

REFERENCE COPY

C.2

# SANDIA REPORT

SAND91-1866 • UC-705

Unlimited Release

Printed December 1991

## AnvilTOL: Interactive Computer-Aided Tolerance Analysis and Synthesis for Anvil 5000

Richard H. Robison

Prepared by  
Sandia National Laboratories  
Albuquerque, New Mexico 87185 and Livermore, California 94550  
for the United States Department of Energy  
under Contract DE-AC04-76DP00789



SAND91-1866  
0002  
UNCLASSIFIED

12/91  
28P

STAC

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

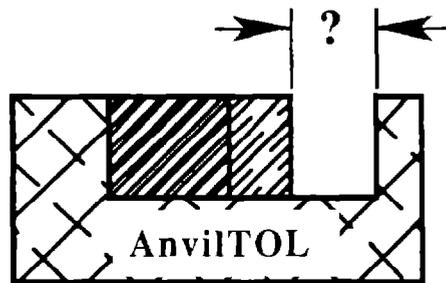
Available to DOE and DOE contractors from  
Office of Scientific and Technical Information  
PO Box 62  
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from  
National Technical Information Service  
US Department of Commerce  
5285 Port Royal Rd  
Springfield, VA 22161

NTIS price codes  
Printed copy: A03  
Microfiche copy: A01

## AnvilTOL: Interactive Computer-Aided Tolerance Analysis and Synthesis for Anvil 5000



Richard H. Robison  
Division 2815 - CAD Software Systems

Sandia National Laboratories  
Albuquerque, New Mexico

### ABSTRACT

This document describes a software application that utilizes an Anvil 5000 CAD model to interactively perform computer-aided tolerance analysis and synthesis. Using AnvilTOL, the designer models a tolerance stack-up as a vector loop by graphically selecting geometric and tolerance data directly from an Anvil5000 CAD model. The tolerance model can then be analyzed by worst case or statistical methods. AnvilTOL also performs tolerance synthesis by re-allocating part tolerances by the methods of proportional scaling or constant precision factor. Precision spacers may also be automatically designed for selective assembly processes.

# Contents

<b>1.0</b>	<b>Introduction .....</b>	<b>1</b>
1.1	AnvilTOL Terminology .....	1
1.2	Tolerance Analysis Using AnvilTOL .....	2
1.3	Limitations of AnvilTOL .....	3
<b>2.0</b>	<b>USING AnvilTOL .....</b>	<b>4</b>
2.1	Create Tolerance Stack .....	4
2.1.1	Defining Stack Parameters .....	4
2.1.2	Creating Dimension Vectors .....	5
2.1.3	Assigning Tolerances .....	8
2.1.4	Termination of the Loop .....	9
2.1.5	Worst Case and Statistical Tolerance analysis .....	9
2.1.6	Allocation By Proportional Scaling .....	10
2.1.7	Allocation By Constant Precision Factor .....	11
2.2	Edit Stack .....	12
2.2.1	Edit Dimension .....	12
2.2.2	Edit Stack Parameters .....	12
2.2.2.1	Change Assembly Tolerance .....	12
2.2.2.2	Change Correction Factor .....	12
2.2.2.3	Change Assembly Sigma .....	13
2.2.3	Delete Stack .....	13
2.2.4	Recall Stack .....	13
2.2.5	List Stacks .....	13
2.3	Option 3: Extended Tolerance Analysis .....	13
2.3.1	Results File .....	14
2.3.2	Generate Tolerance Table .....	14
2.3.3	Precision Spacers .....	14
2.4	Menu Hierarchy Summary .....	15
<b>3.0</b>	<b>SAMPLE PROBLEMS .....</b>	<b>16</b>
<b>4.0</b>	<b>AnvilTOL Programmer's Guide .....</b>	<b>20</b>
4.1	ANVIL 5000 Storage of AnvilTOL Stackups .....	20
4.2	AnvilTOL Subroutine Summary .....	20

# 1.0 Introduction

Tolerance analysis is essential to assure proper assembly functionality and to prevent excessive waste and cost when manufacturing parts, yet it is also tedious and time-consuming. Even with computerized tolerance analysis, data entry tasks can discourage designers from performing adequate tolerance analysis. However, with a design stored in a CAD database, it is possible to extract dimension and tolerance data required for tolerance analysis directly from a CAD system. This possibility is the basis for CAD-based tolerance analysis.

AnvilTOL is a software application that utilizes an Anvil 5000 CAD database to perform interactive, computer-aided linear tolerance analysis and synthesis. AnvilTOL allows the designer to use an existing assembly drawing to graphically select dimensions and tolerances which effect the critical assembly dimension. AnvilTOL then computes both the worst case and statistical tolerance sums, and performs tolerance allocation if a assembly tolerance is specified. Figure 1.1 shows the architecture of the AnvilTOL tolerance analysis system.

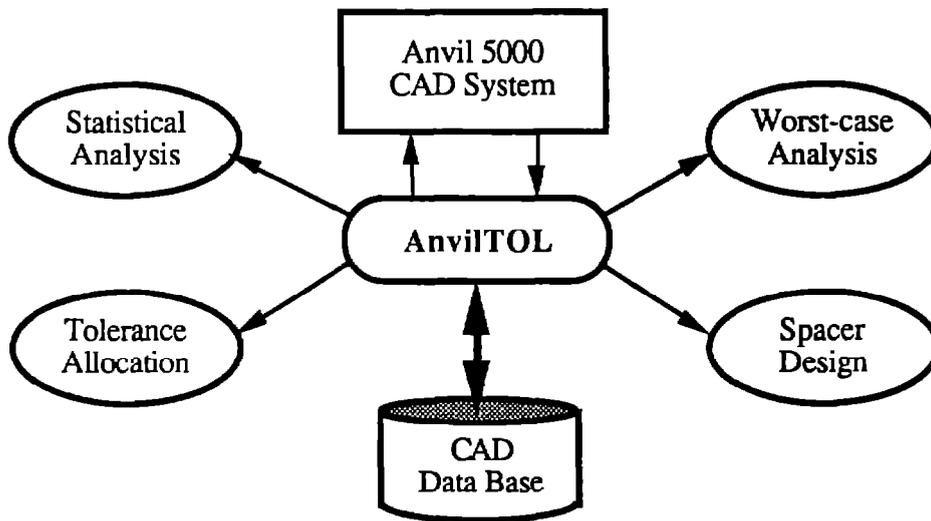


Figure 1.1. AnvilTOL Architecture.

## 1.1 AnvilTOL Terminology

The following terminology is used throughout this document:

**Assembly Function** A mathematical relationship which defines the assembly variable (usually a clearance) in terms of the tolerance variables (usually part dimensions).

**Assembly Variable** An assembly dimension such as a clearance that is determined at assembly and is dependent upon several tolerance variables.

**Dimension Vector** A vector in Euclidean space used to represent a tolerance variable in a vector loop.

**Non-linear Tolerance Analysis** Tolerance analysis where the assembly function(s) is non-linear.

**Tolerance Analysis** Tolerance analysis (or analysis of variation) is the process of predicting the accumulative affect of variations in the independent tolerance variables on the dependent assembly variables.

**Tolerance Allocation** Tolerance allocation is a design task where a designer attempts to rationally assign a specific tolerances to each tolerance variable. A rational tolerance allocation is one where the resulting tolerance stackup does not threaten the assembly's performance while at the same time the tolerances are not so tight that unnecessary manufacturing costs are incurred.

**Tolerance Stackup** The assembly variable's tolerance, which results from an accumulation of variations in the tolerance variables

**Tolerance Variable** An independent variable representing the variation of a part dimension or geometric feature which is limited by a tolerance. Tolerance variables determine the assembly variable via the assembly function.

**Vector Loop** A vector representation of an assembly function where the dimension vectors are connected tip to tail to form a loop or circuit.

## 1.2 Tolerance Analysis Using AnvilTOL

The procedure for performing tolerance analysis using AnvilTOL is as follows (see Figure 1.2):

1. Retrieve or create an wireframe assembly model on which tolerance analysis is to be performed.
2. Execute AnvilTOL.
3. Define a stackup of dimensions and feature controls by selecting existing part geometry.
4. Enter other prompted information for each stackup component such as dimension name, tolerance, etc...
5. Perform automatic tolerance analysis and/or automatic tolerance allocation.
6. Review the analysis results, make design changes, and repeat analysis/allocation
6. Repeat Steps 3 - 6 for each individual tolerance stack to be analyzed.

In automatic tolerance allocation, the allowable assembly variable tolerance is allocated or distributed among the tolerance variables. Tolerance allocation can be based on either a worst case or statistical acculation and is performed according to a specific allocation rule. The two allocation rules available in AnvilTOL are proportional scaling and constant precision factor, both which are discussed in Sections 2.1.6 and 2.1.7.

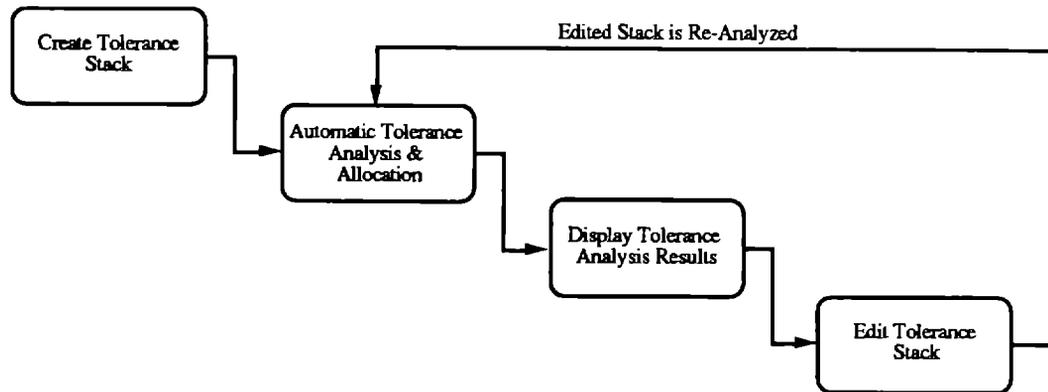


Figure 1.2. Tolerance Analysis Process Using AnvilTOL.

### 1.3 Limitations of AnvilTOL

AnvilTol is limited to one-dimensional linear stackups. In general, assembly variables are usually defined by non-linear assembly functions which include such terms as sines, cosines square roots, etc. But, when assembly functions are linear, the partial derivatives (sensitivity of assembly variable to each tolerance variable) are all equal to 1. In other words, all tolerance variations have a one-to-one linear effect on the assembly tolerance. Many 2-D planar problems can be solved by linear tolerance analysis. The criteria for linear analysis is that all tolerances variables must be in the same plane and in the same direction as the assembly variable. These are the types of problems suitable for AnvilTOL.

Another limitation is that AnvilTOL is implemented in the GRAPL-IV programming language for ANVIL5000 V2.0 and will not run with prior versions. Should the GRAPL language be changed in subsequent versions of ANVIL 5000, AnvilTOL may be rendered incompatible.

## 2.0 USING AnvilTOL

AnvilTOL is a GRAPL-IV program and is executed within Anvil 5000 v2.0 by the ANVIL menu option (Menu 7.2.1): "Run GRAPL Program". Upon execution, AnvilTOL displays the following main menu:

```
AnvilTOL MAIN MENU
1 CREATE STACK
2 EDIT STACK
3 EXTENDED ANALYSIS
4 EXIT AnvilTOL
```

All AnvilTOL menus act identically to the standard Anvil 5000 menus. Menu options may be selected with either the mouse or by entering a option number.

### 2.1 Create Tolerance Stack

The user selects this option to create a vector-loop model of a tolerance stackup.

#### 2.1.1 Defining Stack Parameters

```
ENTER STACK NAME
```

The user should enter a name for the new tolerance loop to be defined. This name will be used to identify the stack if it is recalled in a future session. Should a stackup already exist by the name entered, a message will be given and the prompt will be repeated.

```
DIRECTION OF STACKUP
1 X - DIRECTION
2 Y - DIRECTION
```

The user must select the direction for the tolerance stackup. The Anvil 5000 model may be either a 2-D or 3-D wireframe representation. However, AnvilTOL requires a 2-D planar view of the model. In a 2-D planar workview, the user may define a tolerance stackup in either the x or y direction. The x and y directions are always defined according to the current workview coordinate system. Accordingly, the user must orient the planar view such that dimensions to be used in the stackup are aligned with either the workview's X or Y axis.

```
ASSEMBLY TOLERANCE
ASSMTOL = 0.0000
```

The user can enter an assembly design tolerance (entering 0.0 means no specified assembly tolerance) which limits the tolerance stackup on the clearance. Specifying an assembly tolerance flags AnvilTOL to automatically perform tolerance allocation on part tolerances. Tolerance allocation adjusts each part tolerance to ensure that the resulting tolerance stackup does not exceed

the assembly tolerance. Different analysis and allocation methods are automatically invoked according to the following rules:

- If all parts have specified tolerances, allocation is performed by the method of proportional scaling (Section 2.1.6).
- For part(s) to which no tolerance was assigned, tolerances are allocated by the method of constant precision factor (Section 2.1.7).
- If no assembly tolerance is specified, the worst case and statistical assembly tolerances are calculated. No allocation of component tolerances is performed.

### 2.1.2 Creating Dimension Vectors

Tolerance stackups are created graphically by defining dimension vectors which connect to form a continuous vector loop. Figure 2.1 illustrates this concept by showing a shaft-bearing assembly with a vector loop representation of a tolerance stackup. Dimension vectors are defined by selecting existing CAD geometry which defines datums at the tail and tip of each vector. As dimension vectors are defined, a graphical vector loop representation of the tolerance stackup is super-imposed upon the original Anvil 5000 geometry. It is common practice to begin at one side of the assembly clearance and proceed part-by-part through adjacent parts of the assembly until the other side of the clearance is reached.

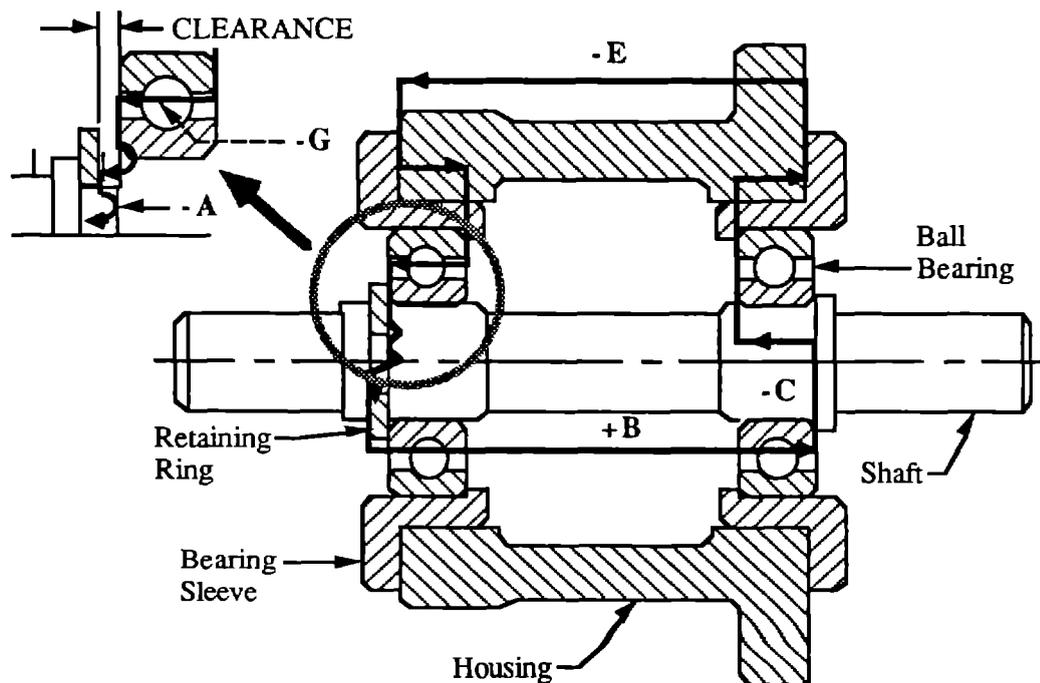


Figure 2.1. Shaft and housing assembly with tolerance loop.

## Place Dimension Vectors Automatically (Y)?

The user has the option to manually place the graphical representation of each dimension vector or let AnvilTOL automatically place the vectors. If 'N' is entered, the user will be prompted to digitize the location of each dimension vector as it is created.

In AnvilTOL, stack dimension vectors are defined by defining datums from existing model geometry which locate the tail and tip of the vector. A datum is defined by identifying a point on an existing Anvil 5000 entity. For the first dimension vector in a stackup, two datums must be defined for the vector's tail and tip positions. For all subsequent vectors, the user only defines one datum for the vector's tip. A tail datum is not required because AnvilTOL automatically places each vector's tail on the tip datum of the last vector. AnvilTOL allows the user to reference points for datums from the following menu:

- DATUM
- 1 EXISTING POINT
- 2 COORDINATE ENTRY
- 3 DELTA
- 4 END STACK

The user selects the appropriate method for defining a point. Note that after all required vectors have been defined, the user ends the stack by selecting Option 4. Also, each defined vector can be rejected in reverse order by entering [reject]. After each vector is defined, the length is calculated as the distance between the two dimension vector datums. Note that for "X loops" the datums are vertical, and for "Y loops", the datums are horizontal. For example, if the tolerance loop was chosen to be in the "x" direction, the part length would be the difference of the two points' "x" components. Figure 2.2 shows how the first dimension vector "A" from Figure 2.1 would be created.

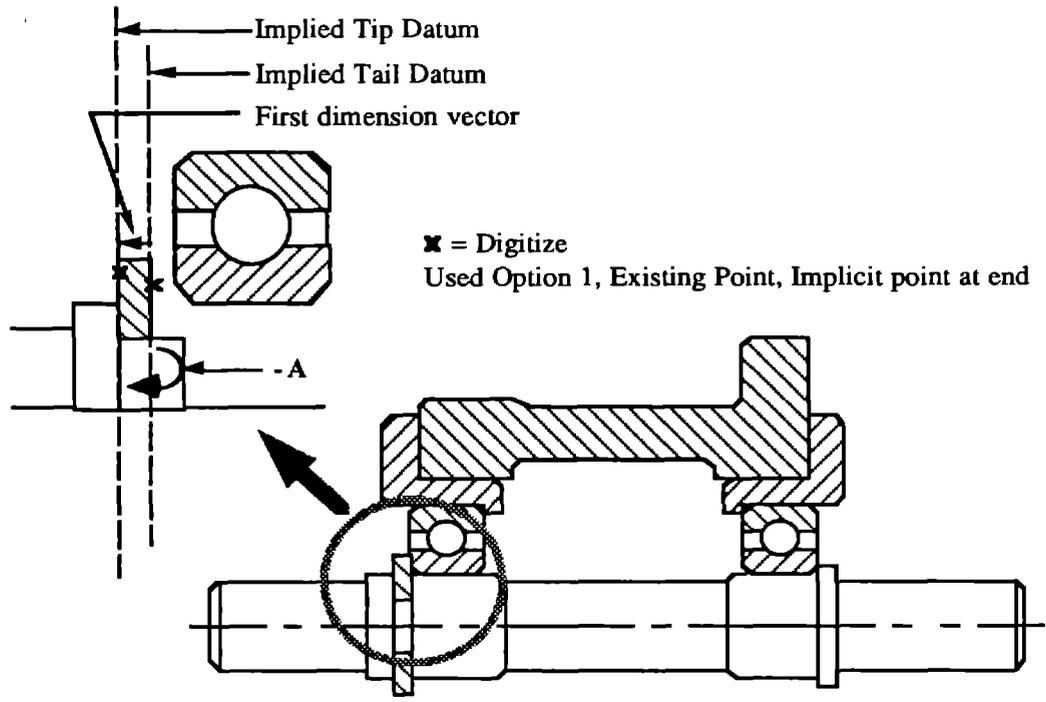


Figure 2.2. Defining dimension "A".

The corresponding dimension vector is then graphically displayed in the vector loop representing the tolerance stack, or in the case of a feature control, the appropriate GD&T symbol is displayed at the correct vector datum as shown in Figure 2.3. A feature control tolerance is defined by digitizing a zero-length vector. This is accomplished by selecting the same datum as the last vector's tip.

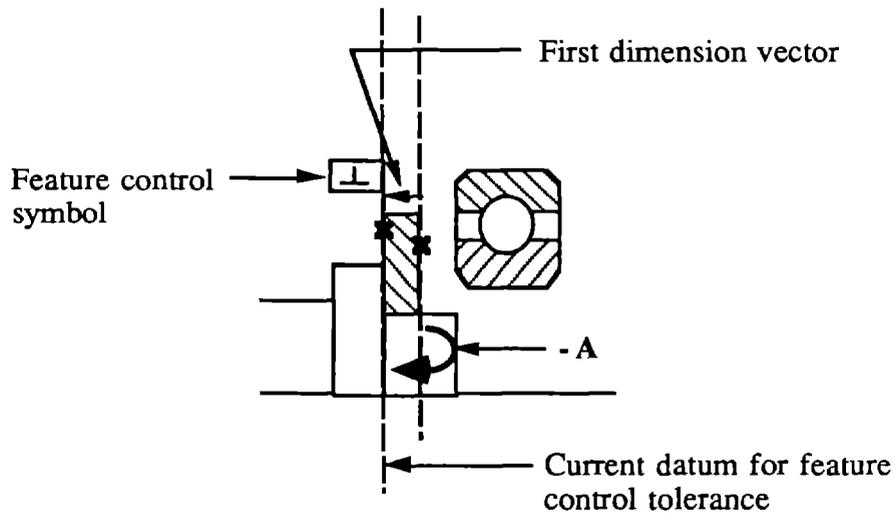


Figure 2.3. Feature control symbol in stackup vector loop.

ENTER DIMENSION NAME (Loopname:V1)

The user should enter a unique name for the dimension. If command complete is selected without entering a name, a default is chosen as the stack name concatenated with the sequential vector number.

### 2.1.3 Assigning Tolerances

As each dimension vector is defined, the user assigns a tolerance. If no tolerance is specified, AnvilTOL will automatically allocate a tolerance for that dimension vector. As mentioned above, ANSI Y14.5 feature control tolerances may also be included in a tolerance stackup as independent manufacturing variables by creating a zero-length dimension vector. This flags AnvilTOL that the dimension is a geometric tolerance. A zero-length dimension is defined by selecting the same datum point as the previous vector's tip datum. If, for example, a perpendicularity tolerance was required for the tip datum surface of dimension vector "A" in Figure 2.2, simply select the same tip datum again, and a zero-length vector is defined.

When AnvilTOL detects a zero-length dimension vector, the following menu of ANSI Y14.5 symbols is presented:

```
FEATURE CONTROL TYPE
1 STR - STRAIGHTNESS
2 FLA - FLATNESS
3 RND - ROUNDNESS
4 CYL - CYLINDRICITY
5 PRL - PROFILE LINE
6 ANG - ANGULARITY
7 PER - PERPENDICULARITY
8 PAR - PARALLELISM
9 POS - TRUE POSITION
10 CON - CONCENTRICITY
```

Although in reality the effects of these feature controls are non-linear, AnvilTOL makes a linear approximation using the tolerance value assigned by the user. The user enters a plus or minus symmetrical tolerance from the following menu:

```
+/- TOLERANCE
1 .001
2 .002
3 .003
4 .005
5 .0001
6 .0005
7 .01
8 Enter Value
```

AnvilTOL provides several common tolerances, or an arbitrary tolerance is entered by selecting Option 8. The same menu of default +/- symmetrical tolerance values is presented for

size tolerances (vector length is non-zero). If option 8 is selected, the user may enter an arbitrary tolerance either as a +/- symmetrical tolerance or as a max/min bi-lateral tolerance at the following prompt:

```
          TOLERANCE
+/- TOL = 0.0000
MAX TOL = 0.0000
MIN TOL = 0.0000
```

The "+/- TOL" field is used to enter a symmetrical tolerance, or the "MAX LIMIT" and "MIN LIMIT" fields are used to specify a bi-lateral tolerance. If the "+/- TOL" field is non-zero, the other fields are ignored. Also, if all three fields are left as zero, this is a zero tolerance and AnvilTOL will know to automatically allocate a tolerance for that dimension vector. If a bi-lateral tolerance is entered, AnvilTOL internally converts it to a symmetrical tolerance and computes a symmetrical nominal length. The symmetrical length is used from then on.

After assigning a tolerance, the user can specify whether or not the tolerance is fixed at the prompt shown below. For any dimension vector that has a fixed tolerance, AnvilTOL will not alter that tolerance during automatic allocation. This is important for representing part dimensions over which the designer has no direct control (vendor supplied parts, existing parts, a critical design specification, etc.).

```
Is tolerance fixed (N)?
```

Tolerances on a feature controls are automatically fixed. This is because allocation by constant precision factor (see Section 2.1.7) cannot be performed on a part with zero length. Also, it is likely that if a designer has chosen to use a feature control tolerance, the tolerance value has already been determined.

#### **2.1.4 Termination of the Loop**

As mentioned above (section 2.1.2), the user terminates the process of defining dimensions vectors and ends the stackup by choosing Option 4 from the DATUM Menu. If the user chooses to end the stack, the vector loop is completed by adding a closing vector representing the clearance labeled with the stack name. Once the stackup is completed, tolerance analysis and/or allocation are automatically executed. The methods for tolerance analysis and synthesis are discussed in Sections 2.1.5 thru 2.1.7.

#### **2.1.5 Worst Case and Statistical Tolerance analysis**

Standard worst case and statistical tolerance analysis are both automatically performed on each newly entered tolerance loop. In a worst limits analysis, the assembly tolerance ( $T_{ASM}$ ) is

determined by a linear sum of the dimension tolerances ( $T_i$ ). Each dimension tolerance is assumed to be at its max. or min. limit, resulting in the worst possible tolerance accumulation.

$$T_{\text{Assm}} = \sum T_i \quad (1)$$

Statistical analysis assumes that component tolerances add as the root sum squared (RSS). The low probability of the worst case combination occurring is taken into account statistically, assuming a Normal or Gaussian distribution for component variations. Tolerances are commonly assumed to correspond to six standard deviations ( $6\sigma$ ):

$$T_{\text{Assm}} = \sqrt{\sum T_i^2} \quad (2)$$

For the general case (other than  $6\sigma$  tolerance distributions):

$$T_{\text{Assm}} = C_f Z \sqrt{\sum (T_i/Z_i)^2} \quad (3)$$

where  $Z$  (referred to as  $z$  factor) is the number of standard deviations desired for the specified assembly tolerance and  $Z_i$  is the expected standard deviations for each dimension tolerance.

$C_f$  is a correction factor frequently added to account for any non-ideal conditions (non-normal tolerance distributions, etc.). When  $C_f$  is 1.0, the result is a simple root sum squared analysis, but experience has shown that a simple RSS can result in tolerances that are too loose. Also, for assemblies with a small number of parts, a simple RSS analysis can actually predict a larger tolerance stackup than worst case. Therefore, typical values used for  $C_f$  are 1.2, 1.5 and 1.7. Initially, AnvilTOL uses  $C_f = 1.2$  for statistical analysis and allocation; however,  $C_f$  can be changed to any value (see Section 2.2.2.2). The following are a general set of guidelines for performing tolerance analysis:

- Use the results from the analysis that predicts the smallest tolerance accumulation out of either worst case, RSS with  $C_f = 1.2$ , RSS with  $C_f = 1.5$ , or RSS with  $C_f = 1.7$ .
- $C_f = 1.5$  is generally best for small to large production lots.
- $C_f = 1.7$  is generally best for prototype production.

These guidelines apply to tolerance allocation as well. The rational allocation of component tolerances requires the establishment of some rule upon which to base the allocation. Sections 2.1.6 and 2.1.7 describe the two allocation methods used in AnvilTOL:

#### 2.1.6 Allocation By Proportional Scaling.

The designer begins by assigning reasonable component tolerances based on process or design guidelines. AnvilTOL sums the component tolerances to see if they meet the specified assembly tolerance. If not, it scales the component tolerances by a constant proportionality factor.

In this way the relative magnitudes of the component tolerances are preserved. The proportionality factor is computed from:

$$\begin{array}{ll}
 \text{Worst Limits} & \text{Statistical} \\
 P = \frac{T_{\text{Assm}}}{\sum T_i} & P = \frac{T_{\text{Assm}}}{\sqrt{\sum T_i^2}}
 \end{array} \quad (4)$$

Then the component tolerances are scaled as

$$T_i = P \cdot D_i \quad (5)$$

### 2.1.7 Allocation By Constant Precision Factor

Parts machined to a similar precision will have equal tolerances if they are the same size. As part size increases, tolerances generally increase approximately with the cube root of size:

$$T_i = P \cdot (D_i^{1/3}) \quad (6)$$

where  $D_i$  is the basic size of the part and  $P$  is the Precision Factor.

Based on this rule of thumb, the tolerances can be distributed according to part size as follows. AnvilTOL computes the Precision Factor:

$$\begin{array}{ll}
 \text{Worst Limits} & \text{Statistical} \\
 P = \frac{T_{\text{Assm}}}{\sum D_i^{1/3}} & P = \frac{T_{\text{Assm}}}{\sqrt{\sum D_i^{2/3}}}
 \end{array} \quad (7)$$

Then the component tolerances are computed by equation 6.

If the user did not specify an assembly tolerance, no allocation takes place. For tolerance allocation, the user must specify an assembly tolerance for the particular loop under study. An assembly tolerance can be entered either when the user first starts creating a new loop, or by editing an existing loop in the Edit and Review tolerance stack submenu.

The type of tolerance allocation performed is determined by whether or not all stack components have been assigned a tolerance. While creating a new loop, the user can choose not to assign any loop component a tolerance. If this is the case, both worse case and statistical allocation will automatically be performed by the method of Constant Precision Factor. If all loop components have been assigned tolerances, then all tolerances on unfixed components will be reallocated by the method of Proportional Scaling.

Allocation flexibility is provided to the user by allowing a component with an unspecified tolerance to be assigned a specified tolerance ( or visa versa) by editing that component. Additional flexibility is available by using the "fix" option. Any component tolerance may be fixed at any time

by means of the editing option. Tolerance allocation is then performed on the remaining component tolerances. Similarly, any fixed tolerance may be "unfixed". Editing is performed under the Edit and Review tolerance stack submenu.

Additional tolerance analysis options are available as menu options under the Edit and Review menu described in the next section.

## 2.2 Edit Stack

This main menu option gives the capability to edit assembly or dimension information for the current stack or to recall an existing tolerance stack. The following submenu is presented:

```
EDIT STACK
1 EDIT DIMENSION
2 EDIT STACK PARAMETERS
3 DELETE STACK
4 RECALL STACK
5 LIST STACKS
```

### 2.2.1 Edit Dimension

The user can select any dimension vector in the current tolerance stack and edit its name, tolerance value, and fix/unfix flag. After selecting the vector to edit, the user is taken through a similar set of input prompts as in Section 2.1.3: Assigning Tolerances. After editing any number of dimension vectors, AnvilTOL automatically re-computes the tolerance analysis and allocation.

### 2.2.2 Edit Stack Parameters

The user can change the current active stack's specified assembly tolerance, statistical correction factor ( $C_f$ ), and/or assembly sigma (number of standard deviations on assembly tolerance - see Section 15.1).

```
EDIT STACK
1 ASSEMBLY TOLERANCE
2 RSS CORRECTION FACTOR
3 ASSEMBLY SIGMA
```

#### 2.2.2.1 Change Assembly Tolerance

The user is given the following data entry/modify list in which to edit the stack's specified assembly tolerance. Note that AnvilTOL will automatically re-analyze the tolerance stack according to AnvilTOL's rules for automatically performing tolerance analysis (see Section 2.1.1).

```
ASSEMBLY TOLERANCE
ASSMTOL = 0.0000
```

#### 2.2.2.2 Change Correction Factor

The user can set or change the correction factor to compensate for the weaknesses inherent in simple root sum squared statistical analysis (see Section 2.1.5). The default for this value is 1.2, which gives a semi-conservative root sum squared analysis. Changes made to  $C_f$  are not

stored in the data base. If Cf is changed and the stackup is subsequently recalled, Cf assumes the default value.

```
RSS CORRECTION FACTOR
Cf = 1.0000
```

### 2.2.2.3 Change Assembly Sigma

The user can edit the stack's assembly sigma which is the number of standard deviations corresponding to the assembly tolerance during statistical analysis and allocation. Changes made to Sigma are not stored in the data base. If Sigma is changed and the stackup is subsequently recalled, Sigma assumes the default value.

```
ASSEMBLY SIGMA
1 +/- 3 SIGMA -.9973 YIELD
2 +/- 2.58 SIGMA -.99 YIELD
3 +/- 3.29 SIGMA -.999 YIELD
4 +/- 3.89 SIGMA -.9999 YIELD
5 +/- 4.42 SIGMA -.99999 YIELD
6 User Defined
```

### **2.2.3 Delete Stack**

This option should be selected if the user has no more use for an existing tolerance stack and wishes to delete it from the ANVIL 5000 data base. Caution! Since this option deletes the tolerance stack data from the ANVIL 5000 data base, it is permanently gone. AnvilTOL allows the user to reaffirm this action by the following prompt:

```
DELETE CURRENT STACK?
```

### **2.2.4 Recall Stack**

The user can recall any pre-defined tolerance stack (granted it has not been deleted from the ANVIL 5000 data base) as the current active stack.

```
ENTER STACK NAME
```

AnvilTOL searches the database for the requested stackup and activates it if found. With the stackup activated, the user can resume exploring design changes, generating tables, etc.

### **2.2.5 List Stacks**

This option lists to the screen the names of all tolerance stacks contained in the ANVIL 5000 data base.

## **2.3 Option 3: Extended Tolerance Analysis**

This option is provided for the output of analysis results and for additional tolerance analysis features to be added into the software. The options in the following extended analysis submenu apply to the current tolerance stack:

EXTENDED ANALYSIS  
1 RESULTS FILE  
2 GENERATE TABLE  
3 PRECISION SPACERS

### 2.3.1 Results File

After creating a tolerance loop, this option writes the tolerance stackup information, including tolerance analysis and allocation results, to an external ASCII data file.

ENTER FILE NAME

The user enters the name of the file (no extension should be entered since AnvilTOL automatically adds the extension ".ATL") to which the analysis results are to be sent. Since the resulting file is in sequential ASCII format, it can be reviewed or edited like any text file. Also, as an alternative to the next option, generate tolerance table, the results file can be used as text for a note entity to place analysis results on the drawing.

### 2.3.2 Generate Tolerance Table

This option allows the user to place a table directly on the CAD drawing containing all the tolerance data for the current tolerance loop. The table also contains the results of tolerance analysis and allocation performed by AnvilTOL.

ENTER DRAFT SCALE  
SCALE = 1.0000

The user defines the draft scale.

ENTER TEXT SIZE  
TEXT SIZE = 0.1500

The user defines the text size of the tolerance table. Then a menu is presented for locating the table's upper left corner.

DEFINE UPPER LEFT CORNER OF TABLE  
1 SCREEN POSITION  
2 ENTER COORDINATES  
3 EXISTING POINT

### 2.3.3 Precision Spacers

This option allows the user to enter a build dimension, build tolerance and spacer tolerance in order to calculate precision spacers to bring the assembly clearance to a desired value and tolerance. This option requires that there is no specified assembly tolerance. If one is specified, simply choose the edit stack option and change the specified assembly tolerance to zero.

PRECISION SPACER BUILD DATA  
ASSEMBLY CLEARANCE = 0.0000  
BUILD DIMENSION = 0.0000  
BUILD TOLERANCE = 0.0000  
SPACER TOLERANCE = 0.0000

For the above data, the assembly clearance must be larger than the build dimension, and the build tolerance must be larger than the spacer tolerance.

## 2.4 Menu Hierarchy Summary

The following outline shows the hierarchy of AnvilTOL's menu structure:

### Main Menu

1. Create Stack
2. Edit Stack
  1. Edit Dimension
  2. Edit Stack Parameters
    1. Assembly Tolerance
    2. RSS Correction Factor
    3. Assembly Sigma
  3. Delete Stack
  4. Recall Stack
  5. List Stacks
3. Extended Analysis
  1. Results File
  2. Generate Tolerance Table
  3. Precision Spacers
4. Exit AnvilTOL

### 3.0 SAMPLE PROBLEMS

The following examples are based on the shaft and bearing assembly in Figure 3.1 reproduced from Figure 2.1. Initial tolerances for dimensions B, D, E, and F are selected from tolerance guidelines for the turning processes. The retaining ring (dimension A) and the two bearings (dimensions C and G) supporting the shaft are vendor-supplied, hence their tolerances are fixed and must not be altered by the allocation process. The critical clearance is the shaft end-play (between the bearing and retaining ring), which is determined by tolerance accumulation in the assembly. In each following analysis example, the design specifications are stated, calculations are given, and the results are presented in a tolerance table generated by AnviTOL:

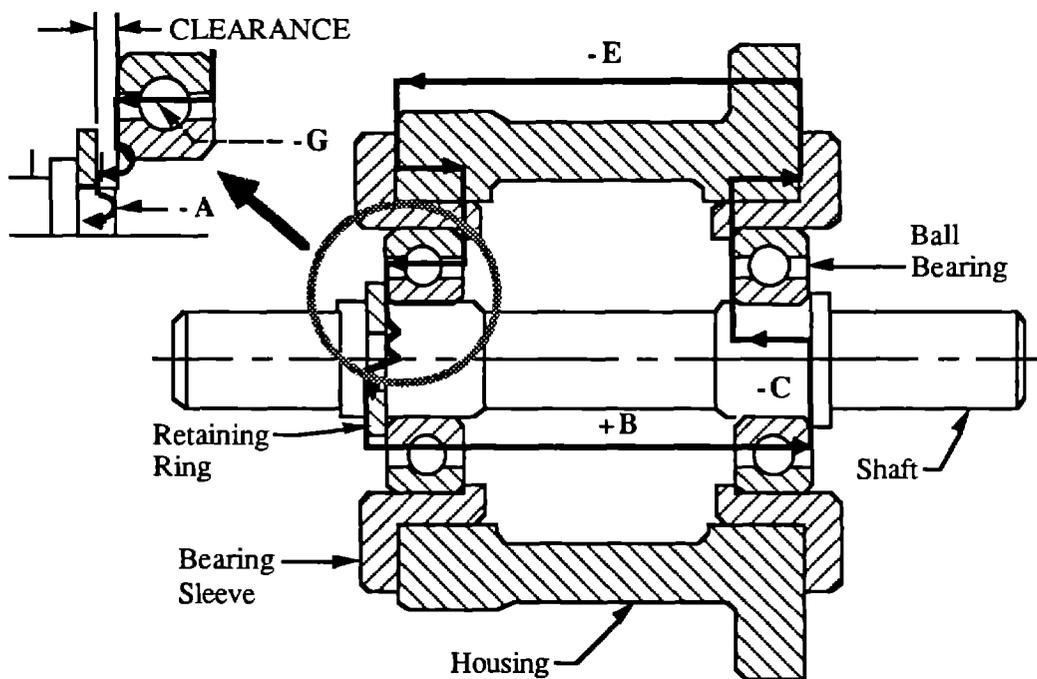


Figure 3.1. Shaft and bearing assembly.

**Example 1.** Worst case and statistical analysis of assembly tolerance.

Initial tolerance specifications:

Dimension	A	B	C	D	E	F	G
Average	.0505	8.000	.5093	.400	7.711	.400	.5093
Tolerances(+/-)							
Design		.008		.002	.006	.002	
Fixed	.0015		.0025				.0025

$$\begin{aligned}
 \text{Average Clearance} &= -A + B - C + D - E + F - G \\
 &= -.0505 + 8.000 - .5093 + .400 - 7.711 + .400 - .5093 \\
 &= .020
 \end{aligned}$$

The following is a tolerance table generated by AnviTOL showing the results of this example; since no assembly tolerance is specified, no tolerance allocation is performed:

Analysis for Stack-up: Housing
CLEARANCE: 0.020 CF: 1.000
ASSM TOL: +/- 0.000 ZFAC: 3.000

#	DIMNAME	TYPE	NOMINAL	FIX	+/- TOL	W.C. ALOC	ST. ALOC
1	RING	SIZ	-0.0505	Y	0.00150	0.00000	0.00000
2	SHAFT	SIZ	8.000		0.00800	0.00000	0.00000
3	BEARING1	SIZ	0.5093	Y	0.00250	0.00000	0.00000
4	SLEEVE1	SIZ	0.400		0.00200	0.00000	0.00000
5	HOUSING	SIZ	7.711		0.00600	0.00000	0.00000
6	SLEEVE2	SIZ	0.400		0.00200	0.00000	0.00000
7	BEARING2	SIZ	0.5093	Y	0.00250	0.00000	0.00000

W.C SUM 0.0245  
ST. SUM 0.0111

**Example 2.** This example illustrates worst case and statistical allocation by proportional scaling showing worst case calculations. Note: to match the results in AnviTOL,  $C_f$  (RSS correction factor) must be set to 1.0.

Design specifications:

Required Clearance = .020 +/- .015 (Edit the Assembly Tolerance to be 0.015)

The clearance tolerance is obtained by computing the worst case tolerance sum:

$$\begin{aligned} T_{ASM} &= + T_A + T_B + T_C + T_D + T_E + T_F + T_G \\ &= + .0015 + .008 + .0025 + .002 + .006 + .002 + .0025 \\ &= .0245 \text{ (too large)} \end{aligned}$$

Solving for the proportionality factor:

$$\begin{aligned} T_{ASM} &= .015 &= .0015 + .0025 + .0025 + P (.008 + .002 + .006 + .002) \\ P &= .47222 \end{aligned}$$

Note that the fixed tolerances were subtracted from the assembly tolerance before computing the scale factor. Thus only the four design tolerances are re-allocated:

$$\begin{aligned} T_B &= .47222 (.008) = .00378 & T_E &= .47222 (.006) = .00283 \\ T_D &= .47222 (.002) = .00094 & T_F &= .47222 (.002) = .00094 \end{aligned}$$

Each of the design tolerances has been scaled down to meet assembly requirements. This procedure could also be followed assuming a statistical sum for the assembly tolerance (equation 2), in which case the tolerances would be scaled up. The results are found in the following AnviTOL tolerance table:

Analysis for Stack-up: Housing CLEARANCE: 0.020 CF: 1.000 ASSM TOL: +/- 0.015 ZFAC: 3.000
---

#	DIMNAME	TYPE	NOMINAL	FIX	+/- TOL	W.C. ALOC	ST. ALOC
1	RING	SIZ	-0.0505	Y	0.00150	0.00150	0.00150
2	SHAFT	SIZ	8.000		0.00800	0.00378	0.01116
3	BEARING1	SIZ	0.5093	Y	0.00250	0.00250	0.00250
4	SLEEVE1	SIZ	0.400		0.00200	0.00094	0.00279
5	HOUSING	SIZ	7.711		0.00600	0.00283	0.00837
6	SLEEVE2	SIZ	0.400		0.00200	0.00094	0.00279
7	BEARING2	SIZ	0.5093	Y	0.00250	0.00250	0.00250

W.C. SUM 0.0245

ST. SUM 0.0111

**Example 3.** Worst case and Statistical Allocation by Precision Factor showing statistical calculations.

Design specifications:

Required Clearance = .020 +/- .015

No user specified tolerances for parts B (shaft), D (sleeve1), E (housing), and F (sleeve2).  
(Edit each of these dimensions and set the tolerances to zero)

Compute the assembly tolerance for the shaft/bearing assembly by a statistical sum:

$$T_{ASM}^2 = T_A^2 + T_B^2 + T_C^2 + T_D^2 + T_E^2 + T_F^2 + T_G^2$$

$$.0152 = (.0015^2 + .0025^2 + .0025^2) + P^2 (8.0^{2/3} + .400^{2/3} + 7.711^{2/3} + .400^{2/3})$$

Again, the fixed tolerances are subtracted from the assembly tolerance before computing the precision factor.

Solving for the precision factor:  $P = .004836$

Re-allocating:

$$T_B = .004836 (8.00)^{1/3} = .00976 \quad T_E = .004836 (7.711)^{1/3} = .00955$$

$$T_D = .004836 (.400)^{1/3} = .00356 \quad T_F = .004836 (.400)^{1/3} = .00356$$

The Precision Factor method is similar to the Proportional Scaling method, except there is no initial allocation required by the designer. Instead, the tolerances are initially allocated according to the nominal size of each component dimension, then scaled to meet the specified assembly tolerance. This procedure could also be followed assuming a worst limits sum for the assembly tolerance (equation 1).

Analysis for Stack-up: Housing  
 CLEARANCE: 0.020 CF: 1.000  
 ASSM TOL: +/- 0.015 ZFAC: 3.000

#	DIMNAME	TYPE	NOMINAL	FIX	+/- TOL	W.C. ALOC	ST. ALOC
1	RING	SIZ	-0.0505	Y	0.00150	0.00150	0.00150
2	SHAFT	SIZ	8.000		0.00000	0.00312	0.00976
3	BEARING1	SIZ	0.5093	Y	0.00250	0.00250	0.00250
4	SLEEVE1	SIZ	0.400		0.00000	0.00115	0.00356
5	HOUSING	SIZ	7.711		0.00000	0.00308	0.00955
6	SLEEVE2	SIZ	0.400		0.00000	0.00115	0.00356
7	BEARING2	SIZ	0.5093	Y	0.00250	0.00250	0.00250

W.C. SUM 0.0065  
 ST. SUM 0.0038

**Example 4.** Extended tolerance analysis to calculate worst case precision spacers.

Initial tolerance specifications:

Dimension	A	B	C	D	E	F	G
Average	.0505	8.000	.5093	.400	7.711	.400	.5093
Tolerances(+/-)							
Design		.008		.002	.006	.002	
Fixed	.0015		.0025				.0025

$$\begin{aligned} \text{Average Clearance} &= -A + B - C + D - E + F - G \\ &= -.0505 + 8.000 - .5093 + .400 - 7.711 + .400 - .5093 \\ &= .020 \end{aligned}$$

Precision spacer design specifications:

- Build dimension (new nominal clearance) = .010
- Build tolerance = +/- .008
- Spacer tolerance = +/- .001

The result is two precision spacers whose nominal sizes are .003 and .017. During assembly spacers would be selectively placed into position until the measured clearance matched the specifications for the build dimension and tolerance.

## 4.0 AnvilTOL Programmer's Guide

This section provides a reference for maintenance of the AnvilTOL software.

### 4.1 ANVIL 5000 Storage of AnvilTOL Stackups

AnvilTOL utilizes ANVIL 5000 entity types as a means for both representing dimension vectors and storing related tolerance information in the ANVIL 5000 database. A dimension vector is either a dimension (Type 13) for size tolerances or a geometric tolerance (Type 16) for GD&T vectors. AnvilTOL stores each dimension vector's name, symmetrical nominal length, tolerance, and fixed/unfixed flag as part of the entity's canonical information available through implied common. As illustrated in Figure 4.1, the dimension vector's name is simply the entity name. EC(35) is used to store the fixed/unfixed flag as either the interger 68 or 69. Instead of 0 and 1, these two numbers are used as a code to identify whether or not a dimension entity is part of a tolerance stackup. Additionally, the dimension vector's symmetrical nominal length is stored in DDEF(J2+4) (see extended GRAPL manual), normally used for xxxxx, and the vector's tolerance is stored in DDEF(1) which is normally the entity's Z-Depth.

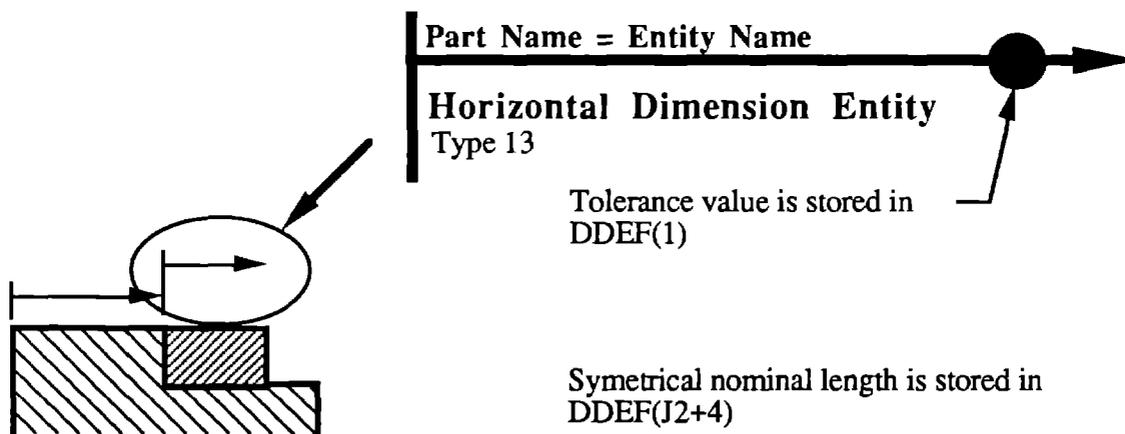


Figure 4.1. Tolerance information stored with dimension entity.

The dimension vector which closes the vector loop and represents the stackup clearance is also a dimension entity. Like above, DDEF(1) is used to store the assembly tolerance, and DDEF(J2+4) contains the stackup clearance.

A complete tolerance stackup is stored as a group entity consisting of all the stackup's dimension vector entities. The group entity's name is the stackup's name.

### 4.2 AnvilTOL Subroutine Summary

AnvilTOL is comprised of several GRAPL-IV programs. The main GRAPL-IV program is called MAINDRIV. The other programs are GRAPL-IV subroutines called from MAINDRIV or other subroutines. Figure 4.2 shows the hierarchy of AnvilTOL's subroutines, and the subroutines and their functions are listed below:

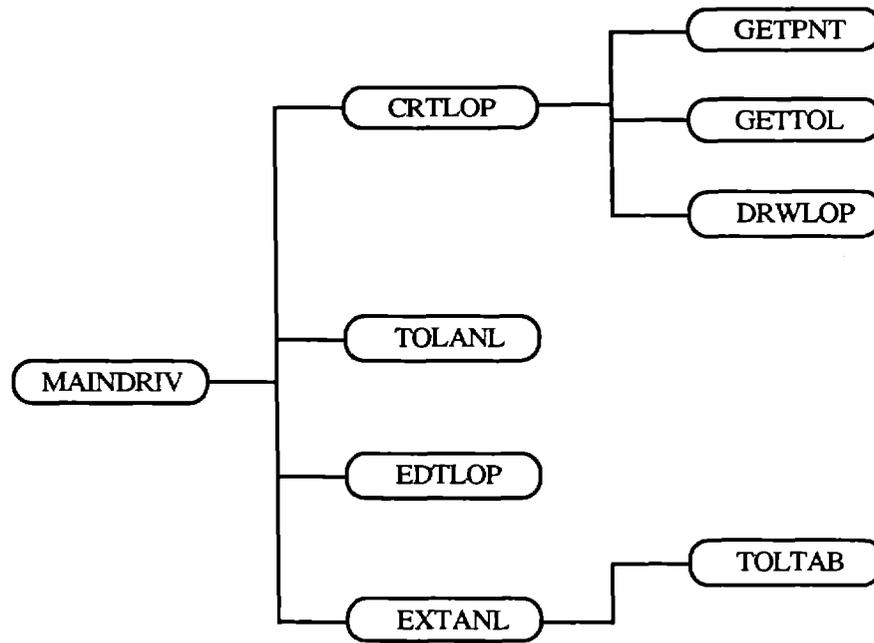


Figure 4.2. AnvilTOL Subroutine Hierarchy

- Maindriv** Main driver program that presents the main menu and branches to appropriate subroutines.  
Calls Subroutines Crtlop, Tolanl, Edtlop, Extanl.
- Crtlop** Subroutine which lets user define a tolerance stackup.  
Called by Subroutine Maindriv.  
Calls Subroutines Getpnt, Gettol, Drawlp.
- Getpnt** Lets user select existing Anvil5000 geometry to define stackup dimensions.  
Called by Subroutine Crtlop.
- Gettol** Prompts user for tolerance information for each stackup dimension, including tolerance type, tolerance value, and whether tolerance is fixed.  
Called by Subroutine Crtlop.
- Drwlop** Displays graphic representation of stackup vector loop, including geometric tolerances, as each stackup dimension is created.  
Called by Subroutine Crtlop.

- Tolanl** Automatically computes worst case and statistical tolerance analysis and allocation for the current loop upon 1) completion of creating a stack, or 2) completion of editing stack or dimension parameters (statistical accuracy, design tolerance, fixed flag, etc.)
- Called by Subroutine Maindrv.
- Edtlop** Allows user to recall an existing stack, edit the current stack's design parameters, or edit the parameters of each dimension in the current stack (name, tolerance, fixed/unfixed, etc.) If changes are made which would affect analysis results, a new tolerance analysis is automatically computed.
- Called by Subroutine Maindrv.
- Extanl** Allows user to display tolerance analysis results either as 1) an ASCII file written to disk, or 2) a table displayed on the drawing. Also allows user to design precision spacers for selective assembly.
- Called by Subroutine Maindrv.  
Calls Subroutines Toltab and Stkfil.
- Toltab** Generates a tolerance table containing all analysis results and tolerance information for the current stackup model.
- Called by Subroutine Extanl.
- Stkfil** Writes to an sequential, ASCII file all analysis results and tolerance information for the current stackup model.
- Called by Subroutine Extanl.

## Distribution

2543 Rick Harris  
2545 Bill Greenwood (2)  
2545 Scott Nicolaysen  
5100 Glen Rackley  
2800 Arlyn Blackwell  
2814 Ross Burchard  
2814 Margaret Olsen  
2815 Dick Thompson  
2815 Sam Cancilla  
2815 Fran Current  
2815 Greg Neugebauer  
2815 Richard Robison (10)  
2850 Tom Cannon  
2851 Larry Grube (3)  
2852 Del Klinetobe (3)  
2853 Ron Williams (3)  
2855 Ron Haid  
2858 Bill Drozdick (3)  
2858 Wilbur Martin  
2858 Mark Platzbecker

3141 S. A. Landenberger (5)  
3145 Document Processing (8) for DOE/OSTI  
3151 G. C. Claycomb (3)  
3400 (1)

7483 Paul Plomp  
8523 Central Technical Files

KCD Curtis Brown  
KCD Charlie Miller  
MCS Gary Kilponen  
MCS Scott Hanratty  
MCS Scott Owens  
MCS Mitchel Clowsner  
MCS Jack Arnold

Eastman Kodack Bill Wallace  
GM Allison Mike Larsen (Transmission Division)  
GM Allison John Toole (Gas Turbine Division)