Jetstream Region Design Variable User Guide

Thomas A. Reist
tom.reist@utoronto.ca

August 1, 2013

Contents

Nomenclature

x, y, z	Stream-wise, span-wise and vertical coordinates in global space
$\bar{x}$, $\bar{y}$, $\bar{z}$	Stream-wise, span-wise and vertical coordinates in local region space
$\mathbf{x}$, $\bar{\mathbf{x}}$	Column three-vector of global and local coordinates, respectively.
j, k	Stream-wise and span-wise coordinate indices
c	Chord
b	Span
S	Projected area
V	Volume
Λ	Sweep angle
Γ	Dihedral angle
W	Weight
RDV	Region design variable

CP	B-spline control point
OML	Outer mold line
LE	Leading edge
TE	Trailing edge
LS	Lower surface
US	Upper surface

1 Introduction

A set of design variables have been implemented in Jetstream which aim to be able to represent a relatively wide range of geometries and optimization problems with little user interaction. These variables are termed Region design variables (RDVs) for reasons discussed later. This document shall outline some of the basics of their implementation as well as their use. The majority of routines relating to the implementation of RDVs are contained in the Jetstream directory under `optimizer/Region_Mod.f90`. Note that, due to the how portions of the code are written in order to have a clean code, `Region_Mod.f90` can be slow to compile. Since not everyone needs the RDV routines, a flag in the main Jetstream `Makefile` called `rgnOpt` determines whether this module is compiled or not. Setting `rgnOpt=on` will compile the full `Region_Mod.f90`, while setting `rgnOpt=off` will only compile a few dummy routines at the end of `Region_Mod.f90` and thus greatly speed up compilation. Thus, `rgnOpt` should be left 'off' so those who do not need RDVs do not waste time compiling the RDV routines.

Much of this manual is quite brief and many details are eliminated so if you have any questions, or would like any features added, you can reach me (Tom) at the email address above.

2 Implementation

This section shall briefly describe some of the inner workings of the RDV code. The basic idea behind the use of RDVs is the use of 'components' and 'regions'.[1] Much of the extraction, processing, interaction with the optimizer, etc. is hidden from the user. The aim of this code is that the user can perform a range of optimization with the use of a simple set of common input files, without the need to develop their own design variables and/or constraints for different problems and blocking topologies. Thus, the following subsections shall outline the basics of each concept.

2.1 Regions

The basic idea behind RDVs is the definition of a 'region'. Based on user input (described in Section 3), a given number of patches are assigned to each region in a (j, k) grid. A sample

[1] Regions were part of the original development, with components coming later. Hence the name region design variables.

wing-body geometry illustrating said patch layout is shown in Figure 1 with components, regions, joints, patches and stitches identified. For RDVs to be used, the following conditions on patch topology are required:

- The chord-wise and span-wise dimensions are x and y, respectively.
- Patches must be aligned in a (j, k) sense, with no holes, etc. and full connectivity between included patches.
- There must be some sense of a lower and upper surface on each region.
- Each region has the same number of patches in the chord-wise direction.
- Each patch has the same number of CPs in each direction.[2]

In addition to regions, structures termed 'joints' are also created. A joint consists of the CPs on a stitch between two regions as well as one k line of CPs inboard and outboard of that stitch, as shown in Figure 1. Joints allow a way of maintaining G^1 continuity between patches automatically. Note that on the root region, the CPs corresponding to $k = 1$ and $k = 2$ are coincident. Those CPs which are interior to the regions and fall on stitches are not included as region variables, but rather are determined by a combination of their neighbours such that G^1 continuity on all stitches is maintained.

The combination of patches into regions allows for the use of arbitrarily fine block topologies with a single set of design variable definitions and input files. After the RDVs are applied to the CPs on the regions, these are mapped back to the patches at the end of a function call.

Each region is defined in its own coordinate system which allows the extraction of intuitive design variables for each region, regardless if the different regions have significantly different orientations in global-space. A sample of these local coordinate systems is shown in Figure 2 for a three-region case. In this local coordinate system, variables are automatically extracted and defined as shown in Figure 3. For each region, those variables which are free (based on user input) appear in the vector of design variables.

Upon specification of these local variables, the global CPs of region R, can be constructed via

$$\mathbf{x}_R = \sum_{r=1}^{R-1} \begin{bmatrix} \bar{x}_{\text{LE}}^{\text{tip}} \\ \bar{b}\cos\Gamma \\ \bar{b}\sin\Gamma \end{bmatrix}_r + \phi(\Gamma_R)\ \bar{\mathbf{x}}_R \tag{1}$$

where Γ is the dihedral variable for each region[3] and ϕ is the rotation matrix about the $\bar{x}$-axis given by

$$\phi(\Gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Gamma & -\sin\Gamma \\ 0 & \sin\Gamma & \cos\Gamma \end{bmatrix}$$

[2]This could easily be changes, however during initial development this condition was required by other portions of Jetstream so did not form a constraint.

[3]This is the only region variable which is expressed in the global coordinate system.

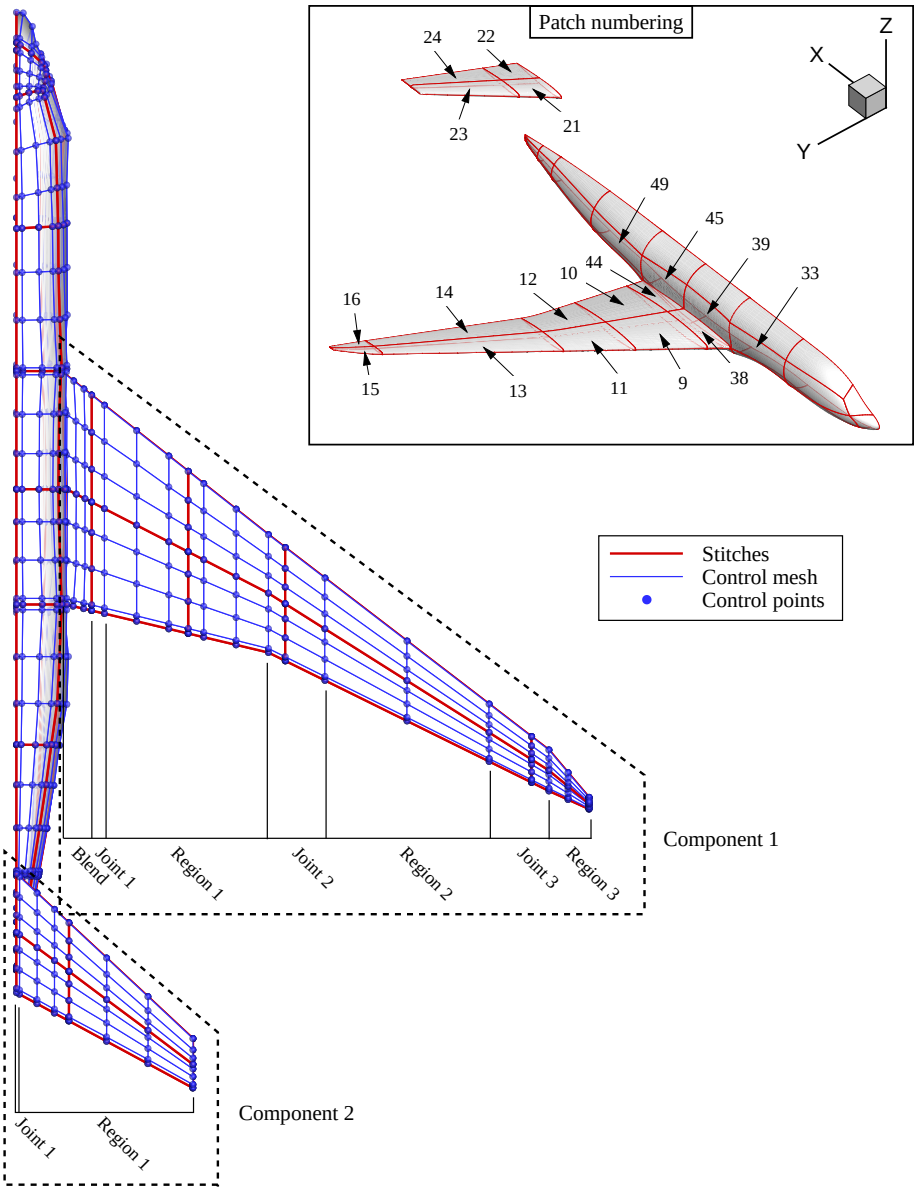

Figure 1: Sample wing-body geometry with patches, stitches, components, regions and joints identified.

This is an overview of the definition of RDVs and how they are used in the local region-space and rebuilt in the global space. Many details, such as implementation of joints, handling of stitch CPs, section angles, etc. are quite involved and thus not described here.

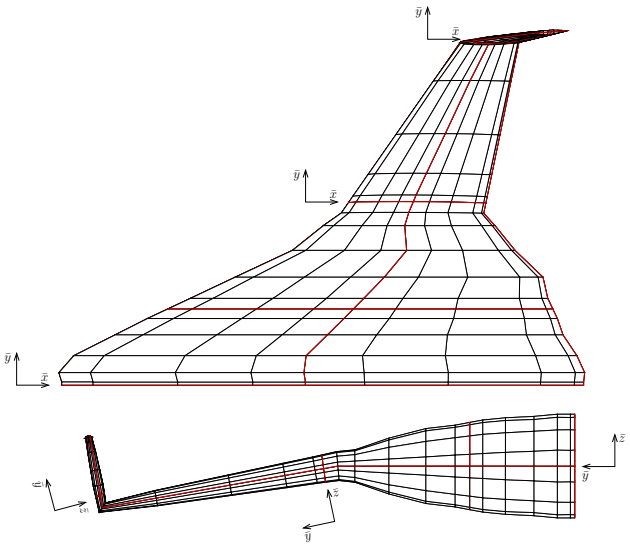

Figure 2: Top and front view of a sample BWB three-region geometry showing local coordinate systems.

2.2 Components

In order to represent more complex geometries, e.g. those with multiple surfaces, the concept of 'components' is introduced. A component is simply a collection of regions. Each component is isolated from other component and contains its own unique set of regions. Components may not intersect or share patches. This is intended, for example, to allow for the optimization of full aircraft configurations, where say, the wing is one component and the horizontal tail is another, and each can be controlled independently based on the role of that component. Such an example is shown for a two-component geometry in Figure 1. The global space referenced to in the previous section is actually the local space of a component. Components can translate and rotate about the span-wise axis, and in doing so, all the regions which are included on that component translate and/or rotate accordingly. Specification of translation and/or rotation are described in Section 3.1. Components also allow for the interaction of patches which are included in the design space (e.g. a wing) and those which are not to be controlled by the optimizer (e.g. a fuselage). Specification of how a component is 'mounted' is described in Section 3.1. If a component is mounted on a symmetry plane no special treatment is needed. If a component is mounted on another set of patches (e.g. as in the wing-fuselage case) the CPs on the mount (e.g. fuselage) are controlled via algebraic movement based on the movement of the CPs on the root of the component via

$$\mathbf{x}_k = \mathbf{x}_k^{\text{orig}} + \Delta\mathbf{x}_{\text{cmp}} \left[\frac{1 + \cos(\pi S_k)}{2} \right]^{\beta} \tag{2}$$

for each control mesh grid line k normal to the component root section. Where $\mathbf{x}_k$ is the new 3-vector of x, y, z CP coordinates on the mount surface, $\mathbf{x}_k^{\text{orig}}$ are the original CPs on the

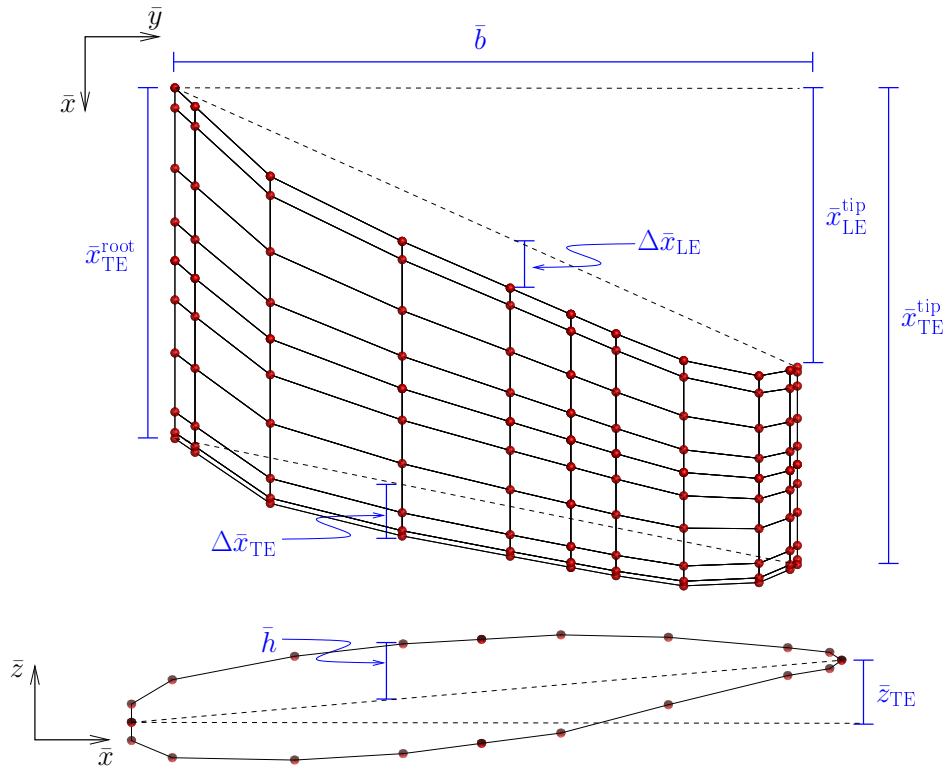

Figure 3: A region in its local coordinate system with RDVs shown. All dimensions marked in blue with an over-bar correspond to design variables for this region. Note that not all design variables are shown in this figure. Namely, a $\Delta\bar{z}_{\mathrm{LE/TE}}$ is also present which corresponds with the marked $\Delta\bar{x}_{\mathrm{LE/TE}}$.

mount surface, and $\Delta\mathbf{x}_{\mathrm{cmp}}$ is the movement of the component root relative to the original shape. S_k is the arc length along the grid line given by

$$S_k = \frac{\sum_{i=2}^{k} ||\mathbf{x}_i - \mathbf{x}_{i-1}||_2}{\sum_{i=2}^{k_{\max}} ||\mathbf{x}_i - \mathbf{x}_{i-1}||_2} \tag{3}$$

and β is a factor which allows control over how far along the grid line the geometry change at the component root is propagated. Large values of β better maintain the initial mount geometry, however if large changes of the component root occur the resulting mount geometry may be severely warped. The default value is $\beta = 4$.

2.3 Code Setup

Outlined here are some of the basics of how the regions and components are treated in the code. Extensive use is made of user-defined types, with a different type specified for components, regions, joints, weights, blend, etc. Component types (`cmp(1:Ncmp)`) contain all of the features which belong to that component, for example components and joints.

Region types (`region(1:2,1:Nrgn)`) contain all of the information associated with that region. The first subscript is the surface number, with 1 being the LS and 2 being the US. The second subscript is the region number. This is the same for the joint types. These are the most important types, and information about the others can be found by looking at the code. Each of these types can also belong to another type of which it is a subset. For example, a problem with `Ncmp` components, each with `Nrgn` regions and `Nwgt` weights on each regions would be constructed as `cmp(1:Ncmp)%region(1:2,1:Nrgn)%weight(1:Nwgt)`

The vector `Xrgn(1:nrgndv)` contains all of the region variables, however not all of these variables are design variables. An intuitive map of the region design variables to `Xrgn` is given in `region(:,:)%indx(1:jmax,1:kmax,1:3)` as explained in the subroutine `initRegionVars`. Those region variables which are design variables (and hence contained in `X(1:ndv)`) are marked by `XrgnDOF(1:nrgndv) = 1` and the total number is contained in `nrgndof`. The map between `X` and `Xrgn` and `Xcmp` is given by the contents of `XrgnID(:)`. The vector `Xcmp(1:ncmpdv)` contains all of the component variables, which are all design variables. Thus, the total number of geometric design variables are given by
`ngeodv = nrgndof + ncmpdv`.

3 Input Files

The following section shall briefly describe the input files required to use the RDVs. In addition to setting `jtstrm% region_dvs = .true.` in the main input file, component and region definition files are required. The component definition file is specified via
`component_file_prefix = <<string>>` in the main input file for a file with the suffix `.cmp`. The region definition files are defined there.

Sample component and region definition files are provided below. These files correspond to the geometry in Figure 1. The patch numbers on the US are identified in this figure, and correspond to the patch numbers listed in the input file. The specification of patch numbers on the LSs follows in a similar manner, and should be evident based on the US example given.

3.1 The Component Input File

This file defines each component, and in turn reads the region definition file which defines the regions on a component. A sample of a two-component file for the wing-body optimization problem in Figure 1 is shown below.

```
 Component definitions for jetstream
-------------------------------------------
 Number of components
        2

Component 1
 ------------------------------------------------------------
 | Variable / Constraint | T/F | Dim |  Minimum  |  Maximum  |
```

```
 | Region geometry file  |  -  |  -  |         wing
 | Region weight file    |  0  |  -  |         wing
 | Internal shape file   |  0  |  -  |         cabin
 | Base type             |  -  |  -  |           2
 | Base LS patches       |  -  |  -  |     31  36  42  47
 | Base US patches       |  -  |  -  |     33  39  45  49
 | Junction LS patches   |  -  |  -  |         35  41
 | Junction US patches   |  -  |  -  |         38  44
 | Junction X deviations |  2  | -D- |    -0.01       +0.05
 | Junction Y deviations |  1  | -D- |    -0.05       +0.00
 | Junction Z deviations |  3  | -D- |    -0.01       +0.05
 | Root point  (x/c)     |  -  |  -  |          0.50
 | Root X movement       |  0  | -D- |    -0.02       +0.02
 | Root Y movement       |  0  | -D- |    -0.02       +0.02
 | Root Z movement       |  0  | -D- |    -0.01       +0.01
 | Root pivot            |  1  | -D- |    -5.00       +5.00
 | Span constraint       |  1  |  S  |    -0.25       +0.25
 | Area constraint       |  1  |  D  |    -0.00       +0.00
 | Volume constraint     |  1  |  D  |    -0.00       +0.00
 | Tip continuity        |  -  |  -  |           1

Component 2
 ------------------------------------------------------------
 | Variable / Constraint | T/F | Dim |  Minimum  |  Maximum  |
 | Region geometry file  |  -  |  -  |         tail
 | Region weight file    |  0  |  -  |         tail
 | Internal shape file   |  0  |  -  |           -
 | Base type             |  -  |  -  |           0
 | Base LS patches       |  -  |  -  |
 | Base US patches       |  -  |  -  |
 | Junction LS patches   |  -  |  -  |
 | Junction US patches   |  -  |  -  |
 | Junction X deviations |  0  | -D- |
 | Junction Y deviations |  0  | -D- |
 | Junction Z deviations |  0  | -D- |
 | Root point  (x/c)     |  -  |  -  |          0.50
 | Root X movement       |  0  | -D- |    -0.05       +0.05
 | Root Y movement       |  0  | -D- |    -0.00       +0.00
 | Root Z movement       |  0  | -D- |    -0.10       +0.10
 | Root pivot            |  1  | -D- |   -10.00      +10.00
 | Span constraint       |  0  |  S  |    -0.25       +0.25
 | Area constraint       |  1  |  D  |    -0.00       +0.00
 | Volume constraint     |  0  |  D  |    -0.00       +0.00
 | Tip continuity        |  -  |  -  |           1
 ------------------------------------------------
NOTES: This area can be used for descriptions, etc.
```

Follows is a description of each option in the component definition file. The number of components are defined first such that all space can be allocated prior to reading. The 'Variable / Constraint' column is simply the name of the option and may not correspond

to a variable or constraint. The 'T/F' column is a binary indicating if the option is used, variable is free, constraint is active, etc. (1 = true, 0 = false). The 'Dim' column consists of a letter code which indicates how the constraint bounds are defined. The options consist of:

A: The bounds are given in absolute units.

$$f_{\text{low}} < f < f_{\text{upp}}$$

D: The bounds are given as a difference, in the relevant units, from the initial value.

$$f_0 + f_{\text{low}} < f < f_0 + f_{\text{upp}}$$

S: The bounds are given as a fraction of change of their initial value.

$$f_0 + |f_0| f_{\text{low}} < f < f_0 + |f_0| f_{\text{upp}}$$

Several of the 'Dim' columns in the sample files are marked with `-A/D/S-`. This is just used here as a reminder to the user that these variables/constraints can only use the type of 'Dim' indicated between the dashes and may not be changed by the user. Different variables/constraints have different 'natural' units, which are noted in the descriptions below. These dimension types also apply to region definitions as described in the following subsection. The final 'Minimum' and 'Maximum' columns indicate either the variable or constraint bounds or simply the value of the input. Subsequently, each component has the following options:

Region geometry file: The string of the region definition file name prefix, which has the suffix `.rgn`.

Region weight file: The string of the region low-fidelity weight model definition file prefix which has the suffix `.wgt`. `T/F = 0,1` indicates the absence or inclusion of a weight model for this component. If `T/F = 0` no file name need be given.

Internal shape file: A file which defines a surface which the OML cannot encroach upon during optimization. See the code for required format. `T/F = 0,1` indicates the absence or inclusion of this internal shape constraint for this component. If `T/F = 1` a file name prefix must be given which has the `.shp` suffix. If `T/F = 0` no file name need be given.

Base type: Indicates how this component is 'mounted'. A value of 0 indicates it is mounted on the symmetry plane, while a value greater than 0 indicates it is mounted on a set of patches (e.g. a wing mounted on a fuselage). A value of 1 indicates it is mounted on a set of patches and the intersection is controlled by including a 'blender' region which projects the root of the first region to the base patches.

Base LS/US patches: A list of patches on the LS and US of the mount (e.g. fuselage) in increasing stream-wise order, which serve as the base for this component. These patches include those which are above/below the component as well as one patch upstream, and one downstream of the component. Thus, the length of this list should be two more than the number of chord-wise patches on the component's regions. If `Base Type = 0` then no list need be given.

Junction LS/US patches: A lift of patches which form the 'blender' for `Base type = 1`. By definition, the 'blender' consists of only one patch in the span-wise direction, and the same number of patches in the stream-wise direction as on the component. The patches in this list are ordered in increasing stream-wise order. If `Base Type = 0` then no list need be given.

Junction X deviations: The amount by which the junction LE and TE can change in the x direction from a linear projection of the first region root. The `T/F` column may take on three different values: `T/F = 0` indicates that the LE and TE are not variable and will simply be a linear projection of the root of the first region. `T/F = 1,2` indicates that the CPs on the LE and TE at the wing-body junction can move in the x direction and the LE and TE on the interior of the junction are determined by an interpolation of the form

$$\Delta x(y) = \Delta x(y = 0)(1 - y/b)^{\alpha}$$

where $\Delta x(y)$ is the x deviation at a given junction span location, α is the value of the `T/F` input, b is the span of the junction, and y is the span-wise coordinate which is increasingly positive along the span. This essentially allows the chord to vary either linearly or quadratically between the root of the first region and the wing-body junction. `Minimum` and `Maximum` need only be given if `T/F > 0`. Measured in units of original junction span.

Junction Y deviations: This is the amount by which the span of the junction can change. The `T/F` input can be either 0 for no span variation, or 1 for span variation. `Minimum` and `Maximum` need only be given if `T/F > 0`. Measured in units of original junction span.

Junction Z deviations: The amount by which the junction LS and US can change in the z direction from a linear projection of the first region root. This effectively allows section changes on the junction. If `T/F = 0` then the sections on the junction are a linear loft of the root section on the first region. As for the junction x deviations, if `T/F = 1,2` then the section at the root of the junction (i.e. wing-body intersection) is also free and the sections on the interior of the junction are interpolated by

$$\Delta z(x, y) = \Delta z(x, y = 0)(1 - y/b)^{\alpha}$$

where $\Delta z(x, y)$ is the z deviation at a given (x, y) location on the junction, α is the value of the `T/F` input. The option `T/F = 3` allows each Δz (section shape) over the span of the junction to be free. Note that currently there are no thickness constraints applied to the junction, so small lower bounds should be used. `Minimum` and `Maximum` need only be given if `T/F > 0`. Measured in units of original junction span.

Root point: This is the point, as a fraction of the root chord, about which the movement and or pivoting of the component occurs.

Root X/Y/Z movement: The x, y, z deviations, in grid units, by which the component can move from its initial position. If `T/F = 0` for any direction component, than there is no design variable and the component base is fixed in this dimension. Currently, only dimension type 'D' can be used for this variable. Measured in grid units.

Root pivot: The allowable change, in degrees, by which the component can pivot about its original angle. This is intended, for example, to allow for a horizontal tail for trimming. If `T/F = 0` then the component cannot pivot. If `T/F > 0` then the component can pivot. The value of `T/F` indicates for which operating points, in a multipoint optimization, the component can pivot. For example, if `T/F = 5` then the component could pivot for the first 5 operating points in a multipoint optimization. Measured in degrees.

Span constraint: A constraint on the total span of this component projected in the $x - y$ plane, if desired. If `T/F = 0` then this constraint is not active. If `T/F = 1` then this constraint is active with the lower and upper bounds specified. If a bound is set to be $<= 0.0$ then the initial span is set as this bound. Thus, by setting both minimum and maximum bounds to 0.0, the original span shall be maintained. Measured in grid units.

Area constraint: A projected area constraint of this component, if desired. If `T/F = 0` then this constraint is not active. If `T/F = 1` then this constraint is active with the lower and upper bounds specified. If a bound is set to be $<= 0.0$ then the initial area is set as this bound. Thus, by setting both minimum and maximum bounds to 0.0, the original area shall be maintained. Measured in grid units.

Volume constraint: A volume constraint of this component, if desired. If `T/F = 0` then this constraint is not active. If `T/F = 1` then this constraint is active with the lower and upper bounds specified. If a bound is set to be $<= 0.0$ then the initial volume is set as this bound. Thus, by setting both minimum and maximum bounds to 0.0, the original volume shall be maintained. Measured in grid units.

Tip continuity: Allows the enforcement of either G^0 or G^1 continuity at the component tip. I.e. a sharp or smooth wing tip.

3.2 The Region Input File

Each component is defined by one or more regions. All of the regions which belong to a component are defined by this file. A sample of a three-region file for the wing-body optimization problem in Figure 1 is shown below.

```
 Region definition file
-------------------------------------------
 Number of regions
        3

 ------------------------------------------------
 | Region | Chordwise patches | Spanwise patches |
   1        2                   2
   2        2                   1
   3        2                   1

 Patch layout

 ---------------------------------------
 | Region | LS/US | Patch (j,k) layout |
```

```
     1        1       1  3
              1       2  4
              2       9 11
              2      10 12
     2        1       5
              1       6
              2      13
              2      14
     3        1       7
              1       8
              2      15
              2      16

Region 1
 ------------------------------------------------------------
 | Contraint Limit   | DOF | Dim |  Minimum  |  Maximum  |
 | Span              |  1  |  S  |    -0.25       +0.25
 | Root chord        |  1  |  S  |    -0.20       +0.20
 | Chord             |  1  |  S  |    -0.20       +0.20
 | Taper             |  1  |  A  |     0.10        1.00
 | Dihedral          |  0  |  A  |    -0.00       +0.00
 | Root twist        |  1  |  A  |    -5.00       +5.00
 | Twist             |  1  |  A  |    -5.00       +5.00
 | Sweep x/c axis    |  -  |  -  |          0.25
 | Sweep limits      |  1  |  D  |    -5.00       +5.00
 | LE linear x dev.  |  0  |  A  |    -0.00       +0.00
 | LE linear z dev.  |  0  |  A  |    -0.00       +0.00
 | TE linear x dev.  |  0  |  A  |    -0.00       +0.00
 | TE linear z dev.  |  0  |  A  |    -0.00       +0.00
 | Section height    |  1  |  D  |    -0.05       +0.05
 | Section thickness |  1  |  S  |    -0.25       +0.50
 | Loft sections     |  1  |  -  |
 | Area constraint   |  0  |  S  |    -0.00       +0.00
 | Volume constraint |  0  |  S  |    -0.00       +0.00
 | Match joint LE x  | +1  |  -  |
 | Match joint LE z  | +1  |  -  |
 | Match joint TE x  | +1  |  -  |
 | Match joint TE z  | +1  |  -  |
 | Flap model        |  1  | -A- |   -30.00      +30.00
 |  -> flap span     |  -  |  -  |     0.00        1.00
 |  -> flap chord    |  -  |  -  |          0.15

Region 2
 ------------------------------------------------------------
 | Contraint Limit   | DOF | Dim |  Minimum  |  Maximum  |
 | Span              |  1  |  S  |    -0.25       +0.25
 | Root chord        |  1  |  S  |    -0.20       +0.20
 | Chord             |  1  |  S  |    -0.20       +0.20
 | Taper             |  1  |  A  |     0.10        1.00
 | Dihedral          |  1  |  A  |    -5.00       +5.00
 | Root twist        |  1  |  A  |    -5.00       +5.00
```

```
 | Twist             |  1  |  A  |    -5.00       +5.00
 | Sweep x/c axis    |  -  |  -  |          0.25
 | Sweep limits      |  1  |  D  |    -5.00       +5.00
 | LE linear x dev.  |  1  |  A  |    -0.00       +0.00
 | LE linear z dev.  |  1  |  A  |    -0.00       +0.00
 | TE linear x dev.  |  1  |  A  |    -0.00       +0.00
 | TE linear z dev.  |  1  |  A  |    -0.00       +0.00
 | Section height    |  1  |  D  |    -0.05       +0.05
 | Section thickness |  1  |  S  |    -0.25       +0.50
 | Loft sections     |  1  |  -  |
 | Area constraint   |  0  |  S  |    -0.00       +0.00
 | Volume constraint |  0  |  S  |    -0.00       +0.00
 | Match joint LE x  | +1  |  -  |
 | Match joint LE z  | +1  |  -  |
 | Match joint TE x  | +1  |  -  |
 | Match joint TE z  | +1  |  -  |
 | Flap model        |  0  | -A- |   -30.00      +30.00
 |  -> flap span     |  -  |  -  |     0.00        1.00
 |  -> flap chord    |  -  |  -  |          0.15

Region 3
 ---------------------------------------------------------
 | Contraint Limit   | DOF | Dim |  Minimum  |  Maximum  |
 | Span              |  1  |  S  |    +0.50       +1.00
 | Root chord        |  1  |  S  |    -0.20       +0.20
 | Chord             |  1  |  S  |    -0.20       +0.20
 | Taper             |  1  |  A  |     0.10        1.00
 | Dihedral          |  1  |  A  |   +80.00      +80.00
 | Root twist        |  1  |  A  |    -5.00       +5.00
 | Twist             |  1  |  A  |    -5.00       +5.00
 | Sweep x/c axis    |  -  |  -  |          0.25
 | Sweep limits      |  1  |  D  |   -15.00      +15.00
 | LE linear x dev.  |  0  |  A  |    -0.00       +0.00
 | LE linear z dev.  |  0  |  A  |    -0.00       +0.00
 | TE linear x dev.  |  0  |  A  |    -0.00       +0.00
 | TE linear z dev.  |  0  |  A  |    -0.00       +0.00
 | Section height    |  1  |  D  |    -0.05       +0.05
 | Section thickness |  1  |  S  |    -0.25       +0.50
 | Loft sections     |  1  |  -  |
 | Area constraint   |  0  |  S  |    -0.00       +0.00
 | Volume constraint |  0  |  S  |    -0.00       +0.00
 | Match joint LE x  | -1  |  -  |
 | Match joint LE z  | -1  |  -  |
 | Match joint TE x  | -1  |  -  |
 | Match joint TE z  | -1  |  -  |
 | Flap model        |  1  | -A- |   -10.00      +15.00
 |  -> flap span     |  -  |  -  |     0.50        0.95
 |  -> flap chord    |  -  |  -  |          0.10

---------------------------------------------------
NOTES: This area can be used for descriptions, etc.
```

First, the number of regions contained in this file is given. Then, the arrangement of patches which define each region are given by the number of patches on either the LS or US in each direction. Remember that the LS and US must have the same number of patches. Following this, the arrangement of patches is described by the 'Patch layout', where the patches which belong to the LS and US are listed in (j, k) format as viewed from the top of the aircraft. In general, the 'Constraint' column indicates which variable/constraint is free. Some of these values are defined by design variables, while others are enforced via linear or nonlinear constraints. The 'DOF' column indicates, via a binary, whether each variable/constraint is free or fixed. The 'Minimum' and 'Maximum' column shows the variable/constraint bounds, as well as other values when applicable. Each input is defined as follows:

Span: The span of each region in local region-space (not including its joint), and its bounds, in grid units. If `DOF = 0` then the initial span is maintained. Measured in grid units.

Root chord: Whether the root chord is free or not, and its bounds, in grid units. If `DOF = 0` then the initial root chord is maintained. Measured in grid units.

Chord: Whether the chords at non-root sections are free or not, and their bounds, in grid units. If `DOF = 0` then the initial chord is maintained. Measured in grid units.

Taper: Whether the taper constraint is active, and its bounds. If `DOF = 0` then the initial taper ratio is maintained.

Dihedral: Whether the dihedral is free or not, and its bounds, in degrees. If `DOF = 0` then the initial dihedral angle is maintained. Measured in degrees.

Root twist: Whether the root twist constraint is active, and its bounds, in degrees. If `DOF = 0` then the initial root twist is maintained. Measured in degrees.

Twist: Whether the twist at non-root sections is free or not, and its bounds, in degrees. If `DOF = 0` then the initial twist is maintained. Measured in degrees.

Sweep x/c axis: The axis, as a fraction of the local chord, about which the sweep angle is measured.

Sweep limits: Whether the sweep constraint is active, and its bounds, in degrees. If `DOF = 0` then the initial sweep is maintained. Measured in degrees.

LE/TE linear X/Z dev.: The x, z deviations, in percent of region span, by which the LE and TE may vary from a straight edge, and their bounds in percent of region span. If `DOF = 0` then the initial deviations are maintained. Measured in fraction of region span.

Section height: The section heights are defined as the distance from the CPs on the surface to the chordline in local region-space. This defines if, and how much they can move, in percent of local chord, from their initial height. If `DOF = 0` then the initial section shapes are maintained. Measured in fraction of local chord.

Section thickness: The thickness is based on the distance between corresponding CPs on the LS and US at a given chord location. This defines if, and how much the thickness can change. The bounds are given as a fraction of the initial thickness. If `DOF = 0` then the initial section thicknesses are maintained. Measured in fraction of local chord.

Loft sections: This linearly interpolates the sections on this region between the root and

tip sections if activated. No bounds are used for this constraint.

Area constraint: Whether the projected area constraint for this region is active and its bounds. If a bound is set to be <= 0.0 then the initial area is set as this bound. Thus, by setting both minimum and maximum bounds to 0.0, the original area shall be maintained. Measured in grid units.

Volume constraint: Whether the volume constraint for this region is active and its bounds. If a bound is set to be <= 0.0 then the initial volume is set as this bound. Thus, by setting both minimum and maximum bounds to 0.0, the original volume shall be maintained. Measured in grid units.

Match joint LE/TE X/Z: This effectively makes each joint an extension of a given region. If `DOF = 0` then the joints are free and their sweep, chords and dihedral are independent. This is not recommended if planform changes are allowed on regions as it adds more design variables and can lead to a large amount of vary localized geometry changes which are 'unrealistic'[4]. Note that on the first region `DOF` must not be 0. Setting `DOF = +1` means that the joint on this region will become an extension of the region outboard of it. If `DOF = -1` this joint will become an extension of the previous region. However, on the first region, if `DOF = -1` then the root joint will be normal to the symmetry plane.

Flap model: A simple control effector (i.e. flap) model is implemented which allows to deflect the TE of a portion of a region to mimic a continuous-mold line flap. Specifying `T/F = 1` activates the flap model, for which bounds on deflection angle are required. The sub-option `flap span` specifies the start and end of the flap in the span-wise direction as fractions of the region span. Sub-option `flap chord` specifies the chord of the flap as a fraction of the region root chord. Thus, the flap has an approximately constant chord along the region at the start of the optimization, but to prevent unrealistic flap geometries resulting from large changes in taper ratio the flap chord is a constant fraction of the local chord for the remainder of the optimization. Deflections are measured in degrees, with positive deflections giving a TE downward deflection.

3.3 The Weight Model Input File

A low-fidelity model which correlates geometric changes with movement of the center of gravity (CG) is implemented when using RDVs. In essence, weights of primary structures, systems, fuel, etc. are modeled as point masses which move and/or scale with region movement. A sample weight input file for a two region case is shown below.

```
1  Region weight definitions for jetstream
2 -------------------------------------------
3  Number of regions
4         2
5  ------------------------------
6  | Region | Number of weights |
```

[4]It has also not been tested as thoroughly as the use of matching

```
   1                5
   2                3

Region 1                                  ---------------------------------------
 ---------------------------------------|             Scaling powers            |
| Type | Weight |  x/c   |   y/b   | Area | Span | Sweep | Volume | Weight |
    0     0.260   -0.01      0.07    0.00   0.00    0.00     0.00     0.00   - Payload
    0     0.160   -0.01      0.07    0.00   0.00    0.00     0.00     0.00   - Structure
    1     0.049    0.50      0.50    1.00   0.00    0.00     0.00     0.00   - Structure
    1     0.134    0.85      0.00    0.00   0.00    0.00     0.00     0.00   - Engines
    1     0.117    0.50      0.50    0.00   0.00    0.00     0.00     0.00   - Fixed

Region 2                                  ---------------------------------------
 ---------------------------------------|             Scaling powers            |
| Type | Weight |  x/c   |   y/b   | Area | Span | Sweep | Volume | Weight |
    1     0.030    0.40      0.50    0.55   1.00   -1.00     0.40     0.00   - Structure
    1     0.132    0.40      0.40    0.00   0.00    0.00     0.00     0.00   - Fuel
    1     0.100    0.50      0.50    0.00   0.00    0.00     0.00     0.00   - Fixed
```

The number of regions is specified (which must correspond to the component this weight file belongs to) and the number of weights on each region are specified. Then, for each region the weights are listed. There are currently two weight 'types'. `Type 0` is a weight which does not move with the geometry. In this case, the `x/c` and `y/b` are absolute grid coordinates, not fractions or chord and span respectively. `Type 1` is a weight which moves with a region at a constant `x/c` and `y/b` location via a bilinear mapping between the four corners of the region. The `Weight` column indicates the initial weight of each weight source. Each weight, i, can scale via the following relation:

$$W_i = W_{0_i} \left(\frac{S}{S_0}\right)_i^{\beta_S} \left(\frac{b}{b_0}\right)_i^{\beta_b} \left(\frac{\Lambda}{\Lambda_0}\right)_i^{\beta_\Lambda} \left(\frac{V}{V_0}\right)_i^{\beta_V} \left(\frac{W}{W_0}\right)_i^{\beta_W} \tag{4}$$

where S, b, Λ, and V are the area, span, sweep angle and volume of each region associated with weight W_i, and the subscript 0 indicates the value of the initial design. The values of β are given by the `Scaling powers` in the weight file. Currently, scaling with respect to the weight (i.e. the last term in Eqn. 4) is not implemented. The labels at the end of each line of the sample weight file are not required, and are only included here for illustrative purposes as examples of components and their scaling powers. Each component for which a weight model is requested must have weights specified for each region of that component. However, setting `Weight = 0.0` for a given region will not results in any inclusion in the weight model. Weight files may be used for each component, and are summed together in order to calculate an overall CG location for the whole aircraft. The weights of the initial design and the most recent design are output to the `weight.dat` file. In addition, some information required for proper scaling when restarts are used is output to the `weight.rstrt` file.

3.4 Input File Notes

Some additional notes to keep in mind when setting up input files or for diagnosing run-time errors:

- During the reading of either the component, region or weight files, an internal read error may result if the file is not formatted correctly. Try to modify existing files to avoid formatting problems.
- Due to the fact that a lot happens behind the scenes, it can be easy to accidentally form a linearly-infeasible problem. Some checks are done on the input to identify the most likely causes of infeasibility, however these are not exhaustive. Thus, if your problem appears to be linearly-infeasible (SNOPT INFO = 11), check the Jetstream `.scr` file for possible causes. If these printed warnings do not indicate the cause of infeasibilities for your problem, pay close attention to:

  1. Matching between regions
  2. Twist freedom and bounds
  3. Chord freedom and bounds
  4. Edge linear deviation freedom and bounds

  In particular, ensure that the combinations of the above four constraints make sense at region junctions.
- Avoid using more regions than are needed. While technically not a problem, excessive regions introduce more design variables which can hinder optimizer convergence. In addition, the use of numerous regions can make identifying the source of infeasibilities quite challenging.
- Significant care and testing have gone in to the method by which regions and RDVs are extracted from the initial patches and CPs. The current setup tries to maintain the original geometry as closely as possible, while being robust enough to do so for a range of geometries. This code has primarily been developed for the optimization of wings, flying-wings and wing-body configurations. Thus, if you are doing something significantly different such as very-highly-nonplanar designs, this may not be the tool for you.

4 Revision History

Date	Author	Description of changes
November, 2011	T.A. Reist*	First implementation of region variables.
January, 2012	T.A. Reist*	Differentiation of free and fixed RDVs, such that only free variables are seen by the optimizer.
July, 2013	T.A. Reist*	Introduction of 'components' for control of more complex geometries. Major overhaul of code including newly formatted input files.
August, 2013	T.A. Reist*	Implementation of control effector model as well as code reformatting.

* tom.reist@utoronto.ca