

# **A User's Guide to MISES 2.63**

Mark Drela, Harold Youngren  
MIT Aerospace Computational Design Laboratory  
February 2008

This is a user manual for the MISES viscous/inviscid cascade analysis and design system.

# Contents

<b>1</b>	<b>Overview</b>	<b>4</b>
<b>2</b>	<b>Internal Reference Quantities</b>	<b>4</b>
<b>3</b>	<b>Streamsurface and Blade geometry definition</b>	<b>5</b>
<b>4</b>	<b>Input Files</b>	<b>6</b>
4.1	Blade coordinate file <code>blade.xxx</code> . . . . .	7
4.2	Geometry parameter file <code>bparm.xxx</code> . . . . .	9
4.3	Modified-geometry parameter file <code>bspec.xxx</code> . . . . .	10
4.4	Geometry parameter specification file <code>bplist.xxx</code> . . . . .	10
4.5	Stream surface file <code>stream.xxx</code> . . . . .	11
4.6	Prescribed-loss file <code>loss.xxx</code> . . . . .	12
4.7	Wall-suction specification file <code>suct.xxx</code> . . . . .	13
4.8	Flow condition file <code>ises.xxx</code> . . . . .	14
4.8.1	Variable,Constraint indices . . . . .	21
4.8.2	Pressure-correction term . . . . .	22
4.8.3	Momentum/Entropy conservation . . . . .	22
4.8.4	Artificial dissipation . . . . .	23
4.8.5	Artificial dissipation level selection . . . . .	25
4.8.6	Dissipation enhancement during convergence . . . . .	27
4.9	Example <code>ises.xxx</code> input-file lines . . . . .	28
4.9.1	Lines 1–4. Variables, constraints, flow conditions. . . . .	28
4.9.2	Lines 6–7. Viscous flow parameters. . . . .	30
4.9.3	Line 8. Isentropy and dissipation . . . . .	31
4.9.4	Line 9. Streamtube thickness mode amplitudes . . . . .	32
4.10	Geometry perturbation mode specification file <code>modes.xxx</code> . . . . .	32
4.11	Design-parameter specification file <code>params.xxx</code> . . . . .	33

<b>5</b>	<b>Program Descriptions</b>	<b>34</b>
5.1	ISET . . . . .	35
5.1.1	Basic Initialization . . . . .	35
5.1.2	Panel solution . . . . .	35
5.1.3	Initial surface gridding . . . . .	36
5.1.4	Grid smoothing . . . . .	38
5.1.5	Initial solution file output . . . . .	38
5.1.6	Grid parameters . . . . .	38
5.1.7	Grid parameter saving, recall . . . . .	39
5.1.8	Smoothing and writing the grid . . . . .	39
5.2	ISES . . . . .	40
5.2.1	Inflow boundary conditions . . . . .	40
5.2.2	Outflow boundary conditions . . . . .	42
5.3	IPLOT . . . . .	43
5.3.1	Blade surface plots . . . . .	43
5.3.2	Suction . . . . .	45
5.3.3	Streamtube plots . . . . .	45
5.3.4	Contour/grid plots . . . . .	47
5.3.5	Wake profile plots . . . . .	47
5.3.6	$r,b$ vs $m'$ stream surface definition plots . . . . .	47
5.3.7	Wheel view . . . . .	47
5.4	EDP . . . . .	47
5.4.1	Surface parameterization . . . . .	48
5.4.2	EDP execution . . . . .	49
5.4.3	Modal-Inverse . . . . .	52
5.4.4	Parametric-Inverse . . . . .	53
5.4.5	Blade Translation, Scaling, Rotation . . . . .	53
5.4.6	Modified-Blade Output . . . . .	53
5.4.7	ISES Parameter Changes . . . . .	54

5.4.8	Inverse Design Session . . . . .	54
5.4.9	Parameter-Modification Design Session . . . . .	55
5.5	POLAR . . . . .	56
5.6	BLDSET . . . . .	58
<b>6</b>	<b>Optimization</b>	<b>59</b>
<b>7</b>	<b>Graphics</b>	<b>59</b>
<b>8</b>	<b>General Hints</b>	<b>59</b>
8.1	Viscous solutions . . . . .	59
8.2	Inverse solutions . . . . .	60
8.3	Grid resolution . . . . .	60
8.4	Execution times . . . . .	61

# 1 Overview

The MISES system is a collection of programs for cascade analysis and design. This includes programs for grid generation and initialization, flow analysis, plotting and interpretation of results, and an interactive program to specify design conditions.

The block diagram for these programs is given at the end of this manual. The basic grid and flow data file for a case is the so-called *state file* named `idat.xxx`, where “xxx” is an extension suffix used to designate the case being run. The state file is initialized using **ISET** from the blade geometry file `blade.xxx` and the optional stream surface geometry file `stream.xxx` and the prescribed loss schedule file `loss.xxx`. The flow solver **ISES** uses the state file and a flow condition file `ises.xxx` that specifies the flow conditions and program configuration flags. The **POLAR** program performs the same calculations as **ISES**, but for a set of specified parameters. Additional design condition information can be interactively added to the state file using the **EDP** pressure edit program. The **IPLOT** program plots the flow and geometry data from the state file in an interactive plotting session.

## 2 Internal Reference Quantities

All flow variables used by MISES are defined in the relative frame. Internally, MISES employs rotation-corrected stagnation density and speed of sound,  $\rho_{oa}$ ,  $a_{oa}$ , as the basic reference flow variables, so that  $\rho_{oa} = 1$  and  $a_{oa} = 1$  by definition. The corresponding rotation-corrected stagnation pressure  $p_{oa}$  and enthalpy  $h_{oa} \equiv I$  (i.e. rothalpy) are then related as follows.

$$\gamma p_{oa} = \rho_{oa} a_{oa}^2 = (\gamma - 1) \rho_{oa} I$$

The Fortran names and assigned values of the internal reference quantities are  $\rho_{oa} = \text{RSTRO} = 1$ , and  $I = \text{HSTRO} = 1/(\gamma - 1)$ . Normally, these are not of concern for the user, since all input and output is typically done via ratios and related dimensionless quantities, following common conventions. For example, the outlet pressure is specified as  $p_2/p_{o1}$ , where  $p_{o1}$  is the conventional relative-frame total pressure at the inlet at the specified radius  $r_1$ .

The  $(\ )_{oa}$  notation means an “absolute” total quantity (not to be confused with an *absolute-frame* total quantity), in the sense that it implies an isentropic process where the fluid is brought to rest in the relative frame, *and* taken to the rotation center  $r = 0$ . Bringing the fluid to rest in the relative frame at a fixed radius  $r$  gives the conventional stagnation quantities  $\rho_o$ ,  $a_o$ , etc. The absolute and conventional stagnation quantities are related by the usual isentropic relations.

$$\begin{aligned} \frac{p_o}{p_{oa}} &= \left( \frac{I + \Omega^2 r^2 / 2}{I} \right)^{\frac{\gamma}{\gamma-1}} \\ \frac{\rho_o}{\rho_{oa}} &= \left( \frac{I + \Omega^2 r^2 / 2}{I} \right)^{\frac{1}{\gamma-1}} \\ \frac{a_o^2}{a_{oa}^2} &= \frac{I + \Omega^2 r^2 / 2}{I} \end{aligned}$$

The reason  $\rho_{oa}$  and  $a_{oa}$  were chosen for the internal reference quantities is precisely because they are independent of radius, and thus considerably simplify the internal formulation of the code. But again, they are transparent to the user, and are only described here in case source-code additions are being contemplated.

### 3 Streamsurface and Blade geometry definition

The blade airfoil and grid domain geometry are defined in the standard  $m' - \theta$  streamsurface coordinate system, shown in Figure 1. With  $z$  denoting the cylindrical axis coordinate and  $r$  the local streamsurface radius, the  $m'$  coordinate is defined by

$$m' = \int \frac{dm}{r} = \int \frac{\sqrt{dr^2 + dz^2}}{r}$$

while  $\theta$  is the usual circumferential angle. The total arc length increment  $ds$  in the stream surface is given by

$$ds = \sqrt{dr^2 + dz^2 + (r d\theta)^2} = \sqrt{dm^2 + (r d\theta)^2} = r\sqrt{dm'^2 + d\theta^2} = r ds'$$

The normalized arc length  $ds' = \sqrt{dm'^2 + d\theta^2}$  will be used later as a spline parameter to define the blade shape in the  $m' - \theta$  plane. The intermediate coordinate  $dm = \sqrt{dr^2 + dz^2}$  is the physical arc length increment projected onto the meridional  $r - z$  plane, and is not used explicitly in MISES.

The transformation from physical space to the  $m' - \theta$  plane is angle-preserving. Hence, no transformation is required for flow angles or surface normal directions. This simplifies imposition of boundary conditions such as a specified inlet flow angle, or the normal-offsetting of the inviscid flow by the viscous displacement thickness.

For 2-D cascades,  $r$  becomes an arbitrary constant scaling length, and hence

$$m' = \frac{z}{r} \quad (2\text{-D cascade}).$$

For a purely radial cascade (e.g. squirrel cage fan)  $z$  is a constant, and in this case

$$\begin{aligned} m' &= \int \frac{dr}{r} = \ln r && (\text{radial outflow cascade}) \\ m' &= \int \frac{-dr}{r} = \ln 1/r && (\text{radial inflow cascade}) \end{aligned}$$

For some other analytically-defined  $r(z)$  distributions, the general  $m'$  definition might also be integrable in closed form. However, for a general blade section ‘‘slice’’ defined in discrete Cartesian  $x_i y_i z_i$  coordinates, numerical integration for the corresponding discrete  $m'_i \theta_i$  is necessary. A simple trapezoidal integration is appropriate.

$$\theta_i = \arctan\left(\frac{y_i}{x_i}\right)$$

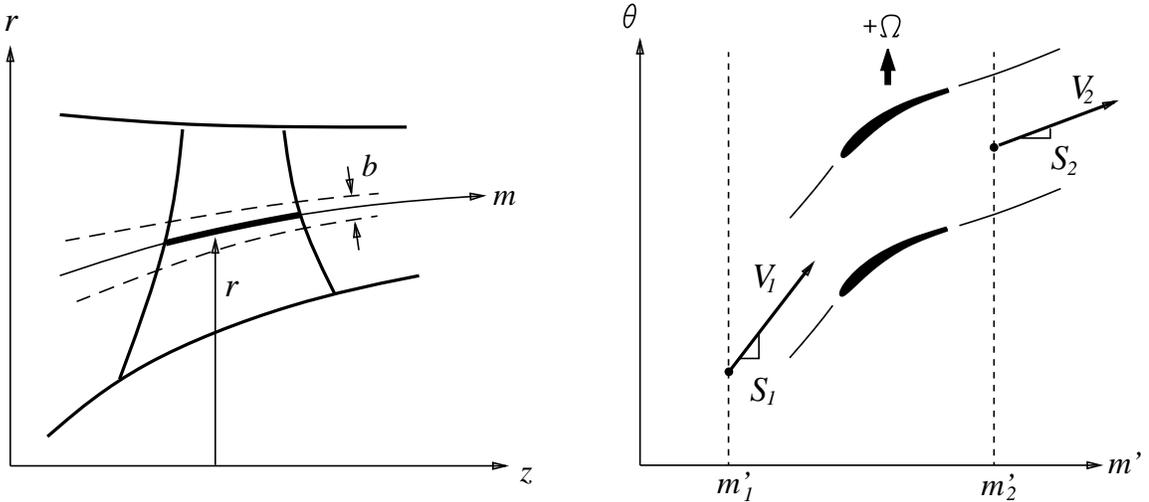


Figure 1: Streamsurface definition.

$$r_i = \sqrt{x_i^2 + y_i^2}$$

$$m'_i = m'_{i-1} + \frac{2}{r_i + r_{i-1}} \sqrt{(r_i - r_{i-1})^2 + (z_i - z_{i-1})^2}$$

If the discrete integration method is used, it may be necessary to slightly adjust  $m'_i \theta_i$  at the  $i = 1$  and  $i = N$  points to obtain the exact trailing edge gap desired. Even if these two points have identical  $x, y, z$  coordinates, as in the case of a sharp trailing edge, they will not generally have identical  $m', \theta$  coordinates due to numerical integration errors. In general, these errors will be quite small if the original  $x, y, z$  coordinates are reasonably dense. The initial  $m'_{i=1}$  coordinate is arbitrary, and merely shifts all the succeeding  $m'_i$  values. Likewise, an arbitrary constant can be added to all  $\theta$  values. These shifts are best done after the integration to position the blade “in space” where desired. Placing the leading edge near the origin is convenient.

## 4 Input Files

The principal input files needed to set up a MISES solution case are listed below. They are described in more detail in subsequent sections. The “xxx” is an arbitrary extension suffix used to designate the case being run.

- `blade.xxx`     *Required* unless `bparm.xxx` is used. Defines blade shape via a relatively large number of coordinate pairs.
- `bparm.xxx`     *Required* unless `blade.xxx` is used. Alternative way to define a blade shape via an arbitrary set of parameters, interpreted by user-supplied routines.
- `bspec.xxx`     *Optional* — used for redesign cases only. Defines a new blade shape to be imposed on an existing flow solution after modest geometry-parameter changes (this is much more efficient than starting a new solution). This file must have the same format as the `bparm.xxx` file.

`bplist.xxx` *Optional* — used in conjunction with `bspec.xxx` to specify which parameters are to be modified.

`stream.xxx` *Required* except for strictly-2D cases. Specifies streamsurface radius and thickness distributions.

`loss.xxx` *Optional*. Defines an additional loss distribution imposed on the solution.

`suct.xxx` *Optional*. Specifies one or more wall-suction locations.

`ises.xxx` *Required*. Specifies the flow conditions and solver control flags.

`modes.xxx` *Optional*. Specifies the blade geometry perturbation mode shapes.

`params.xxx` *Optional*. Specifies new design-parameter values which overrule those specified in the `ises.xxx` file.

#### 4.1 Blade coordinate file `blade.xxx`

The discrete blade airfoil geometry coordinates, however generated, are specified in the formatted file `blade.xxx`, which is used by the initialization program **ISET** to define the initial streamline grid. This file has the following structure.

```

NAME
SINL SOUT CHINL CHOUT PITCH
X(1) Y(1)          ← Blade 1
X(2) Y(2)
. .
. .
X(I) Y(I)
999.              ← Start new blade definition
X(1) Y(1)          ← Blade 2
X(2) Y(2)
X(3) Y(3)
. .
. .
X(I) Y(I)

```

**NAME** is the name of the case, not more than 32 characters.

**SINL** is the initial  $S_1 = \tan(\beta_1)$ , the tangent of the inlet flow angle relative to the axial direction. This is the default inlet flow slope at **ISET** startup, and can be changed interactively.

**SOUT** is the initial  $S_2 = \tan(\beta_2)$ . With the new panel-solver grid generator in **ISET**, **SOUT** is no longer used.

**CHINL** is the distance in  $m'$  from the blade 1 leading edge to the grid inflow plane.  
**CHOUT** is the distance in  $m'$  from the blade 1 trailing edge to the grid outflow plane.  
**PITCH** is the circumferential pitch of the cascade in radians  $= 2\pi/(\text{number of blades})$

The specified inlet flow slope **SINL** and the outlet flow slope **SOUT** calculated in **ISET** are merely the initial values for the flow slope variables  $S_1$   $S_2$ , which will usually change during the flow calculation. For a well-behaved solution process, **SINL** should be comparable to the final inlet flow slope  $S_1$  expected in the flow, although in practice there is quite a lot of room for “error” in the initial value, especially for subsonic flows. For supersonic inlet flows, **SINL** is preferably quite close to the final expected  $S_1$ , since the flow is then very sensitive to small inlet flow angle changes.

The grid inflow and outflow plane locations specified by **CHINL** and **CHOUT** are *not* the locations of the inlet and outlet condition-defining planes  $m_1$  and  $m_2$  shown in Figure 1. The latter are defined separately in the flow condition input file to be described later. It is only necessary that **CHINL** and **CHOUT** be large enough so that  $m_1$  and  $m_2$  fall inside the grid. The only drawback to making them too large is that a larger grid results, which produces correspondingly longer run times. The run time scales very nearly linearly with the number of streamwise grid points.

The blade coordinates **X(1),Y(1)** through to **X(I),Y(I)** are the  $m', \theta$  coordinates of the blade surface, starting at the trailing edge, going round the leading edge in either direction, then going back to the trailing edge. For multiple blades the first blade definition is ended with a 999.0 coordinate. The second blade definition follows the same format as the first and is assumed to be in the same coordinates matching the main blade. If necessary, it will be relocated so that it lies within the passage formed by the main blade and its periodic image at  $Y - \text{PITCH}$ . Note that this differs from previous **MISES** conventions. The new convention is necessary for the new **ISET** grid generator.

A blunt trailing edge is specified by leaving the blade “open”, so that the first and last coordinate points do not coincide. If the actual blade has the common semi-circular trailing edge, it must be “cut off” near the tangency points. The Kutta condition is applied between these two points. **ISES** incorporates a blunt trailing edge model which accounts for the additional losses associated with a blunt trailing edge. Recent investigations indicate that this model underpredicts the losses if significant vortex shedding is present, since the additional Reynolds stresses associated with the vortex motion are not represented by the standard turbulent-wake formulation. A future **MISES** version may address this deficiency.

A sharp leading edge may be specified for either or both blades by repeating the leading edge coordinates. However, sharp leading edges may lead to problems with singular velocities for subsonic or transonic cascades. This is not a problem at supersonic Mach numbers where the “unique incidence” effect aligns the flow with the leading edge.

The first two lines in **blade.xxx** can be omitted, in which case the user will be prompted to enter the missing information from the keyboard.

For 2-D cases,  $\mathbf{X}$ ,  $\mathbf{Y}$  are just the Cartesian coordinates, in units of some arbitrary reference length  $L_{\text{ref}}$  which is also used to define the Reynolds number and the rotational wheel speed, discussed below.

Inside all the programs, the blade shape is defined analytically as a parametric cubic spline of the form  $m'(s')$ ,  $\theta(s')$ , where the spline parameter  $s'$  is the arc length in the  $m'$ - $\theta$  plane.

$$s' = \int \frac{ds}{r} = \int \sqrt{dm'^2 + d\theta^2}$$

The spline parameter  $s'$  and the necessary spline derivatives  $dm'/ds'$  and  $d\theta/ds'$  are calculated directly from the input  $\mathbf{X}$ ,  $\mathbf{Y}$  coordinates, which are used as the spline knots where continuity of second derivatives is enforced. Constant-curvature end conditions (zero third derivative) are used.

## 4.2 Geometry parameter file `bparm.xxx`

MISES supports the ability to define the blade geometry via an arbitrary set of parameters,  $G_k$ ,  $k = 1 \dots K$ . Examples of  $G_k$  might be Bezier coefficients, complex-mapping coefficients, or some set of camber and thickness modes. These parameters are specified in the file `bparm.xxx`, which replaces the raw-coordinate `blade.xxx` file described above.

The following routines, located in the `src/geo/` directory, are programmed by the user to read, write, and process the information in the `bparm.xxx` file:

Routine	source file	Purpose
BPREAD	<code>bpario.f</code>	Reads $G_k$ from file <code>bparm.xxx</code>
BPWRIT	<code>bpario.f</code>	Writes $G_k$ to file <code>bparm.xxx</code>
BLDGEN	<code>bldgen.f</code>	Generates $m', \theta$ points, and optionally $\partial m'/\partial G_k$ , $\partial \theta/\partial G_k$ , from $G_k$
BPCON	<code>bpcon.f</code>	Sets constraint residuals $R(G_k)$ and also $\partial R/\partial G_k$ .

The geometry sensitivities  $\partial m'/\partial G_k$ ,  $\partial \theta/\partial G_k$  allow Parametric-Inverse calculations to be performed, with **ISES** determining the combination of  $G_k$  which best matches a specified pressure distribution in a least-squares sense. This is a generalization of the Modal-Inverse method employed by all previous MISES versions. If Parametric-Inverse calculations are not required, then all  $\partial(\cdot)/\partial G_k$  values can be returned as zero. The residuals  $R(G_k)$  embody constraints which can be imposed on the geometry in Parametric-Inverse calculations. Neither  $R(G_k)$  nor  $\partial R/\partial G_k$  are needed if the constraints are not used.

As can be seen from the call lists of `BPREAD` and `BPWRIT`, these subroutines must return and accept the case name and grid parameters — the same information which is specified in the first two lines of the `blade.xxx` file. Hence, it is natural for the `bparm.xxx` file to have the form shown below, although any format is admissible as long as it is understood by `BPREAD` and `BPWRIT`.

NAME

SINL SOUT CHINL CHOUT PITCH

< geometry parameters >

The geometry parameters read and returned by BPREAD will be passed to BLDGEN for conversion to  $m', \theta$  coordinates. Subsequent processing is then essentially the same as though these  $m', \theta$  coordinates were specified via `blade.xxx`.

BLDGEN itself can of course call other private routines as needed; its internal operations are of no consequence to MISES. For normal analysis calculations, BLDGEN will be called only once by **ISET**. For Parametric-Inverse calculations, it will also be called repeatedly by **ISES** to regenerate the blade geometry if the  $G_k$  parameters have been modified during a Newton iteration.

The BPCON routine defines user-supplied constraints on the geometry parameters. Examples might be constraints on the blade cross-sectional area, bending inertia, trailing-edge angle, etc. These can be imposed in Parametric-Inverse calculations (described later).

### 4.3 Modified-geometry parameter file `bspec.xxx`

This has exactly the same format as file `bparm.xxx`, and contains modified parameter values which are to be imposed on a solution in redesign cases.

### 4.4 Geometry parameter specification file `bplist.xxx`

This file lists the parameters which are to be modified, either by driving them to the modified values in `bspec.xxx`, or by performing a Parametric-Inverse calculation. It has the following format.

```
1    ! Parameter_1_name
-3   ! Parameter_3_name
4    ! Parameter_4_name
8    ! Parameter_8_name
.
.
K    ! Parameter_K_name
```

Only parameters whose indices appear in this list are treated as global variables to be modified. All other parameters are held fixed at their current values. The parameter indices correspond to the  $G(\mathbf{k})$  array indices “ $\mathbf{k}$ ” in the SUBROUTINE BPREAD, BPWRIT, BLDGEN, BPCON call lists. A negative index is ignored, so that parameter 3 in the example above is effectively absent. Also, all input after the “!” is currently ignored, and the parameter-name strings appear only for the user’s convenience. Future MISES versions will likely read the parameter names for use in interactive menus.

## 4.5 Stream surface file `stream.xxx`

This is an optional formatted file which specifies the radius and thickness  $r(m')$  and  $b(m')$  of the stream surface on which the blade-to-blade flow is calculated (see Figure 1). It is used by the initialization program **ISET** to define the flow domain. The actual streamtube thickness used in MISES is defined as

$$b(m') = b_0(m') + B_1 b_1(m') + B_2 b_2(m')$$

where  $b_0$  is the baseline thickness distribution, and  $b_1$   $b_2$  are optional modification modes controlled by the mode amplitudes  $B_1$   $B_2$  (Fortran names: `BVRN(1)` `BVRN(2)`). The purpose of the two modification terms is to more easily allow adjustment of the streamtube thickness to account for effects such as endwall losses and/or cooling mass flow addition. The mode amplitudes  $B_1$   $B_2$  can be set *implicitly* by the flow solver to attain a specified mass flow or outlet pressure, for example. This will be described later. Note: MISES 2.5 is actually coded for an arbitrary number of  $b_k(m')$  modes, although the current array dimension is set as `PARAMETER (NBVRX = 2)` in the `src/INC/STATE.INC` file.

The `stream.xxx` file has the structure

```

ROTREL
X(1) R(1) B_0(1) [ B_1(1) B_2(1) ] ← optional
X(2) R(2) B_0(2) [ B_1(2) B_2(2) ]
.      .      .      .      .
.      .      .      .      .
X(I) R(I) B_0(I) [ B_1(I) B_2(I) ]

```

which specifies the  $m'$ ,  $r$ ,  $b_0$ ,  $b_1$ ,  $b_2$  values. The last two columns can be omitted.

`ROTREL` is the non-dimensionalized wheel speed

$$\text{ROTREL} = \Omega L_{\text{ref}}/a_{o1}$$

with  $\Omega$  positive for the blade row moving “up” in the positive  $\theta = Y$  direction as shown in Figure 1. The normalizing quantities are the previously-described reference length  $L_{\text{ref}}$  and the relative-frame inlet stagnation speed of sound  $a_{o1}$ .

The `X` values ( $m'$  coordinates) are in the same set of axes as those used to define the blade airfoil geometry `X` values in `blade.xxx`. The stream surface radii `R` are in units of the same reference length  $L_{\text{ref}}$  used to define the wheel speed `ROTREL`:

$$R = \frac{r}{L_{\text{ref}}}$$

The streamtube thickness modes `B_0`, `B_1`, `B_2`, can have any length units (only the percentage-wise changes in  $b$  are significant). If `B_1` and/or `B_2` are omitted or all zeros, then default  $b_1$  and/or  $b_2$  distributions will be calculated using SUBROUTINE `BHDEF` (in `src/ises/rbcalc.f`).

The current default distributions are

$$b_1 = \frac{1}{2} [1 - \cos(\pi t)] \quad ; \quad t = \frac{m' - m'_1}{m'_2 - m'_1}$$

$$b_2 = \frac{1}{2} [1 - \cos(\pi t)] \quad ; \quad t = \frac{m' - \frac{1}{2}(m'_1 + m'_2)}{m'_2 - \frac{1}{2}(m'_1 + m'_2)}$$

where  $m'_1$  and  $m'_2$  are the inlet and outlet condition specification locations as described below. If  $t$  falls outside the range  $0 \leq t \leq 1$ , it is reset to the endpoint value, so that  $b_1$  and  $b_2$  are conveniently either zero or unity outside this range, although in practice the mode shapes are quite arbitrary. Care must be taken in defining and using these modes so that the total  $b(m')$  does not become negative anywhere in the grid domain!

The stream surface **R** and **B** definitions are splined in **X** to generate intermediate values and derivatives. A repeated coordinate pair can be used to specify a slope discontinuity in the definitions, although this is not realistic and not generally advised. To avoid extrapolating past the spline endpoints, the stream surface should be defined so that the domain inflow boundaries (typically **X1e-CHINL** to **Xte+CHOUT**) fall well inside the spline **X** limits. The splined  $r(m')$  and  $b(m')$  distributions can be plotted in **IPLLOT** after the initial grid is generated. Any  $b_1(m')$  and/or  $b_2(m')$  mode contributions are shown as well.

#### 4.6 Prescribed-loss file `loss.xxx`

This is an optional formatted file which specifies the total pressure losses added to the flowfield conservation equations. The resulting streamwise momentum equation has the form

$$\Delta(\ln p_{oa}) = \Delta\mathcal{P}$$

where  $\Delta(\ )$  implies a change over a cell along a streamtube. With a constant rothalpy being assumed, the righthand side forcing term is in effect a prescribed entropy change

$$\Delta\mathcal{P} = \Delta(\ln p_{oa})_{\text{prescribed}} = -\frac{\Delta S}{R}$$

which is intended as a means of modeling endwall losses, for example. Naturally,  $\mathcal{P}$  should monotonically decrease for any physical loss-generating process.

The prescribed distribution of  $\mathcal{P}(m')$  is given in the optional file `loss.xxx`, which has the following structure.

```
X(1) P(1)
X(2) P(2)
. . .
. . .
X(I) P(I)
```

The **X** values are  $m'$  coordinates in the same set of axes as those used to define the blade airfoil geometry **X** values in `blade.xxx`. The **P** values are  $\mathcal{P}$  as defined above. Because only changes in  $\mathcal{P}$  are used in the solution, it can have an arbitrary additive constant.

As with the `stream.xxx` file, the `X` value range must contain the entire grid  $m'$  range. The prescribed loss at any  $m'$  location is obtained from the spline representation  $\mathcal{P}(m')$  just like the other streamsurface quantities  $r(m')$  and  $b(m')$ .

#### 4.7 Wall-suction specification file `suct.xxx`

The file defines the location and strength of one or more suction slots on the blade surface(s).

```
CQsuct(1) Sbeg(1) Send(1) Iside(1)
CQsuct(2) Sbeg(2) Send(2) Iside(2)
.
.
```

Each line specifies the parameters for one suction slot:

**CQsuct** = suction mass-flow coefficient  $C_Q$   
**Sbeg** = fractional arc length location  $s'_{\text{beg}}/s'_{\text{side}}$  of front of slot  
**Send** = fractional arc length location  $s'_{\text{end}}/s'_{\text{side}}$  of rear of slot  
**Iside** = blade side containing suction slot

The suction coefficient is the ratio of the suction to total mass flows. These are defined as

$$C_Q = \frac{\dot{m}_{\text{suct}}}{\dot{m}} \quad \dot{m}_{\text{suct}} = \int_{s'_{\text{beg}}}^{s'_{\text{end}}} -\rho_w v_w b r ds' \quad \dot{m} = \rho_1 u_1 b_1 r_1 P$$

where  $P$  is the pitch in radians, and  $()_1$  is a reference inlet quantity. The distribution of the mass flux  $\rho_w v_w$  over the extent of the slot is assumed to be parabolic in  $s'$ , with a maximum in the middle of the slot. A uniform distribution is also implemented, but is currently commented out in the defining SUBROUTINE SETSUCTION (in `isesinit.f`).

The slot extends over the fractional arc length region  $\text{Sbeg} < s'/s'_{\text{side}} < \text{Send}$ , where  $s'_{\text{side}}$  is the distance in the  $m'$ - $\theta$  plane from the stagnation point to the trailing edge. Note that this will make the slot move slightly along the surface as the stagnation point moves. This was done for implementation simplicity, since the slot is then “fixed” on the sliding grid. Future versions may allow specification of the slot at a fixed physical location.

The slot is located on blade side **Iside**. The sides are numbered as follows.

Side 1: Blade 1 top  
Side 2: Blade 1 bottom  
Side 3: Blade 2 top  
Side 4: Blade 2 bottom

## 4.8 Flow condition file `ises.xxx`

The file defines the flow conditions and boundary conditions used by the solver program **ISES**. It also configures the program into its analysis and design modes by specifying appropriate global variables and constraints.

```
GVAR(1) GVAR(2) ... GVAR(N)
GCON(1) GCON(2) ... GCON(N)
MINLin P1PTin SINLin XINL [ V1ATin ] <-- optional
MOUTin P2PTin SOUTin XOUT [ V2ATin ] <-- optional
MFRin HWRATin [ XSHKin MSHKin ] <-- optional
REYNin NCRIT
TRANS1 TRANS2 (TRANS1 TRANS2 for blade 2) ...
ISMOM MCRIT MUCON
BVR1in BVR2in <-- optional (see below)
MOVX MOVY SCAL ROTA (MOVX MOVY ... for blade 2)... <-- optional (see below)
KMOD GMODin <-- optional (see below)
KMOD GMODin <-- optional (see below)
KMOD GMODin <-- optional (see below)
.
.
```

### Global variables and constraints. Lines 1,2

The list of integers `GVAR(1) ... GVAR(N)`, given in any order, specifies the global variables to be treated as formal unknowns in the overall Newton equation system. If any variable is to take on a new value, it must be specified in the list — otherwise it will retain its current value in `idat.xxx`. The global variables selected also indicate to **ISES** which mode it should operate in (direct, inverse, etc).

The list of possible global variables is,

```
1 SINL inlet flow slope (S1)
2 SLEX grid exit slope
5 SBLE LE stagnation point (for each non-sharp LE blade)
6 PREX grid exit static pressure

7 BVR1 streamtube thickness mode 1 DOF
8 BVR2 streamtube thickness mode 2 DOF

10 REYN stagnation Reynolds number

11 PDX0 zeroth mixed inverse prescribed Pi DOF
12 PDX1 first mixed inverse prescribed Pi DOF
13 PDD0 second mixed inverse prescribed Pi DOF
14 PDD1 third mixed inverse prescribed Pi DOF
```

15 MINL inlet Mach number  
 16 MAS1 differential mass fraction DOF  
  
 20 GMODn modal geometry DOF flag n = 1, 2, ... NGMOD  
  
 31 MOVX x-movement DOF for each blade  
 32 MOVY y-movement DOF for each blade  
 33 SCAL scaling DOF for each blade  
 34 ROTA rotation DOF for each blade (in degrees, CCW)  
  
 40 GPARK geometry parameter DOF flag k = 1, 2, ... NGPAR

The list of integers GCON(1) ... GCON(N) , in any order, specifies the global constraints to be used to close the Newton equation system. In effect, these constrain the specified global variables.

The list of possible global constraints is,

1 Drive inlet slope S1 to SINLin  
 2 Drive outlet slope S2 to SOUTin  
  
 3 Set LE Kutta condition (for all non-sharp LE blades)  
 4 Set TE Kutta condition (for all blades)  
 5 Drive over/under splitter mass fraction ratio to MFRin  
 6 Drive inlet POa to PSTr0 ( = 1/gamma )  
  
 7 Drive streamtube thickness mode 1 amplitude to BVR1IN  
 8 Drive streamtube thickness mode 2 amplitude to BVR2IN  
  
 9 Drive inlet v1/ao1 to V1ATin  
 10 Drive outlet v2/ao1 to V2ATin  
  
 11 Fix left endpoint of freewall segment  
 12 Fix right endpoint of freewall segment  
 13 Fix dP/ds2 at left endpoint of freewall segment  
 14 Fix dP/ds2 at right endpoint of freewall segment  
  
 15 Drive inlet Mach M1 to MINLin  
 16 Drive inlet pressure P1/Po1 to P1PTin  
 17 Drive outlet Mach M2 to MOUTin  
 18 Drive outlet pressure P2/Po1 to P2PTin  
  
 19 Drive inlet Reynolds number to REYNIN  
 20 Drive GMODn to GMODnin n = 1, 2, ... NGMOD  
  
 21 Set Xshock from XSHK to XSHKIN  
 31 Drive MOVX to MOVXin  
 32 Drive MOVY to MOVYin  
 33 Drive SCAL to SCALin

34 Drive ROTA to ROTain  
40 Drive GPARk to GPARkin k = 1, 2, ... NGPAR  
41 Set Geometry-Parameter Constraint 1  
42 Set Geometry-Parameter Constraint 2  
.  
.

**Inlet, outlet conditions. Lines 3,4**

MINLin = inlet relative Mach number  $\bar{M}_1$   
P1PTin = inlet static/inlet-total pressure ratio  $\bar{p}_1/p_{o1}$   
SINLin = inlet relative flow slope  $\bar{S}_1 = \tan(\beta_1) = \bar{v}_1/\bar{u}_1$   
XINL = inlet-condition location  $m'_1$   
V1ATin = inlet relative tangential velocity ratio  $\bar{v}_1/a_{o1}$   
  
MOUTin = outlet relative Mach number  $\bar{M}_2$   
P2PTin = outlet static/inlet-total pressure ratio  $\bar{p}_2/p_{o1}$   
SOUTin = outlet relative flow slope  $\bar{S}_2 = \tan(\beta_2) = \bar{v}_2/\bar{u}_2$   
XOUT = outlet-condition location  $m'_2$   
V1ATin = outlet relative tangential velocity ratio  $\bar{v}_2/a_{o1}$

The V1ATin or V2ATin values can be omitted if corresponding constraints (9) or (10) are not specified. This allows the use of MISES 2.4 file format which did not allow specification of  $\bar{v}_1/a_{o1}$  or  $\bar{v}_2/a_{o1}$ . It is useful to note that the inlet Mach number, flow slope, and tangential velocity are related by

$$\frac{\bar{v}_1}{a_{o1}} = \frac{M_1}{\sqrt{1 + \frac{\gamma-1}{2}M_1^2}} \frac{S_1}{\sqrt{1 + S_1^2}}$$

so that setting any two fixes the third.

The inlet and outlet conditions above are in terms of hypothetical mixed-out states at the specified locations  $m'_1$  and  $m'_2$  (XINL, XOUT) shown in Figure 1. The mixed-out state  $\bar{\rho}$ ,  $\bar{u}$ ,  $\bar{v}$ ,  $\bar{p}$  is defined to have the same total mass flow,  $m'$ -momentum,  $\theta$ -momentum, and total enthalpy as the actual flow at that same  $m'$  station. At the inlet station  $m'_1$ , the flow is assumed to be isentropic, so that the “mixed-out” state is obtained analytically from the known stagnation conditions, mass flux, and angular momentum. At the outlet station  $m'_2$ , the mixed-out state is defined implicitly by

$$\begin{aligned} \bar{\rho}\bar{u} Pbr &= \int \rho u br d\theta &= \int d\dot{m} &- \dot{m}_{\text{suct}} \\ (\bar{\rho}\bar{u}^2 + \bar{p}) Pbr &= \int (\rho u^2 + p) br d\theta &= \int u d\dot{m} + \int p br d\theta &- u_e \dot{m}_{\text{suct}} - \rho_e V_e u_e \Theta b \\ \bar{\rho}\bar{u}\bar{v} Pbr &= \int \rho uv br d\theta &= \int v d\dot{m} &- v_e \dot{m}_{\text{suct}} - \rho_e V_e v_e \Theta b \\ \bar{\rho}\bar{u}\bar{h}_o Pbr &= \int \rho u h_o br d\theta &= \int h_o d\dot{m} &- h_{oe} \dot{m}_{\text{suct}} - \rho_e V_e h_{oe} \delta_h b \end{aligned}$$

where  $P$  is the angular pitch,  $u, v$  are the  $m', \theta$  velocity components, and  $V = \sqrt{u^2 + v^2}$  is the speed. The integrals  $\int(\ )d\bar{m}$  on the righthand side are summed over all the inviscid streamtubes at the  $m'_2$  station (in SUBROUTINE MIXOUT), which then requires including the additional terms involving the momentum thickness  $\Theta$  and total-enthalpy thickness  $\delta_h$ . These inviscid streamtubes include the “removed” suction flow (this is discussed later), which then requires subtracting these fictitious contributions explicitly via the  $\dot{m}_{\text{suct}}$  terms.

Downstream of the blade row, the momentum and displacement thicknesses are extracted directly from the solution at the sampled location. The total enthalpy thickness is not directly available, since the thermal energy equation is not solved — the Reynolds analogy is used locally at every surface location instead. Specifically, the surface heat flux  $q_w$  into the wall is determined from the relation

$$\begin{aligned} q_w &= \rho_e u_e (h_{aw} - h_w) C_h \\ h_{aw} &= h_e + f_r \frac{1}{2} u_e^2 = I + \frac{1}{2} (\Omega r)^2 + (f_r - 1) \frac{1}{2} u_e^2 \\ C_h &= 0.22 Re_\theta^{-1} Pr^{-2/3} \end{aligned}$$

where  $h_w$  is the specified wall enthalpy,  $h_{aw}$  is the usual adiabatic-wall enthalpy, and  $f_r$  is the temperature recovery factor. The correlation for the Stanton number  $C_h$  is strictly correct only for zero pressure gradients, although comparisons with finite-difference BL calculations indicate that it is rarely more than 20% in error even for severe pressure gradient cases like high-work turbines.

Since the evolution of the total enthalpy thickness  $\delta_h$  is not tracked, its final value downstream of the blade row must therefore be obtained from the thermal energy equation via the integrated heat flux over the blade surface.

$$\begin{aligned} \rho_e V_e h_{oe} \delta_h &\equiv \int (h_{oe} - h_o) \rho u r d\theta \\ \frac{d}{ds'} (\rho_e V_e h_{oe} \delta_h b) &= q_w b r \\ (\rho_e V_e h_{oe} \delta_h b)_{\text{exit}} &= \int q_w b r ds' \equiv \dot{H}_w \end{aligned}$$

Note that  $\delta_h$  is zero for an adiabatic-wall blade.

Using the state equation, the mixed-out total enthalpy is expressed as

$$\bar{h}_o = \frac{\gamma}{\gamma-1} \bar{p} / \bar{\rho} + \frac{1}{2} \bar{V}^2$$

and the four relations above then form a closed system for the mixed-out state. Note that the mixed-out state is assumed to have the same radius  $r$  and streamtube thickness  $b$  as the station where the integration is performed.

Choosing the defining station to be  $m'_2$  determines the mixed-out flow quantities  $\bar{\rho}_2, \bar{u}_2$ , etc, which are then used to impose global constraints on the outlet flow angles, Mach numbers and/or pressure, and also to compute the mixed-out losses at the exit:

$$\bar{S}_2 = \frac{\bar{v}_2}{\bar{u}_2} \quad \bar{M}_2^2 = \frac{\bar{\rho}_2}{\gamma \bar{p}_2} (\bar{u}_2^2 + \bar{v}_2^2) \quad \bar{p}_{o2} = \bar{p}_2 \left( 1 + \frac{\gamma-1}{2} \bar{M}_2^2 \right)^{\frac{\gamma}{\gamma-1}}$$

It is reassuring to note that for a viscous calculation, the mixed-out exit state and the loss in particular (discussed later) are essentially independent of where it is calculated downstream of the trailing edge. This verifies that the quasi-3D coupled boundary layer formulation is mass-, momentum-, and energy-conserving.

The inlet Mach number constraint (15) using `MINLin` and inlet pressure constraint (16) using `P1PTin` are essentially equivalent, since

$$\bar{p}_1/p_{o1} = \left(1 + \frac{\gamma-1}{2} \bar{M}_1^2\right)^{\frac{-\gamma}{\gamma-1}}$$

Both are provided only for convenience.

Normally, the input parameters `MINLin`, `P1PTin`, `MOUtin`, `P2PTin`, are used only if the corresponding constraints (15),(16),(17),(18), are selected, with the inlet Mach number `MINL` (15) being the appropriate global variable. If *none* of these constraints are selected, then `MINLin` and `SINLin` will be used to simply set the inlet Mach and the mass flow, which is then held fixed in the calculation. Because of this, it is normally not necessary to specify the inlet Mach constraint (15) for subsonic flows, since it will be satisfied anyway if constraints (15)... (18) are all omitted.

It is important to note that choked cases with specified subsonic inlet Mach number or inlet pressure (i.e. a specified mass flow) are ill-posed — physical considerations require that one of the outlet constraints (17),(18) be used instead. Computationally, this does not suffice, however. Normally, the **ISES** Newton procedure adjusts the inlet Mach variable `MINL` to meet any specified conditions, but the inlet Mach and (and all other global variables) are treated as being temporarily fixed when the Newton matrix for the flowfield variables is set up for each iteration. Each such iteration is therefore ill-posed, and **ISES** will complain either with enormous residuals and/or an arithmetic fault due to a nearly-singular Newton matrix. A similar problem occurs in axially-supersonic flows, where the inlet Mach number cannot be influenced by iterating on the downstream conditions.

To allow calculation of choked and/or axially-supersonic cases, it is necessary to select the grid-exit pressure `PREX` (6) as a global variable. The corresponding global constraint is the grid-inlet stagnation pressure constraint (6). This combination enables a computational “trick” by which **ISES** can alter the inlet Mach number *simultaneously* with the flowfield variables, by temporarily allowing variation of the inlet stagnation pressure. This stagnation pressure is then driven back to its correct value by constraint (6). The overall procedure is merely the equivalent of performing partial pivoting in the overall Newton matrix solution process. This produces a different iteration history (without arithmetic faults in particular!), but there is no effect on the final flow solution. The (6),(6) combination can of course be specified for unchoked subsonic flows, but it does produce a  $\sim 10\%$  CPU penalty and therefore should be omitted if no choking is expected.

### Over/Under splitter mass flow ratio, wall temperature ratio. Line 5

`MFRin` = Specified mass flow ratio above/below splitter blade

**HWRATin** = Specified  $h_{\text{wall}}/h_{oa}$   
 = 0.0 for adiabatic cases  
**XSHKin** = Specified shock  $x$  location for constraint (21) (optional)  
**MSHKin** = Specified shock Mach number for constraint (21) (optional)

**MFRin** is used by constraint (5), typically in conjunction with variable **MOVY** (32), to position the splitter blade 2 in the passage.

The wall temperature **HWRATin** is specified as a ratio relative to the rothalpy,  $h_w/h_{oa}$ . If this ratio is specified as zero, the blade surface is taken to be adiabatic, as was assumed in MISES 2.4 and all earlier versions.

**XSHKin** and **MSHKin** are optional, and needed only if constraint (21) is chosen. This can be used to find the exit pressure or inlet Mach which puts the shock at a specified location. Typically, **MSHKin** = 1.0 would be specified.

### Viscous flow parameters. Lines 6,7

**REYNin** = Reynolds number  
 = 0.0 → inviscid calculation  
 (restarting a viscous case with **REYNin** = 0 “freezes” the boundary layers)  
**NCRIT** = (+) critical amplification factor “ $n_{\text{crit}}$ ” for  $e^n$  transition model  
 = (−) freestream turbulence level ( $\tau = -\text{NCRIT}$ , in %) for modified  
 Abu-Ghannam–Shaw bypass transition model  
**TRANS1** = side 1 surface transition trip  $m'$ /chord location  
**TRANS2** = side 2 surface transition trip  $m'$ /chord location

The input Reynolds number **REYNin** is based on the mixed-out static density, viscosity, and relative speed at  $m'_1$ , and the reference length  $L_{\text{ref}}$ .

$$\text{REYNin} = \frac{\bar{\rho}_1 \bar{V}_1 L_{\text{ref}}}{\bar{\mu}_1}$$

The reference length  $L_{\text{ref}}$  is the same as was used used to define the streamsurface radii **R** in the **stream.xxx** file described earlier. If a constant **R**=1 is specified (the default case for 2-D cascades), then  $L_{\text{ref}}$  becomes the length unit of the blade coordinates (**X**,**Y**). If (**X**,**Y**) are also defined so that the blade chord is unity, **REYNin** is then the usual chord-based Reynolds number.

The Reynolds number is used in conjunction with Sutherland’s formula to set the local viscosity as a function of the local temperature. This requires another parameter, namely Sutherland’s constant ( $T_S = 110 \text{ K}^\circ$ ). This is stored internally in variable **TSRAT** as a temperature ratio:

$$\text{TSRAT} = \frac{T_S}{T_{o1}}$$

where  $T_{o1}$  is the relative-frame total temperature at  $m'_1$ . Currently, **TSRAT** is hard-wired to 0.35 in **src/iset/iset.f**, although it can be read as an input parameter via the **ises.xxx**

input file if desired. The ratio of specific heats  $c_p/c_v = \gamma$  (GAM) can likewise be read in as an input parameter. The read statement for both parameters is currently commented out in SUBROUTINE ISESINIT (in `src/ises/isesinit.f`).

MISES incorporates a modified version of the Abu-Ghannam–Shaw (AGS) bypass transition model (see separate document). Both the  $e^n$  and the AGS model are active all the time and either one may be decisive in inducing transition. Their respective input parameters  $n_{\text{crit}}$  and  $\tau$  (Fortran names: ACRIT, FTURB) are always related through a modified Mack’s correlation, and can be input either way. If a positive NCRIT is input, then this is taken as  $n_{\text{crit}}$ , and the % turbulence level  $\tau$  for the AGS model is calculated from the modified Mack’s correlation:

$$\begin{aligned} n_{\text{crit}} &= \text{NCRIT} \\ \tau' &= 100 \exp\left(-\frac{8.43 + n_{\text{crit}}}{2.4}\right) \\ \tau &= \frac{2.7}{2} \ln\left(\frac{1 + \tau'/2.7}{1 - \tau'/2.7}\right) \end{aligned}$$

If a negative NCRIT is input, then this is taken as the % turbulence level, and  $n_{\text{crit}}$  is calculated instead.

$$\begin{aligned} \tau &= -\text{NCRIT} \\ \tau' &= 2.7 \tanh(\tau/2.7) \\ n_{\text{crit}} &= -8.43 - 2.4 \ln\left(\frac{\tau'}{100}\right) \end{aligned}$$

The Mack modification function  $\tau'(\tau)$  prevents negative  $n_{\text{crit}}$  values for large  $\tau$  values, and is deemed reasonable given that Mack’s original correlation was developed for small  $\tau$  levels.

The transition trip locations TRANS are defined in terms of the fractional  $m'$  position on the blade (which is *not* necessarily the fractional arc length position if  $r$  varies along the blade). If an additional blade is present, simply add another TRANS3, TRANS4 pair on the same line. Setting TRANSx  $\geq 1.0$  implies there is no transition strip on that side. Note that the  $e^n$  and AGS criteria are always active, and free transition by either criterion may occur upstream of TRANSx.

### Isentropy and dissipation. Line 8.

- ISMOM = 1 → use S-momentum equation
  - 2 → use isentropic condition
  - 3 → use S-momentum equation, replaced by an isentropic condition  
only near the leading edge to minimize truncation errors there.
  - 4 → use isentropic condition, replaced S-momentum equation only at shocks  
where dissipation is active
- MCRIT = critical Mach number in the definition of bulk viscosity
- = 0.98 usually for weak shocks
  - = 0.85 for exceptionally strong shocks

MUCON = artificial dissipation coefficient (= 1.0 normally)

A negative MUCON value disables 2nd-order dissipation.

The value of MUCON may need to be increased to higher values (up to 1.5 or so) for strong shocks and/or highly sheared grids. It may be appropriate to at the same time reduce MCRIT. Second order dissipation is not recommended when when strong shocks traverse quasi-normal grid lines — e.g. normal shocks on sheared grids, or strong oblique shocks on orthogonal grids. Note that the sign convention on MUCON is reversed from previous MISES versions. This is to make it consistent with the multielement **MSES** code convention from which the new dissipation triggers were taken.

### **Streamtube thickness mode amplitudes. Line 9.**

BVR1in = Specified streamtube thickness mode 1 amplitude BVR1

BVR2in = Specified streamtube thickness mode 2 amplitude BVR2

### **Geometry movement,scaling,rotation mode amplitudes. Line 10.**

MOVXin = Specified x-displacement mode MOVX

MOVYin = Specified y-displacement mode MOVY

SCALin = Specified scaling mode SCAL

ROTAin = Specified rotation mode ROTA

### **Geometry shape mode amplitudes. Lines 11...end**

KMOD = Specifies geometric design mode

GMOD = Specified magnitude of geometric design mode MOD(KMOD)

#### **4.8.1 Variable,Constraint indices**

The use of global variables and constraints gives the user a very flexible means to apply boundary conditions that specify the flow condition or design conditions. In general, any number of global variables can be specified, as long as the same number of global constraints are also specified (except if a Modal- or Parametric-Inverse calculation is to be performed, as explained later). It is only necessary that all the constraints properly constrain all the variables, and that the flow does not admit any non-physical situations. For example, the grid-exit slope variable DSLEX (variable 2) can be constrained either by specifying it directly (constraint 2), or by specifying the trailing edge Kutta condition (constraint 4) instead. Not specifying the Kutta condition is not physical, however, and may produce strong pressure spikes at the trailing edge.

Note that for a multiple-blade case some of the variable and constraint options add a global variable and a global constraint for each blade. For example, the specification of trailing edge

Kutta condition (constraint 4) adds two constraints to the system — one for each blade trailing edge. Specific examples of global variables and constraints are provided in a following section.

The GMODn (20) flag indicates that some number of geometry design mode amplitudes are selected, with their amplitudes specified in lines 11–EOF. The mode shapes associated with each of these amplitude variables are defined in the `modes.xxx` file, described later.

### 4.8.2 Pressure-correction term

The pressure-correction term  $P_{\text{corr}}$  (described in H. Youngren’s thesis) effectively adds tension to all the streamlines, and thus suppresses sawtooth modes in the grid. It is really only required in inverse cases and in viscous cases with boundary layer separation, since the sawtooth modes are adequately constrained by solid-wall and  $\delta^*$ -offset boundary conditions. It is also helpful in cases with strong shocks traversing a sheared grid which has large streamtube widths compared to cell lengths. Since this term is not “smoothing” in the normal sense, and is not dissipative, it is simplest to use it whether it is necessary or not.

Since MISES v 1.4, the  $P_{\text{corr}}$  term has been reformulated slightly from H. Youngren’s form. The dependence on the local streamtube area has also been eliminated, making the term have more or less equal influence throughout the flowfield. Previously, it was often too strong in the thin streamtubes adjacent to the blade, and too weak in the larger interior streamtubes. The result of the reformulation is that a larger  $P_{\text{corr}}$  weighting factor PCWT can now be safely used for difficult cases, particularly those with strong shocks traversing sheared grids. For MISES v 2.5, PCWT is hard-wired in SUBROUTINE ISESINIT for simplicity, since there is little reason to treat it as an input parameter.

### 4.8.3 Momentum/Entropy conservation

The ISMOM flag controls whether and where S-momentum (streamwise momentum) or streamline total pressure (entropy) are conserved. The streamtube-cell equation residuals for each case are:

$$\begin{aligned} \text{ISMOM}=1: \quad \mathcal{R}_1 &\equiv \Delta p + \frac{m}{A} \Delta \tilde{q} - P_s - p \Delta \mathcal{P} = 0 \\ \text{ISMOM}=2: \quad \mathcal{R}_2 &\equiv p \Delta (\ln \tilde{p}_{o_a}) - p \Delta \mathcal{P} = 0 \end{aligned}$$

where  $m$  is the streamtube’s mass flow,  $A$  is the streamtube’s cross-sectional area, and  $\mathcal{P}$  is the prescribed loss described earlier ( $\mathcal{R}_1$  is slightly modified for the case of a curved streamtube). The changes  $\Delta(\ )$  are along the streamwise direction in the cell.  $P_s$  is the streamwise centrifugal force, plus the streamwise pressure force contribution to the cell from streamtubes above and below, arranged so that the net equation is strongly conservative. Dissipation is introduced in the form of an upwinded speed  $\tilde{q}$  described in the next section. The upwinded “absolute” total pressure (described later) is defined in the standard manner, but using the upwinded speed.

$$\tilde{p}_{o_a} = p \left( \frac{I}{I - \tilde{q}^2/2 + \Omega^2 r^2/2} \right)^{\frac{\gamma}{\gamma-1}}$$

ISMOM=1 gives a standard momentum-conserving Euler solver, while ISMOM=2 gives the equivalent of a standard entropy-conserving Full-Potential solver. ISMOM=3,4 are hybrids which attempt to make the use of the best features of the two types of solvers: correct Rankine-Hugoniot shock jumps of the Euler solver, and zero total pressure loss of the Full-Potential solver.

For ISMOM=3, the S-momentum equation  $\mathcal{R}_1 = 0$  is used everywhere except in a region near the leading edge, where the isentropic equation  $\mathcal{R}_2 = 0$  is used instead. The size of the isentropic region is hard-wired into a data statement at the top of SUBROUTINE SETUP (in `src/ises/setup.f`), and typically extends from the inlet plane to  $\sim 10$  cells downstream of the stagnation point, and 4 cells below and above the stagnation streamline. This should cover most typical airfoils, but can be changed if appropriate for an unusual case. The only exception is supersonic inlet casades, in which the bow shock will traverse the isentropic region. **IPLOT** allows the display of the entropy-conserving region on the grid and flow contour plots, the remainder being the momentum-conserving region.

MISES v 2.0 is the first to incorporate a new ISMOM=4 option, which makes *all* cells isentropic except those where artificial dissipation contributes significantly to the momentum flux (described later). In practice, this is nearly the same as ISMOM=3, but it will always give momentum conservation at shocks, and hence is “safer” than ISMOM=3. It also doesn’t rely on the ad-hoc hard-wired isentropic region, and hence is more automatic. Note also that ISMOM=4 is equivalent to ISMOM=2 for subcritical cases. The only possible problem with ISMOM=4 is that because the switching between equations is flow-dependent (as opposed to being hard-wired), there is the possibility of the Newton iteration process stagnating at a limit cycle. This has been observed occasionally in supersonic fans, so ISMOM=3 might be better for these cases.

Because minimizing total pressure losses is much more important than conserving momentum (except at shocks, of course), it is strongly recommended that ISMOM=3 or 4 be used for all flows. Even very small total pressure errors in the vicinity of the leading edge (where they are most likely to occur) can cause dramatic errors in lift and drag of a viscous case at near-stall conditions. This is because for a given imposed streamwise pressure gradient  $dp/ds$ , the edge velocity gradient  $du_e/ds$  which drives the viscous layers depends on the total pressure. Enforcing isentropy (even if only at the leading edge for ISMOM=3) avoids most such problems, and hence significantly reduces grid density requirements. With ISMOM=3, it is only necessary to ensure that no shocks traverse the isentropic region, otherwise incorrect Rankine-Hugoniot jumps will result and the wave drag will not be properly predicted. ISMOM=4 does not have this possible problem, and is recommended for general use.

#### 4.8.4 Artificial dissipation

The artificial dissipation in MISES is a speed-upwinding formulation analogous to bulk viscosity. Instead of the actual speed  $q$ , the momentum and/or isentropy equations are defined using an

upwinded speed  $\tilde{q}$  defined by

$$\tilde{q}_i = q_i - \mu_i^{(1)} (q_i - q_{i-1}) + \mu_i^{(2)} (q_{i-1} - q_{i-2})$$

where  $i$  is the grid node index along a streamtube, and  $\mu_i^{(1)}$ ,  $\mu_i^{(2)}$  are the first- and second-order dissipation coefficients. To maintain numerical stability and allow shock capturing, the following formulas for the dissipation coefficients used, as suggested by a stability analysis.

$$\mu_i^{(1)} = \frac{C_\mu}{\gamma} (1 - M_{\text{crit}}) \log \left[ 1 + \exp \left( \frac{1 - 1/M^2}{1 - M_{\text{crit}}} \right) \right], \quad \text{with } M^2 = \frac{1}{2}(M_i^2 + M_{i-1}^2)$$

$$\mu_i^{(2)} = \begin{cases} \mu_i^{(1)} & ; \text{ 2nd-order dissipation} \\ 0 & ; \text{ 1st-order dissipation} \end{cases}$$

In the limit  $M_{\text{crit}} \rightarrow 1$ , the above formula asymptotes to

$$\mu \rightarrow \max \left( 0, \frac{C_\mu}{\gamma} (1 - 1/M^2) \right)$$

which is the form indicated by a formal stability analysis, and is the dotted curve in Figure 2. For  $M_{\text{crit}} < 1$ , the full formula produces somewhat larger  $\mu$  values, which asymptote rapidly to zero below  $M = M_{\text{crit}}$ . The intent of introducing  $M_{\text{crit}}$  is to provide a user-adjustable margin of safety. Figure 2 shows the variation of  $\mu$  versus  $M_{\text{crit}}$  and the overall scaling constant  $C_\mu$ . Previous MISES versions used the simpler form  $\mu \sim \max(0, 1 - M_{\text{crit}}^2/M^2)$ , which has a slope discontinuity at  $M = M_{\text{crit}}$ . The new form appears to have better shock propagating properties, most likely due to its exponential ‘‘tail’’ for subsonic Mach numbers.

The second-order term in the formula for  $\tilde{q}_i$  above is constructed to cancel the first-order term in the case of a linear variation of  $q(s)$ . Analytically, the net result of either the first-order or second-order upwinding is to add a term to the streamwise momentum equation

$$\frac{\partial p}{\partial s} + \rho q \frac{\partial \tilde{q}}{\partial s} - \rho \Omega^2 r \frac{\partial r}{\partial s} = 0 \quad \longrightarrow \quad \frac{\partial p}{\partial s} + \rho q \frac{\partial q}{\partial s} - \rho \Omega^2 r \frac{\partial r}{\partial s} = -\rho q \frac{\partial(\delta q)}{\partial s}$$

where  $\delta q = \tilde{q} - q$  is the upwinding modification to the speed. This added term is dissipative, producing streamwise changes in the absolute total pressure

$$\frac{1}{p_{oa}} \frac{\partial p_{oa}}{\partial s} = -\gamma M^2 \frac{1}{q} \frac{\partial(\delta q)}{\partial s}$$

and is the mechanism which allows captured shocks to generate total pressure loss. In the ISMOM=4 option, the magnitude of the fractional *upwinded* total pressure change over a cell implied by  $\delta q$  is monitored:

$$\Delta(\ln \tilde{p}_{oa}) \simeq \delta q \Delta \left( \frac{\rho q}{p} \right)$$

If the righthand side term is sufficiently negative, below some small tolerance  $-\epsilon_p$  (typically at a shock), then  $\mathcal{R}_1 = 0$  is used instead of  $\mathcal{R}_2 = 0$ , so that this total pressure loss is realized in the solution. The local velocity gradient is also examined, with acceleration favoring  $\mathcal{R}_2$

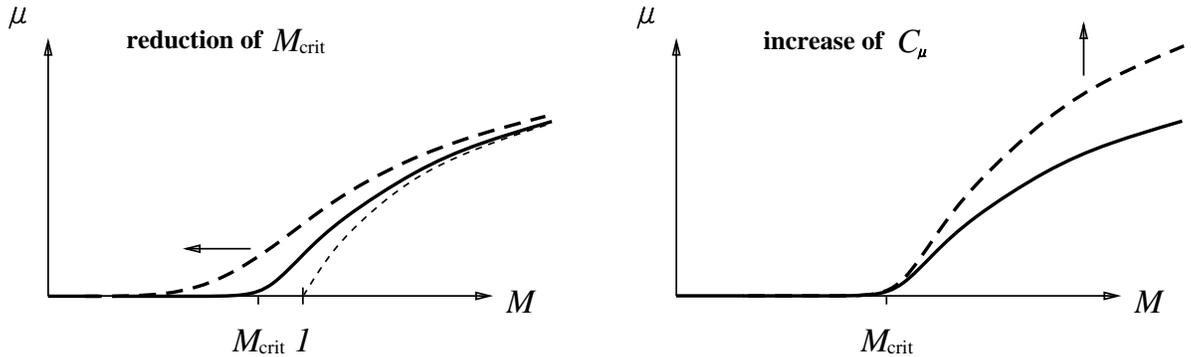


Figure 2: Effect of dissipation parameters  $M_{\text{crit}}$  and  $C_\mu$  on dissipation level.

and deceleration favoring  $\mathcal{R}_1$ . To minimize the tendency for non-convergent limit cycles, the equation switch is implemented as a continuous blend. The actual general equation solved with the ISMOM=4 option is

$$f \mathcal{R}_1 + (1 - f) \mathcal{R}_2 = 0$$

where  $0 \leq f \leq 1$  is a blending fraction which depends on the local magnitude of the total pressure loss, as well as the local speed gradient.

It should be pointed out that this seemingly ad-hoc blending of  $\mathcal{R}_1$  and  $\mathcal{R}_2$  is perfectly legitimate within the accuracy of the numerical scheme, since in smooth flow these two equations are equivalent to within  $\mathcal{O}(\Delta s^2)$ . Hence, away from shocks, any change in  $f$  will produce a solution change of at most  $\mathcal{O}(\Delta s^2)$  — the same as the truncation error of the discretization scheme.

#### 4.8.5 Artificial dissipation level selection

The magnitude of the upwinding (e.g. magnitude of  $\delta q$ ) is controlled by the approximate threshold  $M_{\text{crit}}$  (MCRIT), and the weighting factor  $C_\mu$  (MUCON). If MUCON is specified as negative, then  $C_\mu = |\text{MUCON}|$ , and  $\mu^{(2)} = 0$ .

Lowering  $M_{\text{crit}}$  and increasing  $C_\mu$  both increase the amount of upwinding, but in different ways as shown in Figure 2. The effect of  $C_\mu$  on the numerical normal-shock profile is shown in Figure 3. In general,  $C_\mu \simeq 1$  gives the cleanest normal shocks. For oblique shocks, the effective coefficient is  $C_\mu / \cos^2 \theta$ , with  $\theta$  being the shock angle. This favors somewhat smaller values of  $C_\mu$ . The stability analysis indicates that in general the minimum allowable coefficients are

$$\begin{aligned} \text{1st-order dissipation: } C_{\mu_{\text{min}}} &= 1/2 \\ \text{2nd-order dissipation: } C_{\mu_{\text{min}}} &= 1/4 \end{aligned}$$

with  $M_{\text{crit}} \leq 1$  being required in all cases. Violation of these thresholds will produce numerical instability and a nearly-singular Newton matrix.

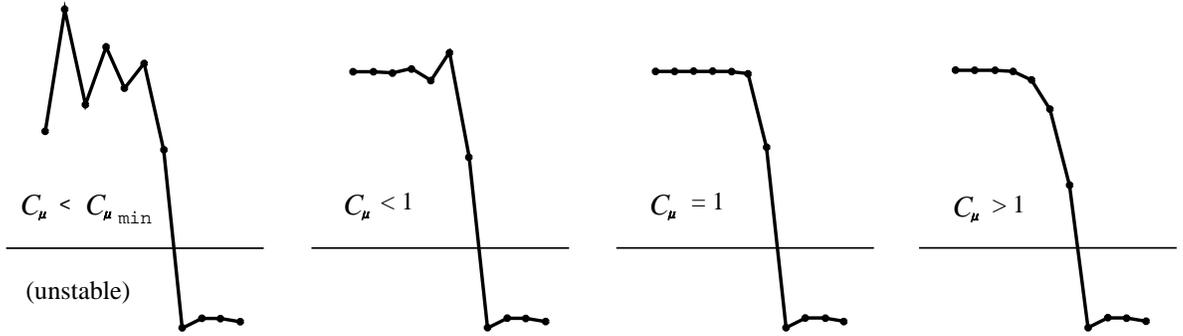


Figure 3: Effect of the dissipation weight  $C_\mu$  on the numerical structure of a captured shock. Assumes  $M_{\text{crit}} \simeq 1$ .

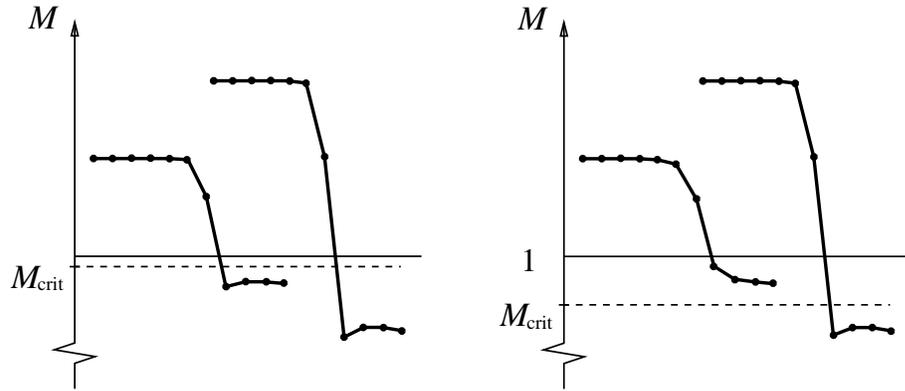


Figure 4: Effect of the dissipation threshold  $M_{\text{crit}}$  on the numerical structure of weak and strong captured shocks. Assumes  $C_\mu \simeq 1$ .

Reduction of  $M_{\text{crit}}$  mainly has an effect where locally  $M \simeq 1$ . Specifically, the sharp subsonic recovery downstream of the shock is smeared if the post-shock Mach number falls significantly above  $M_{\text{crit}}$ , as shown in Figure 4. Hence, reduction of  $M_{\text{crit}}$  tends to smear weak shocks, but has little effect on strong shocks – a rather undesirable situation. It is therefore desirable to set  $M_{\text{crit}}$  just below 1.0 to give a small margin of safety against numerical instability.

The second-order dissipation substantially reduces spurious total pressure errors wherever the S-momentum equation is used, compared to the original first-order dissipation. It also gives crisper shocks for a given value of  $C_\mu$ , gives more reliable wave drag results, and is nice in general. The benefits are greatest for oblique shocks, which tend to be quite heavily smeared with first-order dissipation.

The major possible drawback of second-order dissipation is that it has been found to induce more dispersion noise near a normal or near-normal shock which traverses a sheared grid. This can be suppressed to some extent by increasing PCWT to 2.0 or so, but only up to a point. One might also try a somewhat larger positive value of MUCON (= +1.5, say), which will still give less smearing than a modest 1st-order MUCON (= -1.0, say). In any case, the option of

disabling the second-order dissipation (which then reverts to the old first-order scheme) has been provided. The requirement for extra dissipation is obviously alleviated by using offset grids, which are much less sheared than periodic grids.

At the grid inflow plane, the upstream speeds  $q_{i-1}$  and  $q_{i-2}$  in the definition of  $\tilde{q}$  above are not available. Here, the upwinded speed is defined by

$$i = 1 : \quad \tilde{q}_i = q_i - \mu_i^{(1)}(q_i - V_{\text{inflow}})$$

where  $V_{\text{inflow}}$  is the inflow-plane speed calculated directly from the inlet Mach number  $M_1$ , correcting for any radius and streamtube thickness changes between  $m_1$  where  $M_1$  is defined and the actual grid inflow plane location. This use of the inlet Mach number is the means by which the extra incoming characteristic is set in supersonic-inlet flows.

#### 4.8.6 Dissipation enhancement during convergence

The movement of a captured shock tends to be a slow process, with the movement distance per Newton iteration being limited to the smeared shock's thickness. An ideally-resolved shock can therefore move at most one or two cells per iteration, which is deemed unacceptable. To speed up this process, **ISES** takes the liberty of temporarily reducing  $M_{\text{crit}}$  and/or increasing  $C_\mu$  to smear the shock as much as possible so that it can move as fast as possible. The reduction of  $M_{\text{crit}}$  is especially effective here, since this smears the subsonic side of the shock, allowing it to move many cells per Newton iteration. The reduction is based on the maximum fractional density change from the previous Newton step,

$$\delta\bar{\rho} = |\delta\rho/\rho|_{\text{max}}$$

which typically is large at a moving shock.

The temporary value used for  $M_{\text{crit}}$  is

$$(M_{\text{crit}})_{\text{temp}} = 0.75 + (M_{\text{crit}} - 0.75) \exp(-r^2)$$

$$\text{where } r = \frac{\delta\bar{\rho}}{\varepsilon}$$

$$\text{or } r = \frac{\delta\bar{\rho}^3}{\varepsilon(\delta\bar{\rho}^2 + (\varepsilon/4)^2)}$$

and  $\varepsilon = 0.15$  is a reasonable threshold. When  $\delta\bar{\rho}$  is large, the exponential is negligibly small, giving the decreased value of  $(M_{\text{crit}})_{\text{temp}}$ . As the shock reaches its destination,  $\delta\bar{\rho}$  decreases, and  $(M_{\text{crit}})_{\text{temp}}$  reverts to its prescribed value, which then sharpens the shock. The two forms for the argument  $r$  give nearly the same results, except very close to convergence where  $\delta\bar{\rho}$  is small. The simple form for  $r$  gives  $1 - \mathcal{O}(\delta\bar{\rho}^2)$  for the Gaussian, while the second more complicated form gives  $1 - \mathcal{O}(\delta\bar{\rho}^6)$ . Both forms give terminal quadratic convergence of the overall Newton scheme, but the second form has a larger basin of attraction and is preferred for this reason.

The ‘‘old’’ fractional density change  $\delta\bar{\rho}$  is saved in the `idat.xxx` file, so that the overall process is automated, and is not outwardly visible to the user even if the Newton iteration is halted and then restarted.

For future MISES releases, an automatic grid-sequencing procedure is planned as a replacement for the current dissipation enhancement approach. This should also reduce overall computation times for shocked flow cases, by virtue of mostly eliminating the shock-propagation bottleneck.

#### 4.9 Example ises.xxx input-file lines

Examples of selected lines of the `ises.xxx` input file are given below for a variety of situations. A special feature of the input routine is that it looks for a “!” in the first two lines containing the global variable and constraint indices, causing the subsequent numbers to be ignored. For example, the following two lines are equivalent:

```
1 2 5 ! 11 12
1 2 5
```

This is provided for convenience, since numerous indices must sometimes be repeatedly added and deleted to reconfigure **ISES** for the various types of calculation cases.

##### 4.9.1 Lines 1–4. Variables, constraints, flow conditions.

Quantities not used in the calculation are shown as zero, although `MINLin` should always be set close to the anticipated inlet Mach, since it is sometimes used to initialize the flowfield for iteration.

a) Specified subsonic inlet Mach and inlet slope (fixed mass flow), blunt leading edge.

```
1 2 5 |Global variables
1 4 3 |Global constraints
0.80 0. 1.50 -0.5 |Minl p1/po1 Sinl Xinl
0. 0. 0. 1.1 |Mout p2/po1 Sout Xout
```

b) Specified outlet pressure and inlet slope (mass flow unknown). Choking not permitted.

```
1 2 5 15 |Global variables
1 4 3 18 |Global constraints
0. 0. 1.50 -0.5 |Minl p1/po1 Sinl Xinl
0. 0.50 0. 1.1 |Mout p2/po1 Sout Xout
```

c) Same as b) above, choking permitted.

```
1 2 5 15 6 |Global variables
1 4 3 18 6 |Global constraints
0. 0. 1.50 -0.5 |Minl p1/po1 Sinl Xinl
0. 0.50 0. 1.1 |Mout p2/po1 Sout Xout
```

d) Supersonic/subsonic-axial inflow, specified inlet Mach, outlet pressure. The “unique-incidence” condition will set the inlet slope.

1 2 5 15 6	Global variables
15 4 3 18 6	Global constraints
1.30 0. 0. -0.5	Minl p1/po1 Sinl Xinl
0. 0.50 0. 1.1	Mout p2/po1 Sout Xout

e) Same as d) above, but relative tangential inlet speed ratio  $v_1/a_{o1}$  is imposed rather than the total inlet Mach number  $V_1/a_1$ .

1 2 15 6	Global variables
9 4 18 6	Global constraints
1.30 0. 0. -0.5 0.88	Minl p1/po1 Sinl Xinl v1/ao1
0. 0.50 0. 1.1	Mout p2/po1 Sout Xout

f) Same as d) above, but with sharp leading edge.

1 2 15 6	Global variables
15 4 18 6	Global constraints
1.30 0. 0. -0.5	Minl p1/po1 Sinl Xinl
0. 0.50 0. 1.1	Mout p2/po1 Sout Xout

g) Mixed-Inverse design for case a) above.

1 2 5 11 12	Global variables
1 4 3 11 12	Global constraints
0.80 0. 1.50 -0.5	Minl p1/po1 Sinl Xinl
0. 0. 0.0 1.1	Mout p2/po1 Sout Xout

h) Modal camber design to attain specified outlet slope. Mode 9 is assumed to be a camber-changing mode. Note that mode 9 is selected as a free variable, but the **GMOD** value is ignored, since there is no (20) constraint specified. A negative mode index (e.g. -9) can be used to selectively omit any mode from being imposed via constraint (20).

1 2 5 20	Global variables
1 4 3 2	Global constraints
0.80 0. 1.50 -0.5	Minl p1/po1 Sinl Xinl
0. 0. 0.80 1.1	Mout p2/po1 Sout Xout
.	
.	
9 0.0	KMOD GMOD

i) Modal-Inverse design with 5 modes, driven by least-squares fit to specified surface pressure input via **EDP**.

```

1 2 5 20          |Global variables
1 4 3            |Global constraints
0.80  0.        1.50  -0.5  |Minl  p1/po1  Sinl  Xinl
0.    0.        0.    1.1   |Mout  p2/po1  Sout  Xout
.
.
1  0.0          |KMOD  GMOD
2  0.0          |KMOD  GMOD
3  0.0          |KMOD  GMOD
4  0.0          |KMOD  GMOD
5  0.0          |KMOD  GMOD

```

j) Modal-optimization viscous design, 20 modes. Mode-sensitivity run.

```

1 2 5 20          |Global variables
1 4 3 20         |Global constraints
0.80  0.        1.50  -0.5  |Minl  p1/po1  Sinl  Xinl
0.    0.        0.    1.1   |Mout  p2/po1  Sout  Xout
.
.
1  0.0          |KMOD  GMOD
2  0.0          |KMOD  GMOD
.
.
20 0.0          |KMOD  GMOD

```

k) Parametric-Inverse, driven by least-squares fit to specified surface pressure, with two user-defined constraints. Variable-index 40 takes parameter declarations from required file `bplist.xxx`

```

1 2 5 40          |Global variables
1 4 3 41 42      |Global constraints
0.80  0.        1.50  -0.5  |Minl  p1/po1  Sinl  Xinl
0.    0.        0.    1.1   |Mout  p2/po1  Sout  Xout

```

l) Geometry-parameter sensitivity run.

```

1 2 5 40          |Global variables
1 4 3 40         |Global constraints
0.80  0.        1.50  -0.5  |Minl  p1/po1  Sinl  Xinl
0.    0.        0.    1.1   |Mout  p2/po1  Sout  Xout

```

#### 4.9.2 Lines 6–7. Viscous flow parameters.

a) Inviscid analysis, or freeze current displacement thickness distributions.

```

0.0E6  5.0          |Re   Ncrit
1.0    1.0          |Xtr1 Xtr2

```

b) Viscous analysis, free transition with `Ncrit` specified.

```

1.3E6  5.0          |Re   Ncrit
1.0    1.0          |Xtr1 Xtr2

```

c) Viscous analysis, free transition with `Turb` specified in %.

```

1.3E6 -1.0         |Re   -Turb
1.0    1.0          |Xtr1 Xtr2

```

d) Viscous analysis, BL trips at 50% chord on side 1, and 70% chord on side 2.

```

1.3E6  5.0          |Re   Ncrit
0.5    0.7          |Xtr1 Xtr2

```

e) Two blades, viscous analysis, BL trips only on blade 1.

```

1.3E6  5.0          |Re   Ncrit
0.5    0.7    1.0   1.0 |Xtr1 Xtr2 Xtr3 Xtr4

```

### 4.9.3 Line 8. Isentropy and dissipation

a) Conserve S-momentum. Second-order dissipation.

```

1    0.98  1.0          |Ismom Mcrit Mucon

```

b) Conserve entropy (cannot be used with choked flow).

```

2    0.98  1.0          |Ismom Mcrit Mucon

```

c) Conserve entropy everywhere except at shocks (preferred).

```

4    0.98  1.0          |Ismom Mcrit Mucon

```

d) Increased dissipation, shocks across sheared grid expected.

```

4    0.90  1.5          |Ismom Mcrit Mucon

```

e) First-order dissipation, very strong shocks expected.

```

4    0.95 -1.2         |Ismom Mcrit Mucon

```

#### 4.9.4 Line 9. Streamtube thickness mode amplitudes

a) Streamtube thickness mode 1 amplitude specified directly.

```

1 2 5 7          |Global variables
1 4 3 7          |Global constraints
.
.
0.1  0.          |B1   B2

```

b) Streamtube thickness mode 1 amplitude specified via outlet pressure, mode 2 amplitude specified directly (note: mass flow is fixed here).

```

1 2 5 7 8        |Global variables
1 4 3 18 8       |Global constraints
0.30  0.    1.50  -0.5  |Minl  p1/p01  Sinl  Xinl
0.    0.6   0.    1.1   |Mout  p2/p01  Sout  Xout
.
.
0.    0.1          |B1   B2

```

#### 4.10 Geometry perturbation mode specification file `modes.xxx`

All the geometry deformation modes selected in `ises.xxx` must be described in the file `modes.xxx`. This has the following format.

```

KMOD(1)  IMODE(1)  GWT(1)  ST1(1)  ST2(1)  IEL(1)
KMOD(2)  IMODE(2)  GWT(2)  ST1(2)  ST2(2)  IEL(2)
.
.
KMOD(N)  IMODE(N)  GWT(N)  ST1(N)  ST2(N)  IEL(N)

```

where,

a) `KMOD` ties the mode shape to one of the modal geometry unknowns `GMODn` chosen via variable (20). `KMOD = 1` ties it to `GMOD1`, `KMOD = 2` ties it to `GMOD2`, etc. A mode can be composed of any number of disconnected pieces. Each piece of the mode is described on a separate line, each beginning the same `KMOD` value.

b) `IMODE` specifies the mode (or mode piece) shape. The shapes are implemented in `FUNCTION GFUN` or variants thereof (in `gmodes.f`). The particular shapes currently implemented are:

IMODE n	mode shape
$n = 1 \dots 8$	$\sin(n\pi s/s_{\max})$
$n = 20$	Local bump at leading edge, sized by local curvature
$n = 21 \dots 40$	$T_{n-20}(s/s_{\max})$
$n = 41 \dots \infty$	$\sin^a[\pi(s/s_{\max})^b]$ $\sin^a[\pi(1 - s/s_{\max})^b]$

$T_n$  are Chebyshev polynomials, modified to be zero at the mode piece endpoints  $s/s_{\max} = 0, 1$ . These are a good alternative to the simple sine waves ( $n = 1 \dots 19$ ), since they give more resolution at the endpoints. A good first choice for mode shapes is to use a reasonably large number of the Chebyshev modes (IMODE = 21, 22, ...) over the entire upper and/or lower surface. This allows nearly arbitrary geometry variations. The mode shapes selected can be plotted in **EDP** and in **LINDOP**.

c) GWT is the mode-piece scaling factor. If a mode consists of only one piece, GWT has no effect, as it merely rescales the mode amplitude variable. However, it is needed if the geometry mode is composed of two or more pieces, and each piece must be scaled differently. An example is a pure camber mode, where two identical shapes are placed on opposite sides of the airfoil, and GWT is specified as +1.0 and -1.0 for the two pieces (a positive mode displacement is taken as outward normal to the surface of the airfoil). GWT is also significant in that it will alter the convergence behavior (but not the final answer) of the steepest-descent optimization process. In this respect, it has been found advantageous to set the GWT factors for all the modes so that the mode derivatives with respect to ST are all comparable in magnitude.

d) ST1, ST2 are the mode endpoint locations on the airfoil. These are the normalized arc lengths  $s/s_{\text{side}}$  from the airfoil nose (not the stagnation point like with Mixed-Inverse!), to the trailing edge. For example, specifying ST1, ST2 = 0.0, 0.5 will result in the mode extending over approximately the front half of the airfoil. There is also the option to specify ST1 and ST2 as element  $x/c$  values. The necessary code is presently hibernating in SUBROUTINE GNSET (in `gnset.f`) as comments, and only needs to be enabled.

e) IEL specifies the target blade on which the mode piece acts.

#### 4.11 Design-parameter specification file `params.xxx`

This file specifies new design-parameter values, and is intended only for communicating with the **LINDOP** optimization driver. The file contains values for...

SINLin  
MINLin  
P1PTin  
V1ATin

SOUTin  
MOUTin  
P2PTin  
V2ATin  
REYNin  
BVRNin n=1...NBVRN  
GMODin n=1...NGMOD  
GPARin n=1...NGPAR  
MOVXin, MOVYin, SCALin, ROTain

All these quantities overwrite the values in `ises.xxx` and `bspec.xxx`. A message is printed when this occurs to warn the user.

Normally file `params.xxx` is written only by **LINDOP**. There is no reason to create it by the user directly.

## 5 Program Descriptions

The descriptions for running the code on a UNIX system are given below. Similar, but different commands would be used for VAX/VMS systems. Starting from scratch, the usual program execution sequence is

```
% iset xxx  
% ises xxx  
% iplot xxx
```

with the necessary input and output files for each step shown on the ISES Roadmap data flow diagram. The other programs are executed with the same command syntax, e.g.

```
% edp xxx  
% iprint xxx  
% bldset xxx
```

On UNIX, the execution of all these programs can be more conveniently and more rapidly directed via the shell script `run`. For example, if one is computing case “xxx”, one would invoke the shell script with

```
% run xxx
```

and any program can then be invoked for the `xxx` case with a quick menu selection.

## 5.1 ISET

### 5.1.1 Basic Initialization

**ISET** is the program which initializes the grid, the densities and a variety of other variables. The required and optional input files to **ISET** are shown on the MISES Roadmap. The output file is the main solution state file `idat.xxx`.

**ISET** is menu-driven to allow the user to iteratively generate a good initial grid by tweaking a small number of gridding parameters. The top level **ISET** menu is

- 1 Specify new inlet slope and block off grid
- 2 Generate spacings and initialize grid
- 3 Elliptic grid smoother
- 4 Write `idat.xxx`
- 5 Plot grid
- 6 Plot  $C_p$  vs  $x/c$
- 7 Modify grid parameters
- 8 Write grid parameters to `gridpar.xxx`
- 9 Change plot size
- 10 Read geometry from `blade.xxx` file
- 11 Read geometry from `bparm.xxx` file

Select grid generation option (0 to exit):

The generate the initial grid and write out the solution file, Options 1, 2, 3, 4 (in that order), must be issued as a minimum. Normally, Option 1 is executed automatically when **ISET** is started and can be skipped.

By default, **ISET** tries to read `blade.xxx`, but it can be forced to read and use the `bparm.xxx` geometry parameter file from its menu. The default file is hardwired near the top of PROGRAM **ISET** (in `src/iset/iset.f`), and can be easily changed if desired.

### 5.1.2 Panel solution

Option 1 uses the specified inlet slope to generate an incompressible 2-D panel solution, which is then used to trace a pair of stagnation streamlines just above and below each blade. Iso-potentials emanating from all leading and trailing edges are also located. This divides up the domain into blocks, which are then automatically displayed in a plot. These blocks form the skeleton on which the grid is generated. The specified inlet slope here is of course somewhat arbitrary, since it is only used for initial grid generation. It is a good idea, however, to specify an inlet slope which avoids massive  $C_p$  spikes on the leading edge. This minimizes the start-up trauma with a subsequent viscous **ISES** solution. With small leading-edge radii, the range of

“tolerable” inlet slopes is quite small. Option 6 can be used to examine if the panel solution is reasonable, and a new slope can be specified again with Option 1 if necessary.

To simplify finding a “reasonable” inlet slope for sharp or nearly-sharp leading edges, it is also possible to set the inlet slope implicitly by choosing the leading-edge Kutta condition at the Option 1 prompt.

Enter new inlet slope (or -999 to use LE Kutta):

Entering -999 will result in the inlet slope being set so that there is zero loading at the sharp leading edge point, or at the nose tangency point (described in the **EDP** section) for a blunt-leading edge case. Enforcing the leading-edge Kutta condition will of course minimize any  $C_p$  spikes at the leading edge.

### 5.1.3 Initial surface gridding

Option 2 distributes grid nodes along the streamlines on the blade surfaces. The local arc-length increment  $\Delta s$  between two surface grid nodes is determined from

$$\Delta s \sim \frac{1}{1 + a|\kappa|^b}$$

where  $\kappa$  is the local surface curvature. In regions of high curvature, the spacing is therefore smaller, depending on the curvature exponent  $b$  and the coefficient  $a$ . The exponent is specified directly from the menu described below. A large exponent ( $b = 2$ , say), makes the spacing small in high-curvature regions. A small exponent ( $b = 0.05$ , say), makes the spacing more nearly uniform everywhere. The curvature coefficient  $a$  is indirectly controlled by specifying  $\Delta s$  spacing at the leading edge (or stagnation point, to be more precise) of the blade. Note that if  $\kappa$  is rather small at the stagnation point, the effect of  $a$  is largely shut off in the expression above. If necessary, a fudged additional curvature is added locally very near the stagnation point to allow the spacing requirement to be met. A message is printed when this action is taken. Fudged curvatures are also introduced near the trailing edge, and optionally at selected local-refinement zones on the upper and lower surfaces. The aim is also to control the local spacing.

Since MISES v 2.1, the local/average spacing ratios  $\Delta s/\Delta s_{\text{avg}}$  are specified instead of the actual spacings  $\Delta s$ . The average spacing is defined as  $\Delta s_{\text{avg}} = s_{\text{side}}/N$ , where  $s_{\text{side}}$  is 1/2 of the airfoil perimeter and  $N$  is the number of airfoil-side points. Specifying  $\Delta s/\Delta s_{\text{avg}}$  is more convenient than specifying  $\Delta s$  itself, since the former automatically adjusts the spacing if the number of points is changed.

The curvature exponent, and stagnation-point and trailing-edge spacing ratios and local-refinement parameters are altered from the following menu, which appears each time the spacing is generated and plotted with Option 2:

```

D sLE/dsAvg, dsTE/dsAvg spacing ratios
C urvature exponent
U pper side spacing refinement
L ower side spacing refinement      |   B lowup plot
N umber of points                  |   R eset plot

```

Change what? (<Return> if spacing OK):

For each change request, the current values are displayed after the prompt. Selecting “U”, for example, might produce

```

Enter new upper s/smax limits, local/avg density ratio  0.1000  0.2500  0.800

```

and just hitting <Return> will take the current values as the default input. One can change only some of the required three inputs by using commas. Entering

```
0.15
```

will only change the first value from 0.1 to 0.15, while entering

```
,,0.5
```

will only change the third value from 0.8 to 0.5. The first two “s/smax” values specify the fractional arc length from stagnation point to the trailing edge where the local refinement is to be placed. Tick marks inside the airfoil element contour indicate this fractional arc length in increments of 0.10. The local/avg density ratio specifies the increase in local grid density over the average density which would occur with all points spaced equally.

The point-number option “N” allows one to specify the average point number per side. With offset grids (described shortly), the upper and lower point numbers are different.

After the spacing parameters are altered, the new distribution is generated and displayed. The actual LE, TE, max, min, spacing ratios are also printed out. It must be mentioned that only the stagnation point spacing ratio “dsLE/dsAvg” can be controlled precisely with the input parameters. The other spacing ratios are approximate and may need to be iterated.

Once a good node distribution on each element is obtained, **ISET** proceeds to modify all the spacings to resolve any conflicts between vacing blade surfaces. Basically, all the distributions are automatically fudged to make spacings on element surfaces facing each other match as closely as possible. This prevents massive grid shearing which would otherwise occur. No such action is necessary for single-element airfoils.

Once all the node distributions are finalized, intermediate streamline nodes are generated in the flowfield interior by simple linear interpolation from the stagnation and farfield streamlines. The resulting grid is not yet suitable for **ISES** calculations, but can be viewed with Option 5.

#### 5.1.4 Grid smoothing

Option 3 invokes an elliptic SLOR grid smoother to “clean up” the linearly interpolated grid. This eliminates all kinks, overlaps, and also makes all the grid streamlines correspond to streamlines of incompressible inviscid flow. This is then an excellent initial guess for the **ISES** solver.

#### 5.1.5 Initial solution file output

After the grid is smoothed, Option 4 can be issued to write out the initial solution file `idat.xxx` which is then ready for the **ISES** solver. Before this is done, however, it is a good idea to view the grid with Option 5. Options 2,3 or 1,2,3 can then be repeated if necessary to obtain an acceptable grid before it is written out.

#### 5.1.6 Grid parameters

Option 7 puts up the menu

```
Current grid parameters:
G      t t  inlet,outlet  offset grid flags
I       30  number of inlet  points
O       30  number of outlet points
S       15  number of streamlines
X    0.500  x-spacing parameter
A    1.500  aspect ratio of each cell at stagnation point
```

```
Change what (<return> if done)?:
```

in which each gridding parameter is displayed and can be immediately altered. The type of grid topology is controlled by the inlet and outlet grid flags. A flag set to “t” specifies an offset I-type grid, while “f” specifies a periodic H-type grid. Different grid types can be used over the inlet and outlet. An offset grid is very nearly orthogonal, but it increases the Newton matrix bandwidth and thus significantly increases the CPU requirements for a given grid size. On the other hand, a nearly-orthogonal grid requires much fewer streamlines, which mostly alleviates this CPU penalty.

The great advantage of a nearly-orthogonal grid is that it is much more tolerant of shock waves. Above a certain amount grid shear, supersonic flows are nearly incomputable. Conversely, subsonic flows cause little trouble even with strong grid shear, and are suitable in turbine inlets, for example. The recommended grid topology for common cascade cases are as follows:

Flags	Case
f f	low-speed, low-turning
f t	high-speed turbine
t t	transonic compressor tip
t f	transonic compressor mid-section

The number of inlet and outlet points for periodic grids is set with options “I” and “O”. These are ignored for offset grids, since the inlet and/or outlet points cannot be set independently of the surface points due to topological constraints. The number of streamlines is set with “S” for all cases.

The “X-spacing” parameter controls the repelling-force between the quasi-normal grid lines during the SLOR smoothing phase. There is little reason to adjust it from its default value. The leading-edge cell aspect ratio controls the width of the streamtubes adjacent to surfaces. Again, the default value suffices for most cases.

### 5.1.7 Grid parameter saving, recall

Once a good set of gridding parameters is obtained, including the spacing parameters generated with Option 2, they can all be saved to `gridpar.xxx` by specifying Option 8 at any time. If this file already exists, it is overwritten. `gridpar.xxx` will then be automatically read when **ISET** is executed again for that same `xxx` case, which causes all the gridding parameters to take on their saved values. This allows rapid generation of grids for cases which differ only slightly (e.g. camber, inlet slope), since the same gridding parameters can then be used.

If the trailing edge of the blade is not closed, a constant-thickness wake gap is left extending from the blade base. In inviscid calculations, this gap remains constant in width, but is free to move up and down so that it sustains no pressure jump and hence no lift. For viscous calculations, the gap will collapse down to the local wake displacement thickness, and is still free to move up or down. A special treatment is used to correct for the dead air region immediately behind the blunt base whose length and shape is set in **ISET** to match experimental correlations. This special treatment results in an increase in momentum thickness downstream and accurately accounts for “base drag”, which is also reflected in an increase in the mixed-out loss.

### 5.1.8 Smoothing and writing the grid

**ISET** uses an elliptic grid generator to initialize the grid, which is invoked with Option 3 at the top level menu. The grid can be plotted before or after smoothing with option 4 if desired. If the overall grid is unsatisfactory, option 1 can be repeated as often as necessary.

Once a satisfactory smoothed grid is obtained, executing option 4 will first initialize the flowfield using hard-wired defaults, and then write out the unformatted state file `idat.xxx`. The description of this file can be found from the comments in `STATE.INC`, which declares all the variables in COMMON blocks.

## 5.2 ISES

**ISES** is the main program that solves the Euler equations. It always reads the two required input files, `idat.xxx` and `ises.xxx`, and then writes the output file back to `idat.xxx`. Thus the input file `idat.xxx` can either be a restart file from an old calculation, or a new file created by **ISET**. The optional input files will be accessed as needed, depending on the type of case being run.

New for v 2.63:

Upon termination, **ISES** will also generate the small text file `sens.xxx` which contains the sensitivities of various global variables, with respect to the specified parameters. These sensitivities can be used for optimization procedures, solution interpolation and extrapolation, and a number of other uses. The sensitivity calculation requires negligible CPU time, and is described in the supplemental document *MISES Constrained Least-Squares-Inverse Formulation and Sensitivity Calculation Procedures*.

**ISES** is run by typing `run xxx` and selecting the `ises` option, at which point the user is asked for the number of Newton iterations “n” to be executed. The program response is as follows.

$n = 0$  writes solution to output file and terminates  
 $n > 0$  performs  $n$  iterations and repeats the question  
 $n < 0$  performs  $|n|$  iterations then writes solution to output file and terminates

After each iteration, the r.m.s. and maximum changes of the density, node position, and viscous variables are displayed. Also displayed are the changes of various global variables. Convergence to plotting accuracy occurs when the changes drop to about  $0.1 \times 10^{-3}$  or so. Convergence to machine accuracy is achieved when the changes refuse to go down further with each iteration (about  $0.1 \times 10^{-5}$ ). **ISES** will terminate execution early if convergence is reached. The convergence tolerances are specified in the include-file `EPS.INC`. If the Mach number is low (below 0.1, say), or significant flow separation is present, the changes will not go down as far as they would otherwise. This is due to the Newton matrix being less well-conditioned for these cases. For incompressible viscous cases, it is recommended that a Mach number of at least 0.05 – 0.1 be used. This is effectively incompressible.

**ISES** will automatically select the appropriate inflow/outflow boundary conditions based on the local Mach number. Note that these BCs are distinct from the inlet/outlet mixed-out flow condition specification at  $m'_1$  and  $m'_2$ . Rather, they determine the local nature of the solution at the actual inflow and outflow grid planes. The details are given below.

### 5.2.1 Inflow boundary conditions

The inflow BCs given in the table below use suitable combinations of the local flow variables, depending on the local streamwise and axial components of the mass-averaged inflow Mach number.

Inflow type	Mach number	boundary condition
Subsonic	$M < 1$ , $M_{\text{axial}} < 1$	$r(v + \Omega r) = r_1 \left( \frac{V_1 S_1}{\sqrt{1 + S_1^2}} + \Omega r_1 \right)$
Supersonic/ Subsonic-axial	$M > 1$ , $M_{\text{axial}} < 1$	$\nu(M) \pm \beta = \nu(\hat{M}_1) \pm \arctan(\hat{S}_1)$
Supersonic	$M > 1$ , $M_{\text{axial}} > 1$	$r(v + \Omega r) = r_1 \left( \frac{V_1 S_1}{\sqrt{1 + S_1^2}} + \Omega r_1 \right)$

The subsonic and supersonic inflow BCs simply require that the angular momentum not change between the inflow plane and the defining station at  $m'_1$ . If both the radius and stream-tube thickness are constant over the inlet region, then this is equivalent to imposing the inlet streamline slope:

$$r(v + \Omega r) = r_1(u_1 S_1 + \Omega r_1) \quad \longrightarrow \quad \frac{v}{u} = S_1$$

The supersonic/subsonic-axial BC relates the local flow angle  $\beta = \arctan(v/u)$  and the local Mach number  $M$  via the Prandtl-Meyer function

$$\nu(M) \equiv \sqrt{\frac{\gamma+1}{\gamma-1}} \arctan\left(\sqrt{\frac{\gamma-1}{\gamma+1}} \sqrt{M^2 - 1}\right) - \arctan(\sqrt{M^2 - 1})$$

and thus allows waves to pass out of the grid inflow plane without reflection. The quantities  $\hat{M}_1$  and  $\hat{S}_1$  are effectively the same as  $M_1$  and  $S_1$  defined at  $m'_1$ , but are corrected for any difference between  $r, b$  at the inlet grid plane, and  $r_1, b_1$ . The corrected flow state  $\hat{\rho}, \hat{u}, \hat{v}, \hat{p}$ , is related to the  $m'_1$  flow state  $\rho_1, u_1, v_1, p_1$ , through conservation of mass flow, absolute angular momentum, constant rotation-corrected total pressure, and constant rotation-corrected total enthalpy (i.e. rothalpy):

$$\begin{aligned} \hat{\rho} \hat{u} r b &= \rho_1 u_1 r_1 b_1 \\ r(\hat{v} + \Omega r) &= r(v_1 + \Omega r_1) \\ \hat{p} \left(1 - \frac{\hat{u}^2 + \hat{v}^2}{2I}\right)^{-\gamma/\gamma-1} &= p_{oa} \\ \frac{\gamma}{\gamma-1} \frac{\hat{p}}{\hat{\rho}} + \frac{\hat{u}^2 + \hat{v}^2}{2} &= I + \frac{1}{2}(\Omega r)^2 \end{aligned}$$

The appropriate set of global variables and constraints for each of the three types of inlet flows is listed below (repeating some of the earlier input-file examples).

Inflow type	Variables, constraints
Subsonic	1 15 6 1 18 6
Supersonic/ Subsonic-axial	1 15 6      or      1 15 6 15 18 6      9 18 6
Supersonic	1 15 6 15 18 1

In the supersonic/subsonic-axial case, the inlet slope constraint (1) is discarded in favor of the inlet Mach constraint (15) or the inlet tangential speed constraint (9). Not specifying the inlet slope explicitly is physically correct in light of the “unique-incidence” condition. In the supersonic case, the inlet stagnation pressure constraint (6) is additionally discarded in favor of the inlet slope constraint (1). Now the stagnation pressure will be implicitly constrained by the fully-specified inlet flow, which implies a specified mass flow.

Using an inappropriate variable/constraint set for any inlet flow type will usually still produce a stable solution, but in general will result in some mismatch between the specified and resulting quantity, e.g.  $M_1 \neq \text{MINLin}$ ,  $S_1 \neq \text{SINLin}$ ,  $p_{o1} \neq 1/\gamma$ , etc.

### 5.2.2 Outflow boundary conditions

For the outflow, the boundary conditions again use some combination of local and global variables.

Outflow type	Mach number	boundary condition
Subsonic	$M < 1$ , $M_{\text{axial}} < 1$	$\frac{\partial}{\partial n} [r(v^{\text{isen}} + \Omega r)] = 0$ , $S = S_{\text{exit}}$
Supersonic/ Subsonic-axial	$M > 1$ , $M_{\text{axial}} < 1$	$\frac{\partial}{\partial n} [\nu (M^{\text{isen}}) \pm \beta] = 0$ , $S = S_{\text{exit}}$
Supersonic-axial	$M > 1$ , $M_{\text{axial}} > 1$	$\frac{\partial^2 p}{\partial s^2} = 0$

where  $M^{\text{isen}}$  is an isentropic Mach number defined from the local static pressure  $p$  and radius  $r$ , and the known inlet total pressure  $p_{o1}$  and radius  $r_1$ .

$$(M^{\text{isen}})^2 = \frac{2}{\gamma-1} \left[ \left( \frac{p_o^{\text{isen}}}{p} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \quad p_o^{\text{isen}} = p_{o1} \left( \frac{I + \frac{1}{2}\Omega r^2}{I + \frac{1}{2}\Omega r_1^2} \right)^{\frac{\gamma}{\gamma-1}}$$

The isentropic tangential velocity  $v^{\text{isen}}$  is determined in a similar manner from the local static pressure and the local flow angle  $\beta$ . As with the inflow BCs, these outflow BCs are constructed to be essentially invisible to the flowfield. In particular, the supersonic/subsonic-axial BC allows waves to pass out without spurious reflection, although here the “transparency” isn’t quite as perfect as at the inflow boundary if strong shock losses are present.

The exit flow slope  $S_{\text{exit}}$  (which may be different than the mixed-out flow slope  $\bar{S}_2$ ) is imposed on one streamline at the outflow boundary, and then the flow slopes of the remaining streamlines are constrained by the  $\partial/\partial n$  outflow BCs.  $S_{\text{exit}}$  is typically chosen as a global variable DSLEX (2), and it is normally implicitly constrained by specifying the trailing edge Kutta condition (4). Setting the flow slope explicitly via SOUTin using constraint (2) is not advised, since violating the Kutta condition will result in a large flow disturbance at the trailing edge. Constraint (2) can be used in conjunction with a camber-changing mode variable (20), in a design case where it is desired to modify the blade camber to attain a specified amount of turning.

The simple pressure gradient constancy condition for supersonic-axial flow is in effect a direct extrapolation from the interior to the boundary, and does not influence the blade flow solution in any case. This outflow boundary condition doesn't depend on  $S_{\text{exit}}$ , and hence does not require the outlet slope variable DSLEX (2) / TE Kutta condition (4) pair to be specified as a global variable and constraint. These may be left in, however, with no ill effects.

Besides the TE Kutta condition redundancy, supersonic-axial outflows are treated differently in a number of other ways. Most notable is that the specified outlet pressure or Mach number will generally not be what comes out of the final solution. The local exit Mach number of an axially-supersonic flow is set entirely by the initial conditions at the blade row, and cannot be prescribed. The **ISES** recognizes this and generally ignores any specified mixed-out constraint.

### 5.3 IPLOT

**IPLOT** is the program which displays the solution in `idat.xxx` at any time whether the solution is converged or not. It is executed by the command `run xxx` and selecting the `iplot` option. Note that if the solution in `idat.xxx` is not converged, the results are physically meaningless. The top-level **IPLOT** menu is shown below.

- 1 Blade surface plots
- 2 Streamtube plots
- 3 Contour/grid plots
- 4 Wake profile plots
- 5 r,b,ln(Po) vs m' stream surface definition plots
- 6 Wheel View
- 7 Dump flowfield to text file
- 8 Dump BL quantities to text file

Select IPLOT option (0=Quit):

#### 5.3.1 Blade surface plots

The "Blade surface plots" menu brought up by the top-level option 1 allows plotting of most of the airfoil surface and wake boundary layer variables:

- 1 Mach vs x
- 2 Cp vs x
- 3 Hk vs x
- 7 Ue vs x
- 4 s1 D,T vs x
- 8 A/Ao vs x
- 5 s2 D,T vs x
- 9 Ct vs x
- 6 Cf vs x
- 10 Rtheta vs x

- |    |   |    |                 |
|----|---|----|-----------------|
| 11 | Forces                                    | 13 | Change blade    |
| 12 | Options                                   | 14 | Hardcopy toggle |
| 15 | Change x-axis coordinate type on BL plots |    |                 |
| 16 | Change x-axis limits on BL plots          |    |                 |
| 17 | Change y-axis limits on current BL plot   |    |                 |
| 18 | Cursor blowup of current BL plot          |    |                 |
| 19 | Reset x,y-axis limits for BL plots        |    |                 |
| 20 | Annotation menu                           |    |                 |
| 21 | Plot-page options                         |    |                 |

Select surface plot option for blade 1:

The Mach and menu item 1 is the isentropic Mach number calculated from the local pressure.

The menu items (3 ... 10) display boundary layer quantities for at most one element at a time. Items 3, 6 ... 10 show one variable on both sides of the element, while items 4,5 show  $\delta^*$  and  $\theta$  together for one side only. Items 4,5 also show the total (top + bottom) thicknesses for the wake as dotted lines, and also the inviscid-grid wall-offset distance  $\Delta n$  as a dashed line. Normally,  $\Delta n = \delta^*$ , and the two curves will overlay, but only if the case is fully converged. If the dashed  $\Delta n$  curve can be discerned, the case is not converged. A number of plot coordinate types can be selected with item 15. Items 16,17 allow rescaling of the BL variable plot axes to zoom in on details of interest.

One important feature of **IPLLOT** which needs some elaboration is the normalizing conventions for  $C_p$ , forces, losses, etc., which are listed with the “Forces” menu item 11. In general, all flow quantities are normalized with static isentropic reference values defined from the reference Mach number  $M_{\text{ref}}$ , at the radius  $r_{\text{ref}}$ . When **IPLLOT** is started,  $M_{\text{ref}}$  is initialized to the inlet Mach number  $M_1$ , and  $r_{\text{ref}}$  is initialized to  $r(m_1')$ , so that all quantities are referenced to the usual “1” quantities used for most of the normalization in **ISES**. However,  $M_{\text{ref}}$  and/or  $r_{\text{ref}}$  can be changed to anything using Option 12 in the “Blade surface plots” menu.

Three types of  $C_p$  can be displayed from the surface plots menu:

$$C_p = \frac{p - p_1}{\frac{1}{2}\rho_1 V_1^2}$$

$$\bar{C}_p = \frac{p - p_1}{p_{o1} - p_1}$$

$$C_{p_o} = \frac{p_{o1} - p}{p_{o1} - p_1}$$

The default type is  $C_p$ , but any type can be chosen using Option 12.

Two types of loss coefficients are defined from the hypothetical mixed-out state  $\bar{p}_2, \bar{p}_2 \dots$  at  $m'_2$ , described earlier.

$$\bar{\omega} = \frac{p_{o2}^{isen} - \bar{p}_{o2}}{p_{o1} - p_1}$$

$$\zeta = \frac{p_{o2}^{isen} - \bar{p}_{o2}}{p_{o2}^{isen} - \bar{p}_2}$$

In addition, inviscid shock-loss and viscous-loss components are defined at the grid outflow plane as derived in H. Youngren's thesis:

$$\omega_i = \frac{1}{p_{o1} - p_1} \int (p_o^{isen} - p_o) \frac{d\dot{m}}{\dot{m}}$$

$$\omega_v = \frac{1}{p_{o1} - p_1} \left( \frac{p_o}{p} \frac{\rho V}{\dot{m}} \rho V^2 \Theta b \right)_{\text{exit}}$$

The derivation of these two components assumes a small wake velocity defect, or  $H-1 \ll 1$ , and hence is less rigorous than that of the mixed-out loss  $\bar{\omega}$ . Nevertheless, in most cases  $\omega_i + \omega_v \simeq \bar{\omega}$  to within a few percent, which is reassuring! The cases for which the two approaches differ significantly are supersonic-exit turbines, and radial-outflow blading in general. Here, there is no rigorous way to define the viscous and inviscid loss components, since the momentum defect doesn't asymptote to a nearly-uniform value over the outlet region.

### 5.3.2 Suction

The fluid withdrawn through the suction slot has several parameters which are significant for the overall stage or machine performance. Besides the specified suction coefficient  $C_Q$  defined earlier, the total enthalpy  $h_{os}$  and pressure  $p_{os}$  are also important. The complication here is the difficulty in defining the loss of the blade row. Figure 5 shows the lateral displacement of the flow due to suction being applied at the blade surface. In the computation, the mass is not actually removed from the streamtubes, but rather the "removed" fictitious portion of the flow (shown shaded in the Figure) is made to overlap the real flow downstream of the slot. The same treatment is used in viscous cases, except that an additional shift of  $\delta^*$  is superimposed on top of the suction-induced displacement. The net boundary condition applied to the streamline adjacent to the surface at streamwise location  $s'$  is

$$\Delta n' r = \delta^* - \frac{1}{\rho_e V_e b} \int^{s'} -\rho_w v_w b r ds'$$

where  $\Delta n'$  is the normal distance (in the  $m'-\theta$  plane) from the wall to the streamline. The integral is simply the total mass flow sucked away upstream of the  $s'$  location, and reaches the ultimate value  $\dot{m}_{\text{suct}}$  downstream of the blade row.

### 5.3.3 Streamtube plots

This menu allows the plotting of various flowfield quantities versus  $s'$  (arc length in the  $m'-\theta$  plane). It is mainly a diagnostic tool.

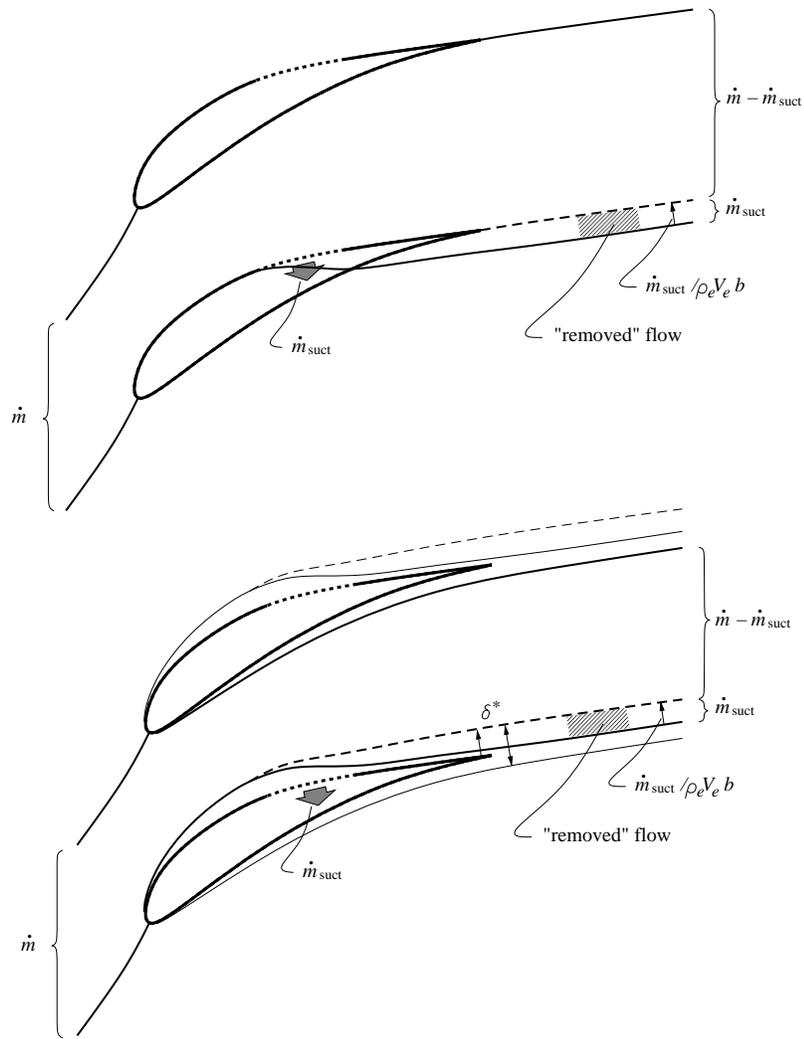


Figure 5: Inviscid and viscous flows with suction.

### 5.3.4 Contour/grid plots

This menu allows plotting of the flowfield. The streamline grid, flow variable contours, and Mach waves can be displayed. There are also miscellaneous options for locating cell indices, shading isentropic cells, plotting BL profiles, etc.

### 5.3.5 Wake profile plots

The wake profile plots allow the display of various inviscid flow quantities versus tangential distance, much like a boundary layer profile. This is mainly a diagnostic tool, useful for checking whether the shock defect wake is adequately resolved, for instance.

### 5.3.6 $r, b$ vs $m'$ stream surface definition plots

This gives a plot of  $r(m')$  and  $b(m')$  as defined in the `stream.xxx` file, showing the individual  $X(I)$ ,  $R(I)$  spline node values as well as the resulting analytic spline function. The individual streamtube thickness modification modes  $B_1 b_1(m')$  and  $B_2 b_2(m')$  are also displayed.

### 5.3.7 Wheel view

This displays a nifty picture of the entire rotor from the side and along the axis of rotation. Only geometries with significant radial changes will make a meaningful picture.

## 5.4 EDP

**EDP** is an interactive menu-driven program used to modify data in `idat.xxx`, primarily for inverse calculations. Three types of inverse methods are provided: Mixed-Inverse, Modal-Inverse, and Parametric-Inverse. The latter two are essentially the same, and differ only in the manner in which the blade shape modification is defined. The table below lists the key features of each method.

Method	Geometry Description	Solution Method
Mixed-Inverse	Pointwise (arbitrary shape)	require $p(s) = p_{\text{spec}} + \dots$
Modal-Inverse	Surface-normal displacement modes	minimize $\int (p - p_{\text{spec}})^2 ds$
Parametric Inverse	User-defined geometric parameters	minimize $\int (p - p_{\text{spec}})^2 ds$

As the input-file examples indicate, Mixed-Inverse is triggered by global variables (11),(12), while Modal- and Parametric-Inverse are triggered anytime the number of global variables exceeds the number of global constraints. With the least-squares minimization, all the global variables are determined so as to obtain the best-possible fit of the computed surface pressure  $p(s)$  to the specified surface pressure  $p_{\text{spec}}$  input in **EDP**, while simultaneously satisfying the global constraints that are declared. These declared constraints would be the usual ones used

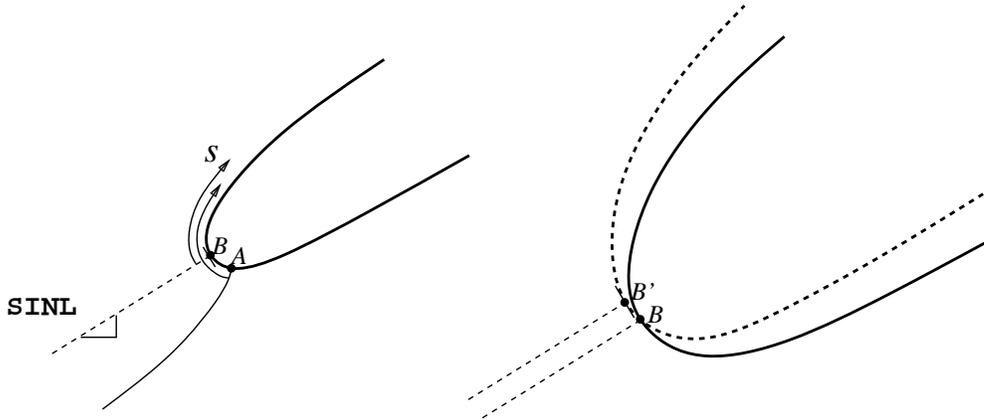


Figure 6: Definition of surface arc length for Mixed-Inverse (from the stagnation point A), and for Modal-Inverse (from the nose tangency point B). Tangency point moves during modal shape changes, but the original point B remains fixed.

for analysis calculations — specified inlet slope (1), TE Kutta condition (4), etc. Any global variable can therefore play the role of a design parameter if it doesn't have a declared constraint associated with it, although this can easily give ill-conditioned problems if such a “free” variable has only a weak effect on the surface pressure. The intent of the generalized least-squares procedure is to drive geometry-related variables for inverse design where the blade shape is restricted to a particular parametric description. In contrast, the Mixed-Inverse method is aimed at efficiently eliminating local aerodynamic defects ( $C_p$  spikes, etc.) in design problems which permit arbitrary blade shapes.

#### 5.4.1 Surface parameterization

The Mixed-Inverse method and Modal-Inverse method use a slightly different way to define the fractional surface arc length parameter  $\sigma = s/s_{\text{side}}$ , illustrated in Figure 6. Here,  $s$  is understood to be the arc length in the  $m' - \theta$  plane.

Since the surface grid points move along with the stagnation point A, Mixed-Inverse essentially defines the surface pressure and geometry in the streamwise grid node index:  $p(i)$ ,  $m'(i)$ ,  $\theta(i)$ . In contrast, Modal-Inverse defines these quantities in terms of the arc length:  $p(s)$ ,  $m'(s)$ ,  $\theta(s)$ , with  $s$  measured from the fixed nose tangency point B. This nose point is set once and for all in **ISET** at the location where the surface-tangent is perpendicular to the initial grid slope SINL. If the blade is deformed by a camber mode which extends all the way to B, then the nose tangency point will move from point B to B' as shown in Figure 6. However, the initial point B is always retained for defining  $s$ , and subsequent points B' are ignored.

Of particular importance are the endpoints of the *target-segment*, which is the part of the surface which is to be modified. In Mixed-Inverse, these endpoints are actually specified as grid point indices  $i_0, i_1$ , while for Modal-Inverse, they are specified as the normalized arc lengths

$\sigma_0 = s_0/s_{\text{side}}$ ,  $\sigma_1 = s_1/s_{\text{side}}$ . The endpoints for either Inverse method are specified along with the surface pressure distributions in **EDP**.

#### 5.4.2 EDP execution

**EDP** is executed by `run xxx` and selecting the `edp` option. **EDP**'s top level menu is:

1	EDIT	pressures	10	Read	flow(spec) data file
2	SET	redesign flags	11	Write	flow(spec) data file
3	WRITE	idat.xxx	12	Write	x,y,z blade file
4	READ	idat.xxx	13	Write	m',theta blade file
5	Change	idat.xxx params	14	Annotate	plot
6	Change	r,b vs x	15	Change	plot size
7	Change	DPO vs x	16	Hardcopy	current plot
8	Print	flow parameters			
9	Plot	mode shapes			

Select EDP option (0=QUIT):

Most of these options are self-explanatory. Options 6 and 7 permit changing the  $r(m')$ ,  $b(m')$ , and/or  $\mathcal{P}(m')$  distributions currently stored in `idat.xxx`. Likewise, option 5 permits changing of several variables and flags which are otherwise inaccessible. The option 5 sub-menu can be easily customized to allow changing of any quantity present in the `STATE.INC` global include file.

The primary purpose of **EDP** is of course the interactive input of specified surface pressures for inverse calculations. MISES is really a blade redesign system rather than a pure design code, and productive use of **EDP** necessitates that `idat.xxx` contain a previously converged case. The surface pressure modification is then done via option 1, which puts the user into the pressure-editing sub-menu:

```

-----
I nitialize Mach (spec)=Mach (wall) on target element
M odify Mach (spec)
D emark inverse segment
S lope-matching at segment endpoints (toggle) ->  F

F low data select (Mach, Cp, P/P0a, P/P01)
T oggle plot type (flow vs s, or vector plot)

Z oom
R eset zoom to original size

```

L imits, set plot limits  
A nnotation menu  
H ardcopy current plot

Select edit option:

The specified pressure array **CPspec** read in from **idat.xxx** is originally zeroed out in **ISET**, so if this is a first inverse editing session, the user must select option **I** to initialize **CPspec** to the current wall pressure coefficient array **CPwall**. Option **M** will then allow the user to “edit” **CPspec** with the screen cursor. This can be done repeatedly if needed. If necessary, the inverse target segment endpoints can be cursor-specified with option **D**. This is done either on the  $C_p(s)$  plot or on the geometry/vector plot, depending on what’s currently on the plot screen (toggle with option **T**). The initial default target segment endpoints for Mixed-Inverse are the first grid point after the front stagnation point, and at the rear trailing edge point. For Modal-Inverse they are the nose and trailing edge locations.

$i_0 = i_{LE} + 1$	$i_1 = i_{TE}$	Mixed-Inverse default
$\sigma_0 = 0.0$	$\sigma_1 = 1.0$	Modal-Inverse default

After the pressure-editing menu is exited, the following redesign-flag menu comes up (it can also be brought up with option 2). The intent is to verify the endpoint locations and several other control flags, and change them from the keyboard if necessary. If a Parametric-Inverse case is to be run, this menu can be ignored.

```
Current redesign flags...
B lade      :      1
M oved-side:      2
P spec-side:      2
I endpoints:      20      78      (for Mixed-Inverse)
S endpoints: 0.0000 1.0000      (for Modal-Inverse)
```

Select flag to change (<return> if OK):

The target-blade **NDES**, moved-side **KSMOVE**, and Pspec-side **KSPRES** flags determine what blade side(s) get redesigned and where the surface pressures are imposed. Figure 7 shows the four possible ways to change the target blade, corresponding to the possible moved-side **KSMOVE** flag settings.

```
-1 Both sides move opposite (camber preserved)
 0 Both sides move together (thickness preserved)
 1 Upper side moves
 2 Lower side moves
```

With Mixed-Inverse, **KSMOVE** = -1 or 0 forces corresponding grid nodes on opposite sides of the blade to move opposite or together. For offset grids, it is not possible to identify “corresponding”

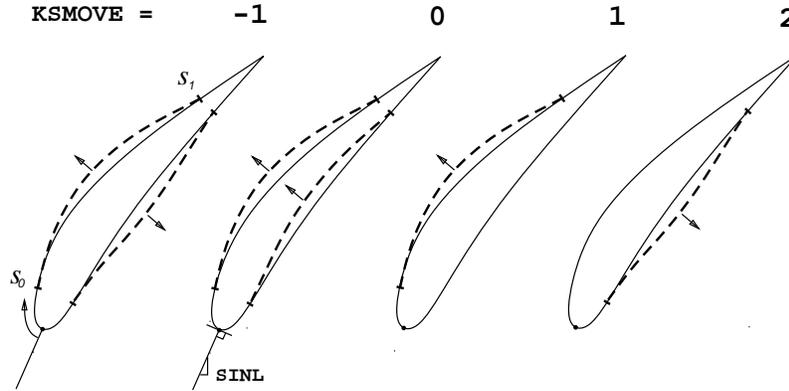


Figure 7: Two-side and single-side geometry changes.

point pairs, and hence Mixed-Inverse with  $KSMOVE = -1, 0$  can only be used on non-offset (periodic) grids. In any case, this option is not recommended on periodic grids either. Camber and thickness changes are best performed using Modal-Inverse.

The geometry changes can be driven by specifying the pressure jump (i.e. the loading) across the blade, or by specifying the pressures on an individual surface. This is controlled by the Pspec-side  $KSPRES$  flag, which can take on the following values.

- 0  $\Delta p$  across blade is specified
- 1 Upper side pressures specified
- 2 Lower side pressures specified

The following combinations produce ill-posed inverse problems and must be avoided.

$KSMOVE$	$KSPRES$
1	2
2	1
-1	0

The last combination is ill-posed since thickness changes have little effect on loading to first order.

If a screen cursor is not available, then top-level option 11 can be used to dump the surface pressures (or equivalent Mach numbers) into a formatted scratch file, which can then be edited manually, or preferably with the user's own software. This file can then be read back into **EDP** with top-level option 12. The scratch file also contains the indices of the requested inverse segment endpoints and the index of the blade side which contains the inverse segment.

However the  $CP_{spec}$  distribution is generated in **EDP**, it must be written out with the `idat.xxx` file so that **ISES** can access the information. This is done with top-level option 3.

**ISES** is configured for a Mixed-Inverse case by selection of the global variables (11) and (12) and the global constraints (11) and (12). If desired, variables (13) and/or (14) can also

be specified, with corresponding constraints (13),(14). These produce a smoother pressure distribution at the inverse segment endpoint(s), although they are rarely necessary in practice.

### 5.4.3 Modal-Inverse

The Modal-Inverse method restricts changes in the blade airfoil to be a sum of some specified set of geometry modes, which deform the surface normal to its current shape. If the modes are smooth, then the modified airfoil is guaranteed to be smooth as well. Modal-Inverse is quite robust, is particularly useful in flows with shocks and/or separation, where the Mixed-Inverse method might produce an irregular airfoil shape or simply fail due to ill-posedness. On the other hand, Modal-Inverse will match CPwall to CPspec in only a least-squares sense, and is intended for changing the overall airfoil shape, rather than removing small geometric defects.

If a Modal-Inverse or modal optimization case is to be run, then the global variables (20) must be chosen. The corresponding geometry modes and their endpoints must be specified in file `modes.xxx` (described in the Optimization section below). The geometry mode shapes are set up in FUNCTION GMODES (in `src/ises/gmodes.f`), and can be altered as desired. Each mode  $f(\hat{\sigma})$  is defined over the interval  $0..\hat{\sigma}..1$ , with  $\hat{\sigma}$  being the fractional arc length over the target segment.

$$\hat{\sigma} = \frac{\sigma - \sigma_0}{\sigma_1 - \sigma_0}$$

The current modes implemented in SUBROUTINE GMODES consist of modified Tchebyshev polynomials, plus one “tail-wagging” mode for changing the overall blade camber. It is essential that the target segment extend to the trailing edge (i.e.  $\sigma_1 = 1$ ) if the “tail-wagging” mode is used. The modes implemented in SUBROUTINE GMODES can be plotted in **EDP** with option 10.

It must be mentioned that the least-squares pressure-matching condition cannot be fully linearized, with the result that the convergence rate will be somewhat more sluggish than usual. Nevertheless, only a few more iterations relative to a usual analysis or Mixed-Inverse case will be required. On the other hand, a viscous Modal-Inverse case can typically handle limited separated regions within the target segment. It is not necessary to temporarily “freeze” the boundary layers as with Mixed-Inverse.

In practice, it is rarely necessary to converge a Mixed-Inverse or a Modal-Inverse case down to the usual analysis convergence tolerance. If the iterations are halted before full convergence is reached, then the new blade shape is still quite usable — it might have been changed 95% of the way towards the “correct” geometry rather than the 99.999% obtainable with full convergence. This slight difference is irrelevant in actual iterative design applications, since the “correct” geometry rarely turns out to be exactly what is required.

#### 5.4.4 Parametric-Inverse

The Parametric-Inverse method is conceptually the same as Modal-Inverse, and differs only in the manner in which the geometry changes are handled. With each update of the user-defined geometry parameters, the entire blade shape is nonlinearly recomputed in **ISES** by calling SUBROUTINE BLDGEN. This allows MISES to be used as an inverse-design engine with *any* geometry-definition system. This flexibility of the inverse-design system can be greatly enhanced by user-supplied geometric constraints coded in SUBROUTINE BPCON, as described earlier.

#### 5.4.5 Blade Translation, Scaling, Rotation

The blade rigid-body movement and scaling variables (31-34) are intended to complement the Modal-Inverse formulation, which can only change the shape of the blade. Scaling variable SCAL (33), for example, can be used to grow or shrink the blade. Since the pitch remains fixed, this in effect modifies the cascade's solidity. For single-blade cases, the blade  $m'$ -translation variable MOVX (31) is meaningful only if the radius and streamtube thickness distributions  $r(m')$ ,  $b(m')$  are nonuniform, and the  $\theta$ -translation variable MOVY (32) has no effect. For multi-blade cases, these variables can be used to move the blades relative to one another.

The rigid-body variables (31-34) can also be used to augment the Parametric-Inverse formulation if the user's blade parameterization does not permit overall blade motion or scaling. Conversely, if the user's parameterization *does* contain any such rigid-body mode, then the corresponding global variable (31-34) must not be used, since the system will have redundant variables and be very ill-conditioned.

The rigid-body variables (31-34) can be driven either directly, via their specified values in the `ises.xxx` file, or indirectly, via some other constraint. For example, the blade rotation variable ROTA (34) can be driven by specifying an outlet slope (2) or outlet tangential velocity (10).

The rigid-body variables can also be used for Modal-Inverse or Parametric-Inverse calculations in conjunction with the wall pressures specified in **EDP**. Adding the rotation variable (34) to the usual mode variables (20) can often produce a better fit to these specified wall pressures, since changing the blade shape alone may not be adequate.

#### 5.4.6 Modified-Blade Output

Option 5 writes out the current  $m'$ ,  $\theta$  coordinates in a new `blade.yyy` file, with the user being prompted for the new filename. This can then be used as the input to a new **ISET**, **ISES** run.

Option 15 writes out the current user-defined geometric parameters in a new `bparm.yyy` file. This can likewise be used to start a new **ISET**, **ISES** run.

Option 13 creates a cartesian coordinate file corresponding to the current  $m' - \theta$  coordinates. This also uses the current radius function  $r(m')$  which was specified in the `stream.xxx` file.

The cartesian coordinates  $x, y, z$  are calculated as follows.

$$\begin{aligned}x &= r \cos(\theta + \Delta\theta) \\y &= r \sin(\theta + \Delta\theta) \\z &= \int \frac{dz}{dm'} dm' = \int \pm \sqrt{r^2 - \left(\frac{dr}{dm'}\right)^2} dm'\end{aligned}$$

As before, the wheel axis coincides with the  $z$  axis, and the wheel lies in the  $x - y$  plane. The transformation assumes that  $z(m')$  is monotonic (i.e. the flow doesn't "curl back" along the  $z$  axis). Otherwise, the sign of the root in the integrand for  $z$  would need to be switched after the turn point. The information needed for this decision is not present in the `idat.xxx` files, and the positive root is always assumed. The result may be a mirror image of the desired blade. The integration for  $z$  is performed around the blade contour via the trapezoidal rule. The constant circumferential angle offset  $\Delta\theta$  is requested from the user. This allows all the blades to be generated in turn by incrementing  $\Delta\theta$  by the pitch angle.

#### 5.4.7 ISES Parameter Changes

Options 6, 7, and 14 are simply convenient means to change some of the contents of `idat.xxx` which are not normally accessible via the usual `ises.xxx` input file. In particular, the quasi-3D stream surface geometry information can be altered using option 6, which prompts for a new file `stream.yyy`. This will replace the previous `stream.xxx` contents which were read during **ISET** execution. Note that the altered `idat.xxx` state file must be written out with option 3 to store any new information.

#### 5.4.8 Inverse Design Session

Below is a sample inverse design calculation sequence, starting from the seed case `xxx`. Program executions as well as option selections within **EDP** are shown.

```
% iset xxx
% ises xxx                (usual analysis run)
% edp xxx
  1 Edit Cp distributions
    I nitialize CPspec
    M odify Cp
    B lowup                (optional)
    M odify Cp            (repeat as needed)
    .
    .
    D emark inverse segment
    return
  2 Set design flags (if necessary)
```

```

    3 Write idat.xxx
    0 Quit
% edit ises.xxx           (add 11,12 to input lines 1 and 2)
% ises xxx               (inverse run)
% iplot xxx              (optional)
% edp xxx
    1 Edit Cp distributions
      I nitialize CPspec   (optional)
      M odify Cp
      .
      .
      return
    3 Write idat.xxx
    0 Quit
% ises xxx               (inverse run)
% edp xxx
    5 Write new airfoil coordinate file(if satisfied with design)

```

The sequence above can be repeated as often as needed. The  $C_p$  plot in **EDP** displays both the current surface pressures and the specified pressures. Any difference after convergence is due to the “error” terms which were added to attain closure at the segment endpoints. **IPLLOT** can of course be used anytime to examine the design in more detail.

The example above shows a Mixed-Inverse calculation. A Modal-Inverse calculation would be done by adding 21,22,... to input line 1 in **ises.xxx**, while a Parametric-Inverse calculation would require the addition of 40 to line 1. The latter also does not require the demarkation of the inverse segment.

#### 5.4.9 Parameter-Modification Design Session

Below is a sample design calculation sequence where the geometry parameters  $G(k)$  are repeatedly modified. The idea here is to implement each parameter change by simply reconverging an existing flow solution maintained in **idat.xxx** with **ISES** only, rather than starting from scratch each time with **ISET**. Just reconverging can easily be an order of magnitude faster if only modest parameter changes are made. The sequence assumes that the initial **ISET** setup is done with the starting parameter file **bparm.xxx**, and that **bspec.xxx** is initially the same as **bparm.xxx**.

```

% iset xxx
% ises xxx               (usual analysis run, with 40,40 variable,constraint flags)
% bldset xxx
    MODI  Modify contour shape
    P arameter modification

```

```

Enter k, Gk:
Enter k, Gk:
Enter k, Gk: (repeat as needed)
.
.
Enter k, Gk: 0 (done)
return
PSAV Write blade parameter file: bspec.xxx
QUIT
% ises xxx (reconverge with new parameters in bspec.xxx)
% iplot xxx (optional)
% bldset xxx
MODI Modify contour shape
P arameter modification
etc.

```

The sequence above can be repeated as often as needed. Again, **IPL** can be used anytime. When finished, `bspec.xxx` contains the new blade geometry description.

## 5.5 POLAR

**POLAR** is a driver program for **ISES** which conveniently sweeps through a specified parameter range, thus generating a loss curve, turning curve, etc. Because it takes full advantage of the quadratic convergence of the Newton method, using **POLAR** is more efficient (and much easier!) than running a sequence of independent cases from scratch with **ISES**.

In addition to the usual **ISES** input files, **POLAR** also requires a `spec.xxx` file, which contains the sequence of operating parameters, with an optional `idat.xxx.nn` file save flag for each parameter:

```

KSPEC
SPEC(1) KSAVE(1)
SPEC(2) KSAVE(2)
.
.
.
SPEC(NA) KSAVE(NA)

```

The first line contains the integer `KSPEC`, which indicates what all the `SPEC(i)` values represent. The following allowed values are currently implemented, matching the global-constraint indices `GCON(.)` described earlier.

KSPEC	SPEC(i)
1	SINLin
7	BVR1in
8	BVR2in
9	V1ATin
15	MINLin
16	P1PTin
17	MOUTin
18	P2PTin

For example, running **POLAR** with the following `spec.xxx` file

```

15
0.70  1
0.72
0.74  2

```

is equivalent to running three separate **ISES** cases with `MINLin = 0.70, 0.72, 0.74`, with the particular `MINLin` in `ises.xxx` being ignored. The 1,2 in the second column above instructs **POLAR** to write out the files `idat.xxx_01` and `idat.xxx_02` for the two particular points, so that they can perhaps be examined later.

Regardless of whether any `KSAVE` flags are specified, **POLAR** always overwrites the basic `idat.xxx` file each time it converges a `SPEC(i)` parameter value.

Everytime **POLAR** converges on a point to the tolerances in `EPS.INC`, it appends the integrated parameters to `polar.xxx`, appends the surface pressure and boundary layer variable distributions to `polarx.xxx`, and overwrites `idat.xxx` with the converged solution. If `KSAVE` for that point is nonzero, then the additional state file `idat.xxx_nn` is also written out, as described above.

*Note: The `polarx.xxx` file is not currently written for *MISES 2.5*, since the necessary plotting program is not yet available.*

If **POLAR** fails to converge on any one point, it will restart from the previously-converged point, and subdivide the offending `SPEC(i)` increment. If that point fails, the `SPEC(i)` increment will be subdivided further. If no convergence is achieved after five subdivisions, **POLAR** will terminate with a “Severe convergence problem detected” message. Failure to converge on any one point may be due to massive separation, or a very dramatic jump in a transition location. Occasionally, a limit cycle occurs, with a transition location oscillating back and forth. This is often caused by inadequate resolution near the transition point. Simply restarting **POLAR** may fix the problem, since such a restart will begin after the offending point if the preceding interval was subdivided previously.

If after **POLAR** execution it is found that the parameter sweep is not complete, points can be added to the `spec.xxx` file, and **POLAR** restarted. The new points will be automatically appended to the save and dump files.

The results in `polar.xxx` can be perused directly, or plotted in an organized manner using program **POLPL**. This is entirely menu-driven, and is simply executed with no argument.

```
% polpl
```

The menu options allow more than one polar file to be read into **POLPL**, and also allow any pair of variables to be cross-plotted, e.g.  $M_1$  vs  $p_2/p_{o1}$ ,  $\omega$  vs  $M_2$ , etc. Experimental data can also be overlaid.

Plotting software for the binary `polarx.xxx` dump file is currently not provided. It is expected to be available in future MISES versions.

## 5.6 BLDSET

**BLDSET** is a geometry manipulation program for modifying geometry files `blade.xxx` and/or `bparm.xxx`. It is fully menu-driven and fairly self-explanatory. It can be run with the case-extension argument,

```
% bldset xxx
```

in which case it will first try to read the geometry in one of four ways, in the following order of precedence:

1. geometry parameters from an `idat.xxx` file
2. x,y coordinates from an `idat.xxx` file
3. geometry parameters from a `bparm.xxx` file
4. x,y coordinates from a `blade.xxx` file

Any one of these can also read in via a menu selection at any time. **BLDSET** can also write out a `blade.xxx` or `bparm.xxx`-type file, the latter being possible only if the parameters are available.

A typical use of **BLDSET** in a design session is to prepare a `blade.xxx` or `bparm.xxx` file for an analysis run — shifting the blade, changing pitch, overlaying another blade, etc. A 2-D inviscid panel method is provided to allow a quick sanity check on the geometry.

Another use of **BLDSET** is to read the geometry parameters from `idat.xxx`, modify them somewhat, and then write them out in a new `bspec.xxx` file. These modified parameters can then be easily and rapidly “implemented” in the flow solution simply by reconverging it with **ISES**. This typically requires only a few Newton iterations, and hence is much faster and more reliable than using the new parameters with **ISET** to start a new case.

## 6 Optimization

Optimization capabilities are no longer part of MISES 2.5 itself, but are now implemented in the external interactive design/optimization driver **LINDOP**, adapted from the isolated-airfoil MISES version **MSES**.

To make use of **LINDOP** requires the specifying some number of geometry deformation modes as additional global unknowns (20), and specifying their corresponding fixing constraints (20). **ISES** sets all the geometry mode amplitudes to zero during a calculation, but it will still calculate the sensitivities of various quantities such as  $S_2$ ,  $p_2/p_{o1}$ ,  $\omega$ , etc. to the mode displacements. These sensitivities are written out to the unformatted file `sensx.xxx`, which is then read in by **LINDOP**.

## 7 Graphics

The plot library used by all the MISES programs is Xplot11, (`libPlt.a`), which is aimed at driving X-terminals. A PostScript file can be generated at any time from the plot visible on the screen. The file `plotlib/Doc` contains much more information on this graphics package.

## 8 General Hints

### 8.1 Viscous solutions

When a viscous case is executed with an initial `idat.xxx` file from **ISET**, two Newton iterations will be performed in the inviscid mode before the boundary layers (BL) are initialized and the viscous mode is turned on. This is to allow the inviscid flow to settle down from the initial guessed flowfield (set up in SUBROUTINE RQINIT), so that the BLs start with a better initial guess. In some cases two iterations may not be sufficient, in which case the initial BL solution might be quite bad, greatly increasing the number of Newton iterations required for viscous convergence. Typical examples are shocked and/or choked flows, and cases with strong streamtube contraction and rotation effects on the streamline pattern. For such cases, it may be better to perform more iterations in inviscid mode (with `REYNIN` temporarily set to zero) before proceeding with the viscous calculation. Alternatively, one can alter the viscous-flag definition in PROGRAM ISES (in `src/ises/ises.f`)

```
LVISC = ICOUNT.GT.2 .AND. REYNIN.GT.0.0
```

to increase the number of initial inviscid iterations from the current 2 iterations.

## 8.2 Inverse solutions

Care must be used when running the Mixed-Inverse mode with a viscous case. It is essential that there is no separation or near-separation anywhere within the target segment. Since the surface pressure imposed in Mixed-Inverse is also imposed on the BL, a physically ill-posed problem results if the BL is separated. In practice, wild changes in the geometry will result under the separated region. Usually, the blade shape will fold up and the calculation will collapse. A fairly simple fix to this problem is to temporarily “freeze” the BLs by specifying (`REYNIN = 0`) when the inverse case is converged:

```
1 2 5 15    11 12
1 3 4 17    11 12
.
.
0.0e6    ...    | REYNin
```

The BLs case then be “unfrozen” and the case reconverged in the usual analysis mode:

```
1 2 5 15 !   11 12
1 3 4 17 !   11 12
.
.
1.0e6    ...    | REYNin
```

The resulting viscous  $p(s)$  will change slightly from the specified  $p_{\text{spec}}(s)$  in the inverse calculation, but this is usually minor, and can be iterated if desired. Note that the use of “!” in the first two lines is convenient for quickly adding and removing the global variable and constraint flags.

The Modal-Inverse option generally does not have the problem with ill-posedness in separated flow, and can be usually be safely used in all situations.

The well-posedness of the Parametric-Inverse method depends a great deal on the action of the user-defined parameters. It is important that all the declared parameters be reasonably independent in their influence on the blade shape, and that they all have an aerodynamically significant influence on the surface pressure distribution. One cannot least-squares fit a parameter if that parameter has no influence on the flow! If such a parameter is declared without any constraints, the result will be a nearly-singular least-squares matrix, producing enormous Newton changes and a certain solution failure.

## 8.3 Grid resolution

Compared to most Euler solvers, **ISES** is usually quite insensitive to grid density. It behaves more like a potential solver in this regard, especially if `ISMOM=3` or `ISMOM=4` are used. In

any case, it is a good idea to check that the leading edges are reasonably well resolved, And that no spurious losses (or gains!) are being generated there. The “Streamtube plots” menu of **IPLOT** can be used to plot the stagnation pressure variations along any streamtube. This should be piecewise-constant, with sudden monotonic drops through the shocks.

Any separation bubble present in the flow must be well-resolved. The default grid is usually adequate for most cases, but maybe not if the bubble is close to the leading edge and very small in streamwise extent. Moderate Reynolds numbers (1-3 million, say) require the finest grid, since the bubbles are then still important, but very small. Fortunately, streamwise grid spacing is “cheap”, increasing the solution time only linearly, so it may be simplest to increase the grid point number parameter **N** in **ISET** to 100 or more. Inadequate bubble resolution often results in a “ragged” or “scalloped” loss vs. incidence curve, so this is easy to spot.

## 8.4 Execution times

Using **ISES** or its variants requires substantial computational resources. A single point solution for a typical grid size ( $160 \times 24$ ) will require anywhere between 3 Newton iterations for a subcritical inviscid case, to  $\sim 12$  iterations for shocked transonic or low-Re viscous cases with separated regions. Each iteration represents 2–10 CPU seconds on a RISC workstation. For a given grid size, the offset-periodic inlet grid topology roughly doubles the required time for a solution. But since  $2/3$  as many streamlines are typically required on an offset grid for the same accuracy, the net CPU time required in practice is similar.

When a sequence of solutions is needed, the CPU requirements per solution can drop substantially due to the quadratic convergence property of the Newton method used by **ISES**. Once a converged solution is obtained, convergence to a “nearby” solution after a small inlet angle or inlet Mach (say) change is quite rapid, requiring only 2 or 3 additional iterations. When inverse design calculations are performed, a minor prescribed-pressure change will typically require only about 2 Newton iterations to converge the new geometry, making interactive design even on a low-end RISC workstation effective.

While the Newton solution method is very efficient for converging small parameter tweaks, it intensely dislikes large changes. Trying to reconverge a solution after a drastic parameter change, such as going from  $S_1 = 0.5$  to  $S_1 = 1.0$ , is definitely not a good idea if a transonic or supersonic viscous case is involved. Clearly, runs should be sequenced so that such large changes are avoided.

# MISES Roadmap

