensure that the **Check Convergence** check boxes are selected.
 - c. Click **OK** to close the dialog box.
4. Run the adjoint solver by opening the **Run Adjoint Calculation** dialog box (Figure 3.10: Adjoint Run Calculation Dialog Box (p. 52)).

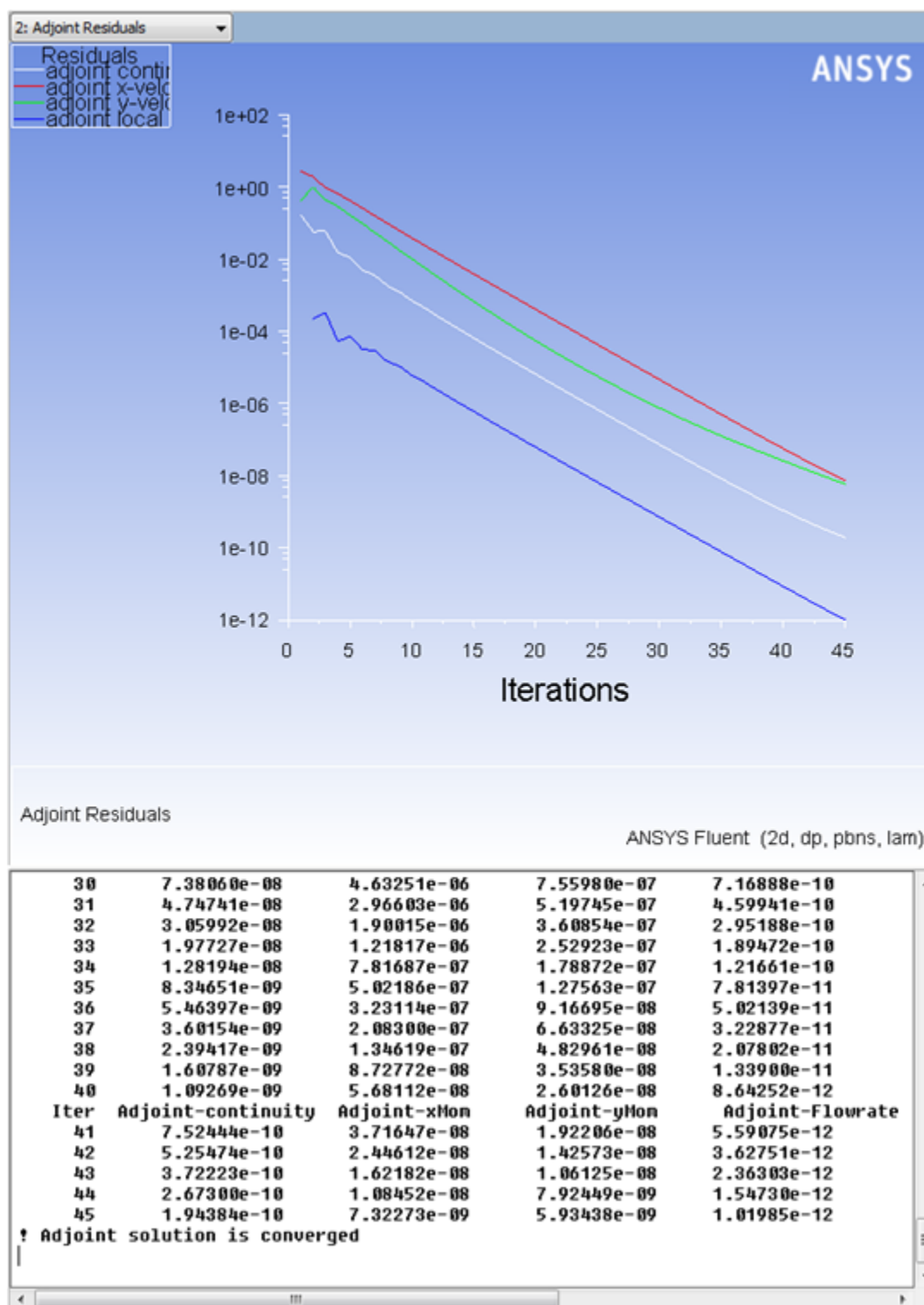
Adjoint → Run Calculation

Figure 3.10: Adjoint Run Calculation Dialog Box

- a. Click the **Initialize** button. This initializes the adjoint solution everywhere in the problem domain to zero.
- b. Set the **Number of Iterations** to 50. The adjoint solver is fully configured to start running for this problem.
- c. Click the **Calculate** button to advance the solver to convergence.

The main graphics window and console window is shown below:

Figure 3.11: Graphics and Console Window After Execution



- d. Once the calculation is complete, click **Close** to close the **Adjoint Run Calculation** dialog box.

3.2.4. Step 3: Postprocessing

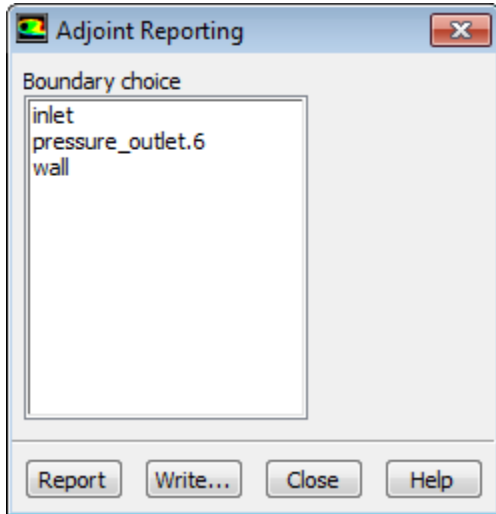
In this section, post-processing options for the adjoint solution are presented.

3.2.4.1. Boundary Condition Sensitivity

1. Open the **Adjoint Reporting** dialog box (Figure 3.12: Adjoint Reporting Dialog Box (p. 54)).

Adjoint → **Reporting**

Figure 3.12: Adjoint Reporting Dialog Box



2. Select **inlet** under **Boundary choice** and click the **Report** button to generate a report to the console of the available scalar sensitivity data on the inlet:

```
Updating shape sensitivity data.
Done.
```

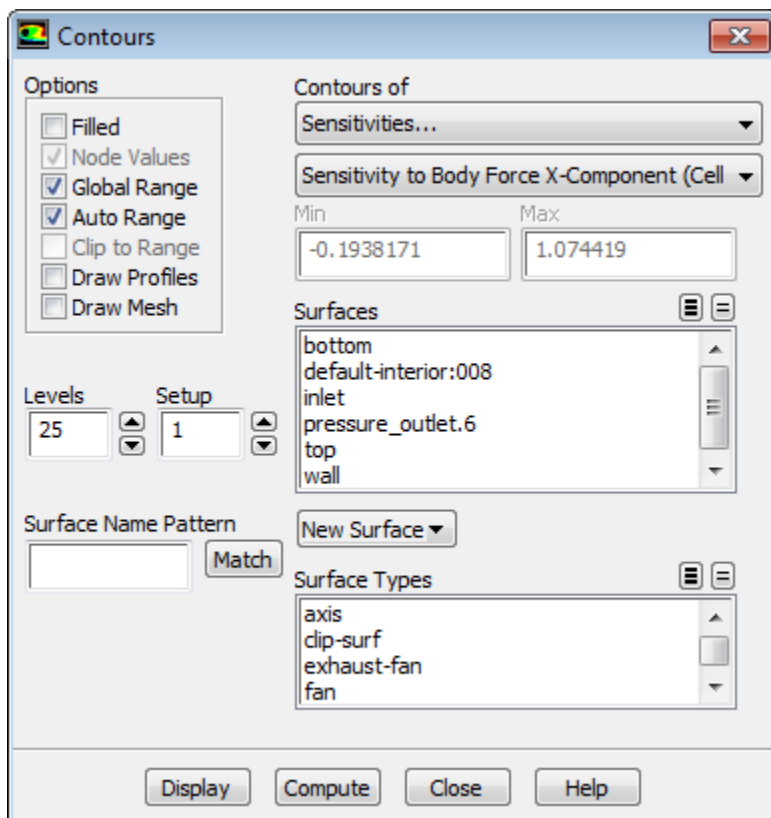
```
Boundary condition sensitivity report: inlet
Observable: force-drag
Velocity Magnitude (m/s) = 40   Sensitivity ((n)/(m/s)) = 54.291264
Decrease Velocity Magnitude to decrease force-drag
```

3.2.4.2. Sources of Mass and Momentum

1. Open the **Contours** dialog box.

 **Graphics and Animations** →  **Contours** → **Set Up...**

2. Under **Contours of**, select **Sensitivities...** and **Sensitivity to Body Force X-Component (Cell Values)** as shown:

Figure 3.13: Contours Dialog Box When Plotting Adjoint Fields

3. Set the **Levels** to 25.
4. Click **Display** to view the contours (Figure 3.14: Adjoint Sensitivity to Body Force X-Component Contours (p. 56)).

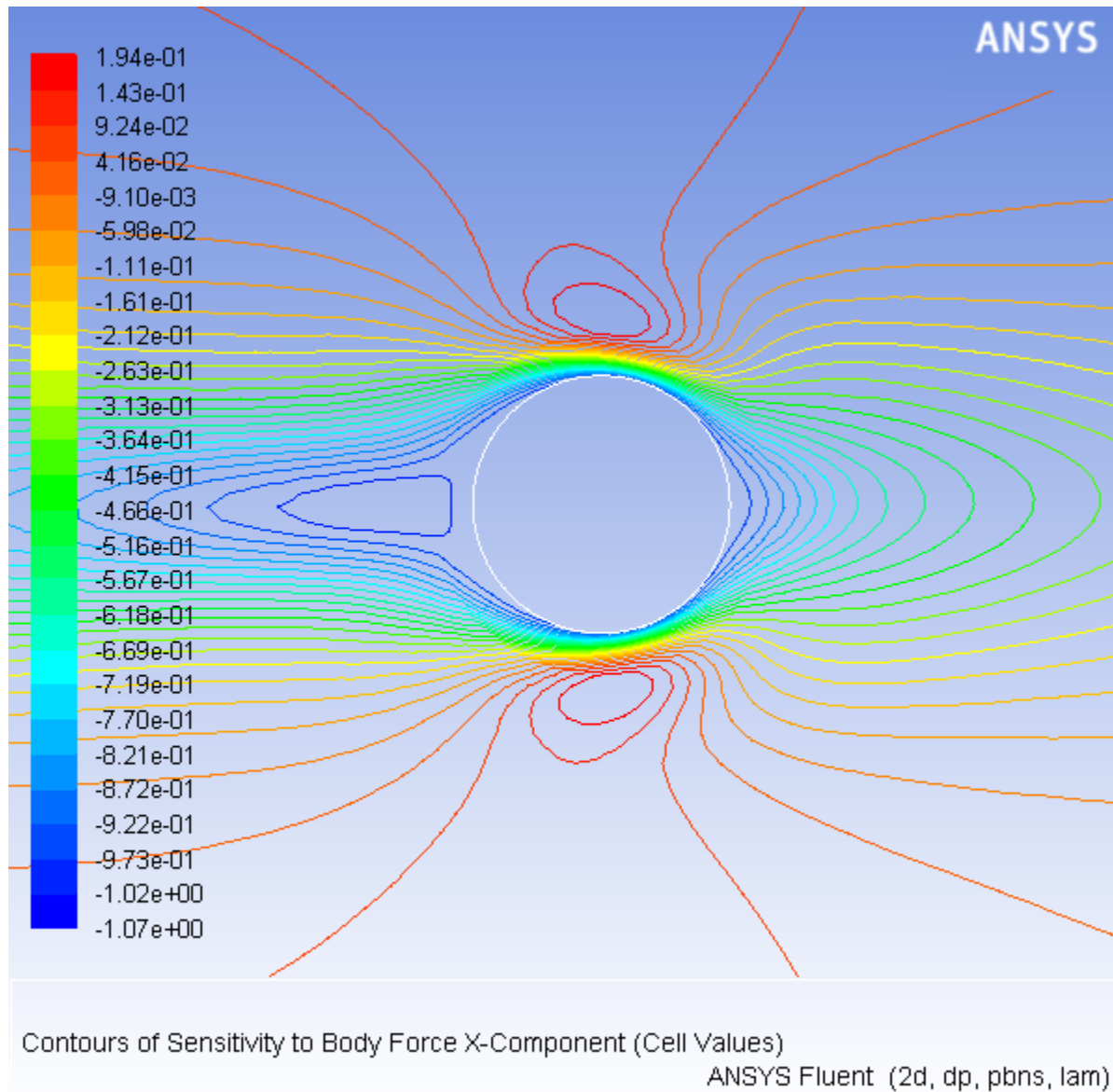
Figure 3.14: Adjoint Sensitivity to Body Force X-Component Contours

Figure 3.14: Adjoint Sensitivity to Body Force X-Component Contours (p. 56) shows how sensitive the drag on the cylinder is to the application of a body force in the X -direction in the flow. Note that the effect of applying a body force downstream of the cylinder is minimal, as expected. If a body force is applied directly upstream of the cylinder, the disturbed flow is incident on the cylinder and modifies the force that it experiences.

The contours also show how the computed drag is affected by lack of convergence in the original X -momentum equation for the flow.

5. Select **Sensitivity to Mass Sources (Cell Values)** and display those contours (Figure 3.15: Adjoint Sensitivity to Mass Sources Contours (p. 57)):

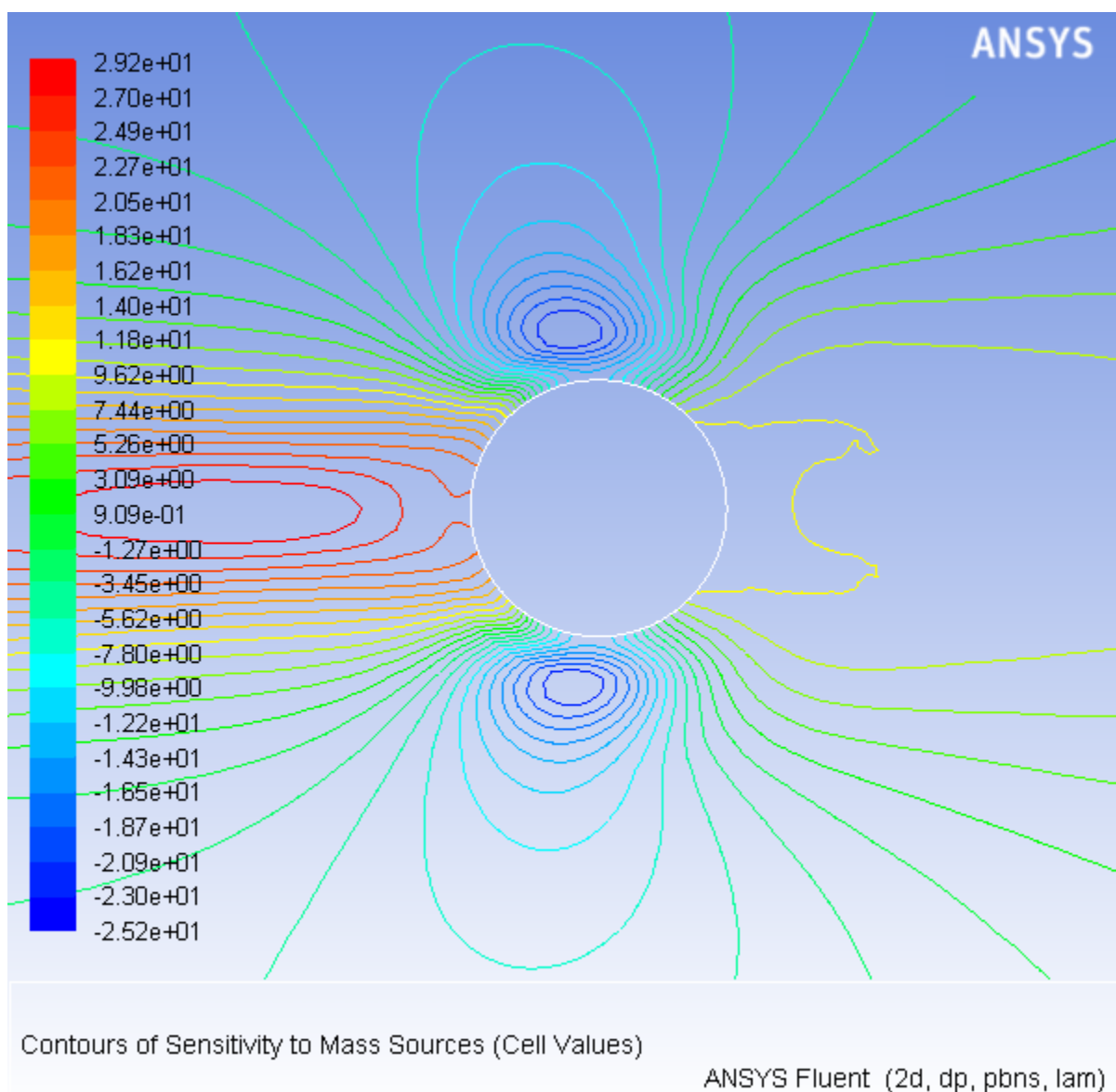
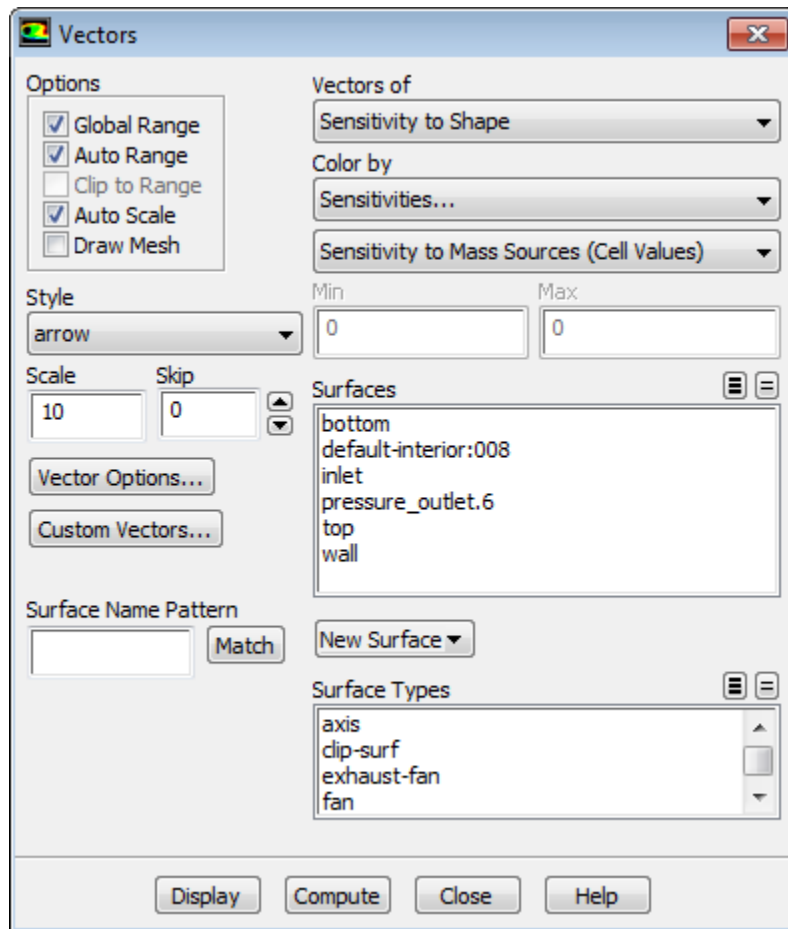
Figure 3.15: Adjoint Sensitivity to Mass Sources Contours

Figure 3.15: Adjoint Sensitivity to Mass Sources Contours (p. 57) shows how the drag on the cylinder is affected by the addition of sources and sinks of mass. If material were to be systematically removed from the flow field upstream of the cylinder, then the drag will be reduced. This can be deduced since the adjoint pressure is negative upstream of the cylinder. In contrast, if material is removed adjacent to the side of the cylinder, then the drag is expected to increase. This is indicated by positive values of the adjoint pressure. Such behavior is expected since removing material at the cylinder wide point will accelerate the flow around the side of the cylinder, thereby increasing the local skin friction drag.

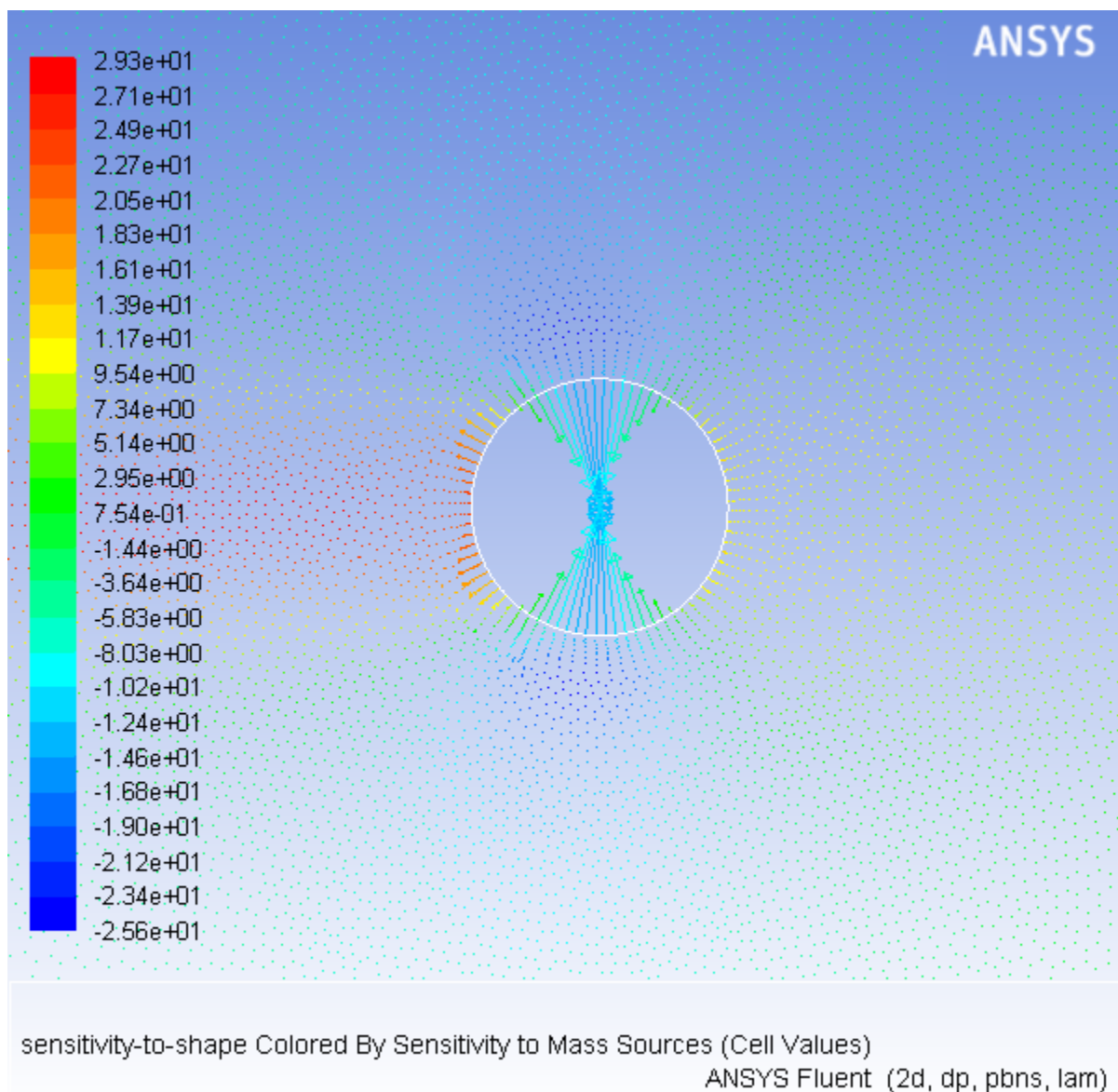
3.2.4.3. Shape Sensitivity

1. Open the **Vectors** dialog box (Figure 3.16: Vectors Dialog Box (p. 58))

🔍 Graphics and Animations → 📊 Vectors → Set Up...

Figure 3.16: Vectors Dialog Box

2. Under **Vectors of**, select **Sensitivity to Shape**.
3. Under **Color by**, select **Sensitivities...**, then select **Sensitivity to Mass Sources (Cell Values)**.
4. Click the **Display** button to view the vectors (Figure 3.17: Shape Sensitivity Colored by **Sensitivity to Mass Sources (Cell Values)** (p. 59)).

Figure 3.17: Shape Sensitivity Colored by Sensitivity to Mass Sources (Cell Values)

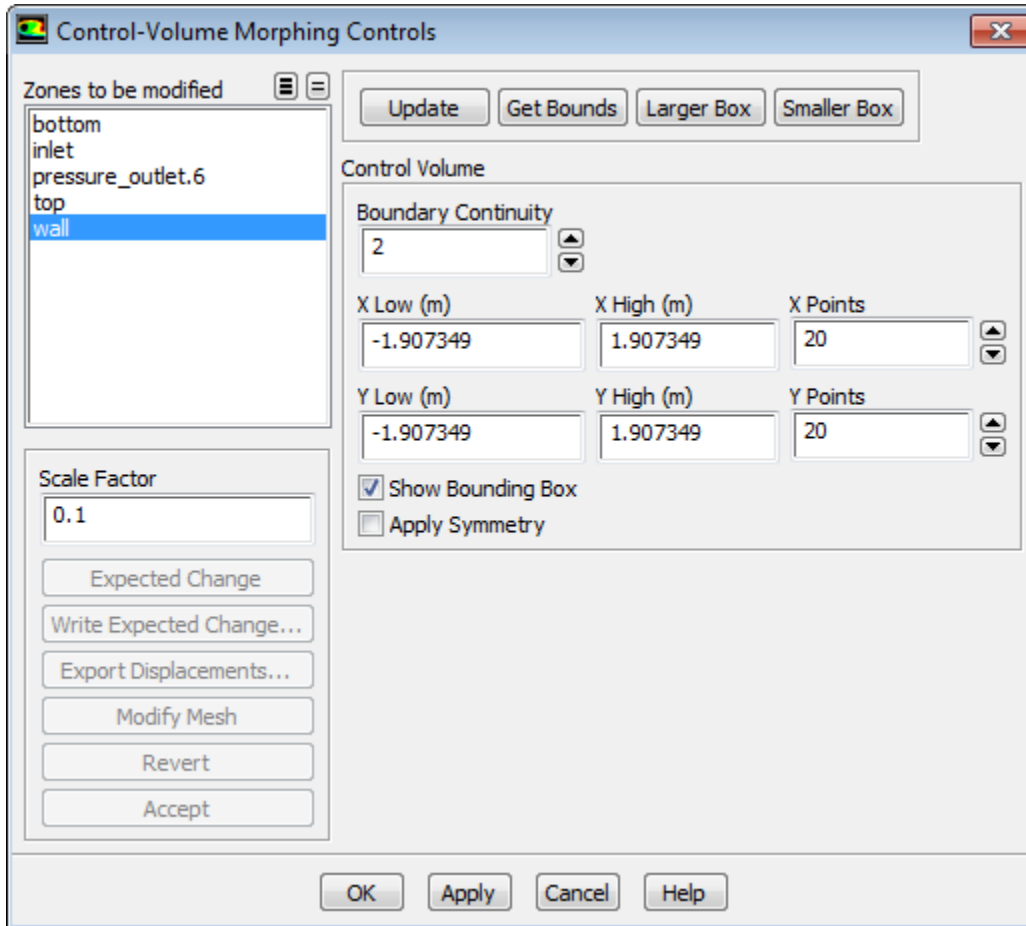
This plot shows how sensitive the drag on the cylinder is to changes in the surface shape. The drag is affected more significantly if the cylinder is deformed on the upstream rather than the downstream side. Maximum effect is achieved by narrowing the cylinder in the cross-stream direction.

3.2.5. Step 4: Shape Modification

In this section, the steps needed to modify the cylinder shape in order to reduce the drag are described.

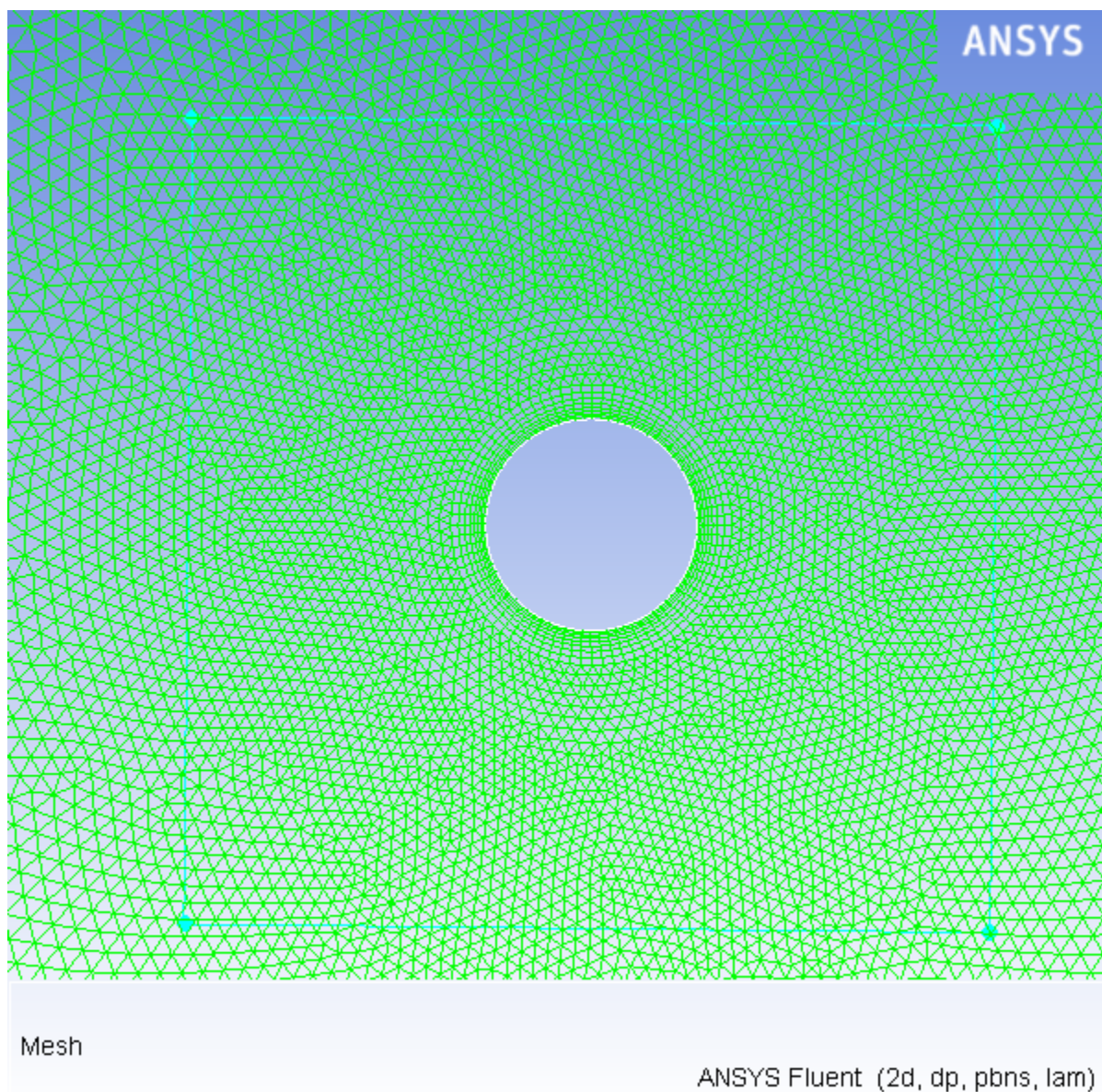
1. Open the **Control-Volume Morphing Controls** dialog box (Figure 3.18: Control-Volume Morphing Controls Dialog Box (p. 60)).

Adjoint → **Control-Volume Morphing**

Figure 3.18: Control-Volume Morphing Controls Dialog Box

2. Perform the following steps to set up the control volume for mesh deformation:
 - a. In the **Zones to be modified** selection box, select **wall**, that is, the cylinder, as the boundary that is to be modified based on the shape sensitivity calculation.
 - b. Click the **Get bounds** button to populate the bounding box for the wall of the cylinder.
 - c. Click the **Larger box** button several times until the bounding box extends from $[-1.907349, 1.907349]$ in both coordinate directions.
 - d. Set **X Points** and **Y Points** to 20. For additional information about control points, see [Defining the Control Volume for Control-Volume Deformation](#) (p. 40).
 - e. Drawing the mesh (using the **Mesh Display** dialog box) at this point will also display the bounding box ([Figure 3.19: Control-Volume Around Cylinder](#) (p. 61)).

◆ **General** → **Display...**

Figure 3.19: Control-Volume Around Cylinder

- f. Set the **Scale factor** to 0.1 and click the **Update** button to propagate the surface sensitivity out to the control points. The sign of the scale factor determines the orientation of the control-point sensitivity vector field that is displayed. In this example, the positive vector indicates that the drag will be reduced (as the observable has been defined as a minimized force in the direction $X=+1, Y=0$).
- g. Click the **Expected Change** button. The expected change in the drag that will result from the modification is printed in the console window:


```
Observable type: force
Expected change (n): -89.326027
```
- h. Click the **Modify mesh** button to apply the mesh deformation that will reposition the boundary and interior nodes of the mesh. Information regarding the mesh modification is printed in the console window:

Modifying walls using control volume deformation.

Updating mesh (steady, mesh iteration = 00001, pseudo time step 1.0000e+00)...

Dynamic Mesh Statistics:

Minimum Volume = 3.64254e-04


Maximum Volume = 6.36270e-01

Maximum Cell Skew = 3.71199e-01 (cell zone 11)

Minimum Orthogonal Quality = 8.11046e-01 (cell zone 11)

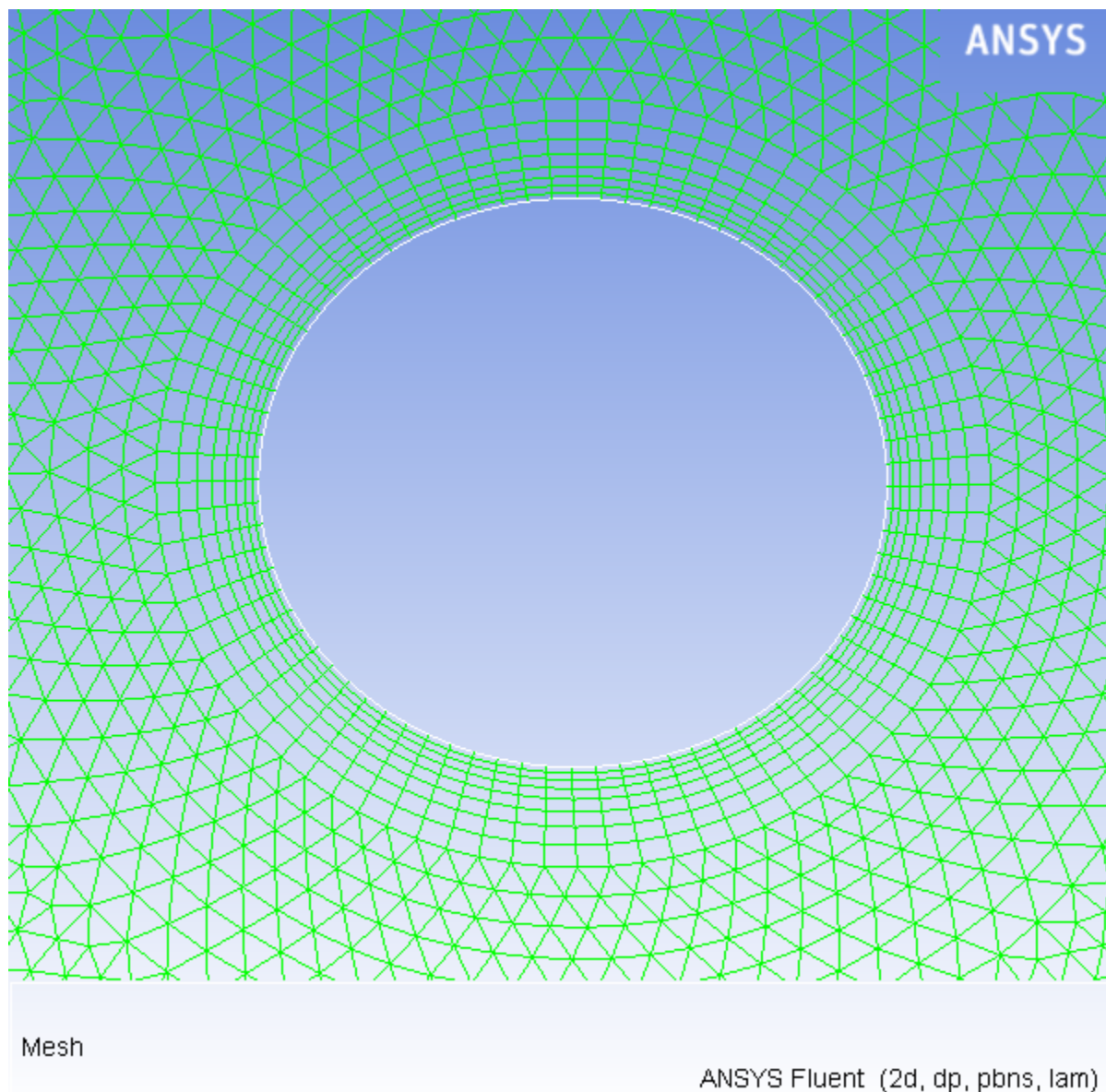
done.

- i. Click **Accept**.
- j. Redraw the mesh.

 **General** → **Display...**

The effect on the mesh is shown in [Figure 3.20: Mesh After Deformation to Reduce Drag \(p. 62\)](#):

Figure 3.20: Mesh After Deformation to Reduce Drag



- k. Re-converge the conventional flow calculation for this new geometry in the **Run Calculation** task page.

Run Calculation

- i. Change the **Number of Iterations** to 100.
- ii. Click **Calculate**.

In order to make a comparison with the change predicted from the adjoint, it is important to converge the modified system well so that the comparison is not polluted due to incomplete convergence. Iterating for 100 iterations is sufficient to converge the drag for the revised geometry. Note that the currently loaded case file already has drag set up as a force monitor.

The new force is observed to be:

Observed value (n) = 1250.243

Note a change of -87.6 compared to the drag on the undeformed cylinder. This value compares very well with the change of -89.3 that is predicted from the adjoint solver.

- l. Once a solution is obtained, you can also use ANSYS Fluent to display a report of the forces in the simulation.

Reports → **Forces** → **Set Up...**

This displays the **Force Reports** dialog box. You can use this dialog box to observe the total force reported in the direction specified.

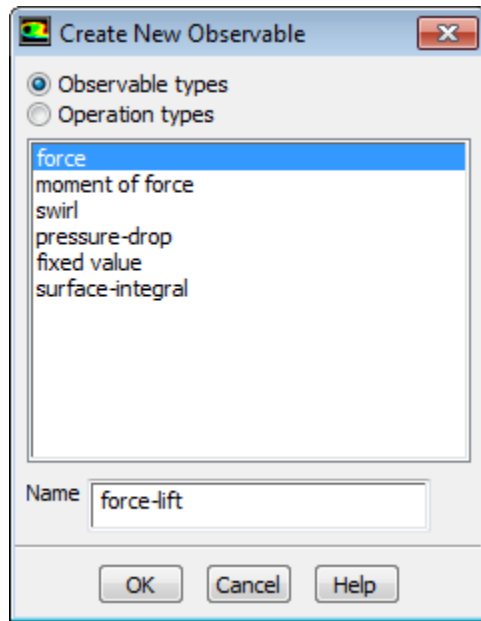
- i. In the **Force Reports** dialog box, make sure **Forces** is selected under **Options**.
- ii. Make sure **X** is set to 1 and that **Y** is set to 0
- iii. Make sure **wall** is selected as a **Wall Zone**.
- iv. Click **Print**.

The force report is printed in the console window.

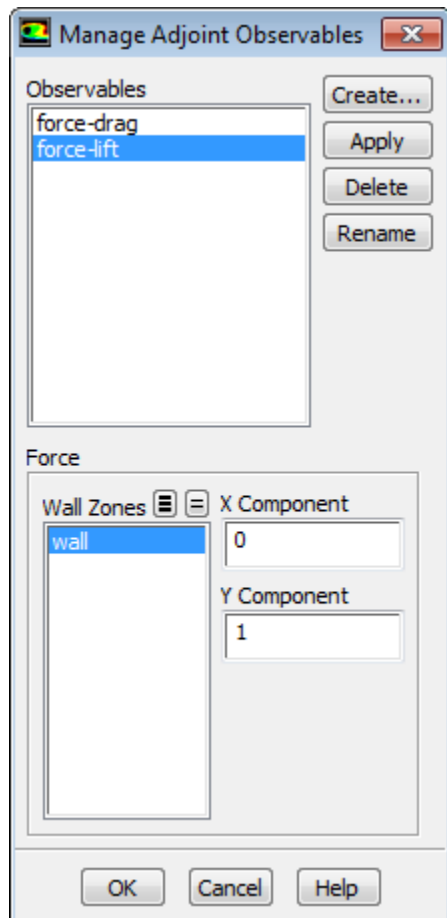
3.2.6. Step 5: Extending the Observable to a Ratio

So far, we have shown how the adjoint solver can be used to reduce drag on the cylinder. Now we will extend this investigation to take lift into account as well, using a single calculation. To do this, we will create a replacement observable representing the ratio of lift and drag, which we will proceed to maximize.

1. Perform the following steps to extend the existing case and create a ratio of two other observables:
 - a. In the **Adjoint Observables** dialog box, click the **Manage...** button to open the **Manage Observable** dialog box.
 - b. In the **Manage Observables** dialog box, click the **Create...** button to open the **Create New Observable** dialog box.
 - c. In the **Create New Observable** dialog box:



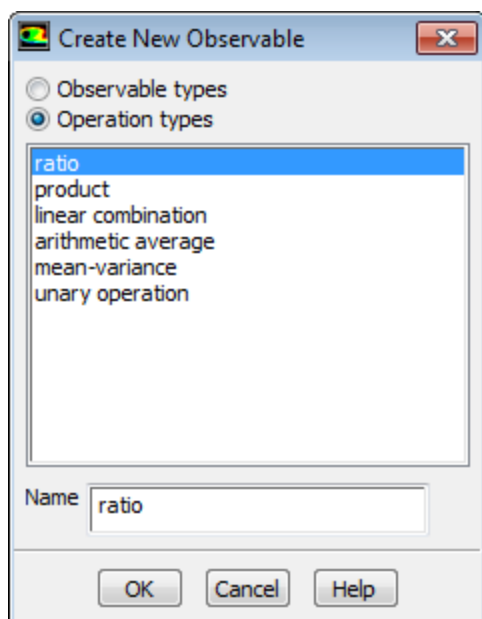
- i. Select **force** in the **Observables types** list.
 - ii. Enter `force-lift` in the **Name** field.
 - iii. Click **OK** to close the dialog box.
- d. In the **Manage Adjoint Observable** dialog box:



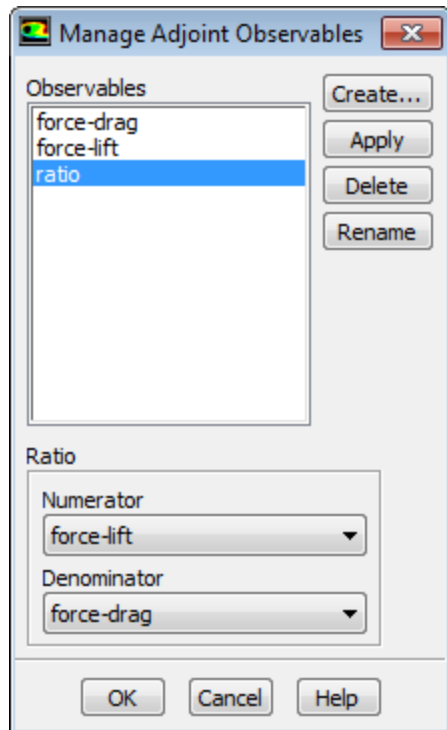
- i. Select **force-lift** in the **Observables** list.

This will expand the dialog box to expose additional controls.

- ii. Select **wall** under **Wall Zones** as being the wall on which the force is of interest - this is the cylinder wall. Set the value for the **Y-Component** direction to 1, and keep the **X-Component** direction set to 0. With these settings, the lift is identified as the observation of interest.
- iii. Click **OK** to close the dialog box.
- e. In the **Manage Observables** dialog box, click the **Create...** button to open the **Create New Observable** dialog box.
- f. In the **Create New Observable** dialog box:



- i. Select the **Operation types** option.
- ii. Select **ratio** as the operation type.
- iii. Keep the default **Name** for the observable.
- iv. Click **OK** to close the **Create New Observable** dialog box.
- g. In the **Manage Observables** dialog box:



- i. Make sure that **ratio** is selected in the list of observables.
 - ii. Select **force-lift** from the **Numerator** drop-down list.
 - iii. Select **force-drag** from the **Denominator** drop-down list.
 - iv. Click **OK** to close the **Manage Observables** dialog box.
- h. In the **Adjoint Observables** dialog box:
- i. Make sure that **ratio** is selected in the list of observables.
 - ii. Make sure that **Maximize** is selected.
 - iii. Click **Close** to close the dialog box.

The case is now set up to attempt to maximize the lift/drag as a ratio.

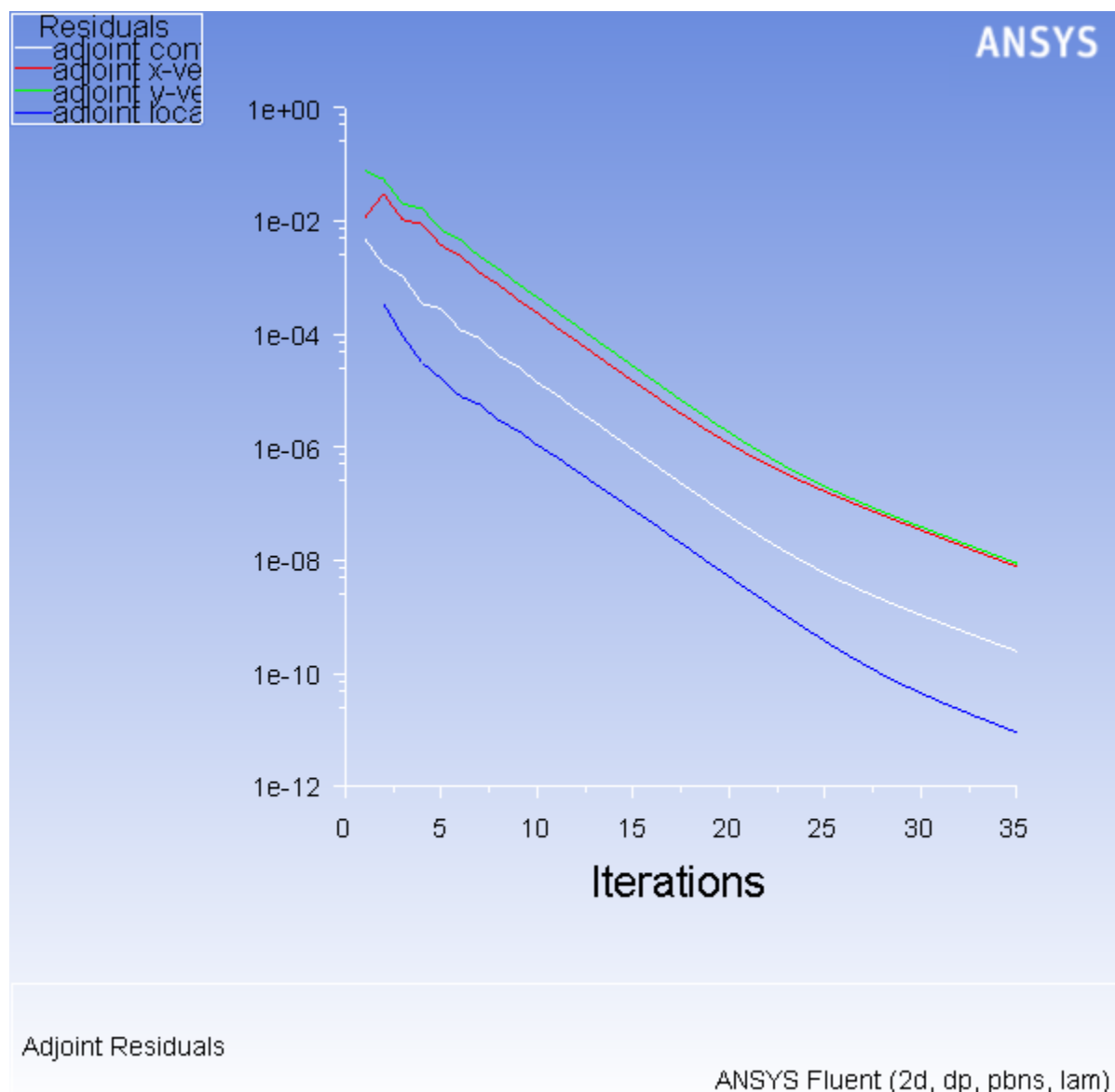
2. Use the previous solver settings to recalculate a solution.
 - a. Run the adjoint solver by opening the **Run Adjoint Calculation** dialog box.

Adjoint → Run Calculation

- b. Click the **Initialize** button. This initializes the adjoint solution everywhere in the problem domain to zero. When you are prompted to confirm whether it is acceptable to overwrite existing data, click **Yes**.
- c. Set the **Number of Iterations** to 50. The adjoint solver is fully configured to start running for this problem.
- d. Click the **Calculate** button to advance the solver to convergence.

The main graphics window displaying adjoint residuals is shown below.

Figure 3.21: Adjoint Residuals



3. Investigate the results and the morphing of the mesh.

a. Open the **Adjoint Reporting** dialog box.

Adjoint → Reporting

b. Select **inlet** under **Boundary choice** and click the **Report** button to generate a report of the available scalar sensitivity data on the inlet to the Fluent console:

```
Updating shape sensitivity data.
Done.
```

```
Boundary condition sensitivity report: inlet
Observable: ratio
Velocity Magnitude (m/s) = 40 Sensitivity ((dimensionless)/(m/s)) = 0.00018563686
Increase Velocity Magnitude to increase ratio
```

This shows that, from the current calculation, a change in inlet flow velocity of 1 m/s is required to change the lift/drag ratio by 0.00018.

- c. Open the **Control-Volume Morphing Controls** dialog box.

Adjoint → Control-Volume Morphing

Use the existing morphing deformation setup, select **wall**, click **Update**, and click **Expected Change** to report the expected change (an undefined unit is expected):

```
Observable type: force  
Expected change (undefined): 0.12046795
```

- d. Click **Modify Mesh** to deform the mesh then click the **Accept** button.
- e. Re-run the fluids calculation.
- i. Return to the **Run Calculation** task page in Fluent, keep the **Number of Iterations** set to 100, and click **Calculate**.
 - ii. Once the calculation is complete, return to the **Control-Volume Morphing Controls** dialog box, click **Update**, **Modify Mesh**, then **Accept**.
 - iii. Repeat this process five times (design iterations) so that you have a scaled residual profile ([Figure 3.22: Scaled Residuals After 5 Design Iterations \(p. 69\)](#)) and deformed mesh ([Figure 3.23: The Deformed Cylinder after 5 Design Iterations \(p. 70\)](#)):

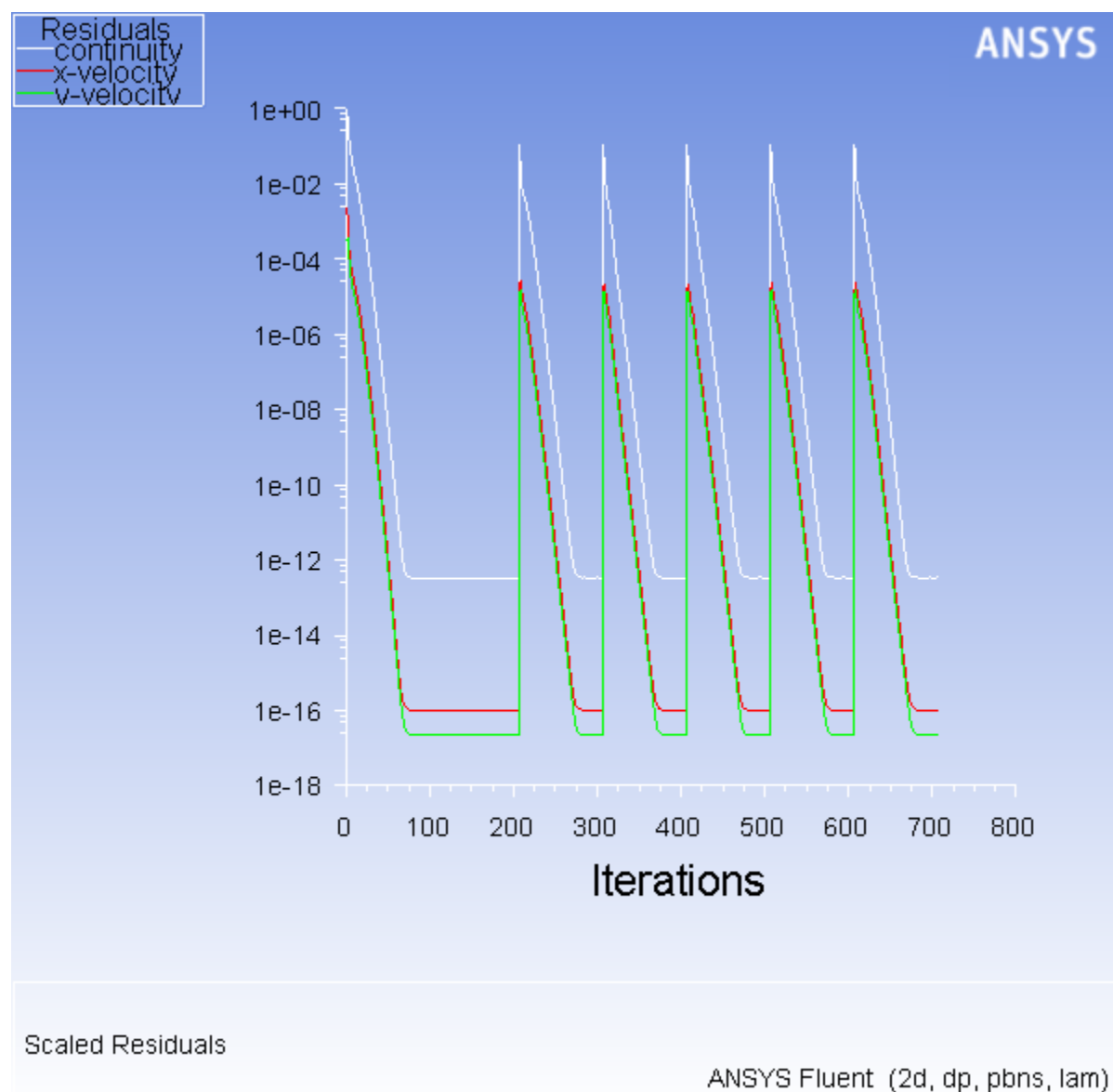
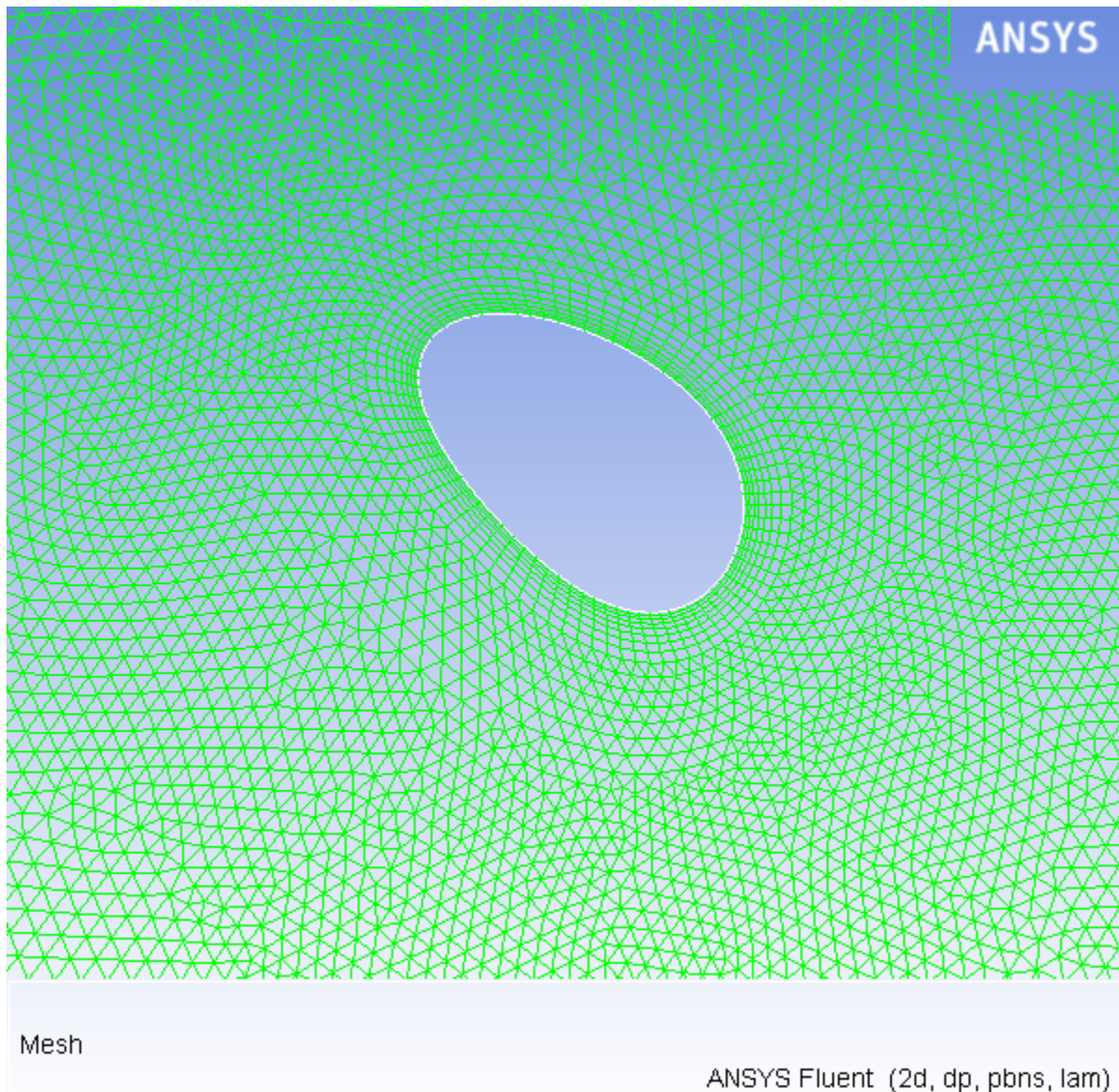
Figure 3.22: Scaled Residuals After 5 Design Iterations

Figure 3.23: The Deformed Cylinder after 5 Design Iterations

- f. Reevaluate the observable value and observe an increasing value.

```
Observable type: ratio  
Expected change (undefined): 0.41788214
```

You can perform additional design iterations to observe that a wing profile begins to take shape. After approximately 10 design iterations, the observable reported becomes greater than 1, indicating that lift is becoming the dominant force.

3.3. Summary

The adjoint solver add-on has been used to compute the sensitivity of the drag on a circular cylinder to various inputs for a previously-computed flow field. The process of setting up and running the adjoint solver has been illustrated. Steps to perform various forms of post-processing have also been described. The sensitivity of the drag with respect to the shape of the cylinder has been combined with mesh morphing to make a change to the design that reduces the drag in a predictable manner. In addition,

the lift/drag ratio observable has been analyzed to show how the free form deformation can be used to optimize the design with consideration for multiple loads.

Index

A

- Adjoint Observables dialog box, 17
- Adjoint Reporting dialog box, 37
- Adjoint Residual Monitors dialog box, 31
- Adjoint Solution Controls dialog box, 24
- Adjoint Solution Methods dialog box, 23
- adjoint solver model
 - control volume deformation, 40
 - discrete solver, 7
 - discrete versus continuous solver, 6
 - installing, 15
 - loading, 16
 - modifying the geometry, 38
 - observable definition, 17
 - overview, 1
 - postprocessing, 32
 - residual monitors, 31
 - running the calculation, 31
 - scope of functionality, 16
 - shape modification, 38
 - smoothing and mesh morphing, 12
 - solution controls, 24
 - solution methods, 23
 - solution-based adaption, 11
 - stabilization scheme settings, 26
 - modal scheme, 26
 - spatial scheme, 29
 - text user interface (TUI), 41
 - theory, 1
 - tutorial, 43
 - using, 15
 - using data to improve design, 11
- arithmetic average observable, 20
- arithmetic average operation, 5, 22

C

- calculating, 31
- control volume deformation, 40
- control volume morphing, 38
- Control-Volume Morphing Controls dialog box, 38
- conventions used in this guide, vi
- Create New Observable dialog box, 22

D

- discrete solver, 7
- discrete versus continuous solver, 6

F

- fixed value observable, 3, 20, 22

- force observable, 3, 20, 22

G

- geometry modification, 38

I

- installing, 15

L

- linear combination observable, 20
- linear combination operation, 5, 22
- loading, 16

M

- Manage Adjoint Observables dialog box, 19
- mean variance observable, 21
- mean variance operation, 5
- mean-variance operation, 22
- moment of force observable, 3, 20, 22

O

- observable definition, 17
- observables, 3
 - arithmetic average, 20
 - creating, 22
 - fixed value, 3, 20
 - force, 3, 20
 - linear combination, 20
 - managing, 19
 - mean variance, 21
 - moment of force, 3, 20
 - product, 21
 - ratio, 21
 - renaming, 23
 - surface integral, 3, 21
 - swirl, 3, 21
 - total pressure drop, 3, 20
 - unary operation, 21
- operations, 5
 - arithmetic average, 5
 - linear combination, 5
 - mean variance, 5
 - product, 5
 - ratio, 5
 - unary, 5
- overview, 1

P

- postprocessing, 32
 - field data, 32
 - scalar data, 37
- pressure drop observable, 22

product observable, 21
product operation, 5, 22

R

ratio observable, 21
ratio operation, 5, 22
Rename Observable dialog box, 23
renaming observables, 23
residual monitors, 31
Run Adjoint Calculation dialog box, 31

S

scope of functionality, 16
shape modification, 38
smoothing and mesh morphing, 12
solution controls, 24
solution methods, 23
solution-based adaption, 11
stabilization scheme settings, 26
 modal scheme, 26
 spatial scheme, 29
Stabilized Scheme Settings dialog box, 26
surface integral observable, 3, 21-22
swirl observable, 3, 21-22

T

text user interface (TUI), 41
theory, 1
total pressure drop observable, 3, 20
tutorial, 43

U

unary operation, 5, 22
unary operation observable, 21
using data to improve design, 11
using the adjoint solver, 15