LA-13784

Comparisons of Wilson-Fowler and Parametric Cubic Splines with the Curve-Fitting Algorithms of Several Computer-Aided Design Systems

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

Edited by Charmian Schaller, Weirich & Associates, for Group IM-1.

LA-13784

Issued: February 2001

Comparisons of Wilson-Fowler and Parametric Cubic Splines with the Curve-Fitting Algorithms of Several Computer-Aided Design Systems

Wilbur D. Birchler
Scott A. Schilling

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

Acknowledgements

The authors would like to express their thanks to all the individuals who contributed to this report. Some individuals provided data files, others contributed to the contents, and some provided moral support.

- Los Alamos National Laboratory, Los Alamos, New Mexico
  - Christopher J. Scully
  - Jacob S. Tafoya
  - Ronald M. Dolin
  - Richard M. Kieltyka
  - Stephen J. Levings
  - Bradley G. Baas
  - Scott D. Parkinson
  - Manuel A. Garcia
  - Edmund E. Kettering
  - Robert F. Montoya
  - Jody B. Niesen
  - Matthew L. Porter
  - Antonio R. Gonzales
  - Ronald V. Karpen
  - Bruce C. Trent
- Los Alamos National Laboratory, Los Alamos, New Mexico (Retired)
  - Donald Hall
- The Y-12 Plant, Oak Ridge, Tennessee
  - Bill D. Cain
- Rocky Flats Plant, Golden, Colorado (Retired)
  - Leroy Mellecker
- Atomic Weapons Establishment (AWE) Hunting-Brae
  - Andy Oughton, Reading, Berkshire, United Kingdom

Table of Contents

List of Figures

List of Tables

Comparisons of Wilson-Fowler and Parametric Cubic Splines with the Curve-Fitting Algorithms of Several Computer-Aided Design Systems

by

Wilbur D. Birchler

and

Scott A. Schilling

Abstract

Summary

The purpose of this report is to demonstrate that modern computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided engineering (CAE) systems can be used in the Department of Energy (DOE) Nuclear Weapons Complex (NWC) to design new and remodel old products, fabricate old and new parts, and reproduce legacy data within the inspection uncertainty limits. In this study, two two-dimensional splines are compared with several modern CAD curve-fitting modeling algorithms. The first curve-fitting algorithm is called the Wilson-Fowler Spline (WFS), and the second is called a parametric cubic spline (PCS). Modern CAD systems usually utilize either parametric cubic and/or B-splines.

Three studies are presented in this report. The first study shows that the accuracy of the minimum-distance algorithm and the ability of both the WFS and PCS to represent analytical data sets (circle, ellipse, and parabola) are well within the inspection uncertainty. A ratio of the calculated deviation to the inspection uncertainty is the metric used in this report. If this ratio is less than one, then the models can be used to design, inspect, and fabricate the parts. Of the 18 numerical analyses presented, the largest ratio of calculated deviation to inspection uncertainty is 0.1489. This ratio is associated with the parabola and is located at the point of highest curvature. The signs of the deviations are shown to be correct. The conclusion from this study is that both the WFS and the PCS can be used to reproduce legacy data and to design new products and redesign old ones.

The second study evaluates three CAD systems—the Parametric Technology, Inc. (PTC) Pro/ENGINEER (Pro/E) system; the Control Data Corporation (CDC) Integrated Computer Engineering and Manufacturing Design, Drafting, and Numerial Control (ICEM DDN) system; and the Computervision Computer-Aided Design and Drafting System (CADDS). It demonstrates their capabilities to model DOE legacy data, and determines that they can be utilized to develop future models. Three nonanalytical shapes (ellipse, lampshade, and a weird shape) are evaluated with the three CAD systems. Of the 18 numerical analyses presented, the largest ratio of calculated deviation to inspection uncertainty is 0.6552. This ratio is associated with the weird-shape curve and is located at the point of high curvature.

This study indicates that all three CAD systems can be used to design, inspect, and fabricate parts from both legacy data and new solid-based models.

The third study establishes upper bounds on the variation of the end angles used to define the geometry of analytical shapes (circle, ellipse, and parabola) and still remain within the inspection uncertainty. When the end angles vary as much as 0.25 degree from the nominal values, the resultant deviations are still within the inspection uncertainly. End-angle effects are very local and damp out in the fourth or fifth segments of the spline data. The conclusion of this study is that if the end angles are within about 0.25 degree, the solid-based models still can be used to design, inspect, and fabricate parts.

Conclusions

The conclusion of this study is that any CAD system that supports either PCS or B-spline can be used to reproduce DOE legacy data and to design, inspect, and fabricate parts with confidence. The NWC should move on to these modern CAD systems, knowing that the legacy data generated by the WFS algorithm can be reproduced well within the inspection uncertainty limit.

Introduction

In the NWC, the models-based engineering (MBE) approach has been introduced to design, inspection, and fabrication operations. MBE is based upon a series of commercially available CAD, CAM, and CAE software packages. The DOE NWC selected the PTC Pro/E family of software packages as its de facto standard.

Many questions about the capabilities and accuracy of a software system must be addressed before it can be utilized in the design, inspection, and production environments. Following are some of the major issues associated with changing or introducing a new CAD system into a facility:

- How well do these CAD systems meet DOE’s needs and requirements?
- How explicitly and precisely is geometry represented?
- How easily can geometry be extracted from the data bases?
- How tedious is the learning curve?
- What changes were made to the algorithms from previous versions?
- How well does the new product reproduce the results of past versions?
- What accuracy can be expected from the CAD system algorithms?
- How well do the CAD systems replicate the legacy data previously used in the DOE NWC?

The question addressed in this report is how well the CAD system replicates the legacy data previously used in the DOE NWC. Specifically, the interest is in the two-dimensional spline curve-fitting routines. DOE has utilized a curve-fitting routine referred to as the WFS for several years (late 1960 to 1998); whereas, the modern CAD systems usually use either parametric cubic and/or B-splines.

Purpose of Study

The purpose of this study is to establish whether the NWC-selected CAD system as delivered (PTC Pro/E) has the capability to replicate the DOE legacy data to the required accuracy. If the selected CAD system meets the accuracy requirements, then it can be used to reproduce legacy data. In addition, it can be deployed with confidence to produce future models with at least the same fidelity as the legacy data. A numerical approach is utilized in this study to determine how well the CAD data replicate the legacy data.

Design, Inspection, and Fabrication Metrics

In order to evaluate the selected CAD system, we established metrics on the design uncertainty, inspection uncertainty, and the fabrication uncertainty. The definitions of these uncertainties are as follows:

- Design uncertainty 0.000254 mm (0.00001 in)
- Inspection uncertainty 0.00254 mm (0.0001 in)
- Fabrication uncertainty 0.0254 mm (0.001 in)

What is important is whether the differences in accuracy among the several CAD systems can be detected through the inspection processes. If the differences cannot be measured, then the models developed with the CAD systems are adequate. The results of numerical experiments presented in the following sections are compared to the inspection uncertainty. A ratio of the calculated deviation to the inspection uncertainty is the metric used in this report. If this ratio is less than one, then the models can be used to design, inspect, and fabricate the parts.

Splines

Splines are mathematical representations of a series of points. They are used to interpolate values at intermediate locations. There are many types of two-dimensional splines. They include the following:

- Linear splines
- Classic cubic splines
- Wilson-Fowler cubic splines
- Parametric cubic splines
- B-splines

Several restrictions are placed on the data sets, which are used to build the mathematical models. The points must be in increasing order, and the end slopes must be known. The bounds set on the formulation of the splines are that the mathematical representation must be continuous in position and smooth in slope. The model also must capture the input data points exactly.

In this study, historically used splines are compared with several modern CAD curve-fitting modeling algorithms. The first curve-fitting algorithm is called the WFS,[1] and the second is called a PCS.[2] These splines are presented in the next two sections.

Wilson-Fowler Spline

The WFS was developed at the DOE Oak Ridge Plant in Tennessee in the early 1960s. It is a cubic spline in the local two-dimensional u-v coordinate system with the chord length as the independent parameter, u. Additional references are available for this spline.[3 and 4]

The mathematical representation of one segment of the WFS in local u-v space is summarized below. The nomenclature is very similar to that used in Ref. 3. Equation (1) defines the WFS in the local u-v space.

$$v(u) = C_1u^3 + C_2u^2 + C_3u + C_4 \quad (1)$$

where $0 \le u \le L$,

$C_1 = \frac{TA + TB}{L^2}$,

$C_2 = \frac{-(2TA + TB)}{L}$,

$C_3 = TA$,

and $C_4 = 0$.

Also, $TA =$ entry slope–measured from the chord,

$TB =$ exit slope–measured from the chord,

and $L = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$, chord length.

The total spline consists of a series (number of spline points minus one) of these cubic segments. A curvature-matching technique is utilized to ensure that the spline is continuous in position, slope, and curvature. Both Ref. 1 and Ref. 3 describe the curvature-matching technique in great detail.

If the WFS is to be utilized in an efficient manner, it must be transformed in the global *x*-*y* coordinate system. The transformation equations are listed below. The nomenclature is very similar to that used in Ref. 3.

The global value of *x* as a function of the *u* is defined in Equation (2).

$$x(u) = A_xu^3 + B_xu^2 + C_xu + D_x \quad (2)$$

where $A_x = -C_1 \sin\gamma$,

$B_x = C_2 \sin\gamma$,

$C_x = \cos\gamma - C_3 \sin\gamma$,

and $D_x = X_A$.

The global value of *y* as a function of *u* is defined in Equation (3).

$$y(u) = A_yu^3 + B_yu^2 + C_yu + D_y \quad (3)$$

where $A_y = C_1 \cos\gamma$,

$B_y = C_2 \cos\gamma$,

$C_y = C_3 \cos\gamma + \sin\gamma$,

and $D_y = Y_A$.

The end points of the segment are defined below.

Where $\gamma =$ chord angle–measured from the global x-axis,

$$\gamma = \tan^{-1}\frac{Y_B - Y_A}{X_B - X_A},$$

$X_A, Y_A =$ beginning chord point,

and $X_B, Y_B =$ ending chord point.

The total spline consists of a series (number of spline points minus one) of these cubic segment pairs.

A complete listing of the FORTRAN source code used to calculate the segment coefficients is given in Appendix A—WFS Routines.

Parametric Cubic Spline

The PCS presented here is smooth in the first derivative and continuous in the second derivative, both within the segment and at the spline points. Ref. 2 gives a complete derivation of the PCS.

The global value of x as a function of t (chord length) is defined in Equation (4).

$$x(t) = A_x t^3 + B_x t^2 + C_x t + D_x \tag{4}$$

where $0 \le t \le L$,

$$A_x = \frac{X''(L) - X''(0)}{6L},$$

$$B_x = \frac{X''(0)}{2},$$

$$C_x = \frac{X_B - X_A}{L} - \frac{L}{6}(2X''(0) + X''(L)),$$

and $D_x = X_A$.

Also, $X''(0) =$ value of the second derivative at the beginning of the segment,

and $X''(L) =$ value of the second derivative at the ending of the segment.

The global value of y as a function of t (chord length) is defined in Equation (5).

$$y(t) = A_y t^3 + B_y t^2 + C_y t + D_y \tag{5}$$

where $A_y = \dfrac{Y''(L) - Y''(0)}{6L}$,

$$B_y = \frac{Y''(0)}{2},$$

$$C_y = \frac{Y_B - Y_A}{L} - \frac{L}{6}(2Y''(0) + Y''(L)),$$

and $D_y = Y_A$.

Also, $Y''(0) =$ value of the second derivative at the beginning of the segment,

and $Y''(L) =$ value of the second derivative at the ending of the segment.

The end points of the segment are defined below.

Where $X_A, Y_A =$ beginning chord point,

$X_B, Y_B =$ ending chord point,

and $L = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$, chord length.

The total spline consists of a series (number of spline points minus one) of these cubic segment pairs.

A complete listing of the FORTRAN source code used to calculate the segment coefficients is given in Appendix B—PCS Routines.

Computer-Aided Design Systems

The DOE national laboratories have been working with CAD, CAM, and CAE tools since the early 1980s. Many of the original CAD tools were designed using a kernel written by Patrick J. Hantatty.[5] These tools were primarily electronic drafting boards. As the CAD systems matured, the next functionality developed was to integrate the manufacturing and machine control using CAM. CADDS was one of the early codes that went though this growth cycle. The tools and computers on which to run them were very expensive. Very few companies were able to afford them. The DOE national laboratories, Boeing, General Motors, and NASA were the type of organizations that purchased these early CAD/CAM tools. In the late 1980s and early 1990s, CAD companies started to develop three-dimensional modeling tools. A few new companies focusing on designing three-dimensional solid models were established. These tools were intended to be used by engineers to design and visualize the products in three dimensions. A solid model could be shaded and spun so that it was visible from any direction. Because the computer program could detect what was material and what was not, the mass properties of the three-dimensional components could be calculated very easily. Pro/E from Parametric Technology is one of these systems.

The nuclear weapons program in the United Kingdom uses Computervision CADDS as its CAD/CAM/CAE tool. The United States DOE NWC uses Pro/E as a de facto standard. Both tools are competitive in the global CAD/CAM/CAE market and are capable of doing all the design and manufacturing work for complex design. One difference between the two companies is the way they compute complex curves and surfaces. Computervision CADDS responded to a request from the NWC in the early 1980s and used the WFS algorithm that was derived by the DOE to define complex curves. Pro/E used industry-standard PCS and B-spline algorithms in its code. CDC's ICEM DDN system was also utilized in the NWC. This CAD system was used at the Rocky Flats Plant in the 1980s and early 1990s to manufacture many of the NWC parts.

Several CAD systems are and have been used in the DOE NWC. The three that are of interest in this study are the Computervision CADDS system, the Pro/E system, and the ICEM DDN system. Below are introductory statements drawn from the CAD/CAM/CAE companies' web pages.

(The legacy data and manufacturing models of interest in this report were developed on earlier versions of these codes. In 1998, PTC purchased Computervision's CADDS and CDC's ICEM DDN systems. For that reason the CAD systems listed below, taken from PTC's Web site, have different names.)

PTC CADDS 5i

CADDS 5i is the current release of the PTC product. It is a fully functional CAD/CAM/CAE system able to create complex geometry. The tool has integrated manufacturing and communication tools to allow many users to work on the same models. CADDS software predated the parametric modeling concepts. This system is able to model without connecting dimensions and features. To alter designs, one must explicitly remodel the geometry. The system is also capable of modeling in parametric form such that one can modify a dimension and the part will redefine the geometry based on the new dimension. Because of the evolutionary changes of this CAD system, it can create wire-frame, surface, or solid-model geometry for the three-dimensional design of any component.

PTC Pro/E

Pro/E is a fully functional, parametric, three-dimensional, solid-modeling system. This CAD system started in the 1980s and was focused on parametric design with three-dimensional solid modeling from its inception. Pro/E is integrated with its manufacturing, inspection, and analysis packages. Information does not need to be transferred or translated to work in any of the packages. Pro/E uses parametric design for all geometry creation. When geometry is created, its dimensions parametrically drive the model. The documentation of the solid model is a two-dimensional drawing that is associated with the three-dimensional solid model. Dimensions can be modified in any location and will be updated in the model and drawing. The geometry of the model is also recalculated and redisplayed.

PTC ICEM DDN

CDC's ICEM, now a subsidiary of PTC, provides vital advanced styling and surfacing technology and expertise to major automotive- and consumer-product manufacturers. Established in the early 1980s, ICEM Technologies joined PTC in June 1998 as the ICEM Surfacing Center of Excellence. ICEM DDN is a fully functional CAD/CAM/CAE software. ICEM DDN started in the 1970s as a two-dimensional system to perform electronic drafting. This system, like Computervision, has evolved over the last 30 years, expanding from two-dimensional drafting into wire-frame, surface, and solid modeling. ICEM DDN has also developed the manufacturing tools to do CAM work. As in Pro/E, the design and manufacturing are integrated, and the users are able to work in either package without transfer or translation of the data. The ICEM DDN system utilizes a version of the WFS algorithm.

Evaluation Programs

We used two analysis programs in this study—WX-Division Integrated Software Tools, and Keyword Graphics Builder Program. Descriptions of these programs are presented in the following sections.

Integrated Software Tools

This study began by using the WX-Division Integrated Software Tools[6] (WIST) developed at Los Alamos National Laboratory (LANL). The WIST capabilities included the ability to rotate, mirror, and translate spline data; to calculate data normal to curves; to determine the minimum distance of data points from the spline representation of the data; and to calculate the WFS coefficients from the spline points. A graphics package was not available in WIST.

A peer review of WIST was performed. The WFS algorithms were reviewed. Minor changes were made to the source code. These changes were limited to making all the variables double precision, changing the convergence tolerances, and increasing the number of loops in the iteration algorithms. All the changes made to the source code were provided with comment blocks that started and ended with “Cwb” (which stands for “Comment by Wilbur Birchler”).

In addition, the minimum-distance algorithm was reviewed. A searching technique was utilized to determine which segment encompassed the data point. Problems arose when the data point was past the ends of the spline. No warnings were issued by the program. In addition, this technique would sometimes select the wrong segment. The approach used was very computationally intensive.

Because of the peer review and after evaluating many test cases, we concluded that only the coefficient generation part of the WIST software package should be used in this study. The FORTRAN source code is listed in Appendix A—WFS Routines.

Keyword Graphics Builder Program

After performing many analyses, going over the peer reviews, and determining that WIST had no graphics capabilities, we decided to use the Keyword Graphics Builder Program[7] (KGB) to evaluate the CAD systems. KGB was developed at LANL, already had graphics options, and had many analysis capabilities. This software is written in several modules, and it is very easy to add new capabilities as needed. The program’s capabilities include standard mathematical operations such as adding, subtracting, multiplying, dividing, integrating, and differentiating two-dimensional data sets. Other data operations available are maximum/minimum value reporting, data chopping, data smoothing, data extrapolating, data merging, data scaling, data mirroring, and data translating. An excellent report-graphics package and presentation-quality graphics are major parts of this software.

KGB is a command-line-driven program. Commands can be entered interactively and/or from a command file. The Help File is on line and has several levels of detail. It utilizes an extensive Label File for graphics, for tracking units, for converting between units, and for assuring that all the operations are consistent. This software has excellent error-trapping and error-warning capabilities.

This program had a complete set of data-fitting algorithms before the WFS module (Appendix A—WFS Routines) from WIST was added. PCS capability was added to allow for direct comparisons with the WFS algorithms. The relevant FORTRAN software is listed in Appendix B—PCS Routines.

A new minimum-distance algorithm was developed and programmed. This new algorithm is an iterative solution based on the fact that the minimum distance of a point from a curve is normal to the curve. The FORTRAN software we used is listed in Appendix C—Minimum-Distance Routines.

Accuracy Study of Minimum-Distance Algorithms

The first step in evaluating CAD systems is to ensure that the selected evaluation (KGB, in this case) is accurate and correct. We took a numerical approach to this step, selecting and evaluating three analytical shapes. The results of this evaluation are presented in the following sections.

Goals

The goals of this study were to establish the accuracy of the minimum-distance software and to demonstrate that the directions of the deviations were correct.

Analytical Shapes: Circle, Ellipse, and Parabola

Three analytical shapes were utilized to determine how well the minimum-distance algorithm worked. These analytical shapes were a circle, an ellipse, and a parabola. The spline data and the evaluation data were generated with Mathcad.[8] Equations were written for each shape. We generated spline data every two degrees from the x-axis to the y-axis. The spline data points are listed in Appendix D—Analytical Spline-Point Data. All data were rounded to six digits past the decimal point.

The procedure used to evaluate the minimum-distance algorithm was as follows:

- Generate the mathematical representations of the spline points for both WFS and PCS
- Generate three sets of evaluation data for each shape
- Develop Data Set 1—exact data on the curves
- Develop Data Set 2—exact data offset normal to the base curves by a positive value (increased radius)
- Develop Data Set 3—exact data offset normal to the base curves by a negative value (decreased radius)
- Calculate the minimum distances of the exact data from the mathematical representation results
- Summarize the results
- Compare the results

Spline Data

Three analytical shapes were utilized to establish the accuracy of the minimum-distance algorithm. These analytical shapes were a circle, an ellipse, and a parabola. The equations used to generate the spline data were programmed in Mathcad and are listed in the following three sections.

Analytical Circle Equations

The global x and y values as a function of θ are defined by Equation (6) and Equation (7), respectively.

$$x_k = R_0 \cos(\theta_k) \tag{6}$$

and

$$y_k = R_0 \sin(\theta_k) \tag{7}$$

where $\theta_k = \frac{\pi}{180.0} 2k$,

$k = 0,1;45$,

and $R_0 = 100$ mm.

Analytical Ellipse Equations

The global x and y values as a function of β are defined by Equation (8) and Equation (9), respectively.

$$x_k = a_0 \cos(\beta_k) \tag{8}$$

and

$$y_k = b_0 \sin(\beta_k) \tag{9}$$

Where $\beta_k = \tan^{-1}[\frac{a_0 \sin(\theta_k)}{b_0 \cos(\theta_k)}]$,

$\theta_k = \frac{\pi}{180.0} 2k$,

$k = 0,1;45$,

$a_0 = 100$ mm,

and $b_0 = 80$ mm.

Analytical Parabolic Equations

The global x and y values as a function of θ are defined by Equation (10) and Equation (11), respectively.

$$x_k = [\frac{\cot(\theta_k) + \sqrt{\cot^2(\theta_k) - 4d}}{2c}]\cot(\theta_k) \tag{10}$$

and

$$y_k = \frac{\cot(\theta_k) + \sqrt{\cot^2(\theta_k) - 4d}}{2c} \tag{11}$$

where $\theta_k = \frac{\pi}{180.0} 2k$,

$k = 0,1;45$,

$c = \frac{-x_p}{y_e^2}$,

$d = x_p$,

$x_p = 100\,\text{mm}$,

and $y_e = 60\,\text{mm}$.

Spline Data Set Parameters and Plots

Table 1 lists the spline data set parameters of these analytical shapes. The number of points, point spacing, number of digits past the decimal point, and the end angles are summarized.

Table 1. Analytical Shapes: Spline Data Set Parameters

Analytical Spline	Number of Points	Spacing of Points (Degrees)	Digits Past Decimal Point	Beginning End-Angle (Degrees)	Ending End-Angle (Degrees)
Circle	46	2.0	6	90.0	180.0000
Ellipse	46	2.0	6	90.0	180.0000
Parabola	46	2.0	6	90.0	163.3008

These three analytical shapes are shown in Figure 1. The naming conventions for files are as follows:

- First field—shape
    - Circle
    - Ellipse
    - Parabola
- Second field—normal offset direction
    - _n—no normal offset
    - _p—plus normal offset (increasing radius)
    - _m—minus normal offset (decreasing radius)

- Third field—type of file
    - .spn—spline points
    - .pts—evaluation points

Figure 1 shows the three spline curves. The solid red curve is the analytical circle. The analytical ellipse is shown as the small-dash black curve. The large-dash green curve is the analytical parabola.

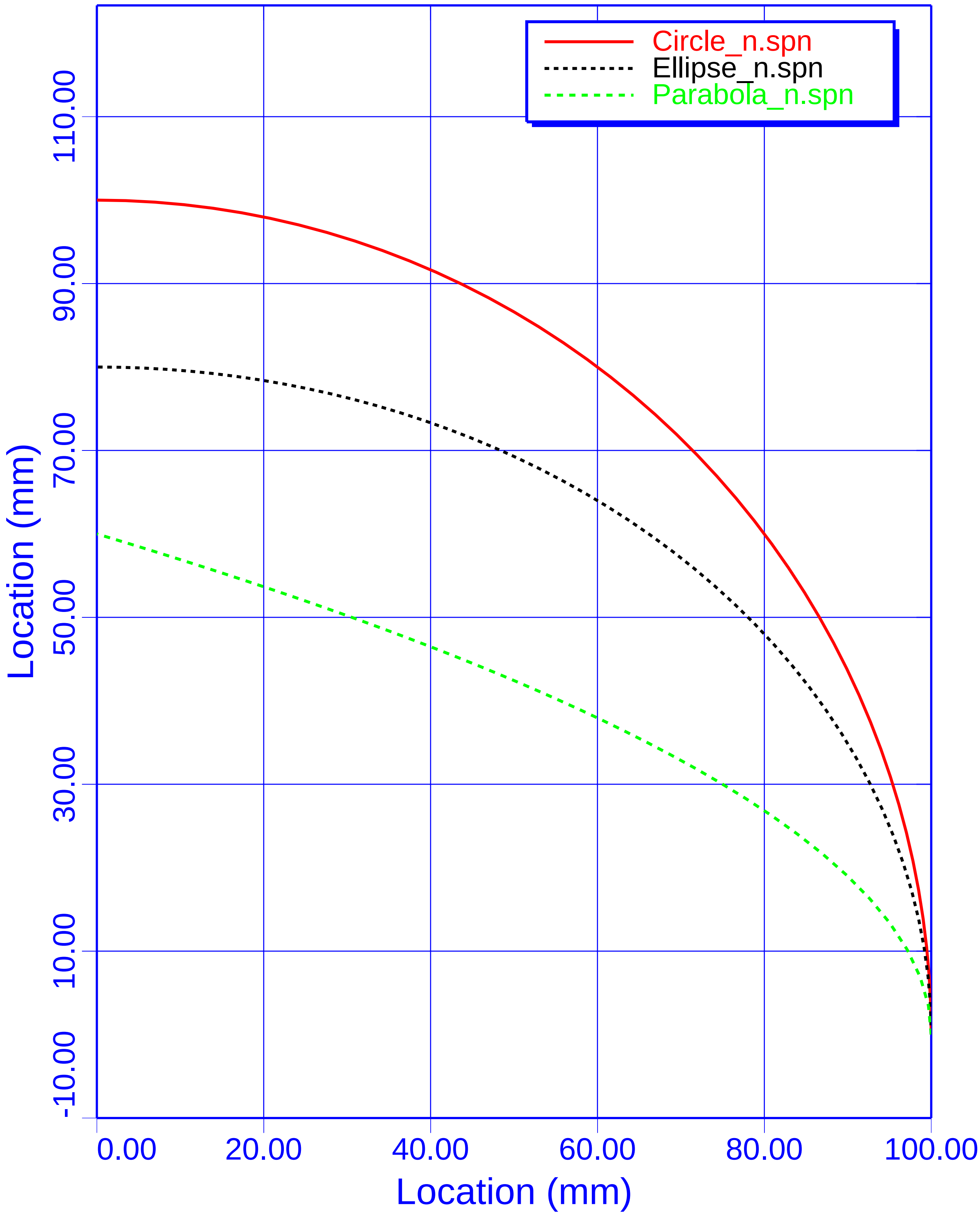

Figure 1. Analytical Spline Data: Circle, Ellipse, and Parabola

Evaluation Data

In order to evaluate the accuracy of the minimum-distance software, three "exact" data sets were generated for each of the analytical shapes. The first data set has a normal offset of 0.0000 mm. The second and third data sets have normal offsets of +0.0254 mm and –0.0254 mm. The equations used to generate the evaluation data are listed in the following three sections. Mathcad was used to generate all the evaluation points.

Analytical Circle Equations

The global x and y values as a function of θ are defined by Equation (12) and Equation (13), respectively.

$$x_j = (R_0 \pm \Delta)\cos(\theta_j) \tag{12}$$

and

$$y_j = (R_0 \pm \Delta)\sin(\theta_j) \tag{13}$$

where $\theta_j = \frac{\pi}{180.0} 0.25 j$,

$j = 0,1;360$,

$R_o = 100$,

and $\Delta = 0.0254\text{mm}$.

Analytical Ellipse Equations

The global x and y values as a function β are defined by Equation 14 and Equation 15, respectively.

$$x_j = a_o \cos(\beta_j) \pm \frac{\Delta \cos(\beta_j)}{\sqrt{(a_o \sin(\beta_j))^2 + (b_o \cos(\beta_j))^2}} \tag{14}$$

and

$$y_j = b_o \sin(\beta_j) \pm \frac{\Delta \sin(\beta_j)}{\sqrt{(a_o \sin(\beta_j))^2 + (b_o \cos(\beta_j))^2}} \tag{15}$$

where $\beta_j = \tan^{-1}[\frac{a_0 \sin(\theta_j)}{b_0 \cos(\theta_j)}]$,

$\theta_j = \frac{\pi}{180.0} 0.25 j$,

$$j = 0,1;360,$$

$$a_o = 100\text{mm},$$

$$b_o = 80\text{mm},$$

and $$\Delta = 0.0254\text{mm}.$$

Analytical Parabola Equations

The global x and y values as a function of θ are defined by Equation (16) and Equation (17), respectively.

$$x_j = [\frac{\cot(\theta_j) + \sqrt{\cot^2(\theta_j) - 4d}}{2c}]\cot(\theta_j) \pm \Delta\cos(\zeta_j) \quad (16)$$

and

$$y_j = \frac{\cot(\theta_j) + \sqrt{\cot^2(\theta_j) - 4d}}{2c} \pm \Delta\sin(\zeta_j) \quad (17)$$

where $$\zeta_j = \frac{\pi}{2} - \tan^{-1}(\frac{-1}{\cot(\theta_j) + \sqrt{\cot^2(\theta_j) - 4d)}}),$$

$$j = 0,1;360,$$

$$c = \frac{-x_p}{y_e^2},$$

$$d = x_p,$$

$$x_p = 100\text{mm},$$

$$y_e = 60\text{mm},$$

and $$\Delta = 0.0254\text{mm}.$$

Evaluation Data Set Parameters

Table 2 lists the parameters of these analytical shapes. The number of points, point spacing, number of digits past the decimal point, and the normal offsets are summarized. Evaluation numbers were generated every 0.25 degree, using the equations listed in the above three sections.

Table 2. Analytical Shapes: Evaluation Data Set Parameters

Analytical Spline	Number of Points	Spacing of Points (Degree)	Digits Past Decimal Point	Normal Offset (mm)
Circle	361	0.25	6	0.0
Circle	361	0.25	6	+0.0254
Circle	361	0.25	6	–0.0254
Ellipse	361	0.25	6	0.0
Ellipse	361	0.25	6	+0.0254
Ellipse	361	0.25	6	–0.0254
Parabola	361	0.25	6	0.0
Parabola	361	0.25	6	+0.0254
Parabola	361	0.25	6	–0.0254

Keyword Graphics Builder Program Command File

The command files utilized to perform the following calculations are listed in Appendix F—Keyword Graphics Builder Program— Command files—Analytical Shapes—Accuracy Study.

Analytical Circle

The next six figures show the comparisons of WFS and PCS representations with the exact analytical circle data generated by Mathcad. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the fabrication uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 1,000.

Deviation Plots

Figure 2 is a graph of the analytical circle modeled with the WFS algorithm. The normal offset is 0.0000 mm. Mathcad was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.0613e–6 mm.

Figure 3 is a graph of the analytical circle modeled with the PCS algorithm. The normal offset is 0.0000 mm. Mathcad was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 1.3075e–6 mm.

Figure 4 is a graph of the analytical circle modeled with the WFS algorithm. The normal offset is +0.0254 mm. Mathcad was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.5401e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 5 is a graph of the analytical circle modeled with the PCS algorithm. The normal offset is +0.0254 mm. Mathcad was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 2.5401e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 6 is a graph of the analytical circle modeled with the WFS algorithm. The normal offset is –0.0254 mm. Mathcad was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.5402e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 7 is a graph of the analytical circle modeled with the PCS algorithm. The normal offset is –0.0254 mm. Mathcad was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 2.5401e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

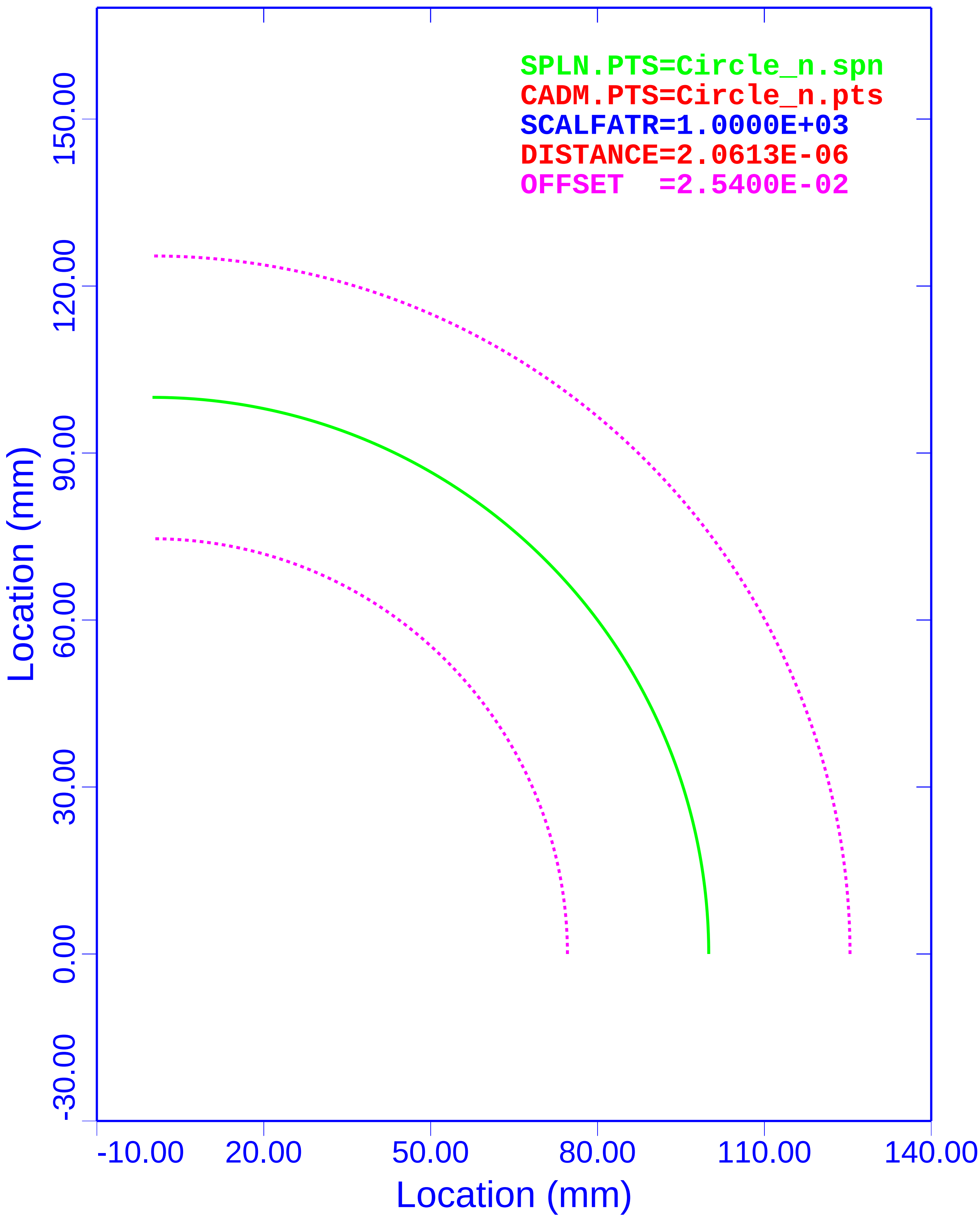

Figure 2. Analytical Circle: WFS, Deviations with 0.0000 mm Normal Offset

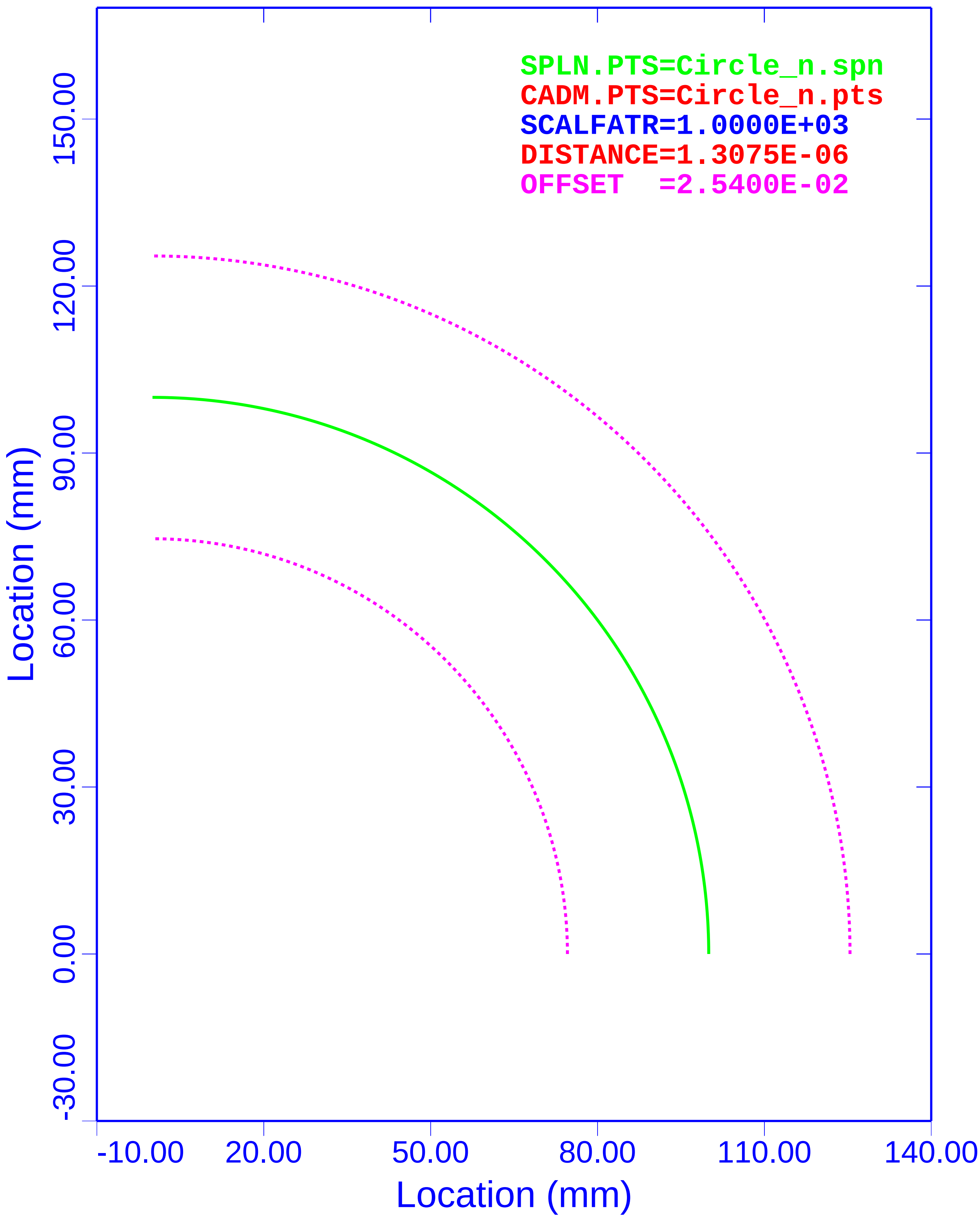

Figure 3. Analytical Circle: PCS, Deviations with 0.0000 mm Normal Offset

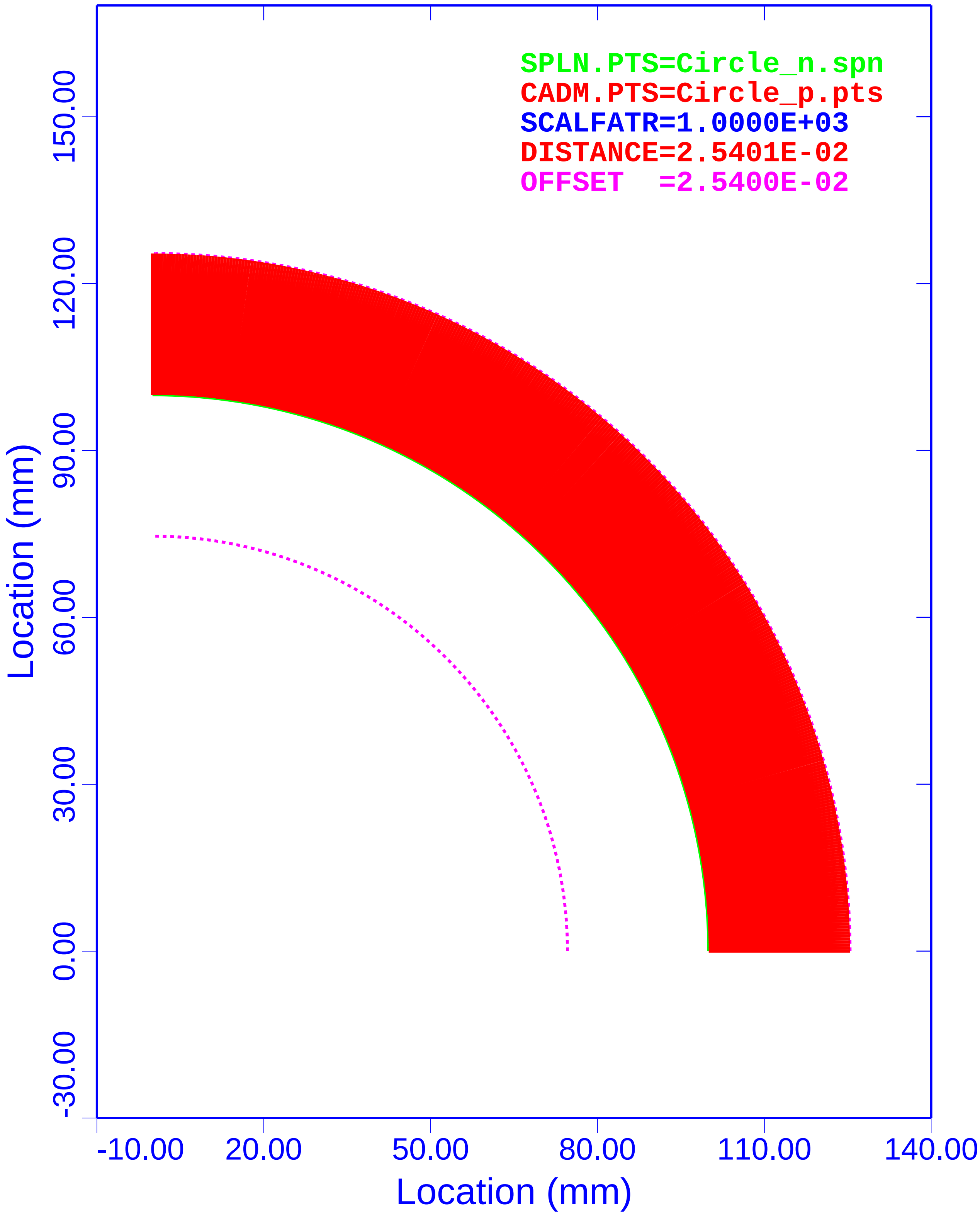

Figure 4. Analytical Circle: WFS, Deviations with +0.0254 mm Normal Offset

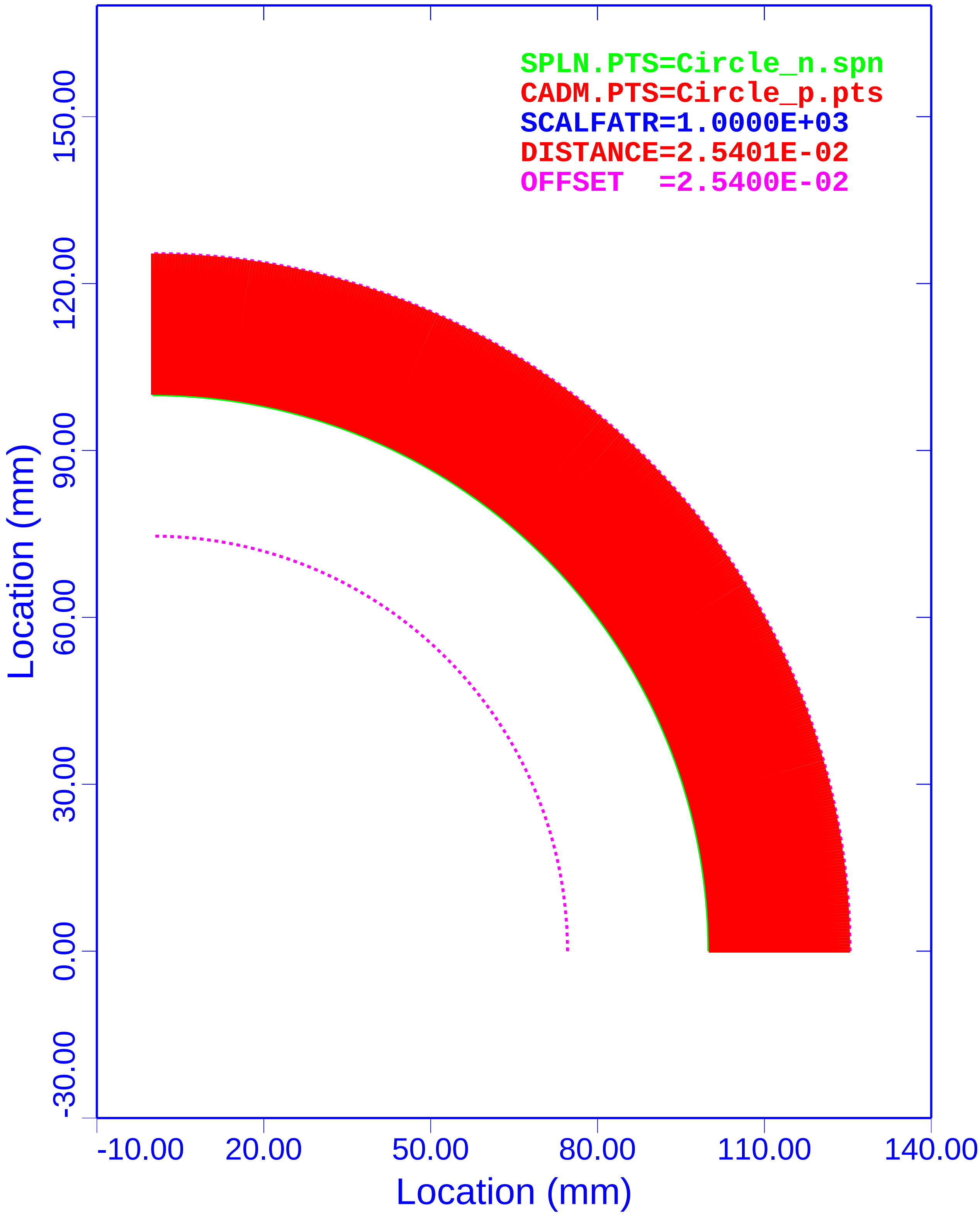

Figure 5. Analytical Circle: PCS, Deviations with +0.0254 mm Normal Offset

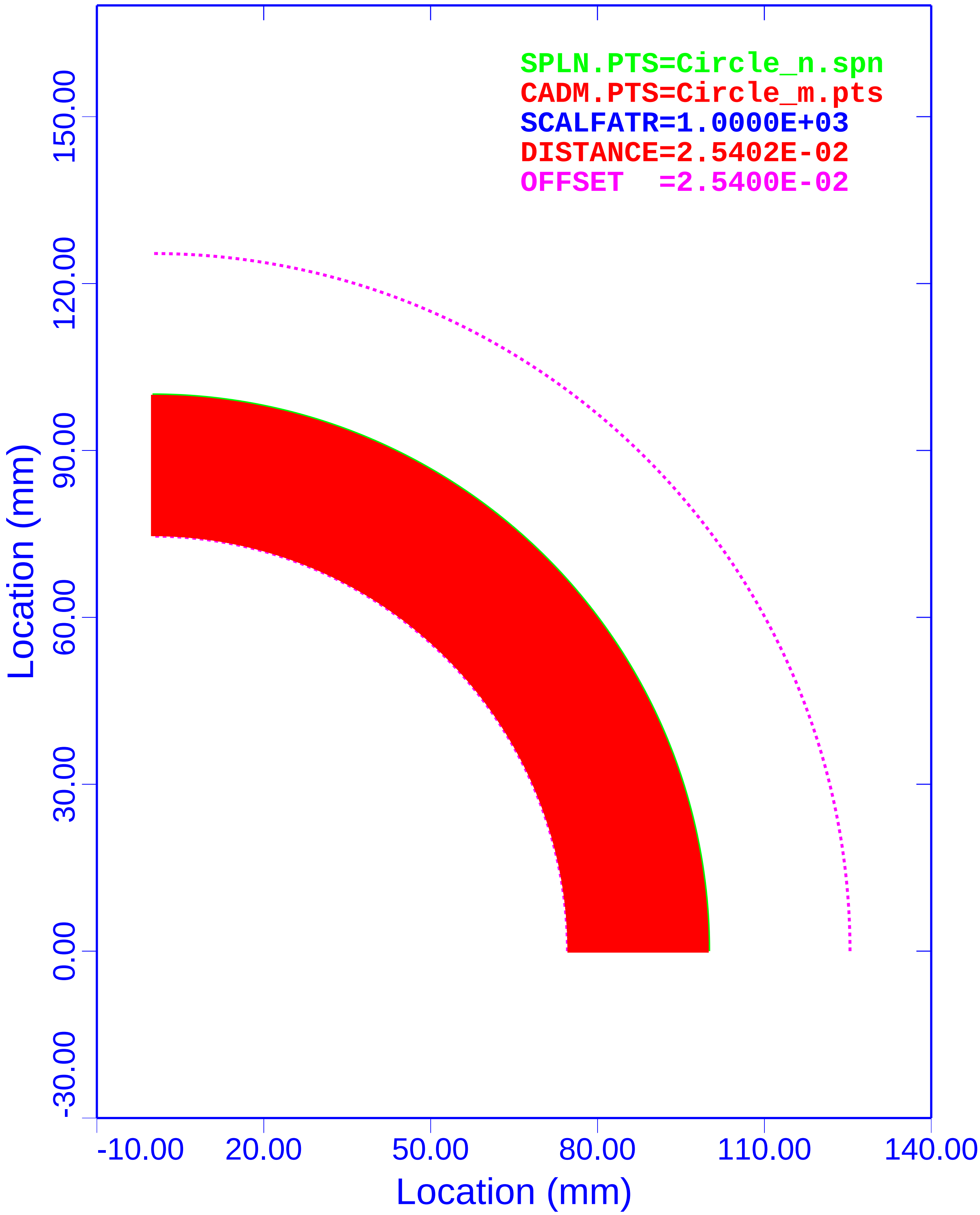

Figure 6. Analytical Circle: WFS, Deviations with –0.0254 mm Normal Offset

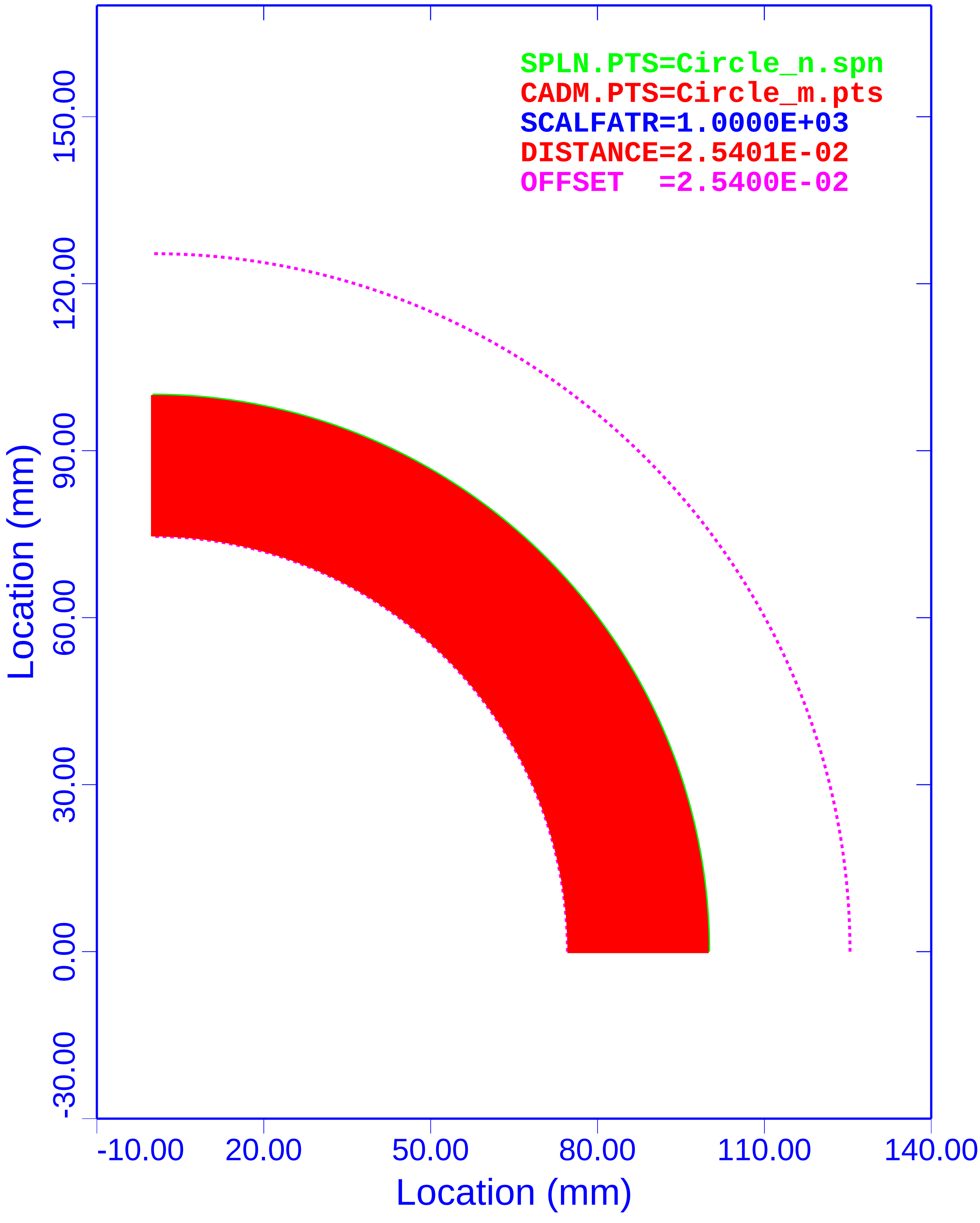

Figure 7. Analytical Circle: PCS, Deviations with –0.0254 mm Normal Offset

Maximum and Minimum Deviation Summary

Table 3 is a summary of the maximum and minimum deviations for the analytical circle. Column 1 lists the associated figure in this document that displays the results. The second column lists the type of evaluation spline. Columns 3 and 4 are the maximum and minimum deviations with the normal offset included. The fifth column gives the normal offset values. Columns 6 and 7 are the deviations with the normal offsets removed.

Table 3. Analytical Circle: Summary of Deviations

Figure	Spline Type	Maximum Deviation (mm)	Minimum Deviation (mm)	Normal Offset (mm)	Maximum Deviation with Offset Removed (mm)	Minimum Deviation with Offset Removed (mm)
Figure 2	WFS	+5.980261e–7	–2.061326e–6	+0.0000	+5.980261e–7	–2.061326e–6
Figure 3	PCS	+1.307488e–6	–9.389013e–7	+0.0000	+1.307488e–6	–9.389013e–7
Figure 4	WFS	+2.540065e–2	+2.539813e–2	+0.0254	+6.500000e–7	–1.870000e–6
Figure 5	PCS	+2.540140e–2	+2.539915e–2	+0.0254	+1.400000e–6	–8.500000e–7
Figure 6	WFS	–2.539907e–2	–2.540206e–2	–0.0254	+9.300000e–7	–2.060000e–6
Figure 7	PCS	–2.539966e–2	–2.540083e–2	–0.0254	+3.400000e–7	–8.300000e–7

A review of columns 6 and 7 of Table 3 shows that the absolute largest deviation is associated with the WFS algorithm and has a value of 2.061326e–6 mm (Figure 2). The ratio of the calculated deviation to the inspection uncertainty is 8.115e–4.

In column 6 of Table 3, Maximum Deviation with Offset Removed, the WFS deviations range from +5.9++e–7 mm to +9.3++e–7 mm. Similarly, the PCS deviations range from +3.4++e–7 mm to +1.4++e–6 mm. The WFS model yields slightly better results than the PCS model for maximum deviations.

In column 7 of Table 3, Minimum Deviation with Offset Removed, the WFS deviations are almost identical and have values of –2.0++e–6 mm. Similarly, the PCS deviations are almost identical with values of –8.5++e–7 mm. The WFS model yields slightly poorer results than the PCS model for minimum deviations.

The results shown in columns 6 and 7 of Table 3 indicated that a minimum accuracy of about five and three-quarters digits past the decimal point could be expected for an analytical circle. In general, the accuracy of these splines is the number of digits past the decimal point minus one-quarter digit.

In column 6 of Table 3, Maximum Deviation with Offset Removed (mm), the WFS deviations are about an order of magnitude better than the PCS deviations. However, the deviations listed in column 7 of the same table, Minimum Deviation with Offset Removed (mm), show the opposite results.

The results of these analyses show that the WFS and PCS produce results that meet all of DOE’s MBE requirements.

A review of the six figures listed in Table 3 shows that the minimum-distance algorithm produces the proper sign on the deviations.

Analytical Ellipse

The next six figures show the comparisons of WFS and PCS representations with the exact analytical ellipse data generated by Mathcad. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the fabrication uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 1,000.

Deviation Plots

Figure 8 is a graph of the analytical ellipse modeled with the WFS algorithm. The normal offset is 0.0000 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 3.4457e–6 mm.

Figure 9 is a graph of the analytical ellipse modeled with the PCS algorithm. The normal offset is 0.0000 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 4.4447e–6 mm.

Figure 10 is a graph of the analytical ellipse modeled with the WFS algorithm. The normal offset is +0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.5401e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 11 is a graph of the analytical ellipse modeled with the PCS algorithm. The normal offset is +0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 2.5405e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 12 is a graph of the analytical ellipse modeled with the WFS algorithm. The normal offset is –0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.5403e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 13 is a graph of the analytical ellipse modeled with the PCS algorithm. The normal offset is –0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 2.5401e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

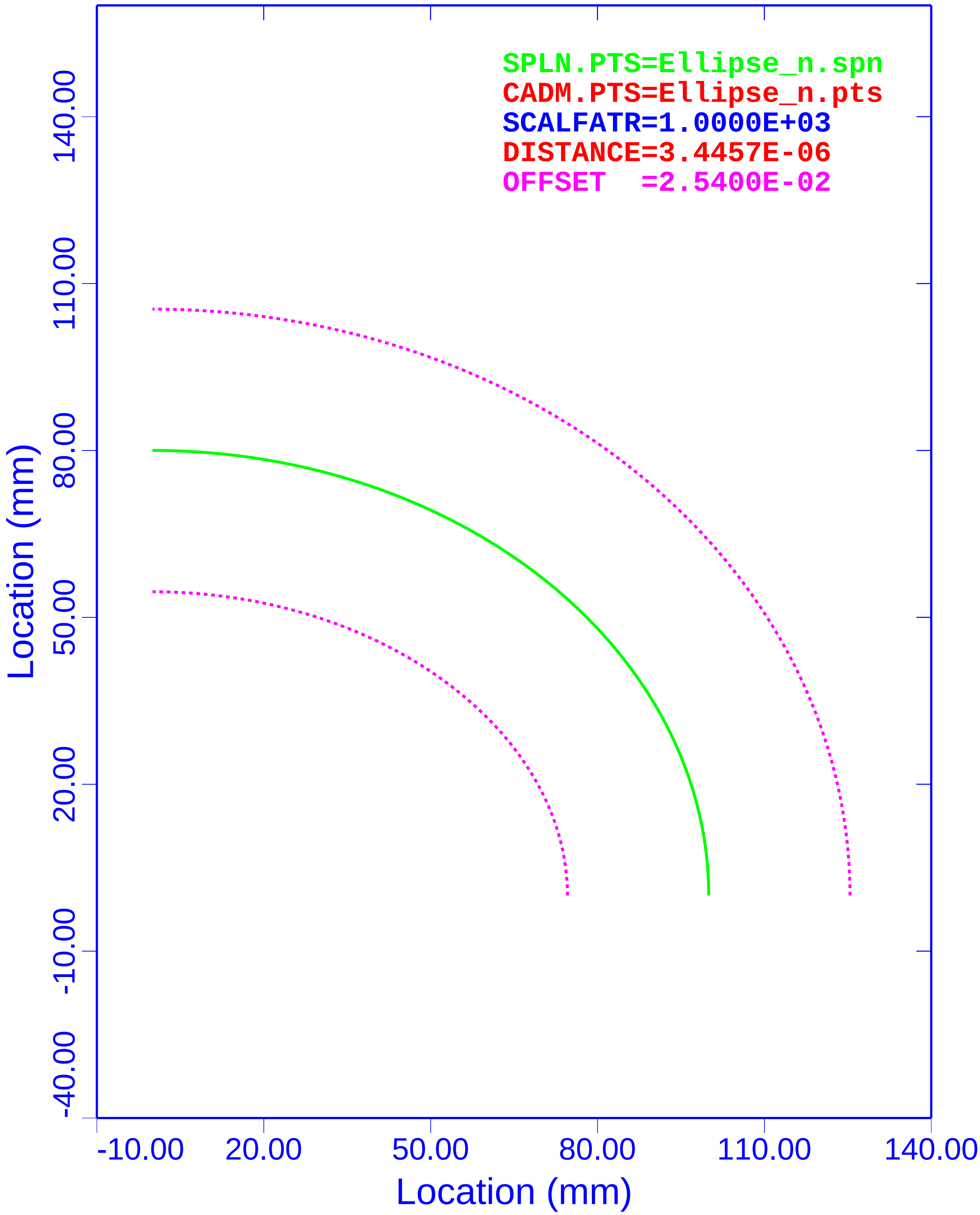

Figure 8. Analytical Ellipse: WFS, Deviations with 0.0000 mm Normal Offset

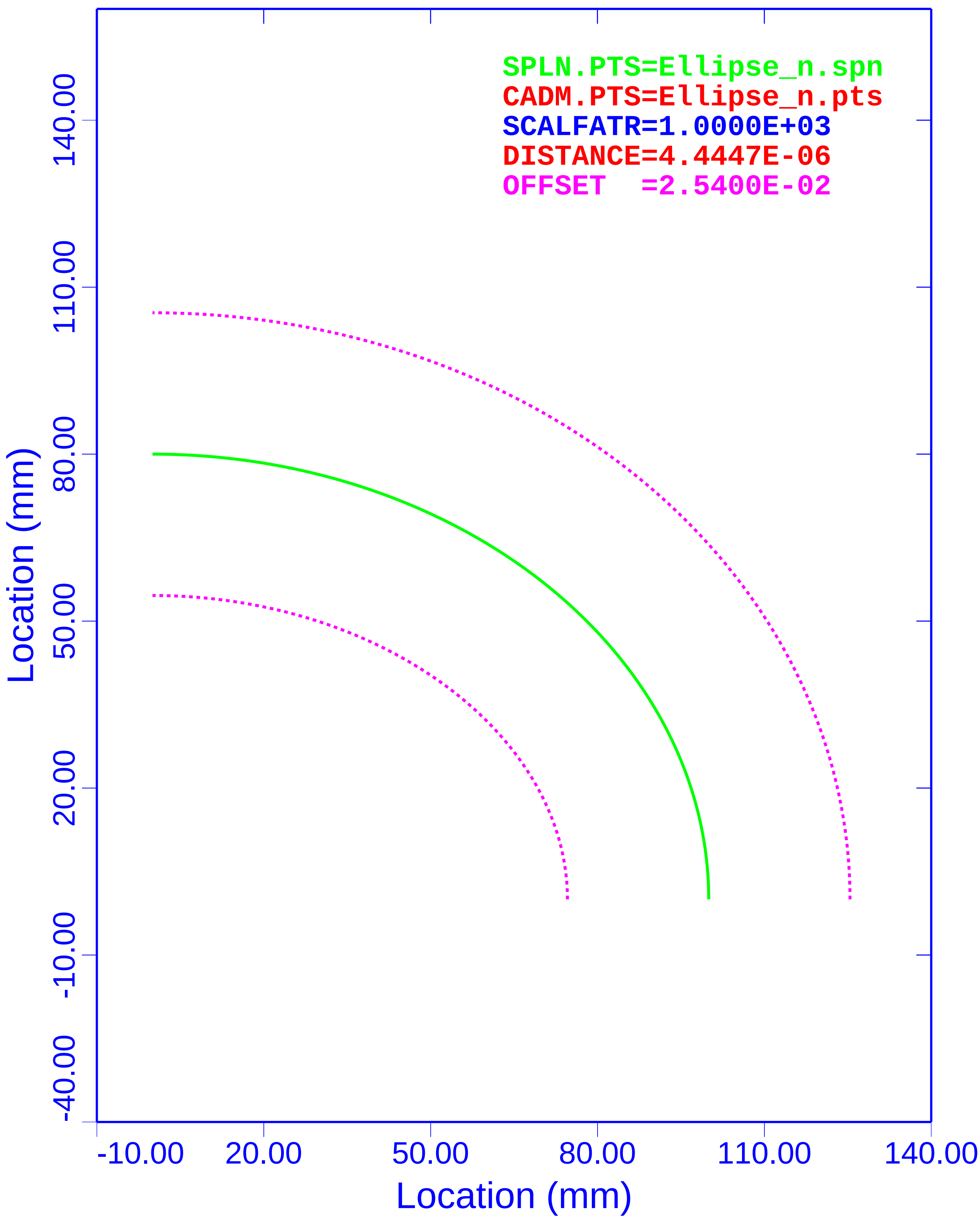

Figure 9. Analytical Ellipse: PCS, Deviations with 0.0000 mm Normal Offset

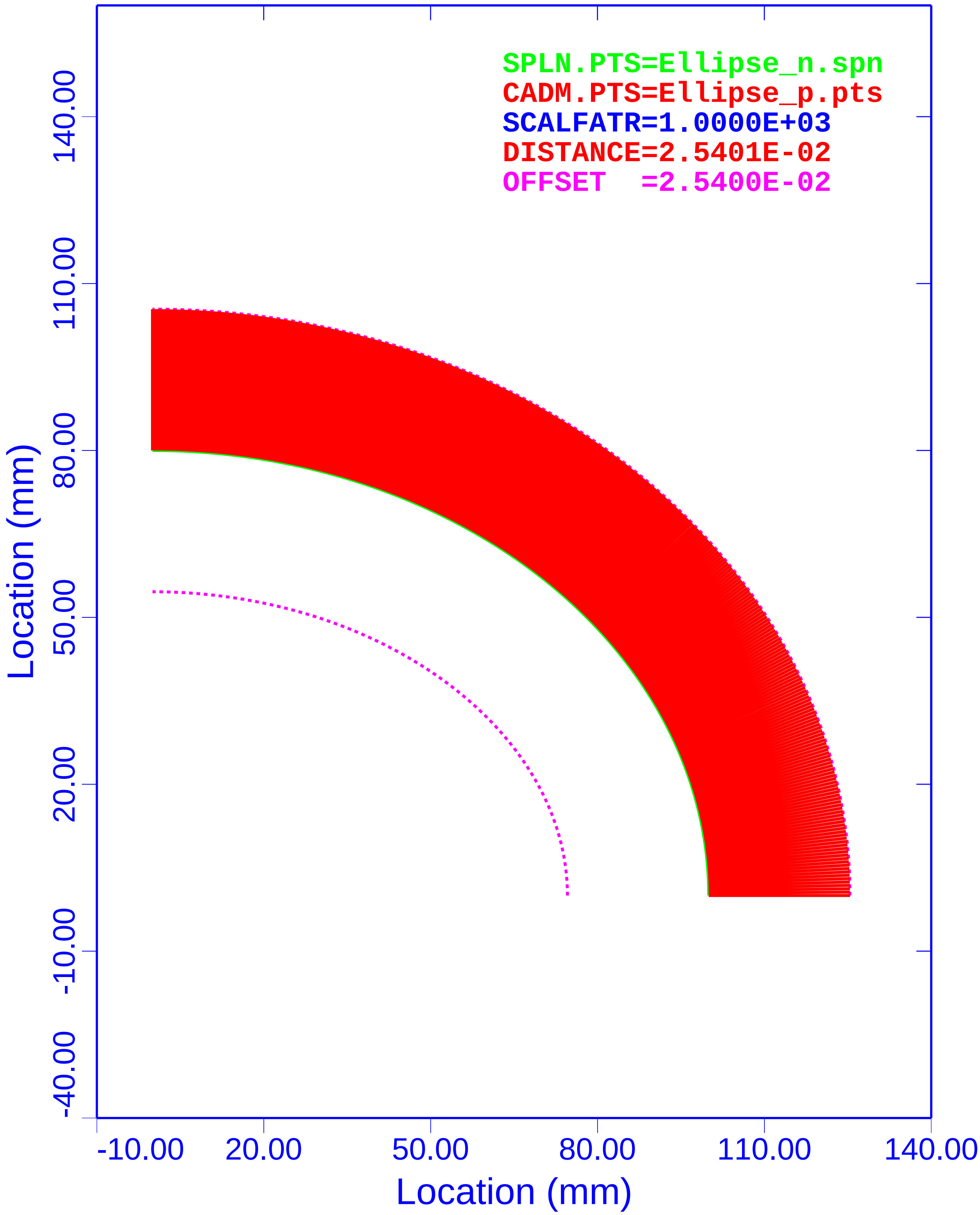

Figure 10. Analytical Ellipse: WFS, Deviations with +0.0254 mm Normal Offset

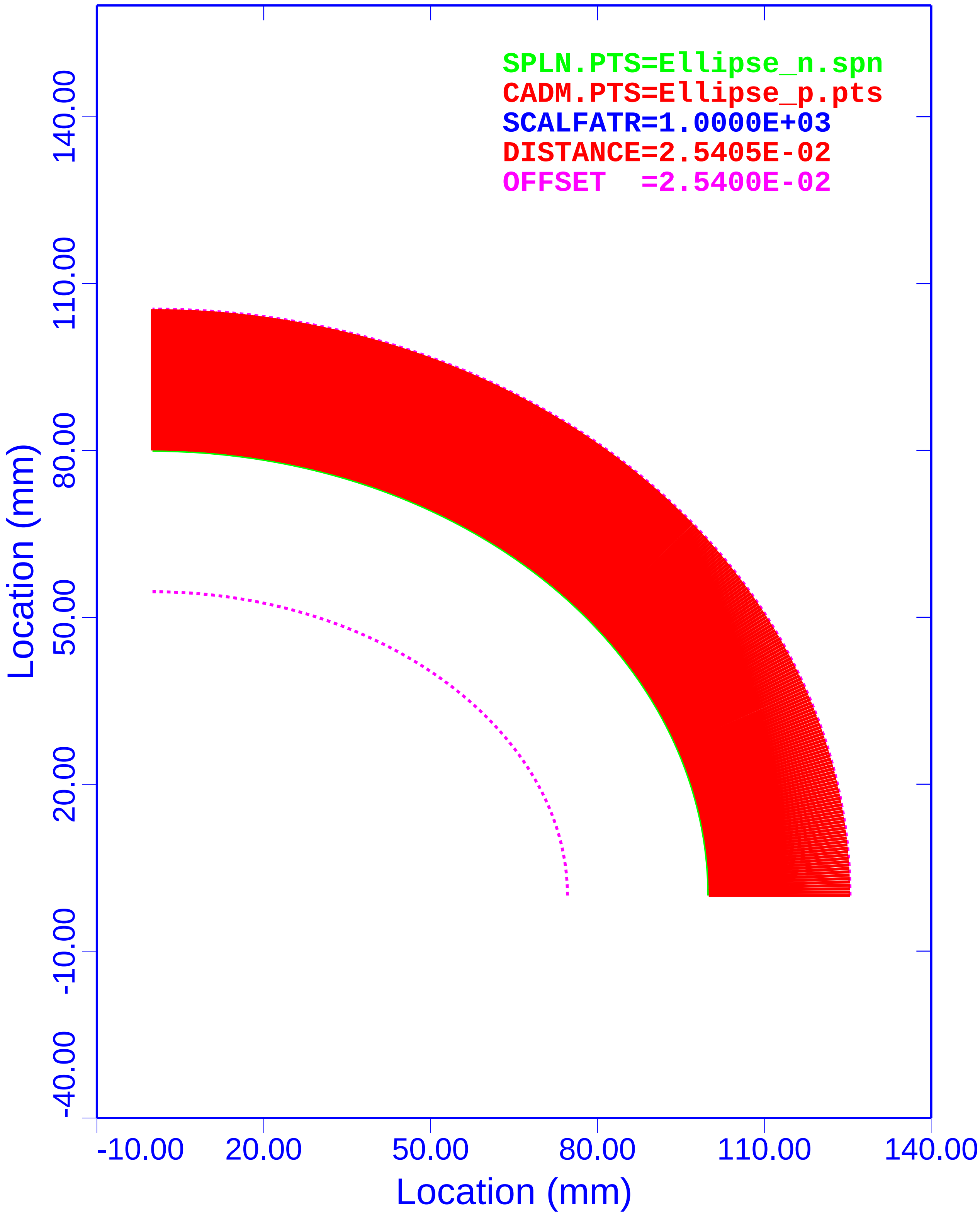

Figure 11. Analytical Ellipse: PCS, Deviations with +0.0254 mm Normal Offset

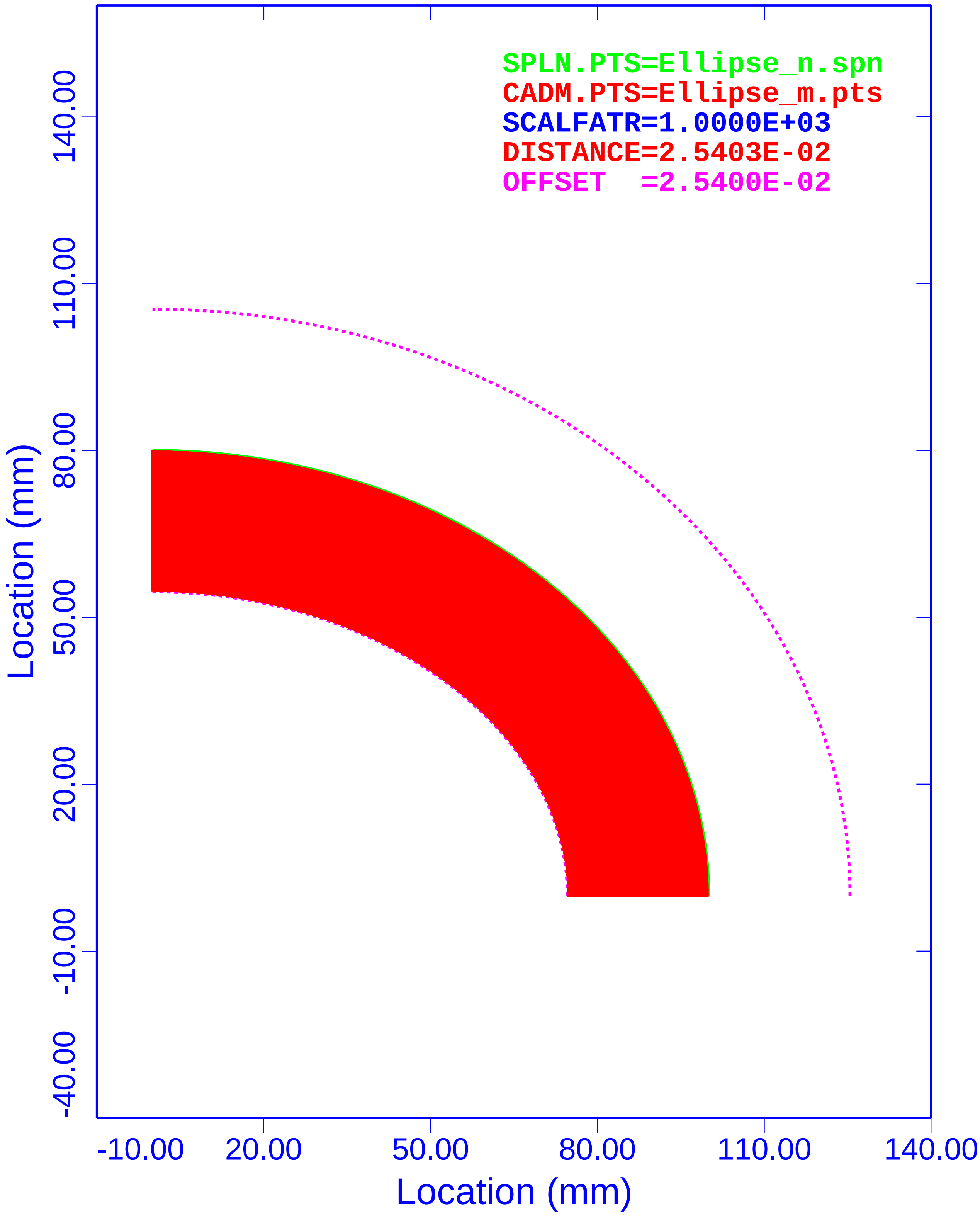

Figure 12. Analytical Ellipse: WFS, Deviations with –0.0254 mm Normal Offset

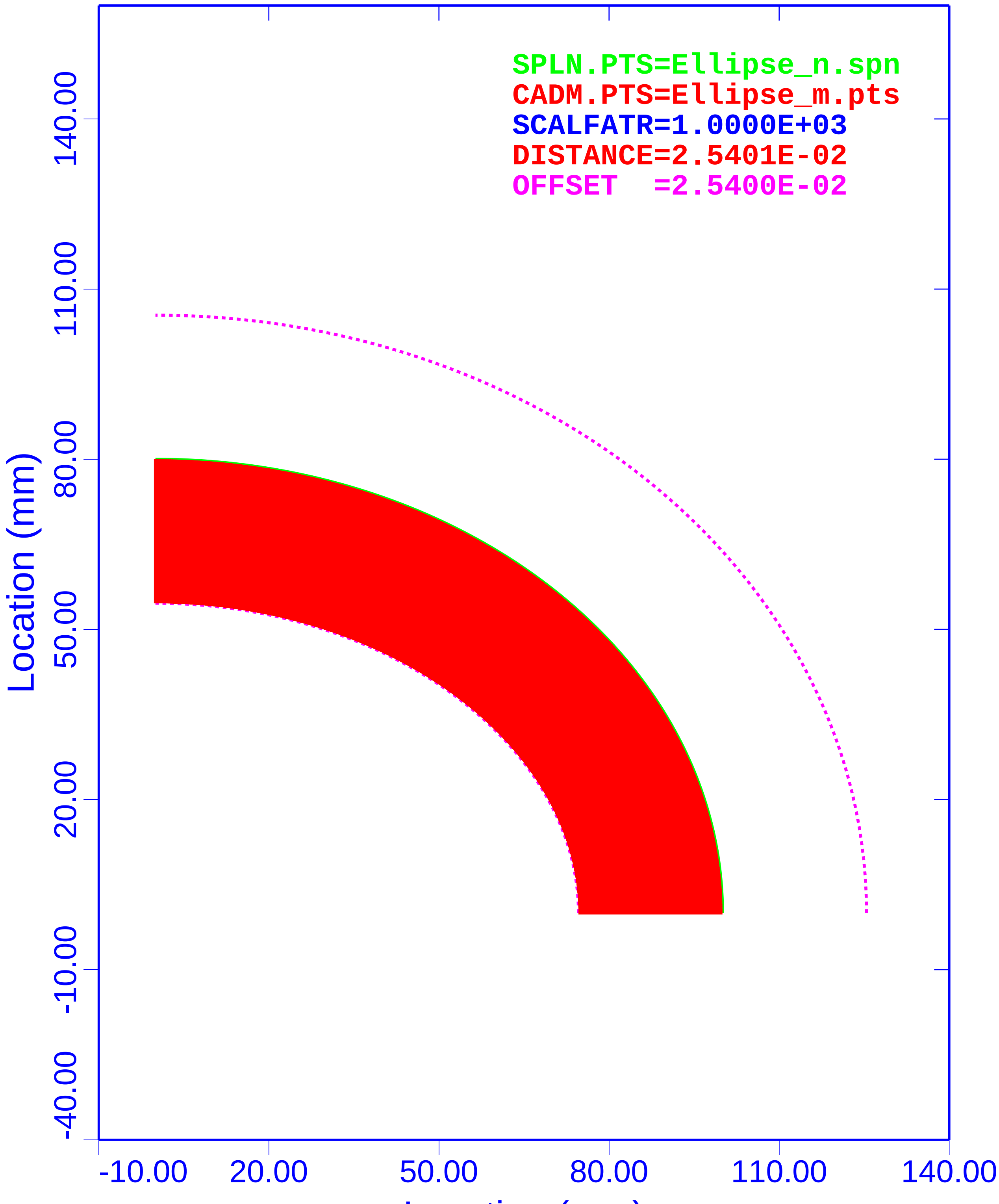

Figure 13. Analytical Ellipse: PCS, Deviations with –0.0254 mm Normal Offset

Maximum and Minimum Deviations

Table 4 is a summary of the maximum and minimum deviations for the analytical ellipse. Column 1 lists the associated figure that displays the results. The second column lists the type of evaluation spline. Columns 3 and 4 are the maximum and minimum deviations with the normal offset included. The fifth column gives the normal offset values. Columns 6 and 7 are the deviations with the offsets removed.

Table 4. Analytical Ellipse: Summary of Deviations

Figure	Spline Type	Maximum Deviation (mm)	Minimum Deviation (mm)	Normal Offset (mm)	Maximum Deviation with Offset Removed (mm)	Minimum Deviation with Offset Removed (mm)
Figure 8	WFS	+4.916562e–7	–3.445712e–6	+0.0000	+4.916562e–7	–3.445712e–6
Figure 9	PCS	+4.444657e–6	–8.693426e–7	+0.0000	+4.444657e–6	–8.693426e–7
Figure 10	WFS	+2.540087e–2	+2.539657e–2	+0.0254	+8.700000e–7	–3.430000e–6
Figure 11	PCS	+2.540481e–2	+2.539896e–2	+0.0254	+4.810000e–6	–1.040000e–6
Figure 12	WFS	–2.539895e–2	–2.540339e–2	–0.0254	+1.050000e–6	–3.390000e–6
Figure 13	PCS	–2.539508e–2	–2.540121e–2	–0.0254	+4.920000e–6	–1.210000e–6

A review of columns 6 and 7 of Table 4 shows that the absolute largest deviation is associated with the PCS and has a value of 4.920000e–6 mm (Figure 13). The ratio to the calculated deviation and the inspection uncertainty is 1.937e–3.

In column 6 of Table 4, Maximum Deviation with Offset Removed, the WFS deviations range from +8.7++e–7 mm to +4.9++e–6 mm. However, the PCS deviations are almost identical with values of +4.++e–6 mm. The WFS model yields slightly better results than the PCS model for maximum deviations.

In column 7 of Table 4, Minimum Deviation with Offset Removed, the WFS deviations are almost identical and have values of –3.3++e–6 mm. However, the PCS deviations range from –8.6++e–7 mm to –1.2++e–6 mm. The WFS model yields slightly poorer results than the PCS model for minimum deviations.

The results of the analyses shown in columns 6 and 7 of Table 4, Maximum Deviation with Offset Removed (mm) and Minimum Deviation with Offset Removed (mm), reveal that the WFS and PCS models do equally well representing the data.

The results shown in columns 6 and 7 of Table 4 indicate that a minimum accuracy of about five and one-half digits past the decimal point could be expected for an analytical ellipse. In general, the accuracy of these splines is the number of digits past the decimal point minus one-half digit.

These analyses show that both the WFS model and the PCS model produce results that meet all of DOE’s MBE requirements.

A review of the six figures listed in Table 4 also shows that the minimum-distance algorithm produces the proper sign on the deviations.

Analytical Parabola

The next six figures show the comparisons of WFS and PCS representations with the exact analytical parabola data generated by Mathcad. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the fabrication uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 1,000.

Deviation Plots

Figure 14 is a graph of the analytical parabola modeled with the WFS algorithm. The normal offset is 0.0000 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS model reproduces the exact results within 1.2481e–4 mm.

Figure 15 is a graph of the analytical parabola modeled with the PCS algorithm. The normal offset is 0.0000 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 3.7820e–4 mm.

Figure 16 is a graph of the analytical parabola modeled with the WFS algorithm. The normal offset is +0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.5421e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 17 is a graph of the analytical parabola modeled with the PCS algorithm. The normal offset is +0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 2.5778e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 18 is a graph of the analytical parabola modeled with the WFS algorithm. The normal offset is –0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the WFS algorithm reproduces the exact results within 2.5525e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

Figure 19 is a graph of the analytical parabola modeled with the PCS algorithm. The normal offset is –0.0254 mm. Mathcad was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their large magnitude, they show as a solid band on the plot. The analysis shows that the PCS algorithm reproduces the exact results within 2.5471e–2 mm. Notice that this value includes the normal offset of 2.54e–2 mm.

In all six of these figures, the inside normal offset curve has a small discontinuity at the pole. This condition is very pronounced in Figure 18 and Figure 19. The deviation curves overlap at the pole.

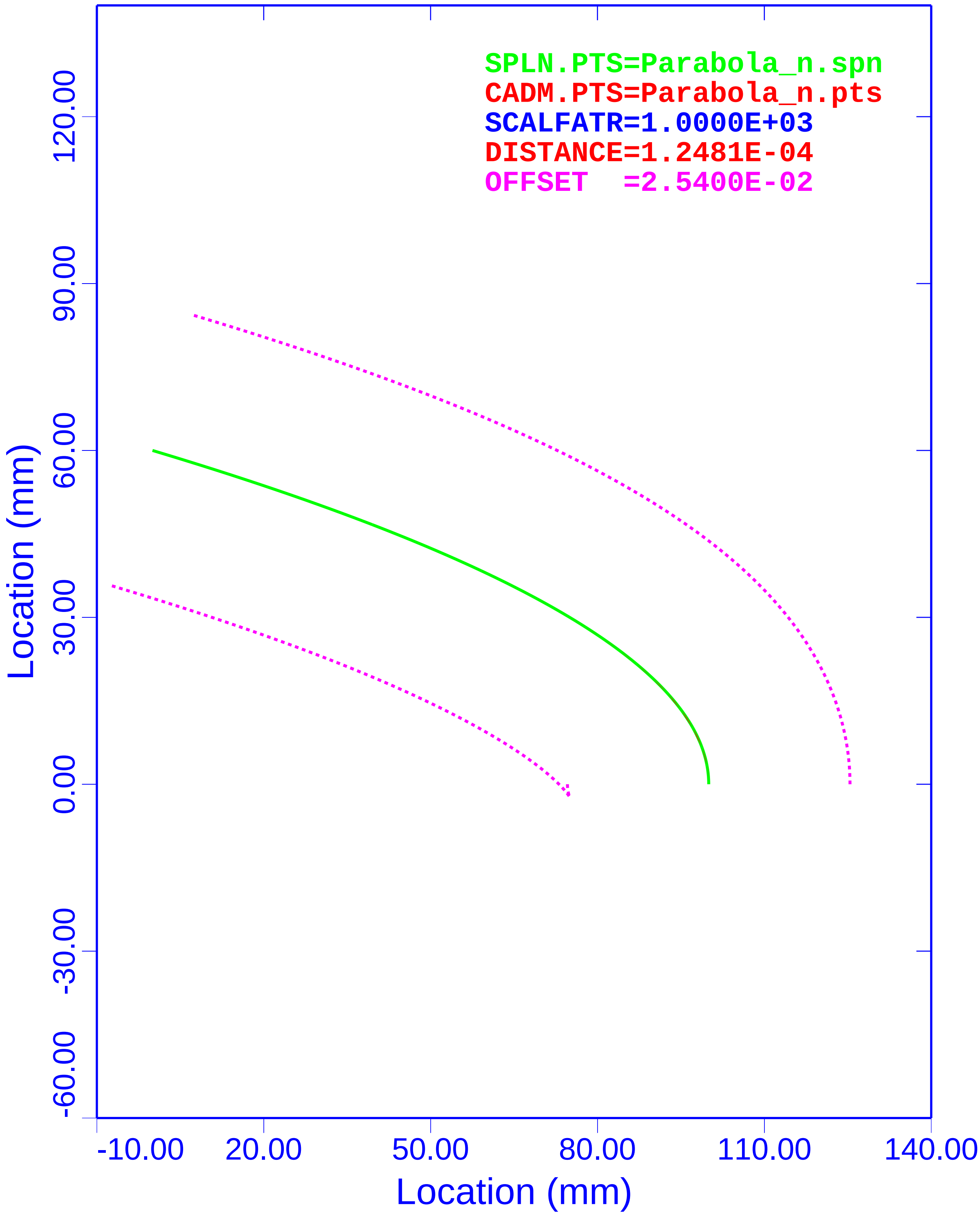

Figure 14. Analytical Parabola: WFS, Deviations with 0.0000 mm Normal Offset

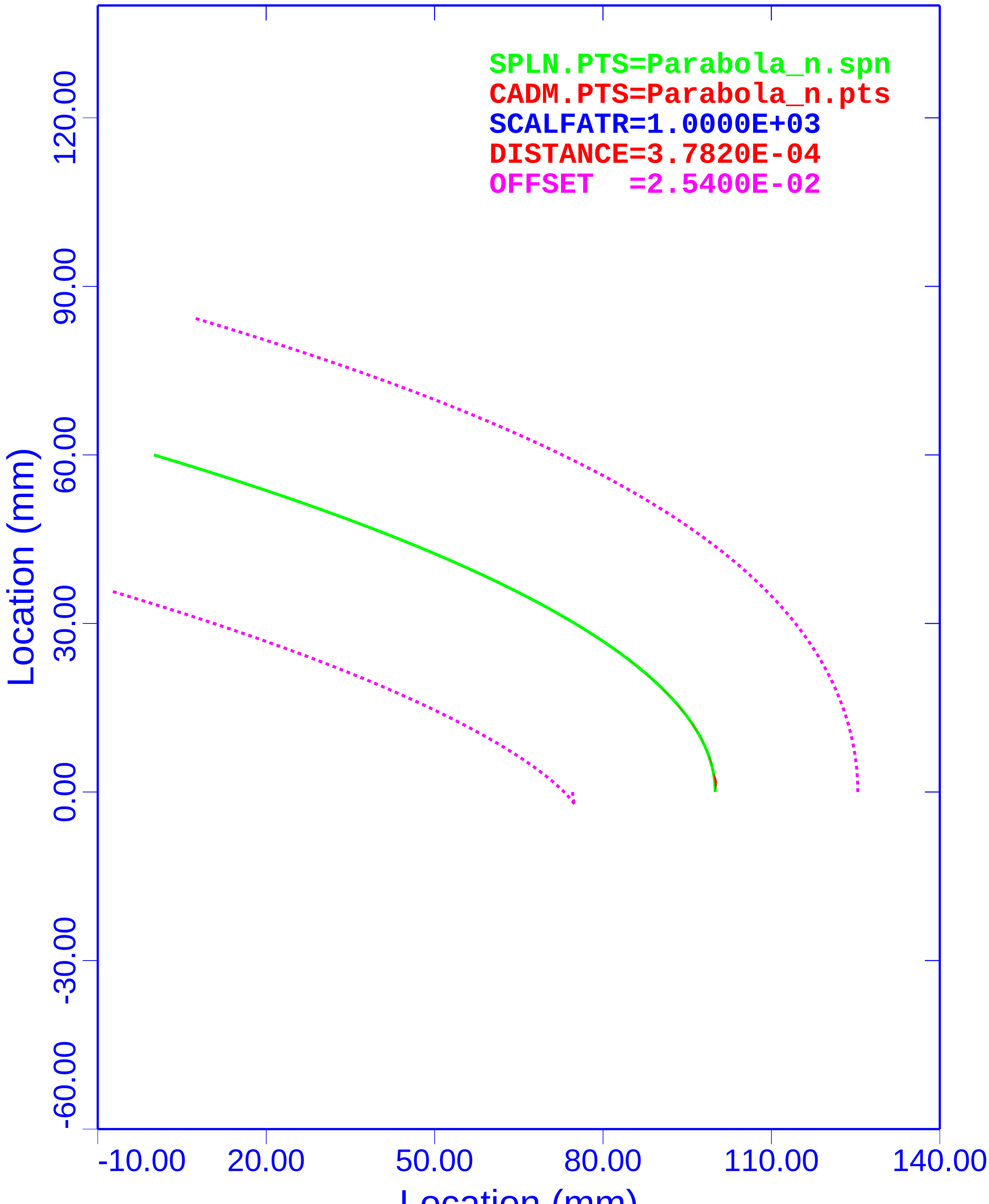

Figure 15. Analytical Parabola: PCS, Deviations with 0.0000 mm Normal Offset

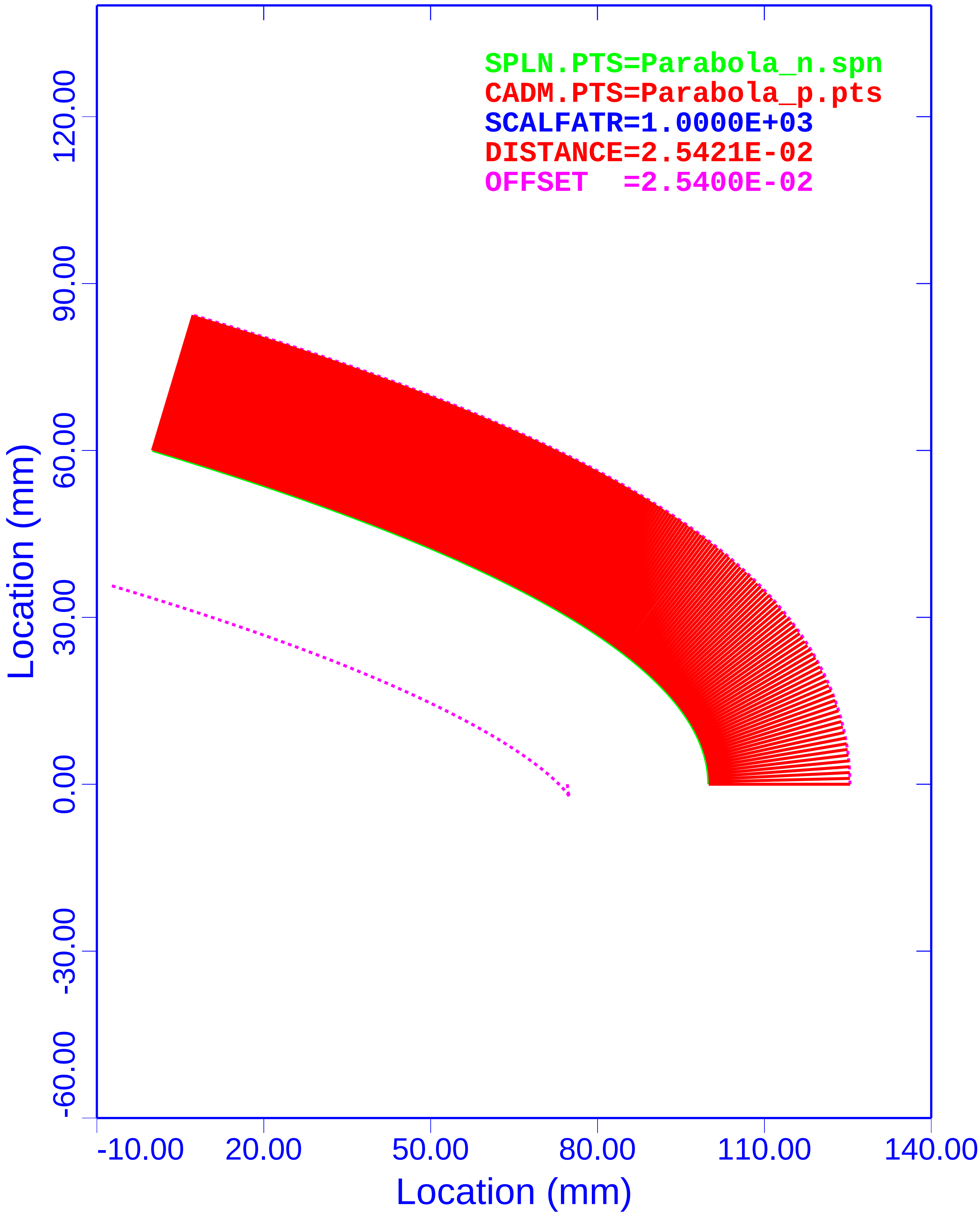

Figure 16. Analytical Parabola: WFS, Deviations with +0.0254 mm Normal Offset

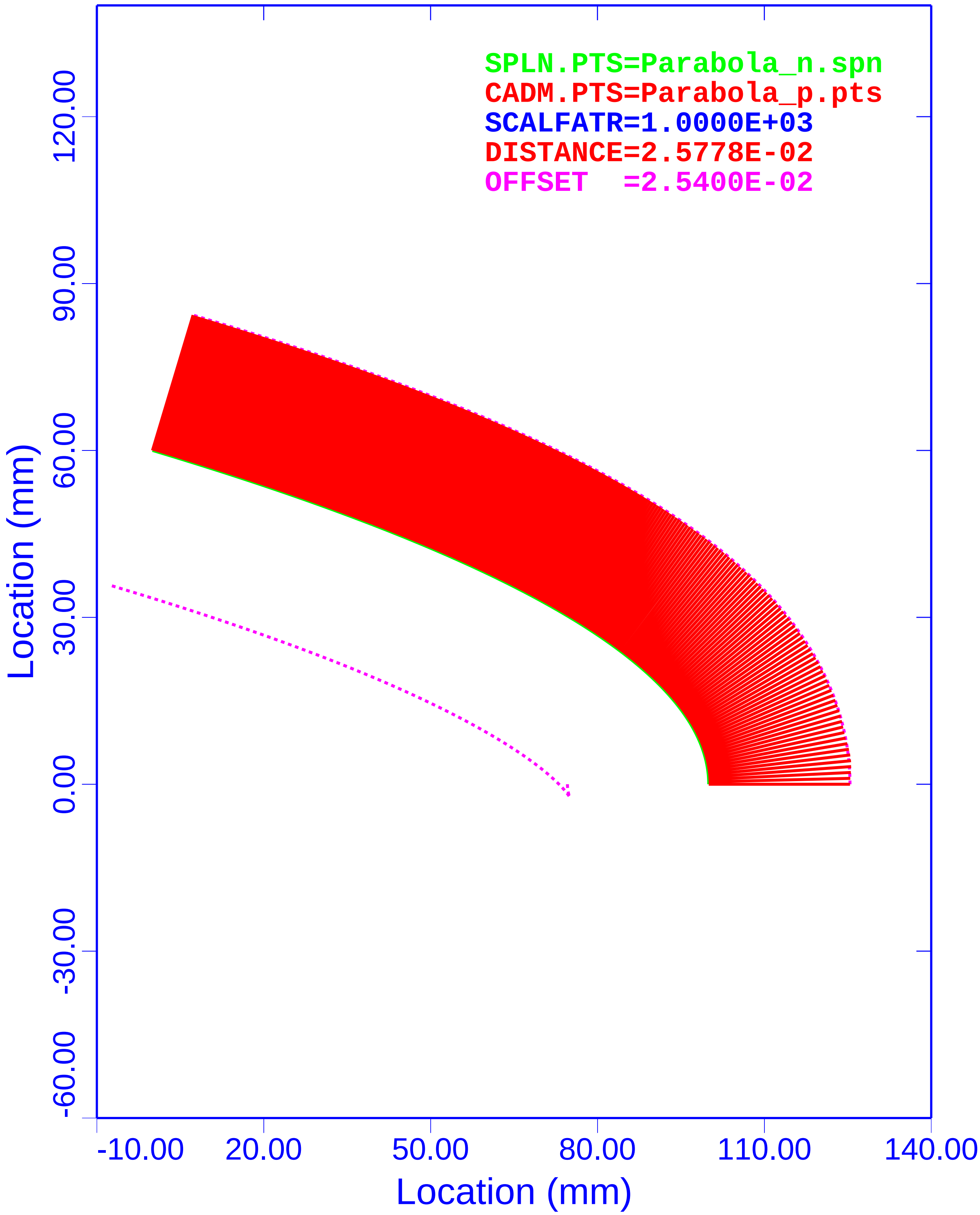

Figure 17. Analytical Parabola: PCS, Deviations with +0.0254 mm Normal Offset

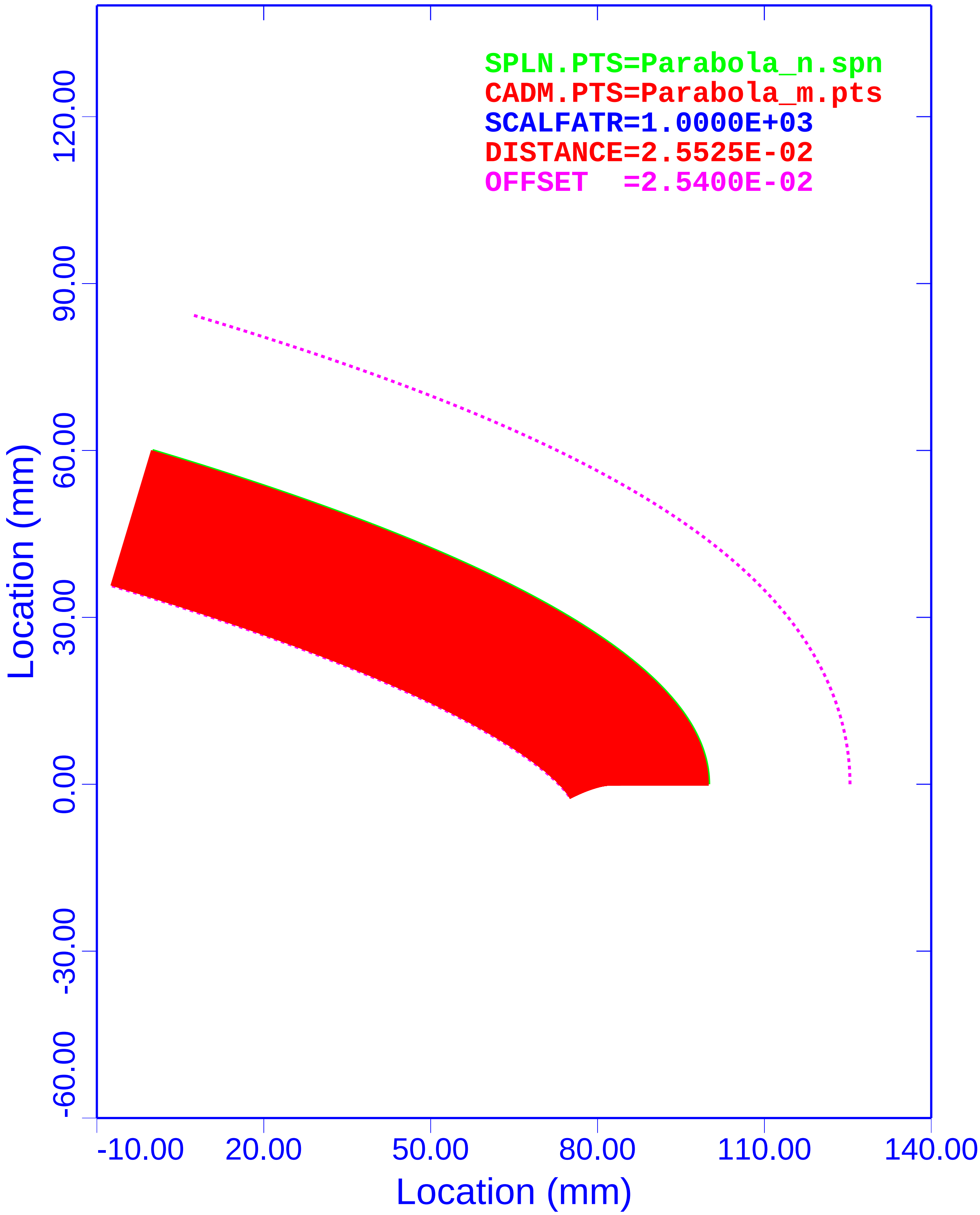

Figure 18. Analytical Parabola: WFS, Deviations with –0.0254 mm Normal Offset

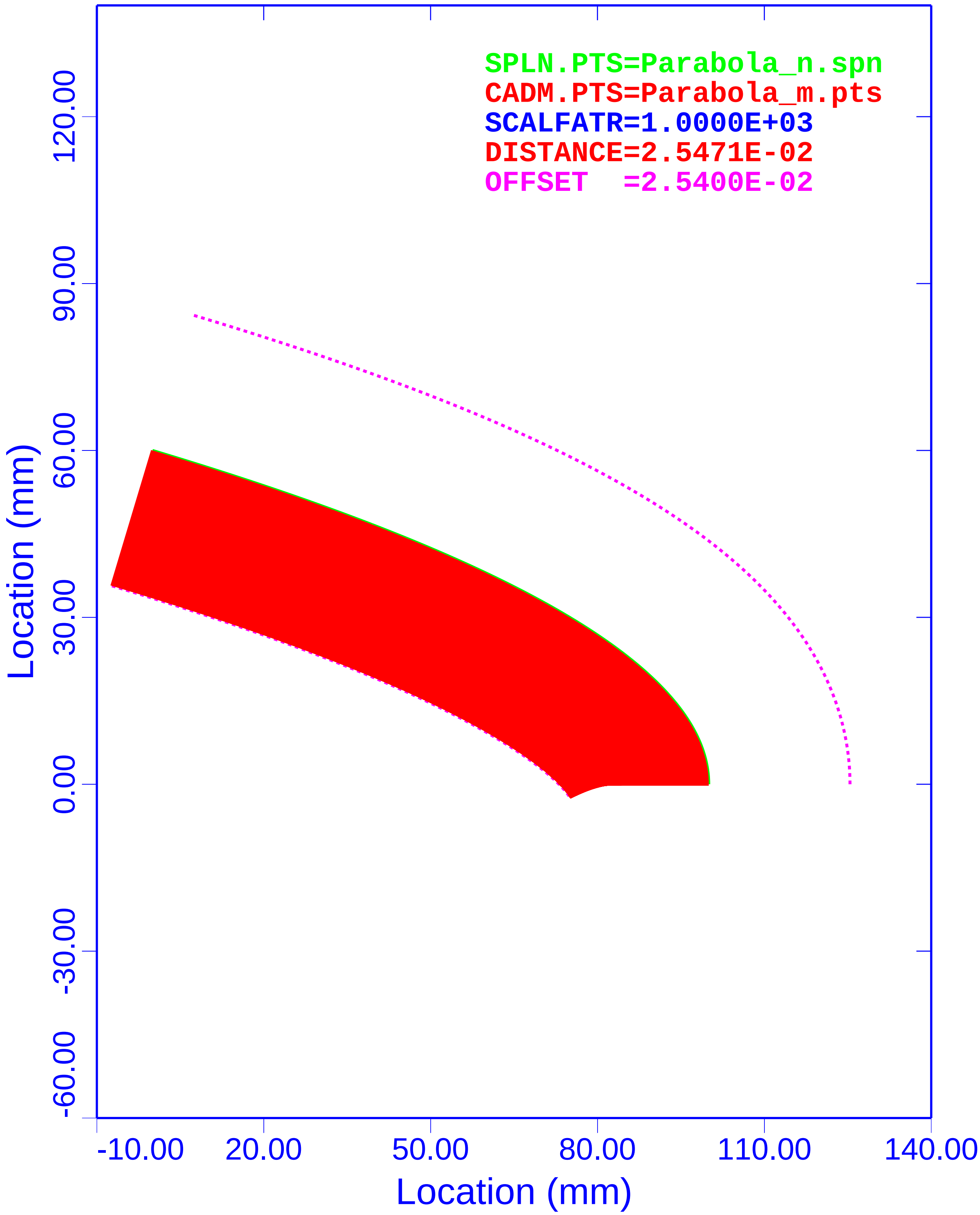

Figure 19. Analytical Parabola: PCS, Deviations with –0.0254 mm Normal Offset

Maximum and Minimum Deviations

Table 5 is a summary of the maximum and minimum deviations for the analytical parabola. Column 1 lists the associated figure that displays the results. The second column lists the type of evaluation spline. Columns 3 and 4 are the maximum and minimum deviations with the normal offset included. The fifth column gives the normal offset values. Columns 6 and 7 are the deviations with the normal offsets removed.

Table 5. Analytical Parabola: Summary of Deviations

Figure	Spline Type	Maximum Deviation (mm)	Minimum Deviation (mm)	Normal Offset (mm)	Maximum Deviation with Offset Removed (mm)	Minimum Deviation with Offset Removed (mm)
Figure 14	WFS	+2.035613e–5	–1.248098e–4	+0.0000	+2.035613e–5	–1.248098e–4
Figure 15	PCS	+3.782016e–4	–7.172041e–5	+0.0000	+3.782016e–4	–7.172041e–5
Figure 16	WFS	+2.542097e–2	+2.527537e–2	+0.0254	+2.097000e–5	–1.246300e–4
Figure 17	PCS	+2.577754e–2	+2.532846e–2	+0.0254	+3.775400e–4	–7.154000e–5
Figure 18	WFS	–2.537913e–2	–2.552455e–2	–0.0254	+2.087000e–5	–1.245500e–4
Figure 19	PCS	–2.502223e–2	–2.547146e–2	–0.0254	+3.777700e–4	–7.146000e–5

The absolute minimum and maximum deviations listed in Table 5 are 1.248098e–4 mm (Figure 14) and 3.782016e–4 mm (Figure 15), respectively. The ratios of the minimum and maximum deviations to the inspection uncertainty are 0.0491 and 0.1489, respectively.

In column 6 of Table 5, Maximum Deviation with Offset Removed, the WFS deviations are almost identical and have values of +2.0++e–5 mm. Similarly, the PCS deviations are almost identical with values of +3.7++e–4 mm. The WFS model yields slightly better results than the PCS model for maximum deviations.

In column 7 of Table 5, Minimum Deviation with Offset Removed, the WFS deviations are almost identical and have values of –1.2++e–4 mm. Similarly, the PCS deviations are almost identical with values of –7.1++e–5 mm. The WFS model yields slightly poorer results than the PCS model for minimum deviations.

The results shown in columns 6 and 7 of Table 5 indicate that a minimum accuracy of about three and three-quarters digits past the decimal point could be expected for an analytical parabola. In general, the accuracy of these splines is the number of digits past the decimal point minus two and one-quarter digits.

These analyses show that both the WFS and PCS produce results that meet all of DOE’s MBE requirements.

A review of the six figures in Table 5 shows that the minimum-distance algorithm produces the proper sign on the deviations.

Conclusions

Of the 18 numerical analyses presented above, the largest ratio of calculated deviation to inspection uncertainty is 0.1489. This ratio is associated with the parabola and is located at a point of high curvature. In addition, the signs of the deviations were shown to be correct. The conclusion from this study is that both the WFS and PCS algorithms can be used to model legacy data and to design new models.

Deviations of Nonanalytical Shapes

We had to know the accuracy of the minimum-distance software algorithms to evaluate the data generated by the different CAD systems (PTC/Pro/E, Computervision/CADDS, and CDC/ICEM DDN). The accuracy was established by the work shown in the section of this report entitled "Accuracy Study of Minimum-Distance Algorithms." Because the mathematics actually used in the commercial CAD systems is not always known, such systems must be evaluated from the "outside."

Each CAD system has its own method of representing the spline data points. Table 6 lists the type of two-dimensional, curve-fitting methods available in each CAD system, along with the end-angle options.

Table 6. CAD System Spline Types and End-Angle Options

CAD System	Spline Type	Default End-Angle Option	Specified End-Angle Option	Other End-Angle Option
Pro/E	PCS	Natural	Yes	–
CADDS	WFS	Circular	Yes	–
ICEM	WFS	Circular	Yes	Parabolic

Goals

The goals of this study are to establish and to compare the accuracy of these three CAD software systems.

Nonanalytical Shapes: Ellipse, Lampshade, and Weird Shape

Three nonanalytical shapes were utilized to determine how well the three CAD systems could model various shapes. These shapes were an ellipse, a lampshade, and a weird shape. The data for these analyses were obtained from Atomic Weapons Establishment (AWE) Hunting-Brae.[9]

The procedure used to evaluate the accuracy of the CAD systems was as follows:

- Generate the mathematical representations of the spline points for both WFS and PCS
- Generate a data set of evaluation points for each shape from each CAD system
- Calculate the minimum distances of the evaluation data from the mathematical representation results of both the WFS and PCS models
- Summarize the results
- Compare the results

Spline Data

Table 7 is a summary of the parameters used to characterize the three nonanalytical shapes. This table contains the number of points used to define the shape, the data-point spacing ranges, the number of digits past the decimal point, and the beginning and ending angles. These data are used exactly as received (Ref. 9).

Notice in Table 7 that the spacing of the spline data points of the lampshade and weird shape vary from less than one degree to more than six degrees. Also, the number of points used to define these data is about half the customary number.

Table 7. Nonanalytical Spline Parameters

Non-analytical Splines	File Names	Number of Points	Spacing of Points (Degrees)	Digits Past Decimal Point	Beginning End Angle (Degrees)	Ending End Angle (Degrees)
Ellipse	c01762.spn	46	2.0	6	90.0000	180.0000
Lampshade	c01763.spn	28	0.92 – 6.94	6	99.2767	171.5234
Weird Shape	c01764.spn	24	0.78 – 5.04	6	174.6875	177.5217

Appendix E—Nonanalytical Spline-Point Data has a listing of the spline definitions. All data were rounded to six decimal points.

Figure 20 shows these three spline curves. The solid red curve is the nonanalytical ellipse. The lampshade is shown as the small-dash black curve. The large-dash green curve is the weird shape.

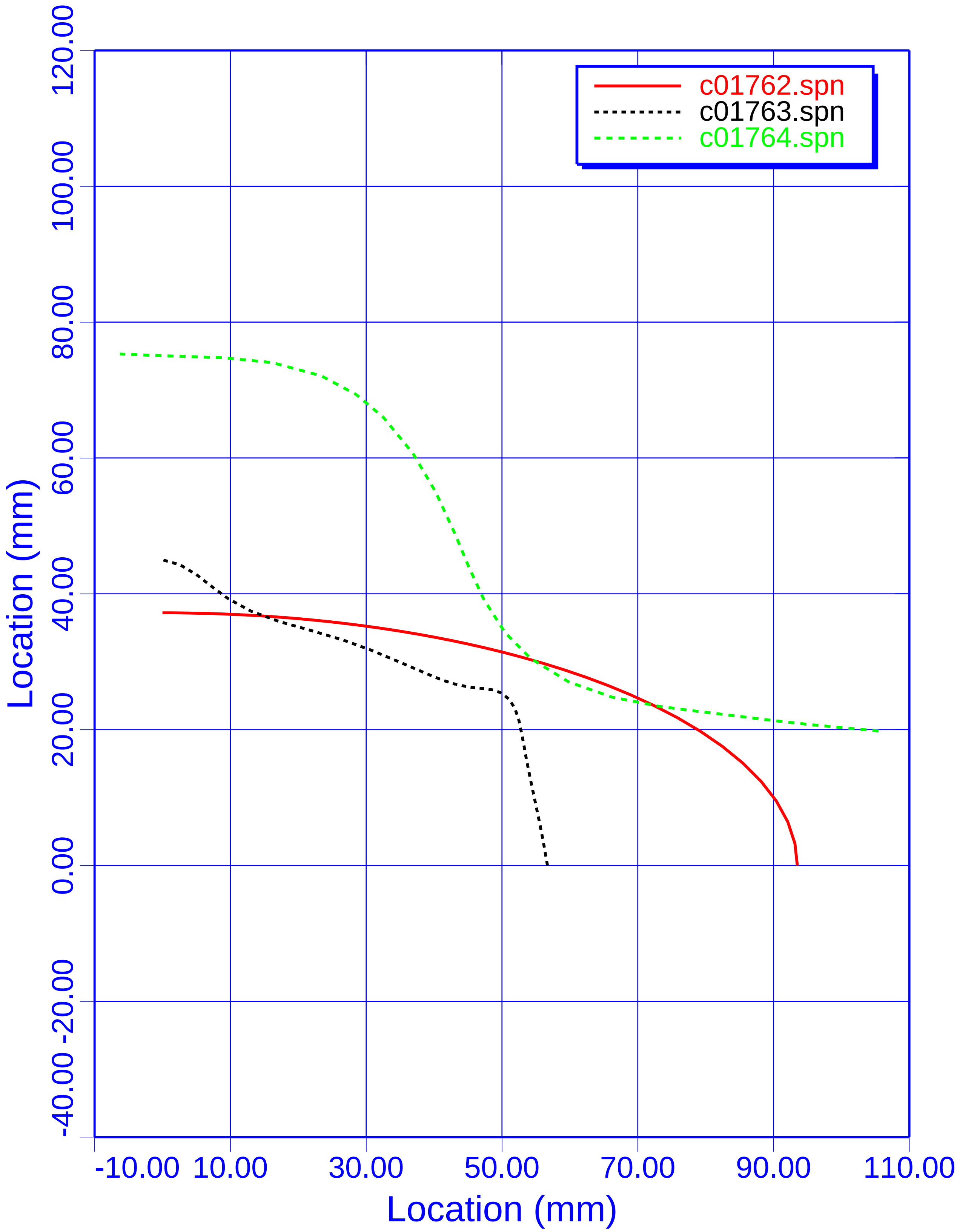

Figure 20. Nonanalytical Shapes: Ellipse, Lampshade, and Weird Shape

Evaluation Data

Three data sets of evaluation points were generated for each of the nonanalytical curves. Table 8 lists the file names, the number of points used in the evaluation, the approximate point spacing, and the accuracy of the data. The number was not identical in each of the evaluation data sets. All the data points were rounded to six digits past the decimal point.

Table 8. Nonanalytical Shapes: Evaluation Data Parameters

Nonanalytical Evaluation Data	File Name	Number of Points	Spacing of Points (Approximate)	Digits Past Decimal Point
Ellipse	c01762_CAD.pts	361	0.25	6
Ellipse	c01762_PRO.pts	351	0.25	6
Ellipse	c01762_ICM.pts	350	0.25	6
Lampshade	c01763_CAD.pts	361	0.25	6
Lampshade	c01763_PRO.pts	351	0.25	6
Lampshade	c01763_ICM.pts	350	0.25	6
Weird Shape	c01764_CAD.pts	339	0.25	6
Weird Shape	c01764_PRO.pts	351	0.25	6
Weird Shape	c01764_ICM.pts	350	0.25	6

Keyword Graphics Builder Program Command File

The command file used to perform the following calculations is listed in Appendix F—Keyword Graphics Builder Program— Command Files—Nonanalytical Shapes—Deviation Study.

Nonanalytical Ellipse

The next six figures show the comparisons of WFS and PCS representations of the nonanalytical ellipse curve with the evaluation data generated by the three different CAD systems. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the inspection uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 5,000.

Deviation Plots

Figure 21 is a graph of the nonanalytical ellipse modeled with the WFS algorithm. CADDS was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS model yields a maximum deviation of 1.1841e–5 mm.

Figure 22 is a graph of the nonanalytical ellipse modeled with the PCS algorithm. CADDS was used to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur near the pole where the curvatures of the data are the largest. Note that the deviations damp out in about four segments of the spline data. The analysis shows that the PCS model yields a maximum deviation of 5.0234e–4 mm.

Figure 23 is a graph of the nonanalytical ellipse modeled with the WFS algorithm. Pro/E was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur near the pole where the curvatures of the data are the largest. Note that the deviations damp out in about four segments of the spline data. The analysis shows that the WFS model yields a maximum deviation of 4.9707e–4 mm.

Figure 24 is a graph of the nonanalytical ellipse modeled with the PCS algorithm. Pro/E was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the PCS model yields a maximum deviation of 6.3625e–8 mm.

Figure 25 is a graph of the nonanalytical ellipse modeled with the WFS algorithm. ICEM DDN was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS model yields a maximum deviation of 2.4593e–5 mm.

Figure 26 is a graph of the nonanalytical ellipse modeled with the PCS algorithm. ICEM DDN was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur near the pole where the curvatures of the data are the largest. Note that the deviations damp out in about four segments of the spline data. The analysis shows that the PCS model yields a maximum deviation of 5.1432e–4 mm.

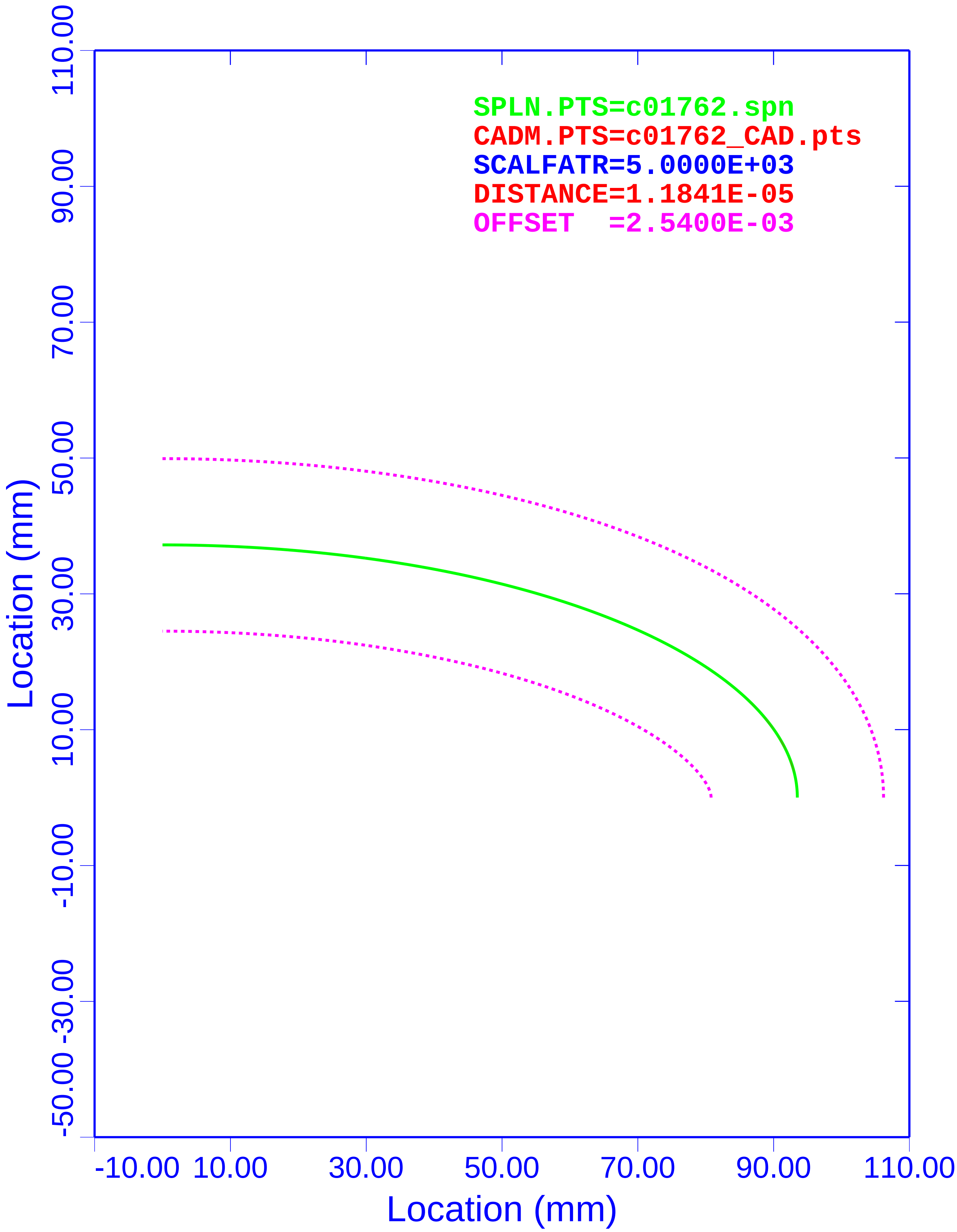

Figure 21. Nonanalytical Ellipse: WFS, CADDS Deviations

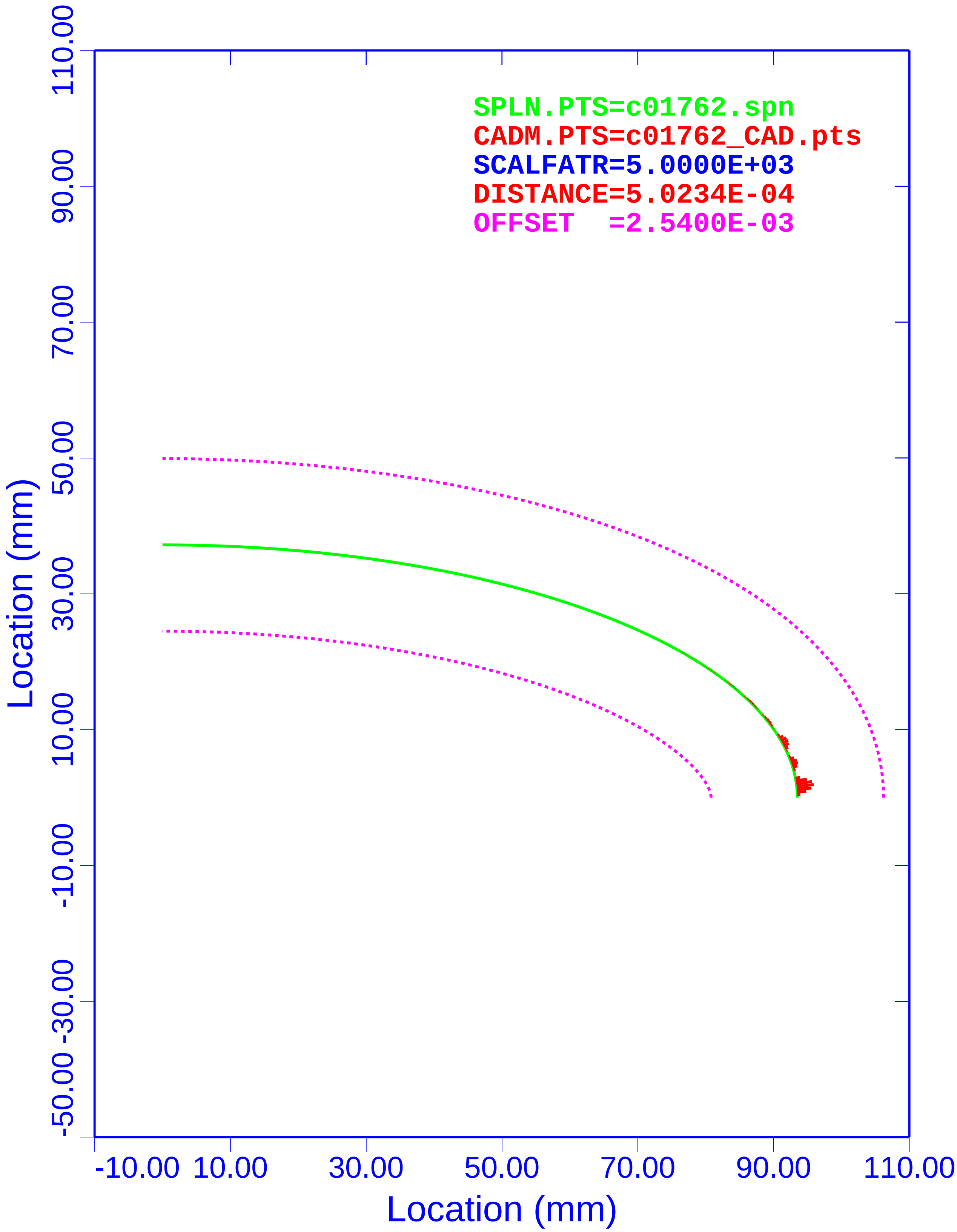

Figure 22. Nonanalytical Ellipse: PCS, CADDS Deviations

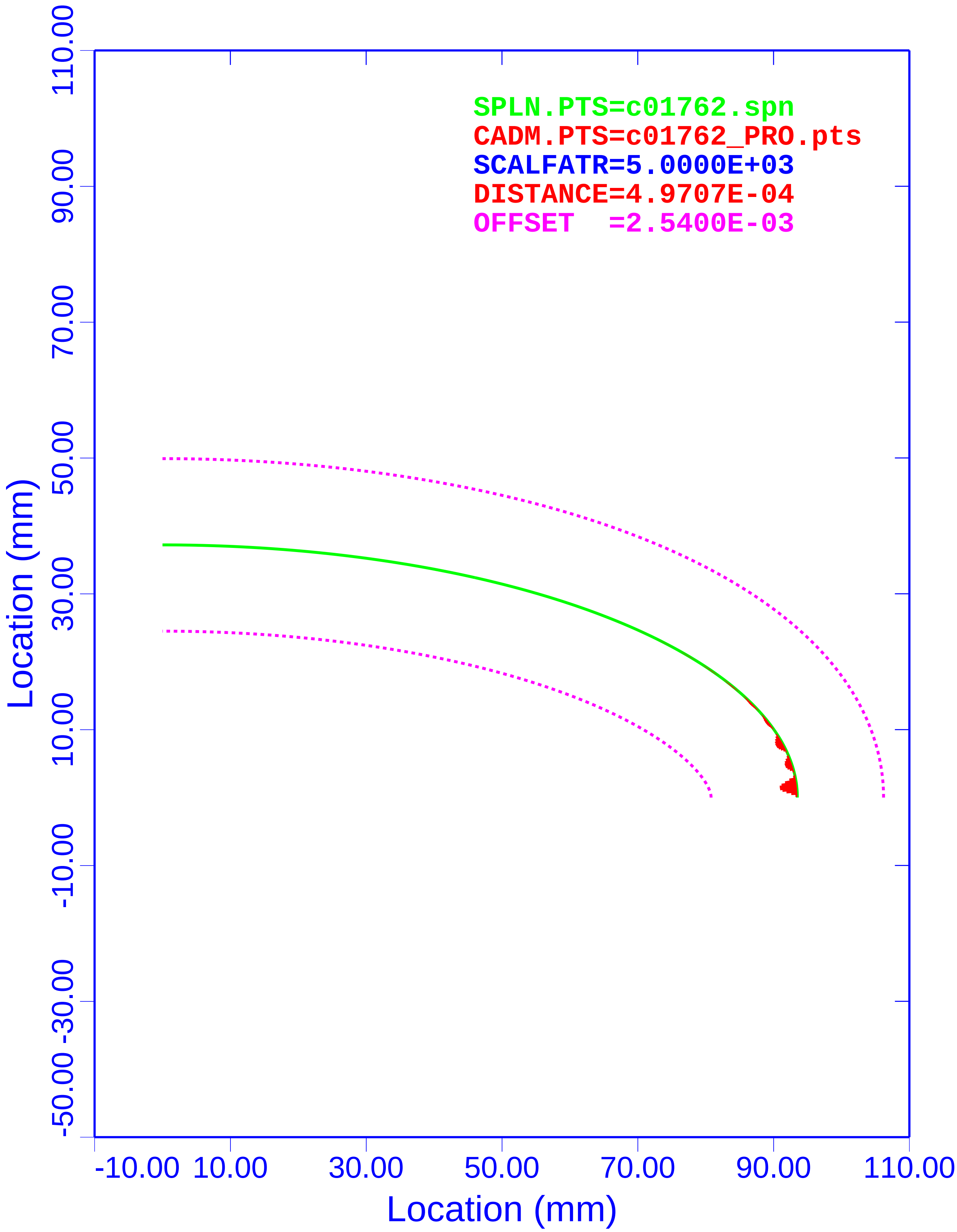

Figure 23. Nonanalytical Ellipse: WFS, Pro/E Deviations

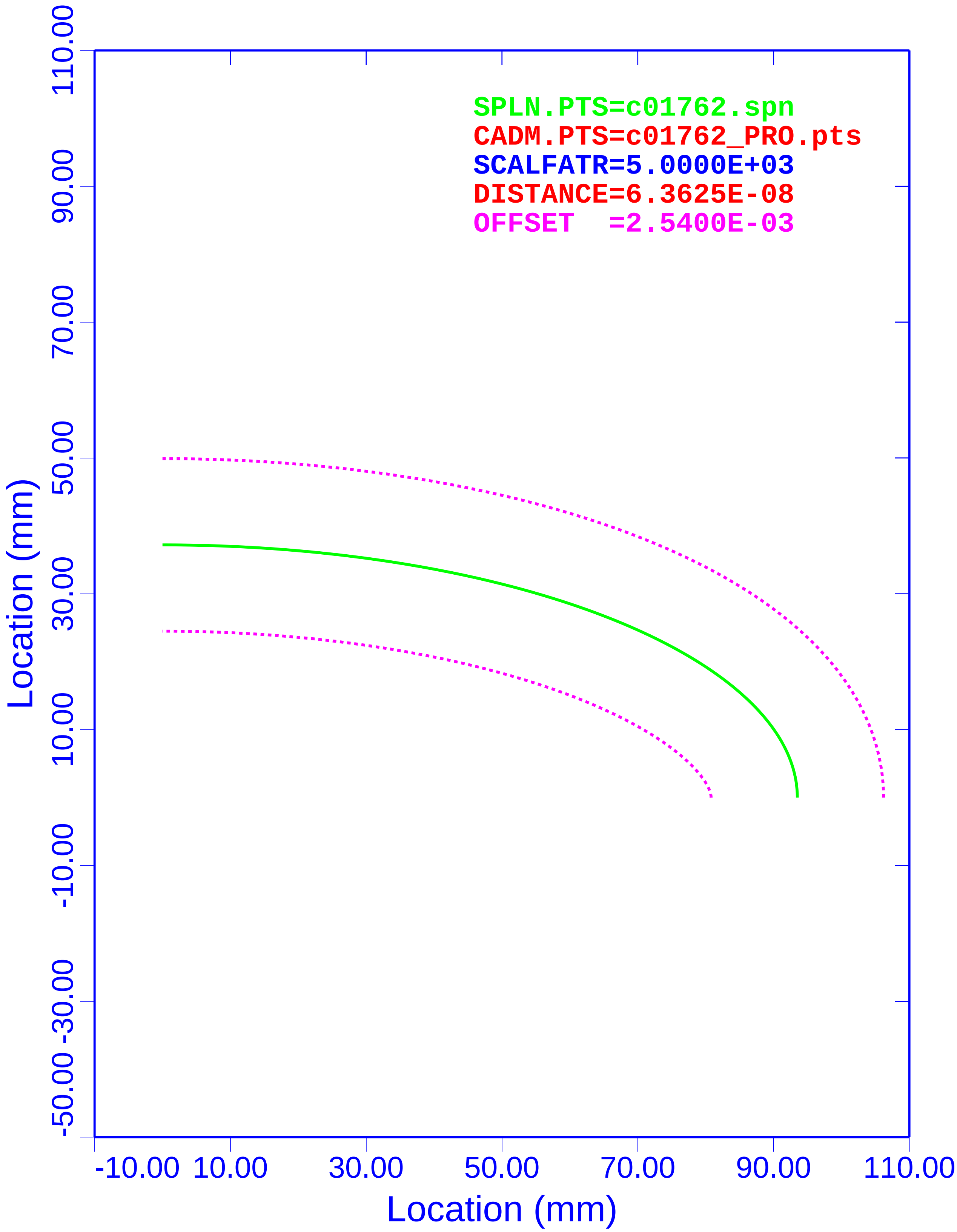

Figure 24. Nonanalytical Ellipse: PCS, Pro/E Deviations

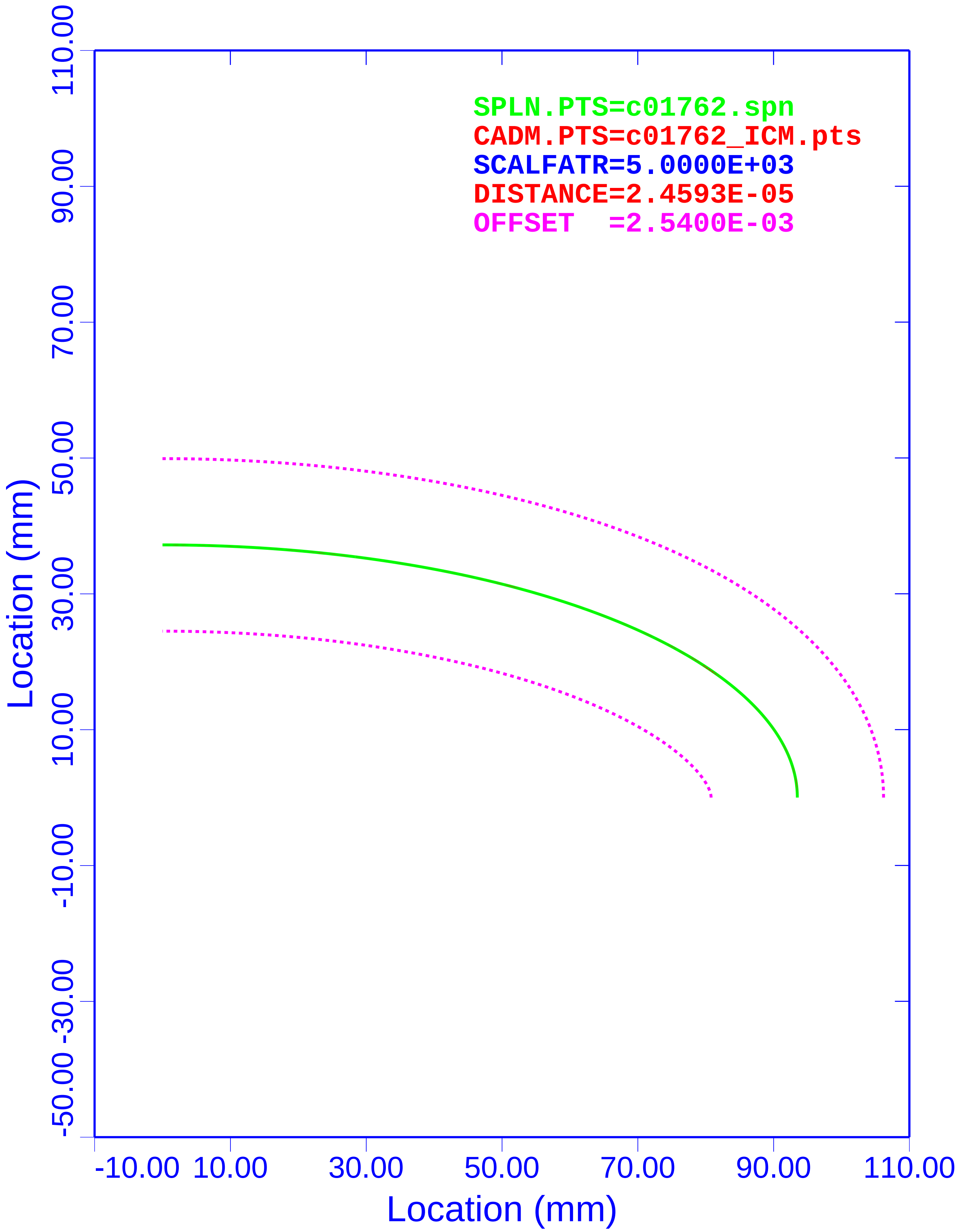

Figure 25. Nonanalytical Ellipse: WFS, ICEM DDN Deviations

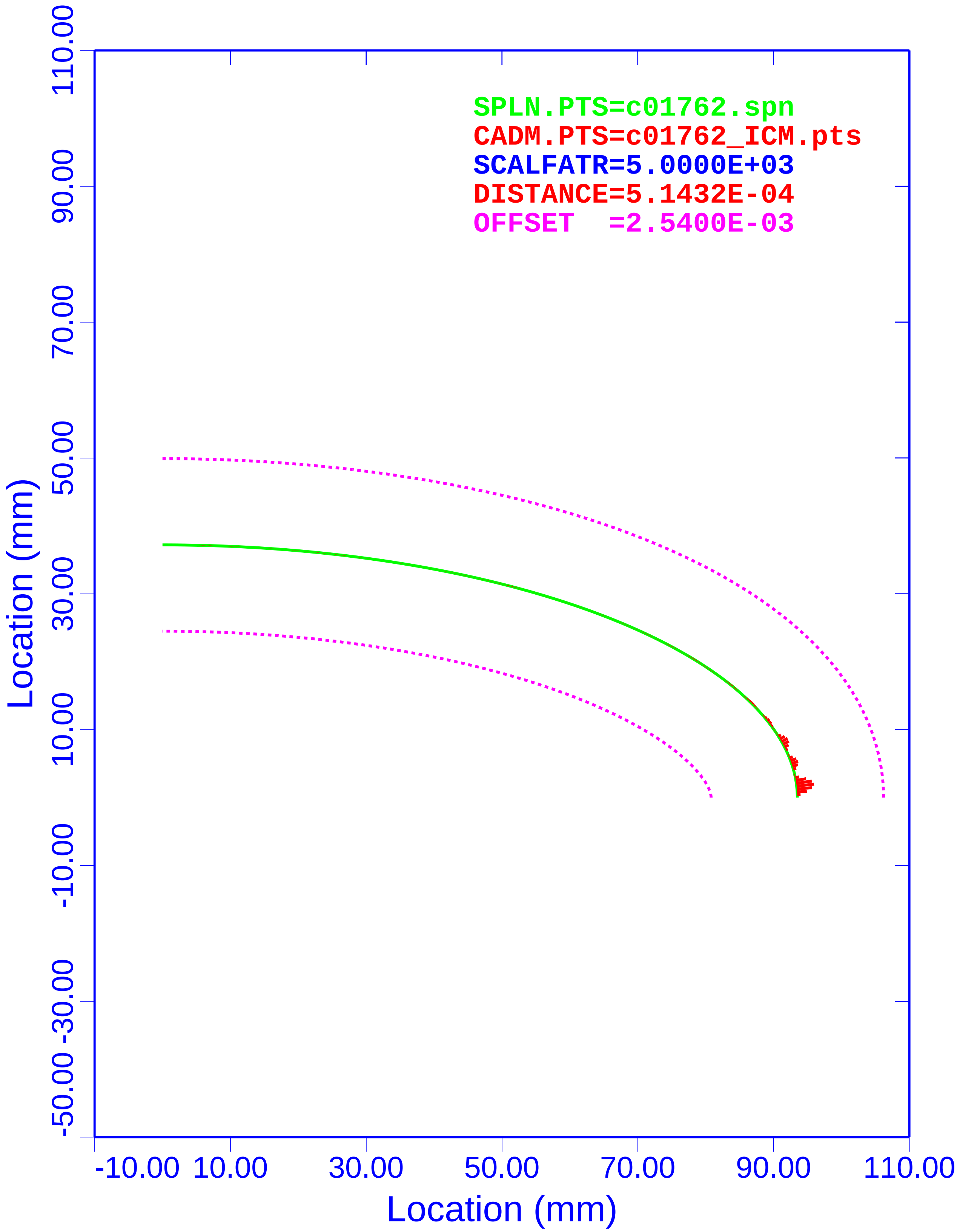

Figure 26. Nonanalytical Ellipse: PCS, ICEM DDN Deviations

Maximum and Minimum Deviations

Table 9 is a summary of the maximum and minimum deviations for the nonanalytical ellipse. Column 1 lists the associated figure that displays the results. Column 2 is the CAD system used to generate the evaluation points. The third column gives the type of evaluation spline. Columns 4 and 5 are the maximum and minimum deviations.

Table 9. Nonanalytical Ellipse: Summary of Deviations

Figure	CAD System	Spline Type	Maximum Deviation (mm)	Minimum Deviation (mm)
Figure 21	CADDS	WFS	+9.934505e–7	–1.184080e–5
Figure 22	CADDS	PCS	+5.023381e–4	–9.812279e-6
Figure 23	Pro/E	WFS	+1.376314e–5	–4.970730e–4
Figure 24	Pro/E	PCS	+6.362479e–8	–5.822224e–8
Figure 25	ICEM	WFS	+1.620899e–5	–2.459316e–5
Figure 26	ICEM	PCS	+5.143201e–4	–1.657802e–5

The absolute minimum and maximum deviations listed in Table 9 are 4.970730e–4 (Figure 23) and 5.143201e–4 (Figure 26), respectively. The ratios of deviations to the inspection uncertainty are 0.1957 and 0.2025, respectively.

As was stated in the section of this document entitled "Computer-Aided Design Systems," both CADDS and ICEM DDN have WFS modules, and Pro/E uses a PCS in its sketcher option. Figure 21 and Figure 25 reflect the existence of the WFS modules. Figure 24 verifies the use of a PCS in Pro/E. The other three figures show opposite results. In essence, like modules yield better results, and unlike modules yield poorer results. The deviations for Figure 22 and Figure 26 at their poles are opposite in direction and almost equal in magnitude to those shown in Figure 23.

A review of Table 9 reveals that the best combination of CAD system and spline model is Pro/E compared to the PCS model. The least desirable representations are Pro/E compared to the WFS model and ICEM DDN with the PCS model.

The results of the these analyses show that the both the WFS and PCS produce results that meet all of DOE's MBE requirements.

Nonanalytical Lampshade

The next six figures show the comparisons of WFS and PCS representations of the nonanalytical lampshade curve with the evaluation data generated by the three different CAD systems. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the inspection uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 5,000.

Deviation Plots

Figure 27 is a graph of the nonanalytical lampshade modeled with the WFS algorithm. CADDS was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS model yields a maximum deviation of 7.3857e–5 mm.

Figure 28 is a graph of the nonanalytical lampshade modeled with the PCS algorithm. CADDS was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur where the curvatures of the data are the largest. Note, in the regions of high curvatures, the deviations damp out in about four to five segments of the spline data. The analysis shows that the PCS model yields a maximum deviation of 1.2690e–3 mm.

Figure 29 is a graph of the nonanalytical lampshade modeled with the WFS algorithm. Pro/E was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur where the curvatures of the data are the largest. Note, in the regions of high curvature, the deviations damp out in about four to five segments of the spline data. The analysis shows that the WFS model yields a maximum deviation of 1.2734e–3 mm.

Figure 30 is a graph of the nonanalytical lampshade modeled with the PCS algorithm. Pro/E was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the PCS model yields a maximum deviation of 6.5001e–8 mm.

Figure 31 is a graph of the nonanalytical lampshade modeled with the WFS algorithm. ICEM DDN was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS model yields a maximum deviation of 5.9263e–5 mm.

Figure 32 is a graph of the nonanalytical lampshade modeled with the PCS algorithm. ICEM DDN was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur where the curvatures of the data are the largest. Note, in the regions of high curvature, the deviations damp out in about four to five segments of the spline data. The analysis shows that the PCS model yields a maximum deviation of 1.2687e–3 mm.

Notice that in all six of these figures, the inside normal offset curve has a large discontinuity in the region of high curvature.

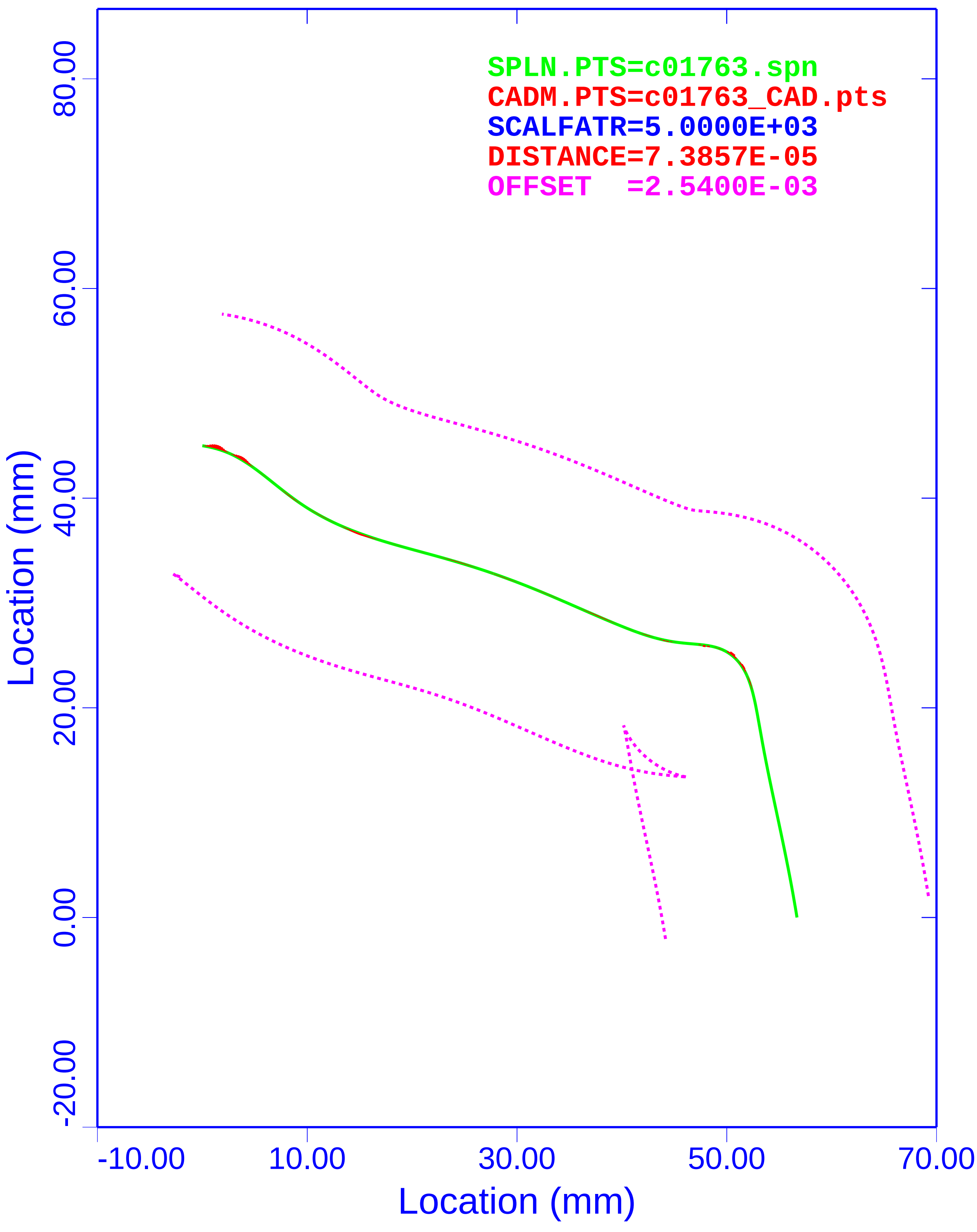

Figure 27. Nonanalytical Lampshade: WFS, CADDS Deviations

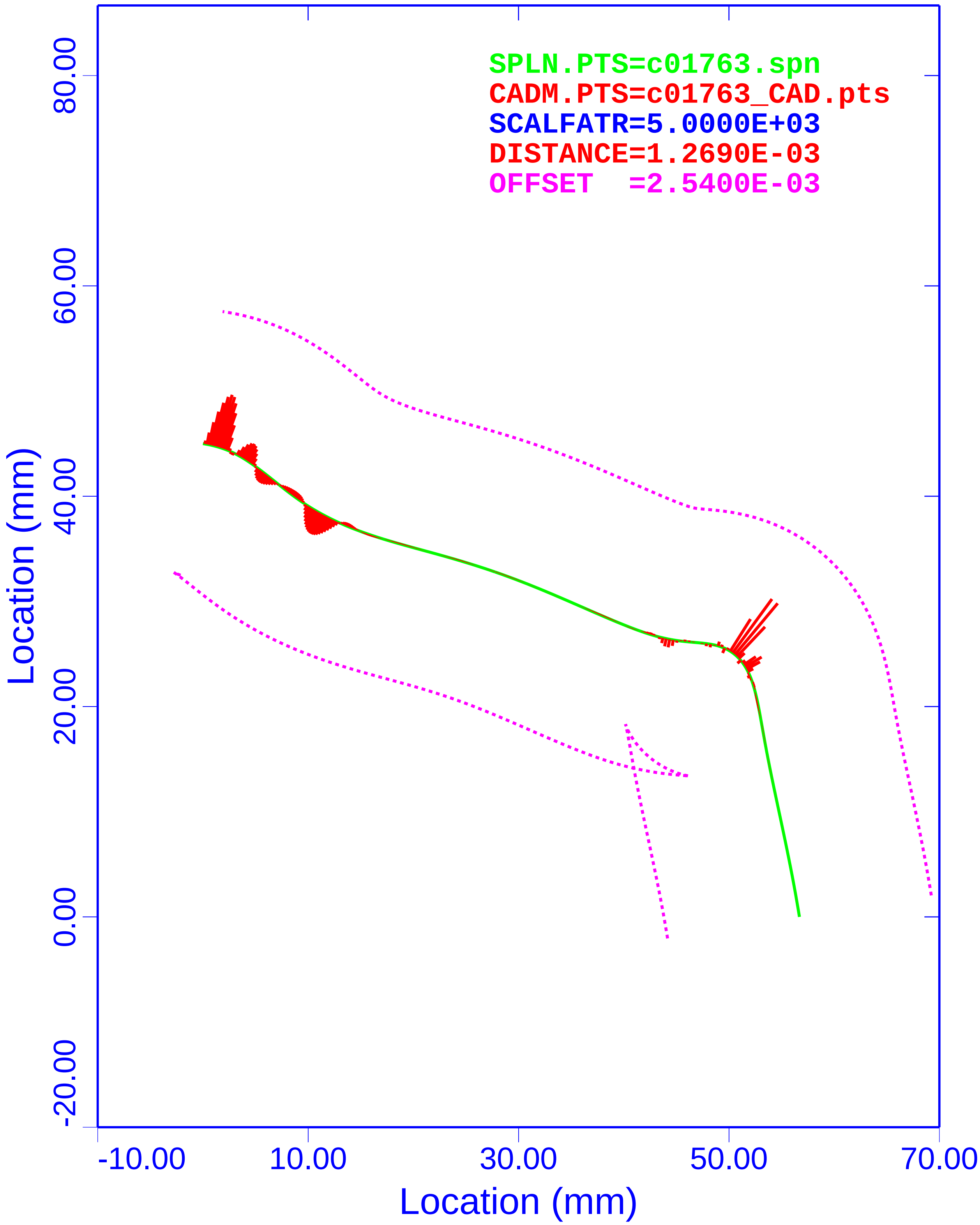

Figure 28. Nonanalytical Lampshade, PCS, CADDS Deviations

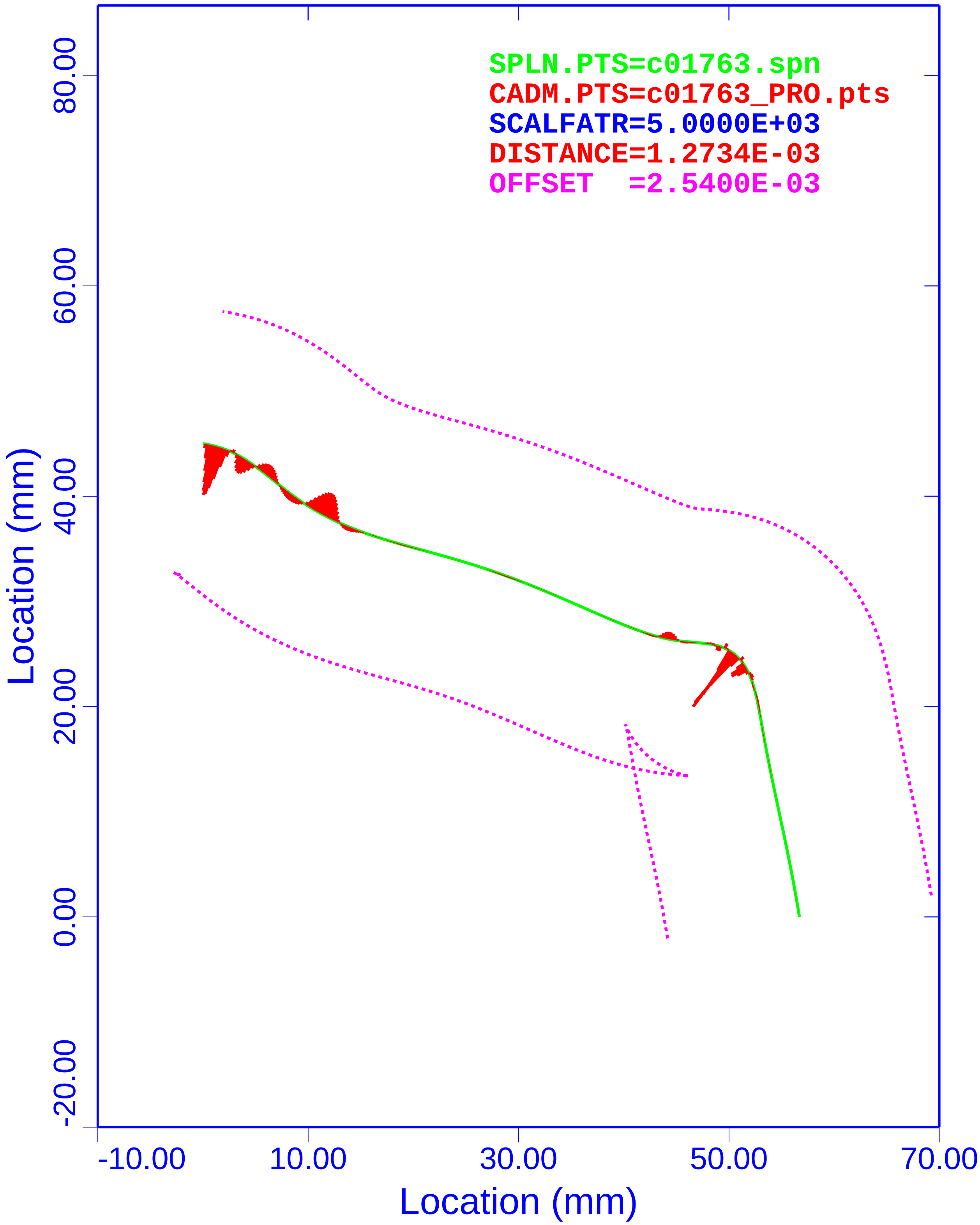

Figure 29. Nonanalytical Lampshade: WFS, Pro/E Deviations

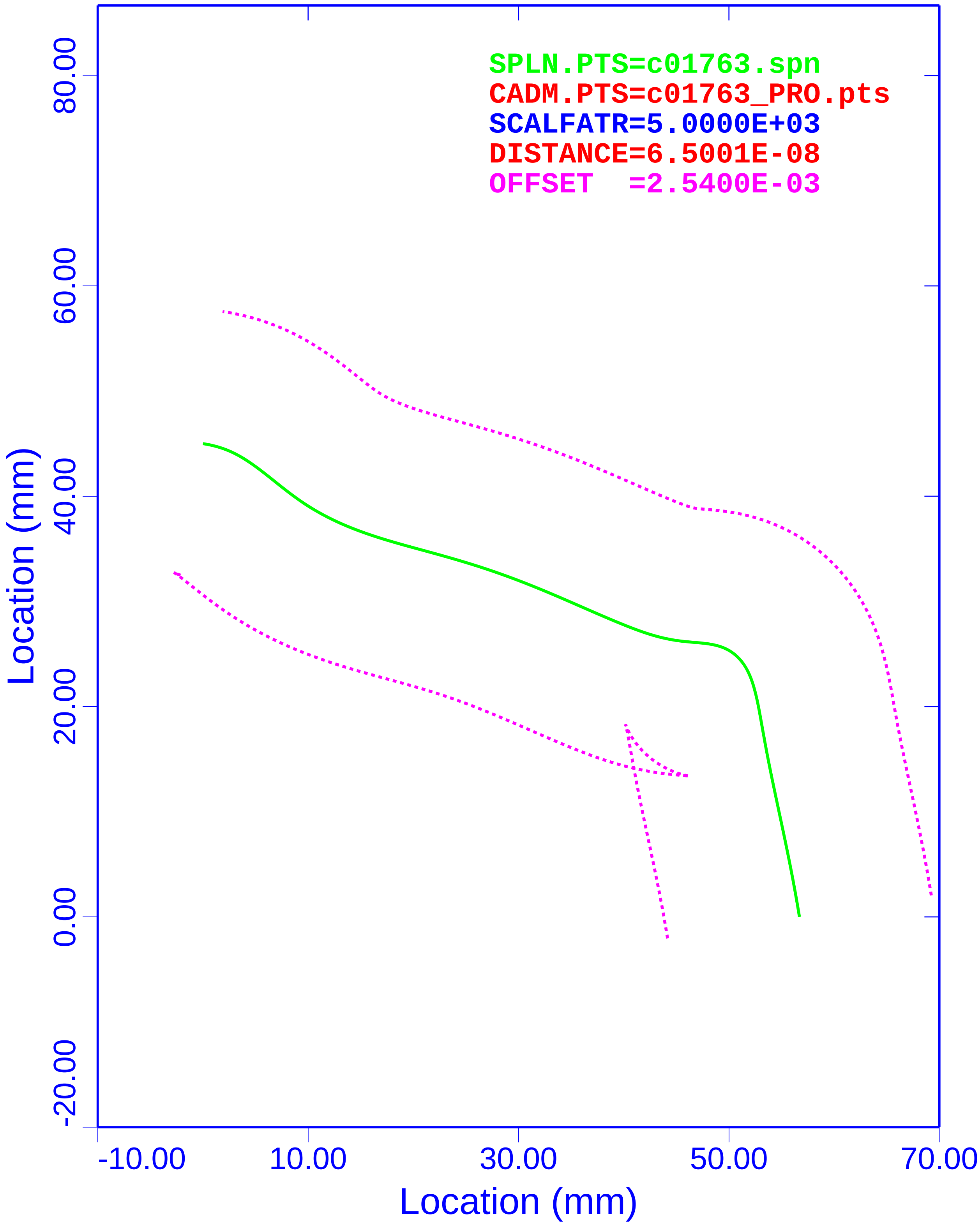

Figure 30. Nonanalytical Lampshade: PCS, Pro/E Deviations

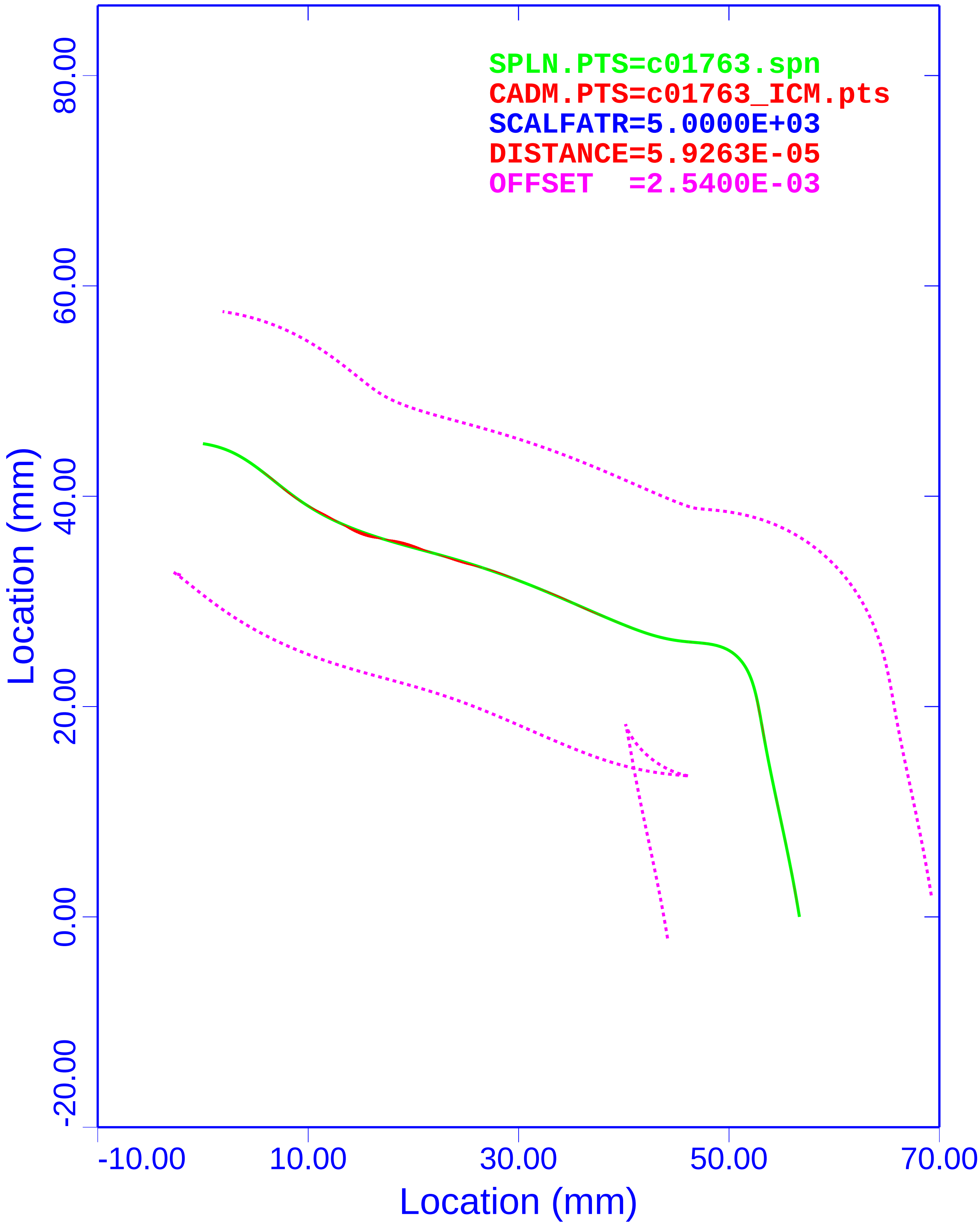

Figure 31. Nonanalytical Lampshade: WFS, ICEM DDN Deviations

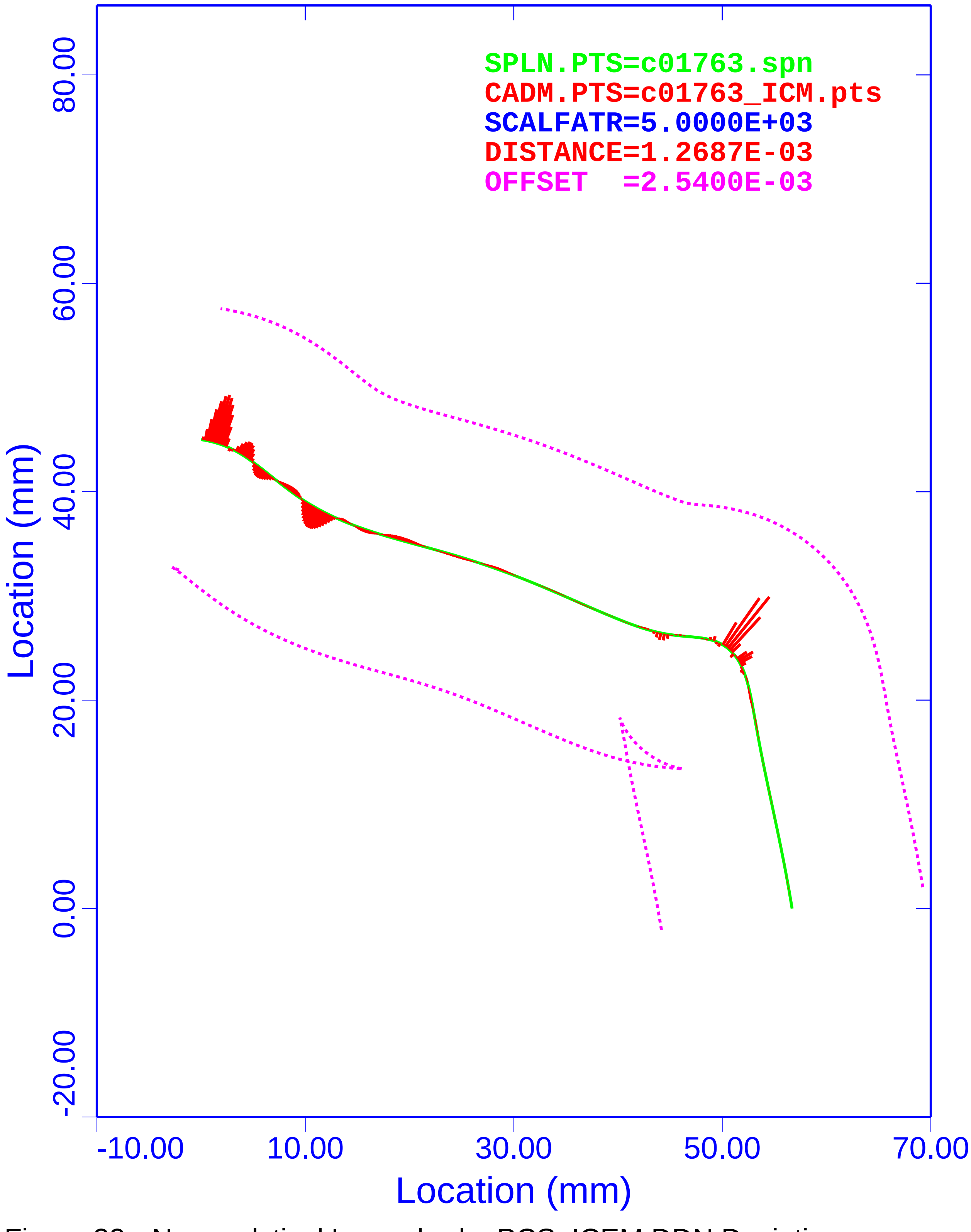

Figure 32. Nonanalytical Lampshade: PCS, ICEM DDN Deviations

Maximum and Minimum Deviations

Table 10 is a summary of the maximum and minimum deviations for the nonanalytical lampshade. Column 1 lists the associated figure that displays the results. Column 2 is the CAD system used to generate the evaluation points. The third column gives the type of evaluation spline. Columns 4 and 5 are the maximum and minimum deviations.

Table 10. Nonanalytical Lampshade: Summary of Deviations

Figure Number	CAD System	Spline Type	Maximum Deviation (mm)	Minimum Deviation (mm)
Figure 27	CADDS	WFS	+7.385725e–7	–3.636758e–5
Figure 28	CADDS	PCS	+1.269041e–3	–4.318652e–4
Figure 29	Pro/E	WFS	+4.393061e–4	–1.273383e–3
Figure 30	Pro/E	PCS	+5.684791e–8	–6.500090e–8
Figure 31	ICEM	WFS	+5.604529e–5	–5.926299e–5
Figure 32	ICEM	PCS	+1.268650e–3	–4.072064e–4

The absolute minimum and maximum deviations listed in Table 10 are 1.273383e–3 (Figure 29) and 1.269041e–3 (Figure 28), respectively. The ratios of deviations to the inspection uncertainty are 0.5013 and 0.4996, respectively.

A review of Table 10 reveals that the best combination of CAD system and spline model is Pro/E compared to the PCS model. The least desirable representations are CADDS compared to the PCS model and ICEM DDN with the PCS model.

As was stated in the section of this document entitled “Computer-Aided Design Systems,” both CADDS and ICEM DDN have WFS modules, and Pro/E uses a PCS in its sketcher option. Figure 27 and Figure 31 reflect the existence of these WFS modules. Figure 30 verifies the use of a PCS in Pro/E. The other three figures show opposite results. In essence, like modules yield better results, and unlike modules yield poorer results. The deviations for Figure 28 and Figure 32 at the equator and the region of high curvature are opposite in direction and almost equal in magnitude to those shown in Figure 29.

The results of these analyses show that the both the WFS and PCS produce results that meet all of DOE’s MBE requirements.

Nonanalytical Weird Shape

The next six figures show the comparisons of WFS and PCS representations for the nonanalytical weird shape curve with the evaluation data generated by the three different CAD systems. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the inspection uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 5,000.

Deviation Plots

Figure 33 is a graph of the nonanalytical weird shape modeled with the WFS algorithm. CADDS was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. This analysis shows that the WFS model yields a maximum deviation of 1.3288e–4 mm.

Figure 34 is a graph of the nonanalytical weird shape modeled with the PCS algorithm. CADDS was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur where the curvatures of the data are the largest. Note, in the regions of high curvature, the deviations do not damp out very well. This analysis shows that the PCS model yields a maximum deviation of 1.6642e–3 mm.

Figure 35 is a graph of the nonanalytical weird shape modeled with the WFS algorithm. Pro/E was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur where the curvatures of the data are the largest. Note, in the regions of high curvature, the deviations do not damp out very well. The analysis shows that the WFS model yields a maximum deviation of 1.6099e–3 mm.

Figure 36 is a graph of the nonanalytical weird shape modeled with the PCS algorithm. Pro/E was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the PCS model yields a maximum deviation of 2.3008e–5 mm.

Figure 37 is a graph of the nonanalytical weird shape modeled with the WFS algorithm. ICEM DDN was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red, and because of their small magnitude, they do not show on the plot. The analysis shows that the WFS model yields a maximum deviation of 1.0311e–4 mm.

Figure 38 is a graph of the nonanalytical weird shape modeled with the PCS algorithm. ICEM DDN was utilized to generate the evaluation data. The deviations of the evaluation data from the spline model are plotted in red. The most deviations occur where the curvatures of the data are the largest. Note, in the regions of high curvature, the deviations do not damp out very well. The analysis shows that the PCS model yields a maximum deviation of 1.6128e–3 mm.

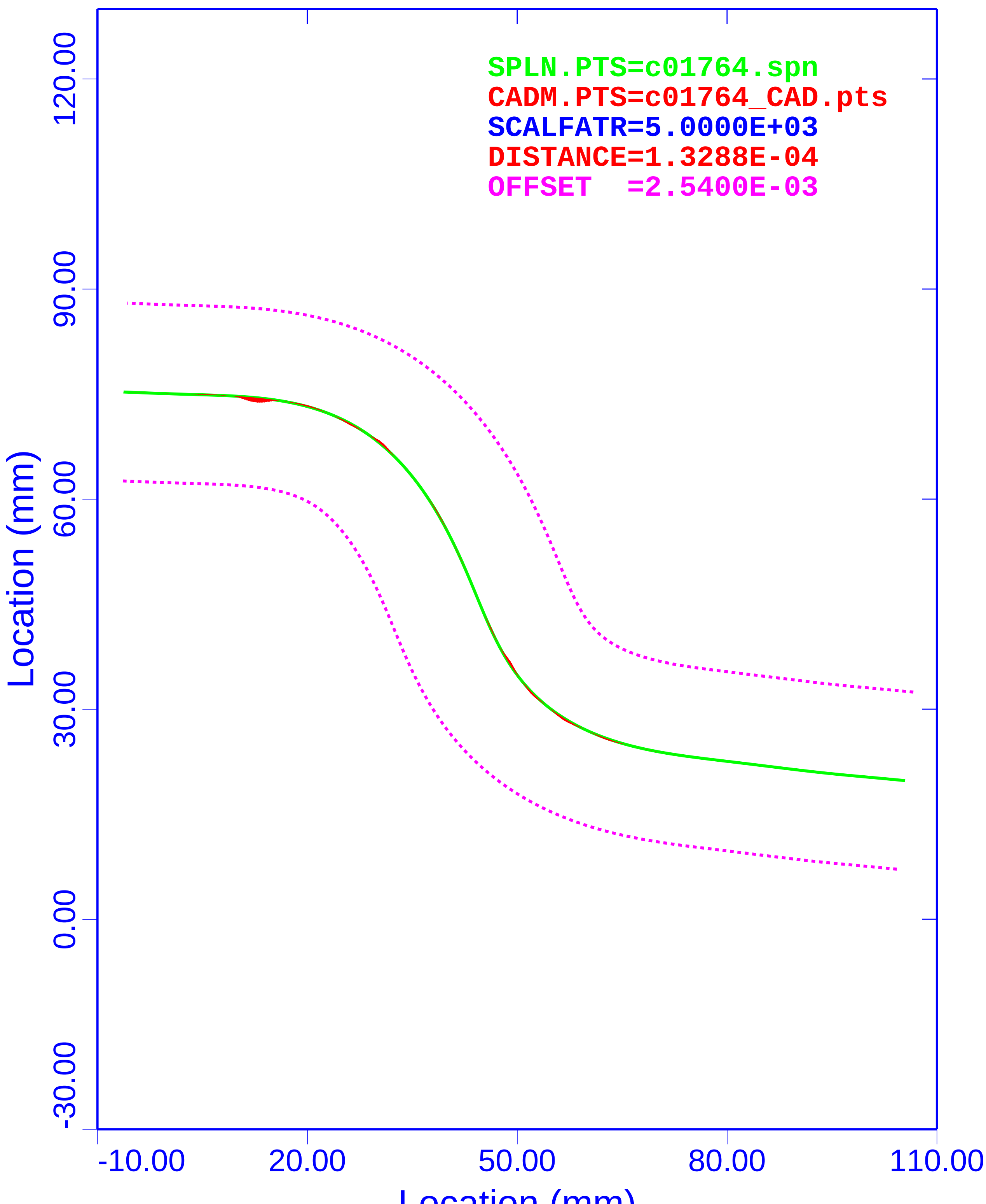

Figure 33. Nonanalytical Weird Shape: WFS, CADDS Deviations

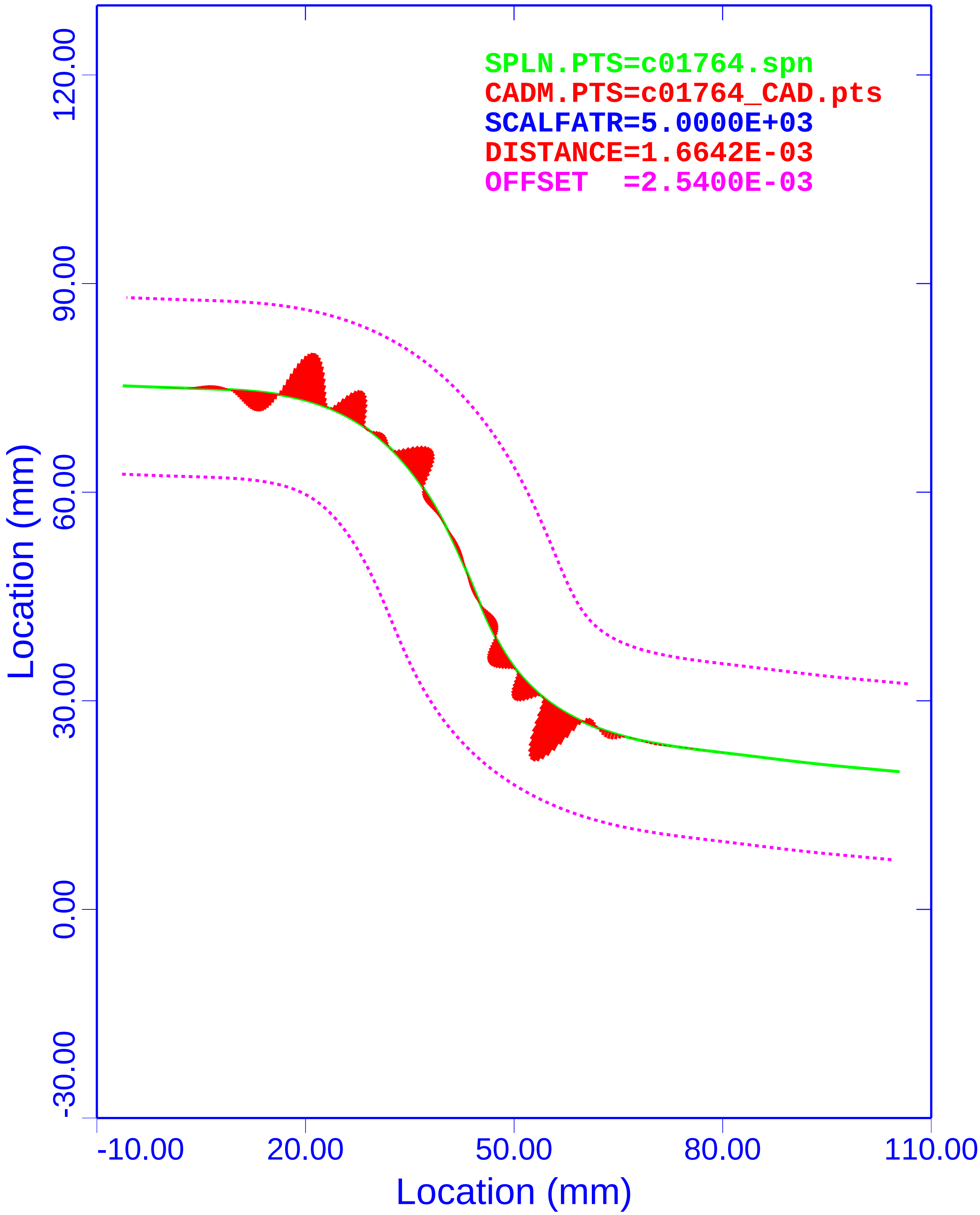

Figure 34. Nonanalytical Weird Shape: PCS, CADDS Deviations

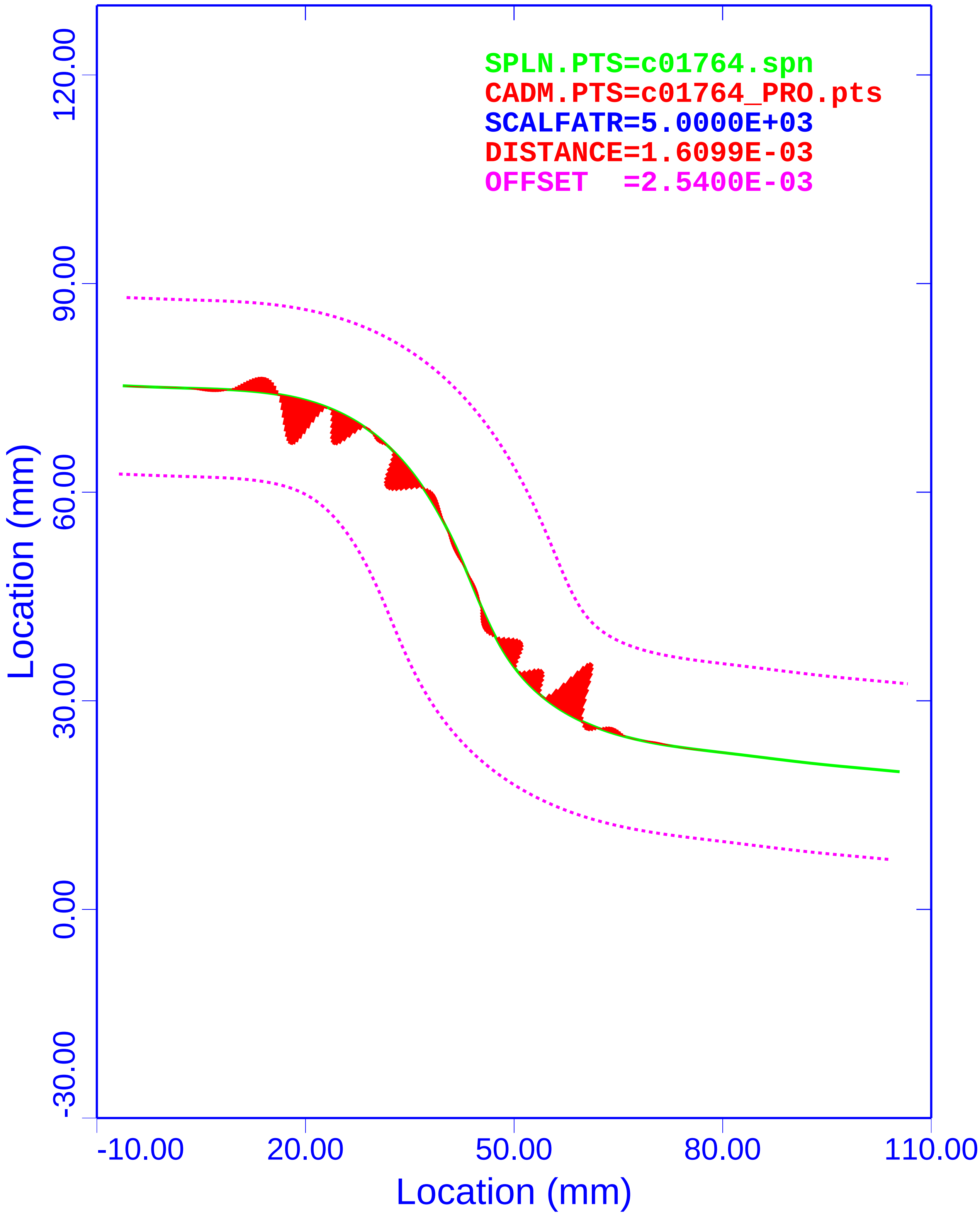

Figure 35. Nonanalytical Weird Shape: WFS, Pro/E Deviations

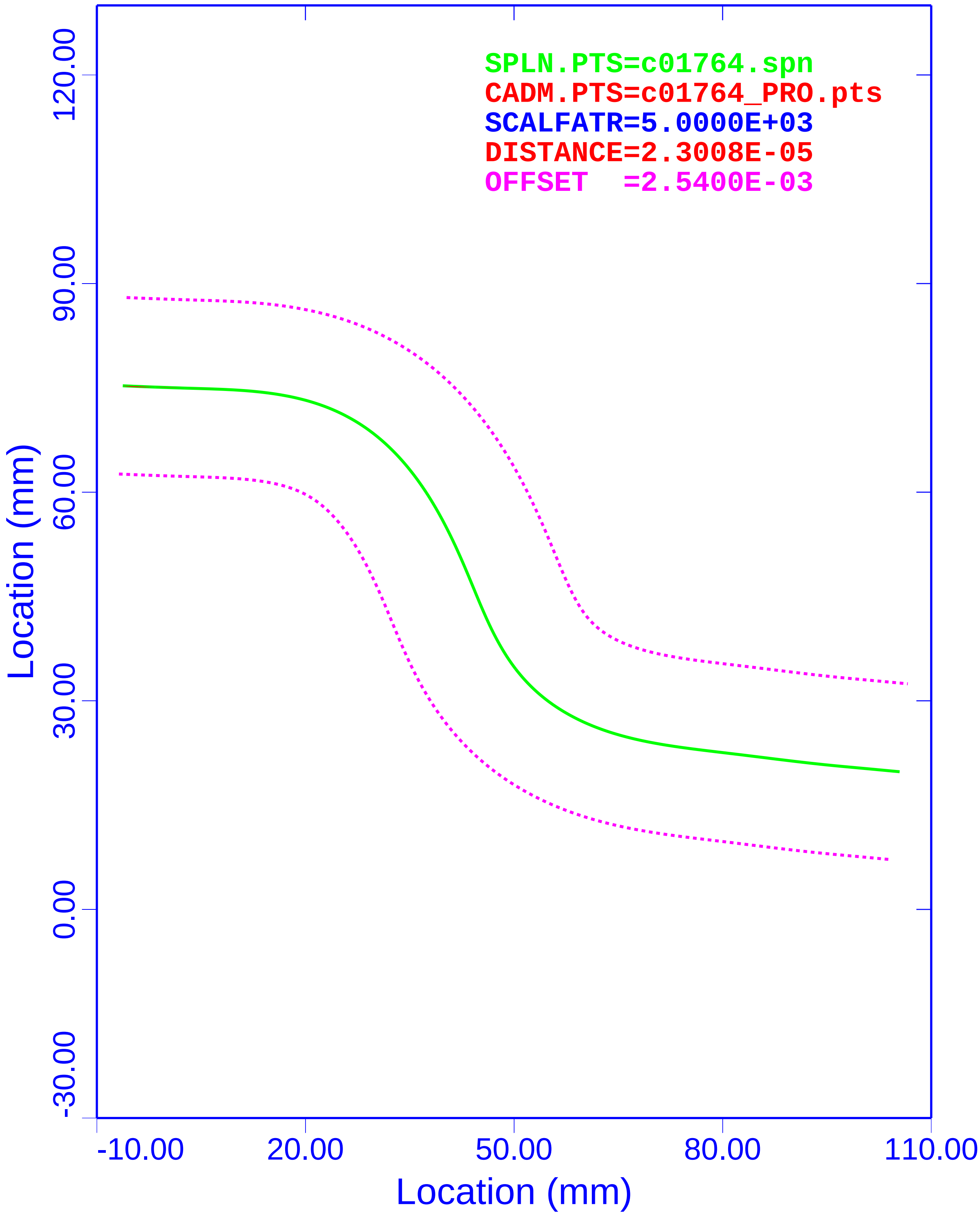

Figure 36. Nonanalytical Weird Shape: PCS, Pro/E Deviations

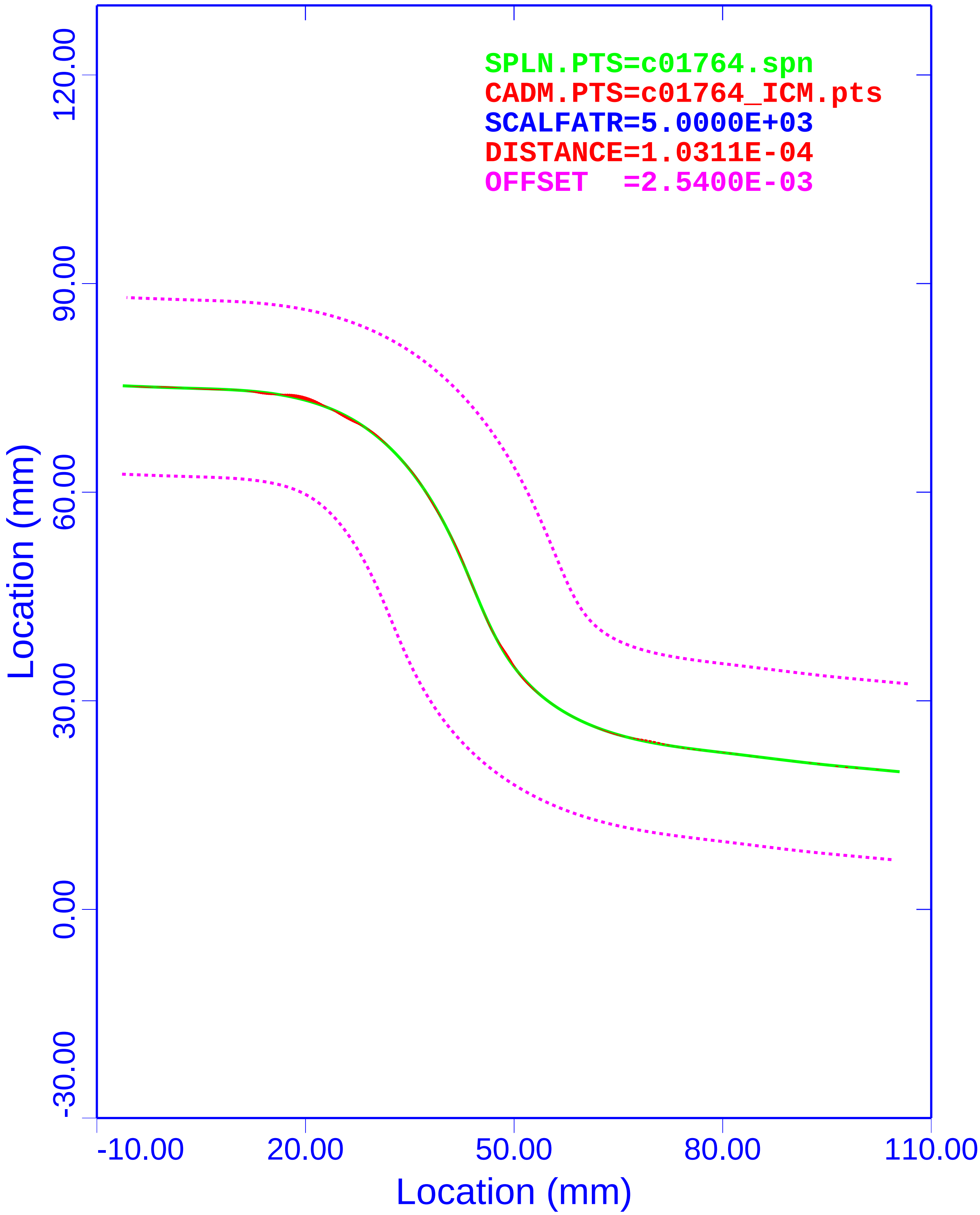

Figure 37. Nonanalytical Weird Shape: WFS, ICEM DDN Deviations

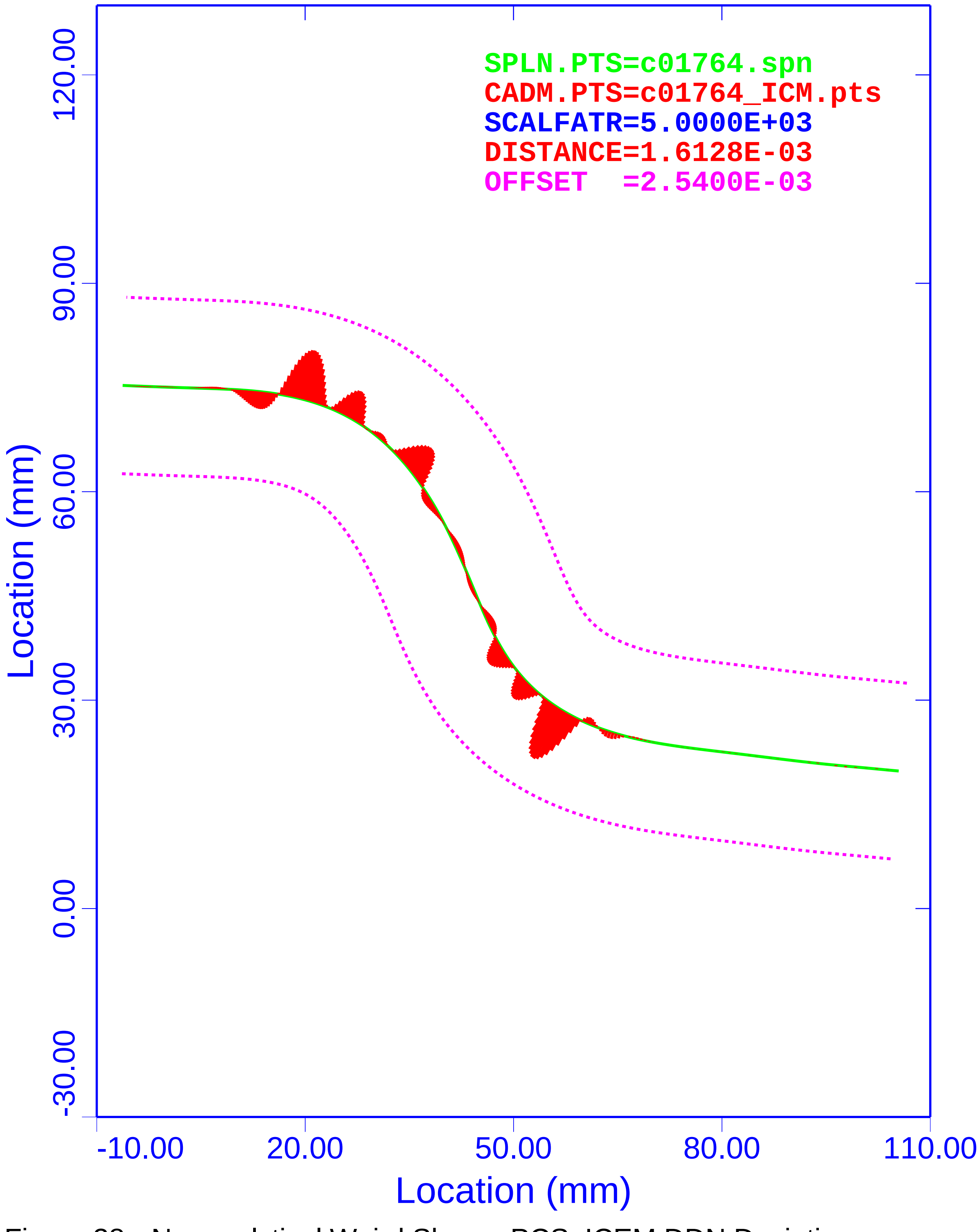

Figure 38. Nonanalytical Weird Shape: PCS, ICEM DDN Deviations

Maximum and Minimum Deviations

Table 11 is a summary of the maximum and minimum deviations for the nonanalytical weird shape. Column 1 lists the associated figure that displays the results. Column 2 is the CAD system used to generate the evaluation points. The third column gives the type of evaluation spline. Columns 4 and 5 are the maximum and minimum deviations.

Table 11. Nonanalytical Weird Shape, Summary of Deviations

Figure Number	CAD System	Spline Type	Maximum Deviation (mm)	Minimum Deviation (mm)
Figure 33	CADDS	WFS	+6.153635e–5	–1.328788e–4
Figure 34	CADDS	PCS	+1.377073e–3	–1.664233e–3
Figure 35	Pro/E	WFS	+1.609922e–3	–1.343000e–3
Figure 36	Pro/E	PCS	+6.048408e–6	–2.300825e–5
Figure 37	ICEM	WFS	+1.031108e-4	–6.984389e–5
Figure 38	ICEM	PCS	+1.445499e–3	–1.612766e–3

The absolute minimum and maximum deviations listed in Table 11 are 1.664233e–3 (Figure 34) and 1.609922e–3 (Figure 35), respectively. The ratios of deviations to the inspection uncertainty are 0.6552 and 0.6338, respectively.

A review of Table 11 reveals that the best combination of CAD system and spline model is Pro/E compared to the PCS model. The least desirable representations are CADDS compared to the PCS model and ICEM DDN with the PCS model.

As was stated in this document in the section entitled “Computer-Aided Design Systems,” both CADDS and ICEM DDN have WFS modules, and Pro/E uses a PCS in its sketcher option. Figure 33 and Figure 37 reflect the existence of these WFS modules. Figure 36 verifies the use of a PCS in Pro/E. The other three figures show opposite results. In essence, like modules yield better results, and unlike modules yield poorer results. The deviations for Figure 34 and Figure 38 at the regions of high curvature are opposite in direction and almost equal in magnitude to those shown in Figure 35.

These analyses show that both the WFS and PCS produce results that meet all of DOE’s MBE requirements.

Conclusions

Of the 18 numerical analyses presented above, the largest ratio of calculated deviation to inspection uncertainty is 0.6552. This ratio is associated with the weird-shape curve and is located at a point of high curvature. This study shows that all three CAD systems can be used to design, inspect, and fabricate parts for both legacy data and new models.

Effects of End Angles

To assure that the three-dimensional solid models are independent of the CAD system, the end angles of the splines must be defined. All three CAD systems being evaluated have the option of specifying the end angles. One of the difficulties associated with building solid models from legacy data is that in a few cases, the end angles were not recorded. In these cases, estimates of the end angles must be made. The purpose of this study is to establish a bound on these angles such that the part definition will be within the inspection uncertainty.

Both the WFS and PCS formulations require that the end angles be defined. There are several methods of defining these angles, including the following approaches:

- Specified end angles
- Linear end angles
- Parabolic end angles
- Cubic end angles
- Circular end angles
- Natural end angles

All three CAD systems evaluated in this study have the capability of using the specified end angles. The designer determines the desired end angles and inputs the values in the system.

Linear end angles are defined by the slopes of linear lines passing through the first and last two points of the data set, respectively. None of the three CAD systems have this option.

Parabolic end angles are defined by the slopes of a parabolic curve passing through the first and last three points of the data set, respectively. None of the three CAD systems have this option. However, the ICEM DDN system does utilize a parabolic end-angle option. In the ICEM DDN WFS option, the parabolic end-angle conditions have very complex definitions. The first and last segments of the spline are forced to be parabolic (not cubic), and the angles at the first (last) and second (last minus one) are defined to be equal and opposite, respectively. These angles become part of the solution of the WFS cubic coefficients.

Cubic end angles are defined by the slopes of a cubic curve passing through the first and last three points of the data set, respectively. None of the three CAD systems have this option.

Circular end angles are defined by the slopes of a circular curve passing through the first and last three points of the data set, respectively. Many of the parts fabricated in the NWC from the 1980s to the early 1990s used a circle end-condition default definition. Both CADDS and ICEM DDN have this option.

Natural end angles are defined by setting the curvatures to zero at the ends. Table 6. CAD System Spline Types and End-Angle Options, shows that Pro/E uses natural end angles as its default.

Goal

The goal of this study is to establish upper bounds on how much the end angles may vary and still have inspection results within the inspection uncertainty.

Analytical Shapes, Circle, Ellipse, and Parabola

Three analytical shapes were utilized to establish the end-angle bounds such that the deviations were within the inspection uncertainty. These shapes were a circle, an ellipse, and a parabola. The spline data and the evaluation data were generated with Mathcad. The spline and evaluation data for this study are defined in the section of this document entitled “Accuracy Study of Minimum-Distance Algorithms.”

Spline Data

Table 12 is a summary of the parameters used to characterize the six analytical shapes. This table contains the number of points used to define the shape, the data-point-spacing ranges, and the beginning and ending end angles.

Table 12. Analytical Spline Parameters

Analytical Shape	Number of Points	Spacing of Points (Degrees)	Beginning End Angle (Degrees)	Ending End Angle (Degrees)
Circle-Plus	46	2.0	90.25	180.25
Circle-Minus	46	2.0	89.75	179.75
Ellipse-Plus	46	2.0	90.25	180.25
Ellipse-Minus	46	2.0	89.75	179.75
Parabola-Plus	46	2.0	90.25	163.5508
Parabola-Minus	46	2.0	89.75	163.0508

Evaluation Data Set Parameters

Table 13 lists the parameters of these analytical shapes. The number of points, point spacing, and number of digits past the decimal point are summarized. The evaluation figures were generated at every 0.25 degree using the equations listed in the section of this document entitled “Accuracy Study of Minimum-Distance Algorithms.”

Table 13. Evaluation Data Set Parameters

Analytical Spline	Number of Points	Spacing of Points (Degree)	Digits Past Decimal Point
Circle	361	0.25	6
Ellipse	361	0.25	6
Parabola	361	0.25	6

Keyword Graphics Builder Program Command File

The command files utilized to perform the following calculations are listed in Appendix F—Keyword Graphics Builder Program— Command Files—Analytical Shapes—End-Angle Effects.

Analytical Circle

The next four figures show the comparisons of WFS and PCS representations of the analytical circle with the evaluation data generated with Mathcad. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the inspection uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 5,000.

Deviation Plots

Figure 39 shows the results of the comparison of the WFS analytical circle model to the Mathcad data. The end angles are increased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. The analysis shows that the WFS model yields a maximum deviation of 2.5917e–3 mm.

Figure 40 shows the results of the comparison of the PCS analytical circle model to the Mathcad data. The end angles are increased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. This analysis shows that the PCS model yields a maximum deviation of 2.5904e–3 mm.

Figure 41 shows the results of the comparison of the WFS analytical circle model to the Mathcad data. The end angles are decreased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. The analysis shows that the WFS model yields a maximum deviation of 2.5917e–3 mm.

Figure 42 shows the results of the comparison of the PCS analytical circle model to the Mathcad data. The end angles are decreased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. This analysis shows that the PCS model yields a maximum deviation of 2.5904e–3 mm.

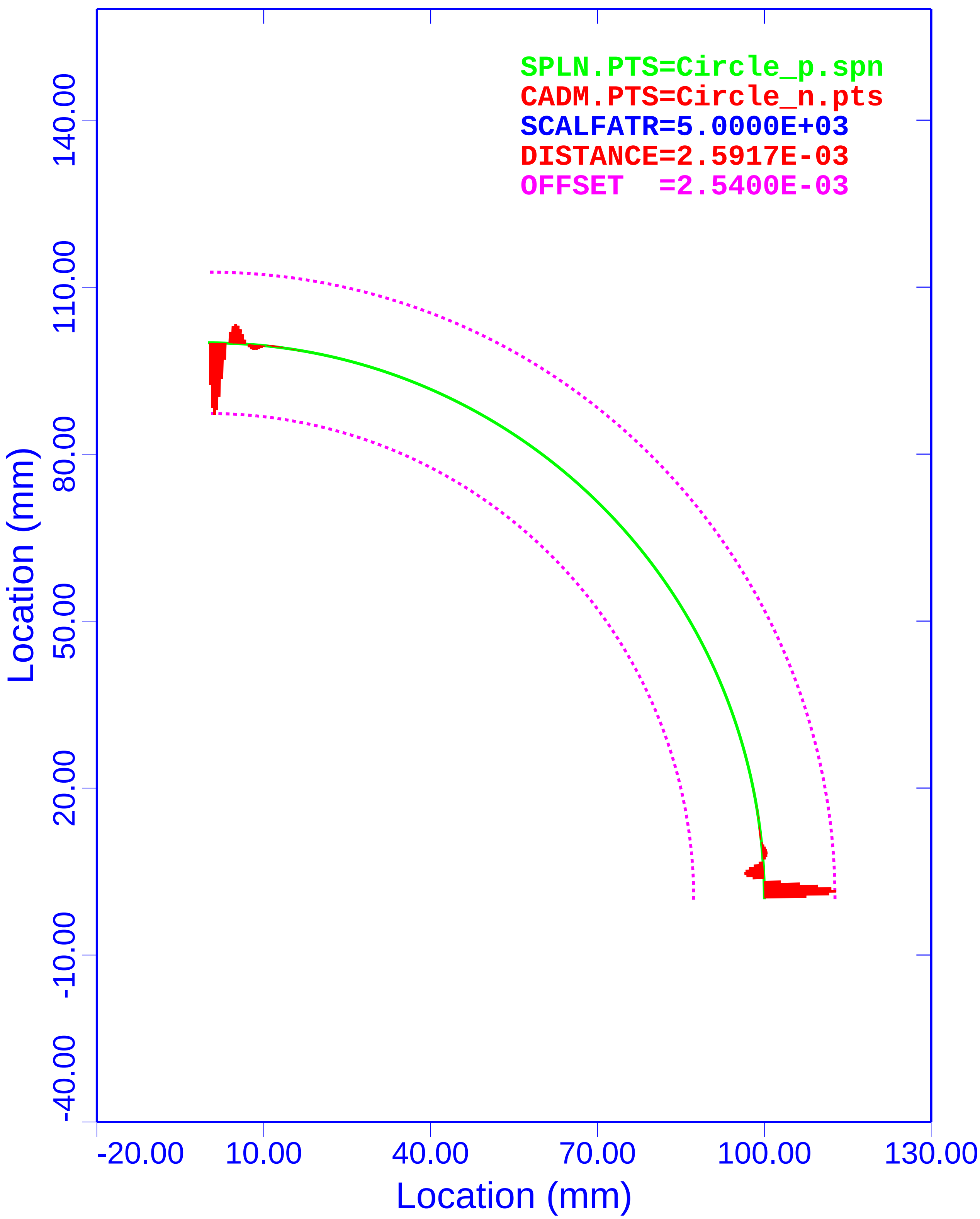

Figure 39. Analytical Circle: WFS, Deviations with +0.25 Degree End Angles

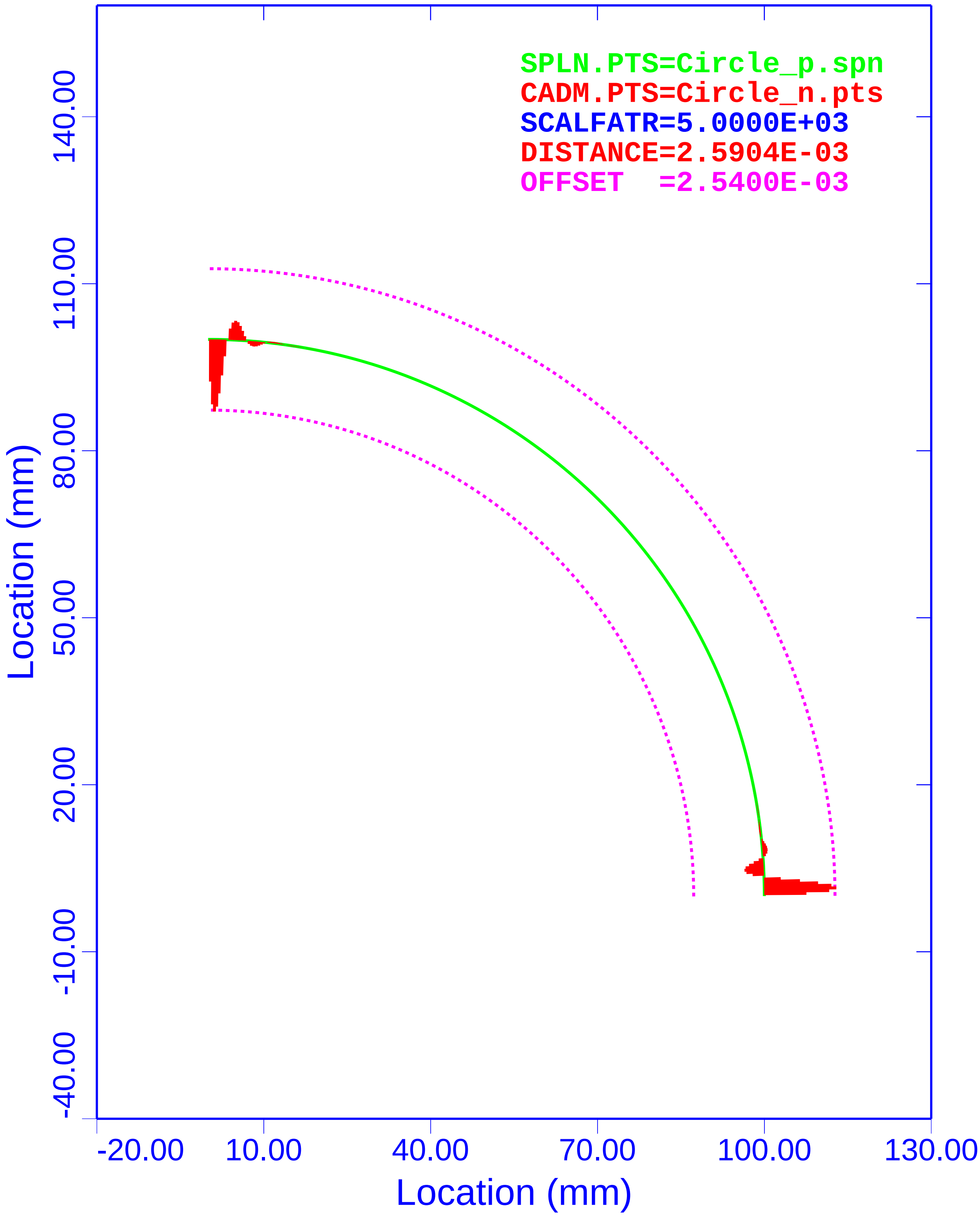

Figure 40. Analytical Circle: PCS, Deviations with +0.25 Degree End Angles

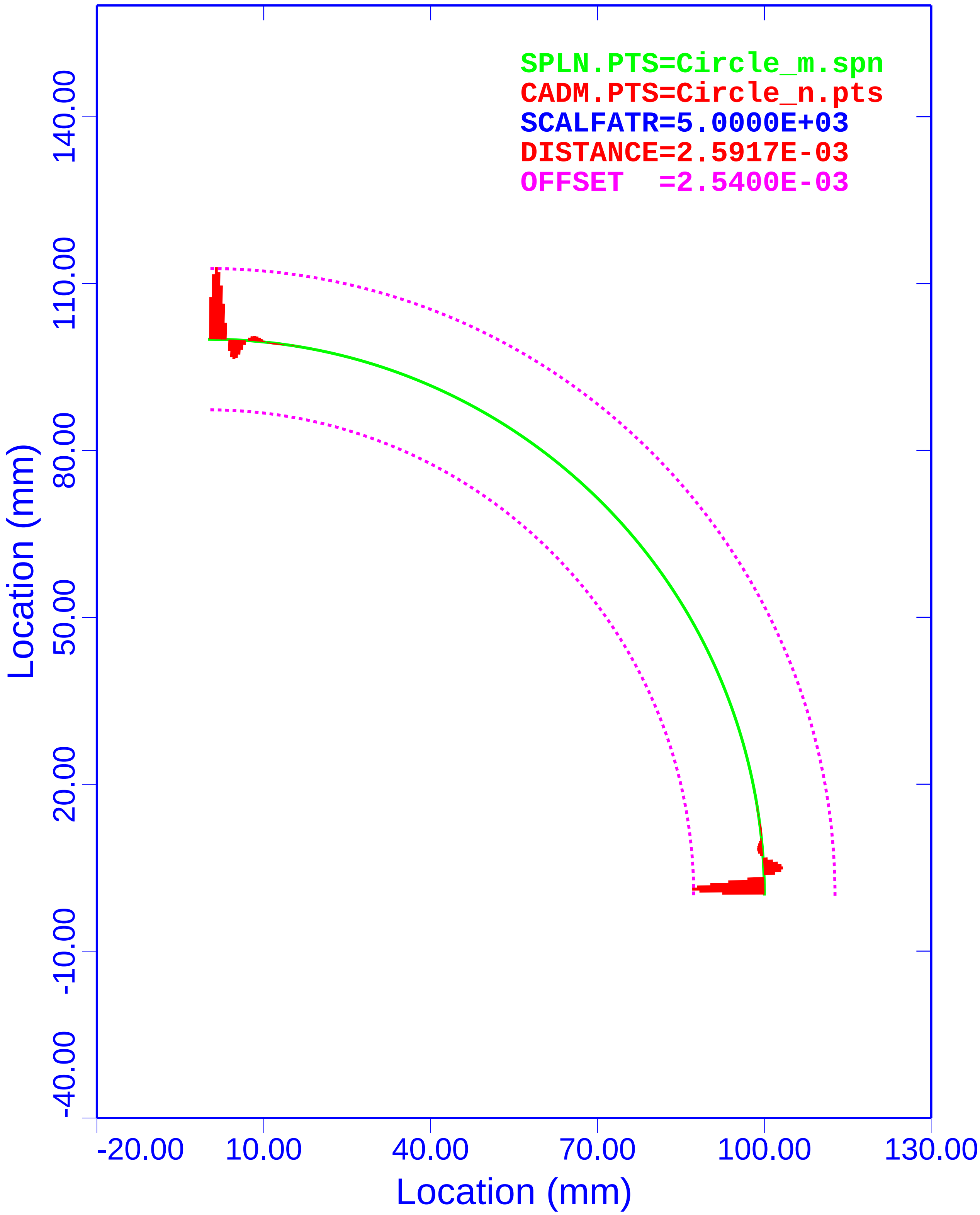

Figure 41. Analytical Circle: WFS, Deviations with –0.25 Degree End Angles

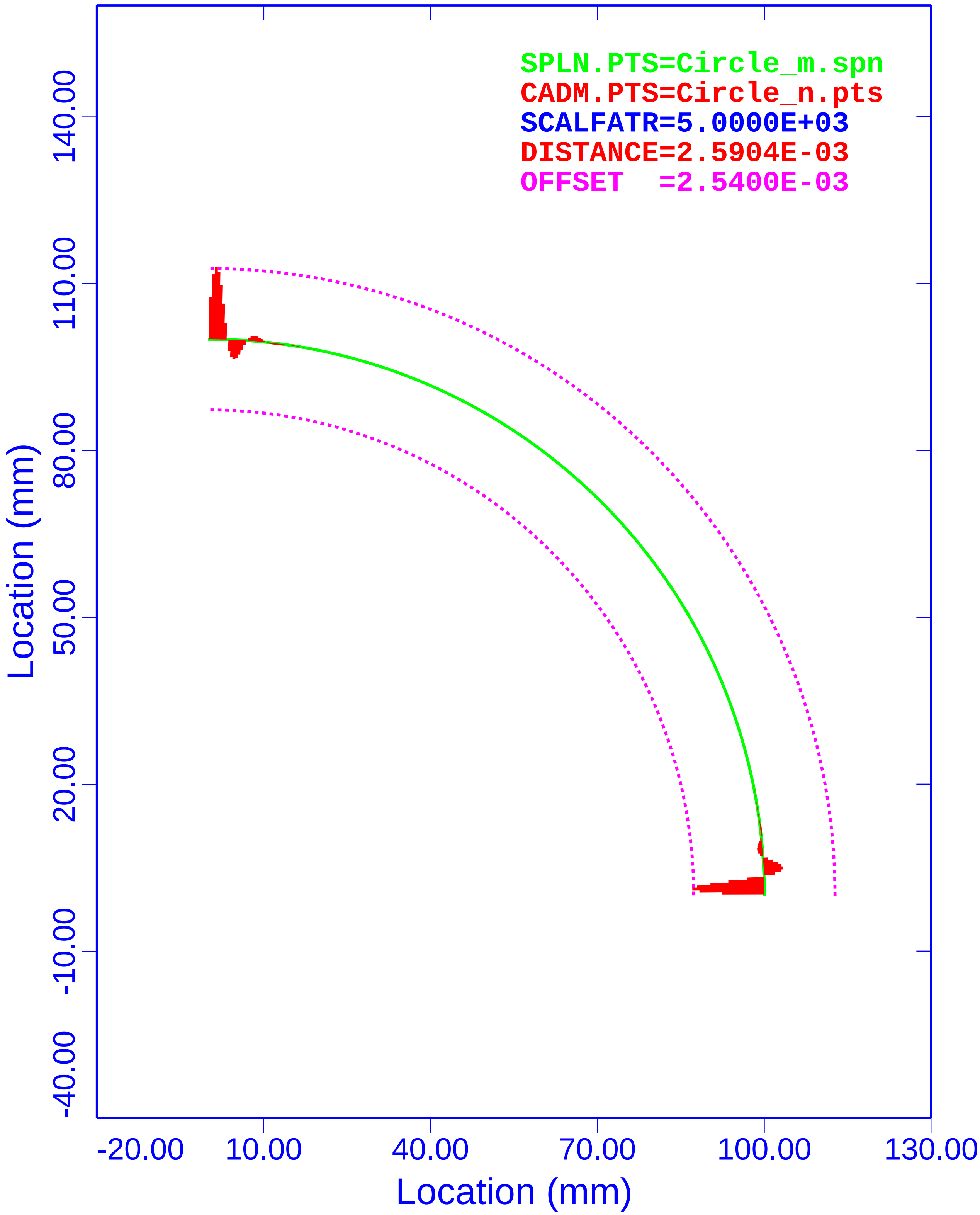

Figure 42. Analytical Circle: PCS, Deviations with –0.25 Degree End Angles

Maximum and Minimum Deviations

Table 14 is a summary of the maximum and minimum deviations for the analytical circle. Column 1 lists the associated figure that displays the results. Column 2 is the curve-fitting algorithm used to generate the evaluation points. The third column lists the end-angle changes. Columns 4 and 5 are the maximum and minimum deviations.

Table 14. Analytical Circle: Deviations with Modified End Angles

Figure	Spline Type	End Angles (Degree)	Maximum Deviation (mm)	Minimum Deviation (mm)
Figure 39	WFS	+0.25	+2.589556e–3	–2.591670e–3
Figure 40	PCS	+0.25	+2.590368e–3	–2.588462e–3
Figure 41	WFS	–0.25	+2.589556e–3	–2.591670e–3
Figure 42	PCS	–0.25	+2.590368e–3	–2.588462e–3

The absolute minimum and maximum deviations listed in Table 14 are 2.591670e–3 (Figure 41) and 2.590368e–3 (Figure 40), respectively. The ratios of deviations to the inspection uncertainty are 1.0203 and 1.0198, respectively. These calculations show that end angles may vary almost 0.25 degree and the deviations will still remain within the inspection uncertainty.

A review of the above four figures shows the effects of the end-angle changes on the deviations. Notice that at both ends of the spline, the deviations damp out between the third and fourth segments. This situation exists for both the WFS and PCS.

A review of Table 14 reveals that the deviations are antisymmetric about the ends for both spline representations. Also, the differences between the WFS algorithm and the PCS model are very small, and they are well within the accuracy of the calculations.

Analytical Ellipse

The next four figures show the comparisons of WFS and PCS representations of the analytical ellipse with the evaluation data generated by the Mathcad. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the inspection uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 5,000.

Deviation Plots

Figure 43 shows the results of the comparison of the WFS analytical ellipse model to the Mathcad data. The end angles are increased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. This analysis shows that the WFS model yields a maximum deviation of 2.5891e–3 mm.

Figure 44 shows the results of the comparison of the PCS analytical ellipse model to the Mathcad data. The end angles are increased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. This analysis shows that the PCS model yields a maximum deviation of 2.5933e–3 mm.

Figure 45 shows the results of the comparison of the WFS analytical ellipse model to the Mathcad data. The end angles are decreased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. The analysis shows that the WFS model yields a maximum deviation of 2.5941e–3 mm.

Figure 46 shows the results of the comparison of the PCS analytical ellipse model to the Mathcad data. The end angles are decreased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about four segments of the spline data. The analysis shows that the PCS model yields a maximum deviation of 2.5842e–3 mm.

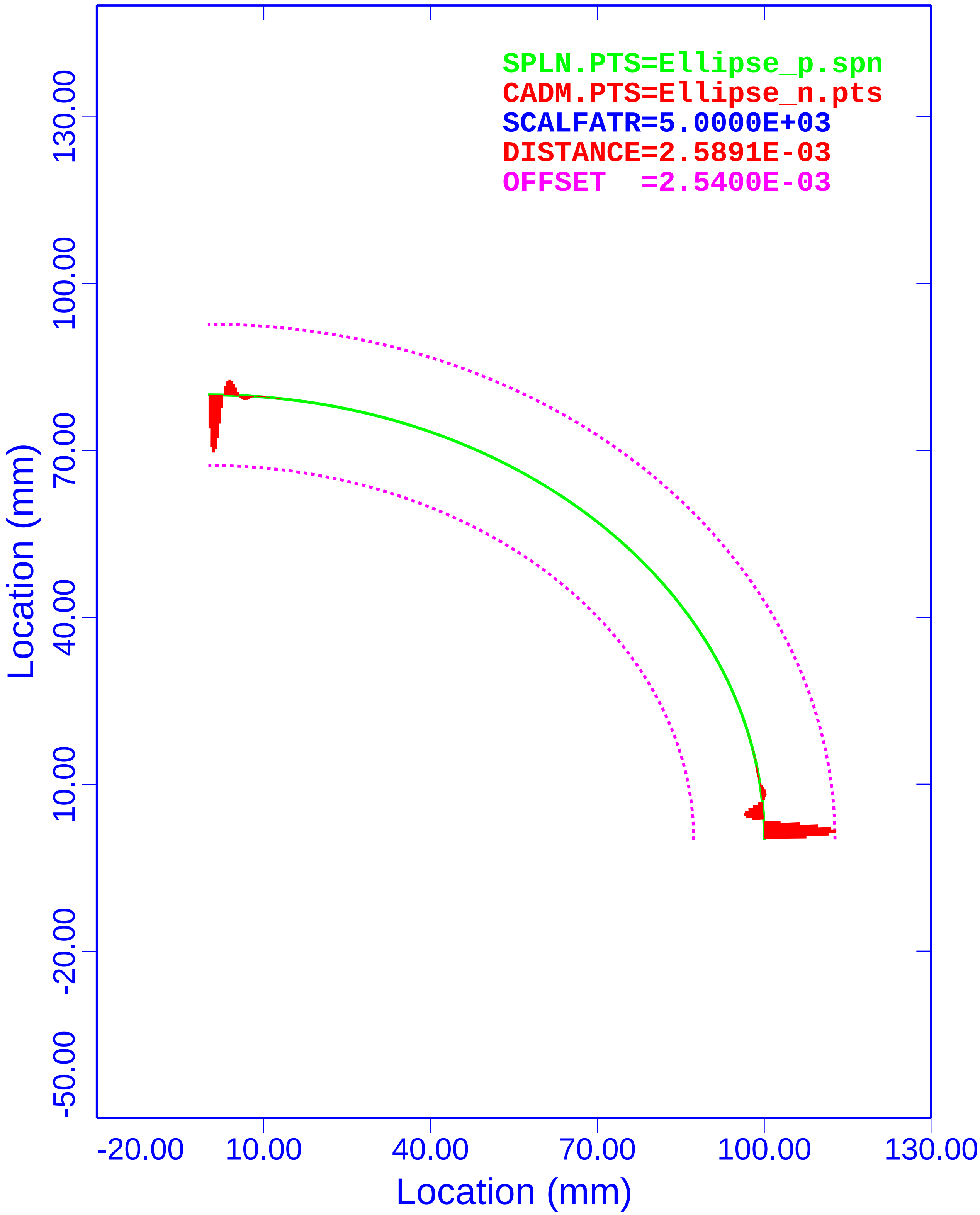

Figure 43. Analytical Ellipse: WFS, Deviations with +0.25 Degree End Angles

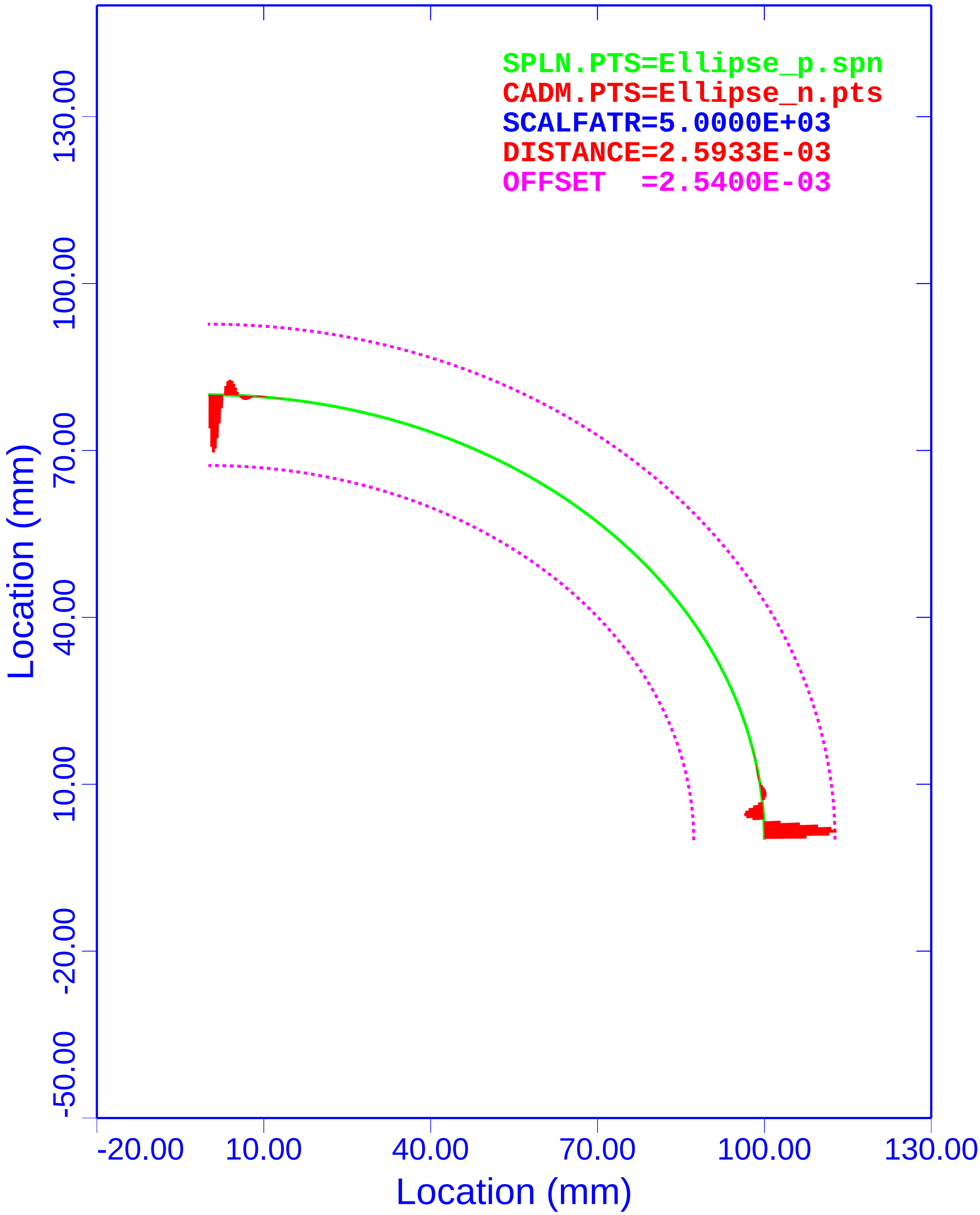

Figure 44. Analytical Ellipse: PCS, Deviations with +0.25 Degree End Angles

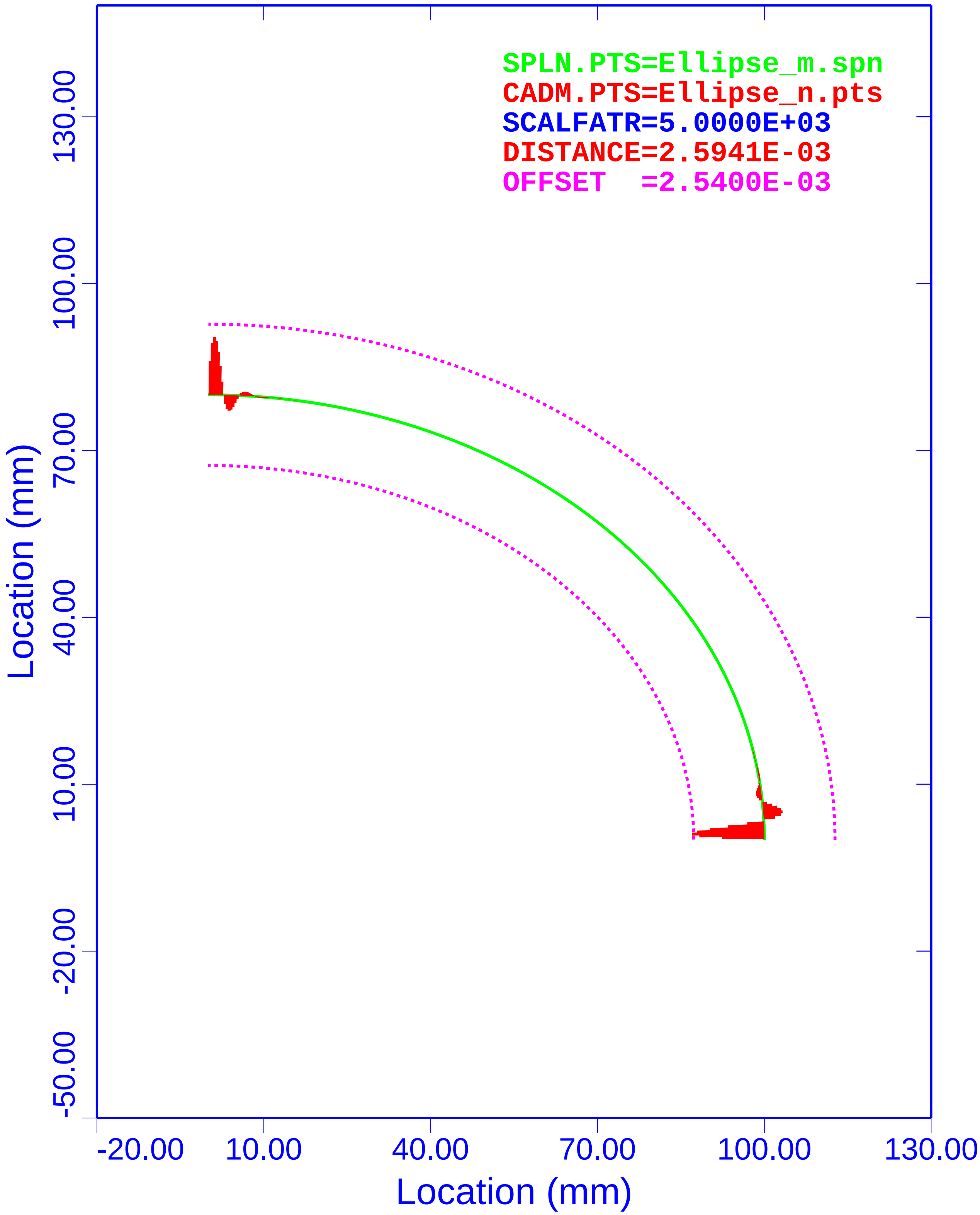

Figure 45. Analytical Ellipse: WFS, Deviations with –0.25 Degree End Angles

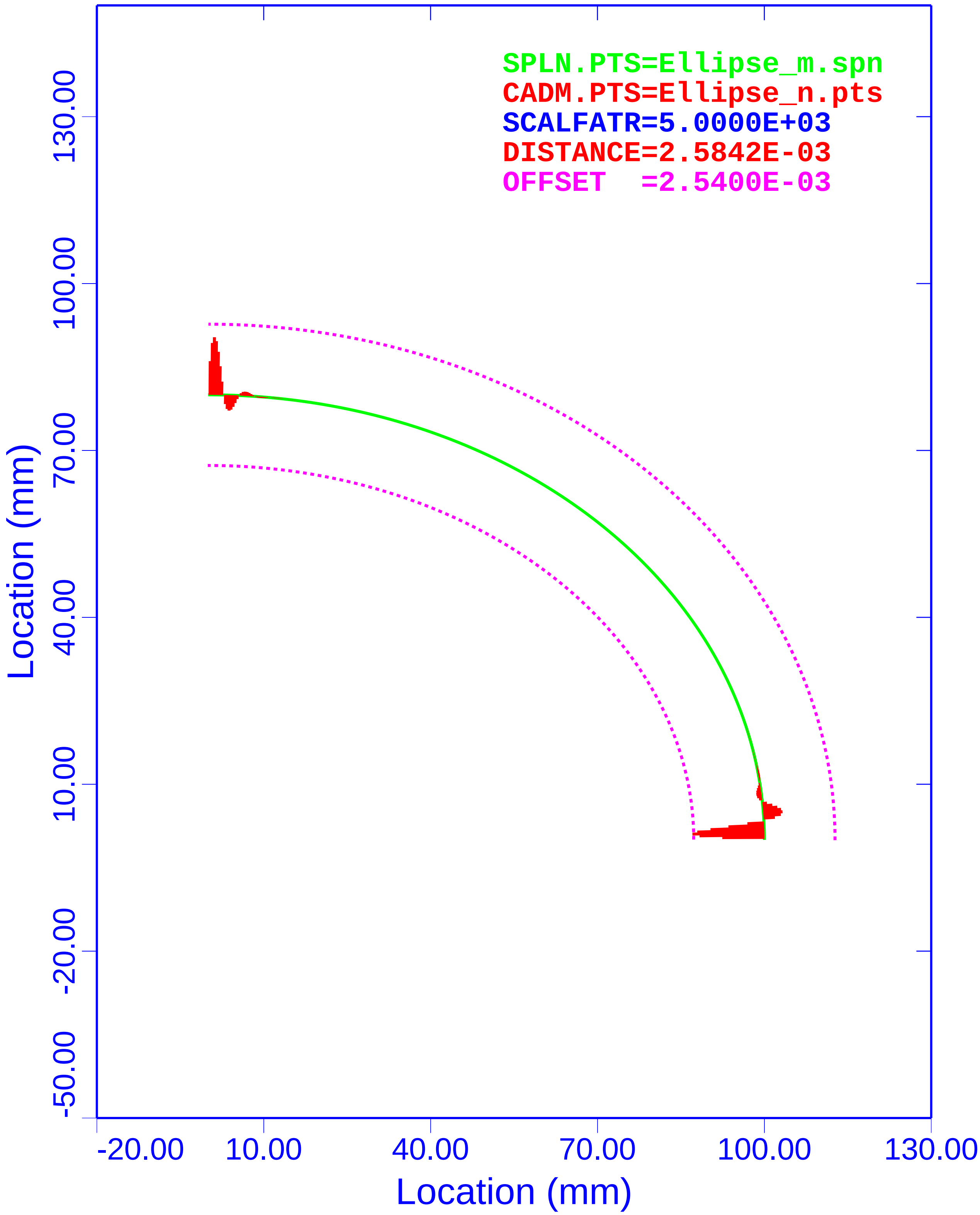

Figure 46. Analytical Ellipse: PCS, Deviations with –0.25 Degree End Angles

Maximum and Minimum Deviations

Table 15 is a summary of the maximum and minimum deviations for the analytical ellipse. Column 1 lists the associated figure that displays the results. Column 2 is the curve-fitting algorithm used to generate the evaluation points. The third column lists the end angle changes. Columns 4 and 5 are the maximum and minimum deviations.

Table 15. Analytical Ellipse: Deviations with Modified End Angles

Figure	Spline Type	End Angles (Degree)	Maximum Deviation (mm)	Minimum Deviation (mm)
Figure 43	WFS	+0.25	+2.589069e–3	–2.073179e–3
Figure 44	PCS	+0.25	+2.593280e–3	–2.072276e–3
Figure 45	WFS	–0.25	+2.071614e–3	–2.594148e–3
Figure 46	PCS	–0.25	+2.071708e–3	–2.584165e–3

The absolute minimum and maximum deviations listed in Table 15 are 2.594148e–3 (Figure 45) and 2.593280e–3 (Figure 44), respectively. The ratios of deviations to the inspection uncertainty are 1.0213 and 1.0210, respectively. These calculations show that end angles may vary almost 0.25 degree and still be within the inspection uncertainty.

A review of the above four figures shows the effects of the end-angle changes on the deviations. Notice that at both ends of the spline, the deviations damp out between the third and fourth segments. This situation exists for both the WFS and PCS.

A review of Table 15 reveals that the deviations are almost antisymmetric about the ends for both spline representations. Also, the differences between the WFS and PCS models are very small and are well within the accuracy of the calculations.

Analytical Parabola

The next four figures show the comparisons of WFS and PCS representations of the analytical parabola with the evaluation data generated with Mathcad. These plots show the spline curve (SPLN.PTS), the CAD points set (CADM.PTS), the deviation scale factor (SCALFATR), the maximum deviation (DISTANCE), and the inspection uncertainty bands (OFFSET). The SPLN.PTS are plotted to scale, and all the other curves are distorted by the scale factor of 5,000.

Deviation Plots

Figure 47 shows the results of the comparison of the WFS analytical parabola model to the Mathcad data. The end angles are increased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about five segments of the spline data. This analysis shows that the WFS model yields a maximum deviation of 2.6329e–3 mm.

Figure 48 shows the results of the comparison of the PCS analytical parabola model to the Mathcad data. The end angles are increased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about five segments of the spline data. This analysis shows that the PCS model yields a maximum deviation of 2.9093e–3 mm.

Figure 49 shows the results of the comparison of the WFS analytical parabola model to the Mathcad data. The end angles are decreased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about five segments of the spline data. The analysis shows that the WFS model yields a maximum deviation of 2.6194e–3 mm.

Figure 50 shows the results of the comparison of the PCS analytical parabola model to the Mathcad data. The end angles are decreased by 0.25 degree. The deviations of the evaluation data from the spline model are plotted in red. Note, at the ends of the spline, the deviations damp out in about five segments of the spline data. The analysis shows that the PCS model yields a maximum deviation of 2.2716e–3 mm.

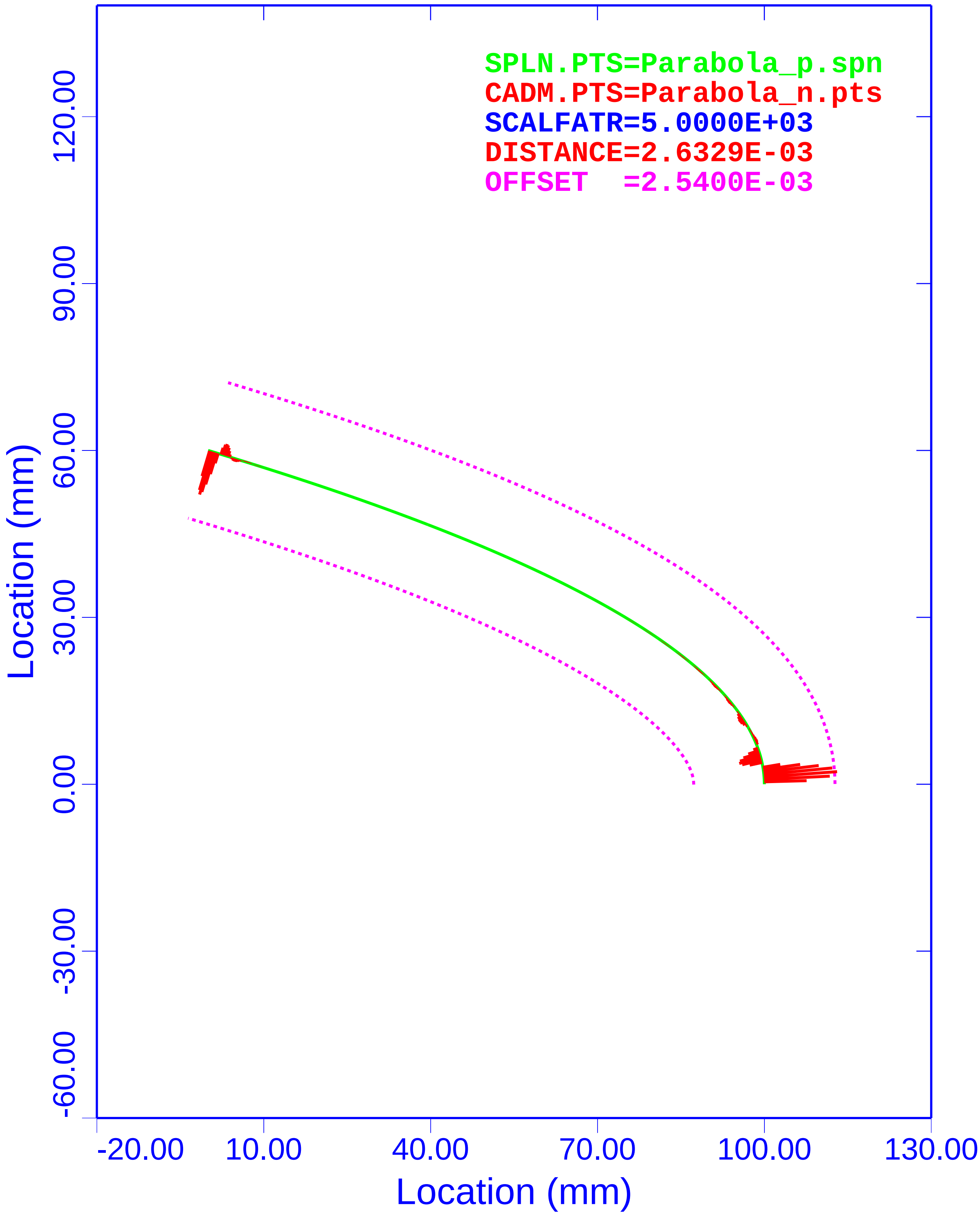

Figure 47. Analytical Parabola: WFS, Deviations with +0.25 Degree End Angles

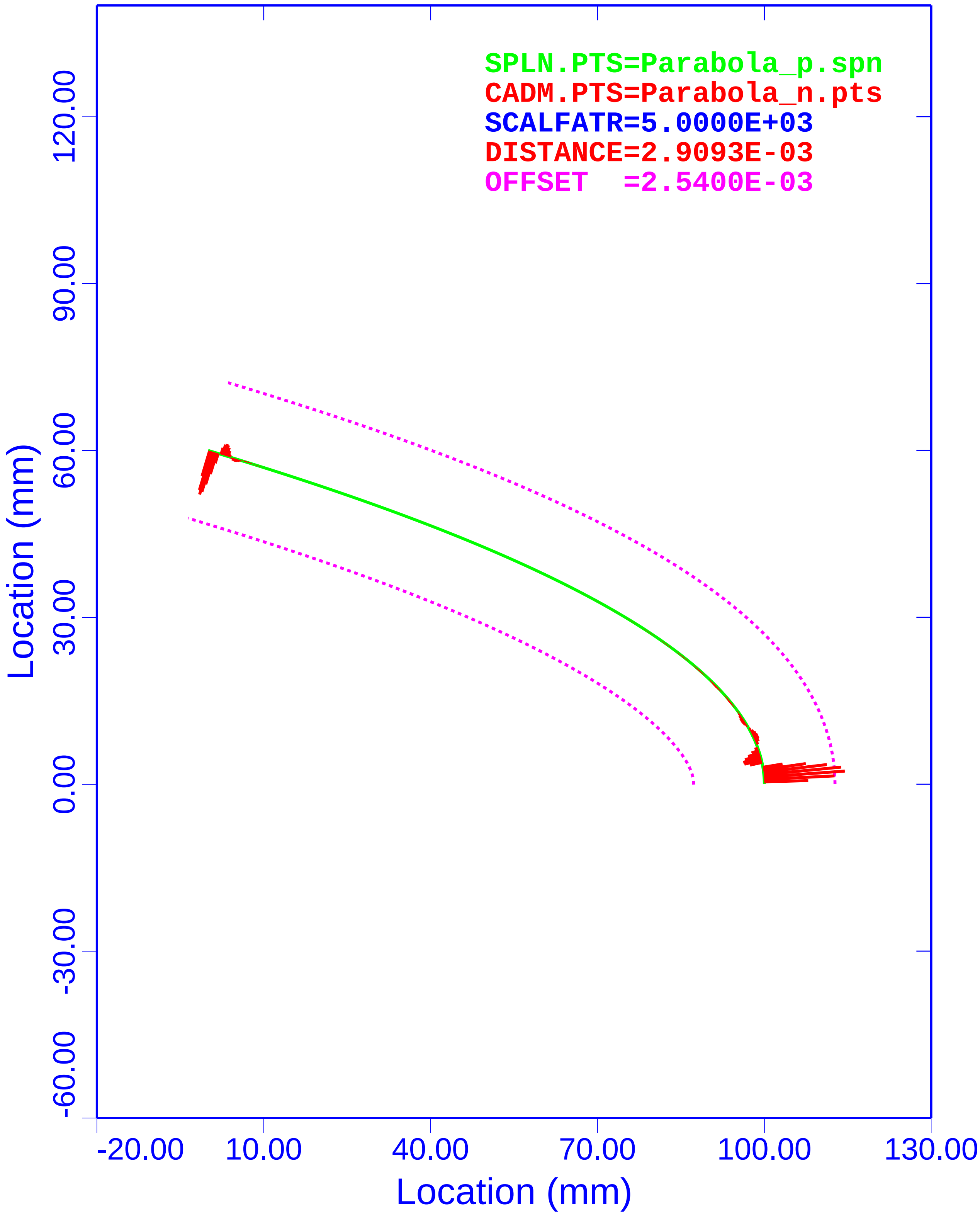

Figure 48. Analytical Parabola: PCS, Deviations with +0.25 Degree End Angles

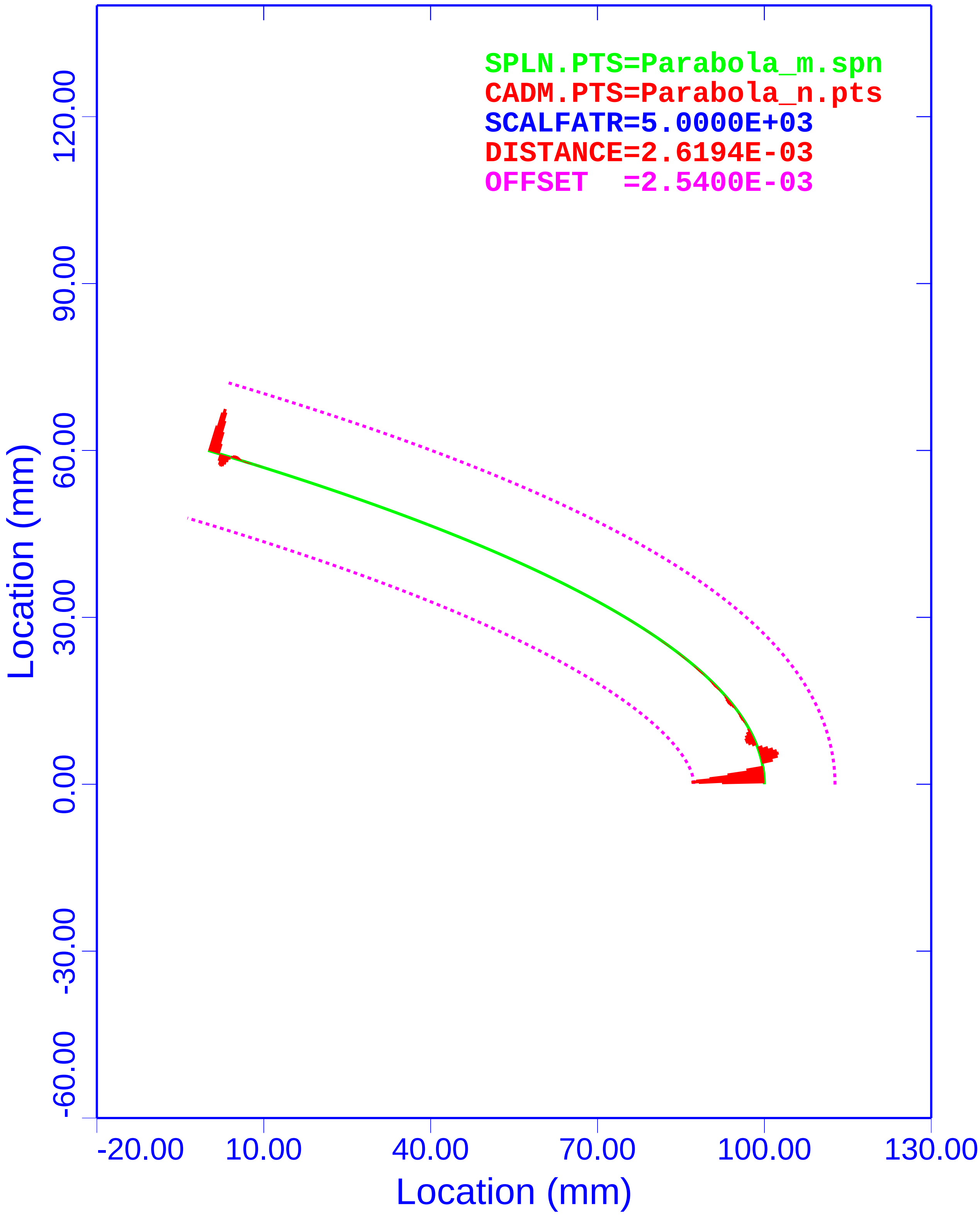

Figure 49. Analytical Parabola: WFS, Deviations with –0.25 Degree End Angles

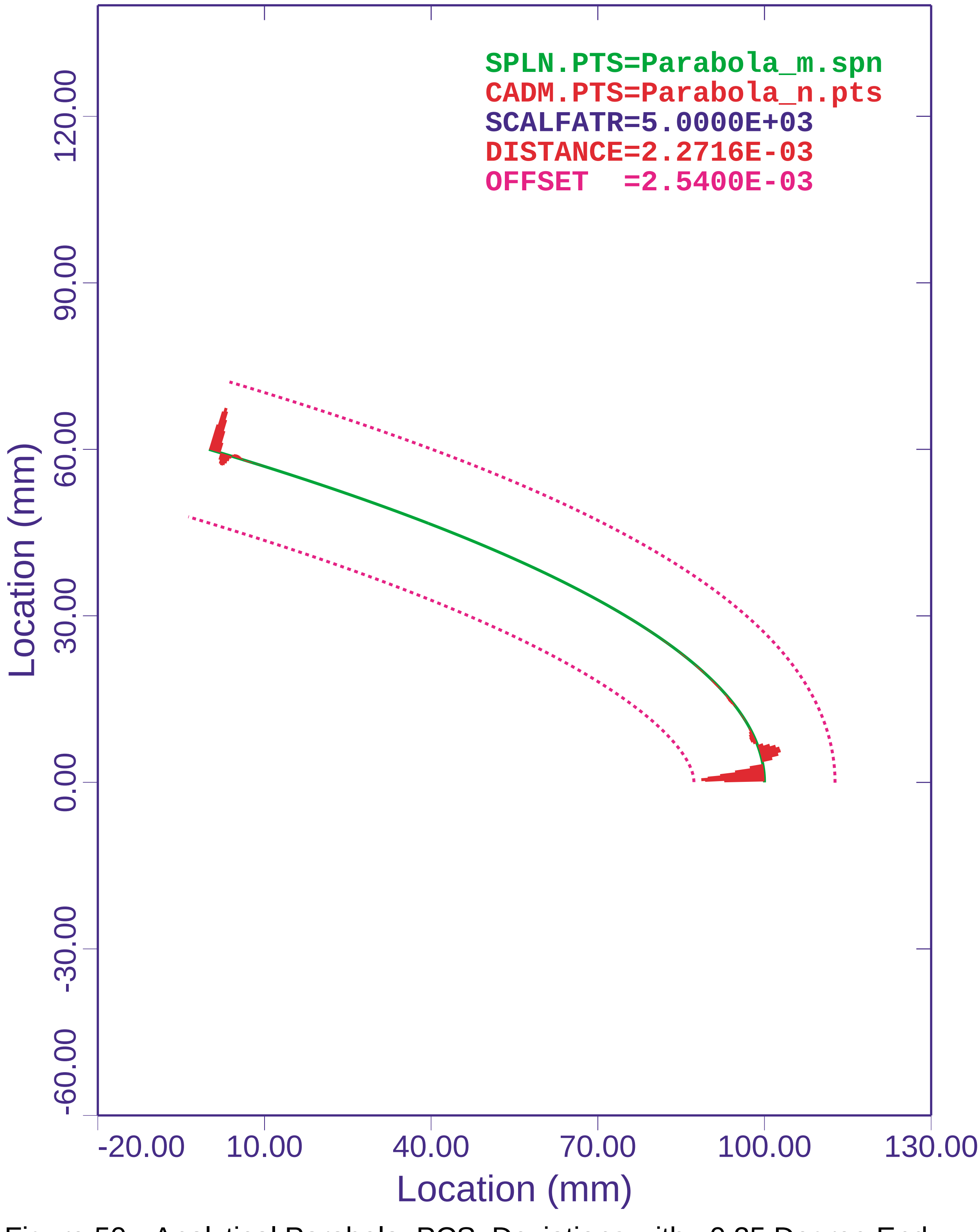

Figure 50. Analytical Parabola: PCS, Deviations with –0.25 Degree End Angles

Maximum and Minimum Deviations

Table 16 is a summary of the maximum and minimum deviations for the analytical parabola. Column 1 lists the associated figure that displays the results. Column 2 is the curve-fitting algorithm used to generate the evaluation points. The third column lists the end-angle changes. Columns 4 and 5 are the maximum and minimum deviations.

Table 16. Analytical Parabola: Deviations with Modified End Angles

Figure	Spline Type	End Angles (Degree)	Maximum Deviation (mm)	Minimum Deviation (mm)
Figure 47	WFS	+0.25	+2.632867e–3	–1.605319e–3
Figure 48	PCS	+0.25	+2.909316e–3	–1.605285e–3
Figure 49	WFS	–0.25	+1.604538e–3	–2.619395e–3
Figure 50	PCS	–0.25	+1.604532e–3	–2.271586e–3

The absolute minimum and maximum deviations listed in Table 16 are 2.619395e–3 (Figure 49) and 2.909326e–3 (Figure 48), respectively. The ratios of deviations to the inspection uncertainty are 1.0313 and 1.1454, respectively. These calculations show that end angles may vary almost 0.25 degree and still be within the inspection uncertainty.

A review of the figures above shows the effects of the end-angle changes on the deviations. Notice that at both ends of the spline, the deviations damp out between the fifth and sixth segments. This situation exists for both the WFS and PCS.

A review of Table 16 reveals that the deviations are almost antisymmetric about the ends for both spline representations. Also, the differences between the WFS and PCS models are very small and are well within the accuracy of the calculations.

Conclusions

The above analyses show that the end angle may vary almost 0.25 degree and still be within the inspection uncertainly. End-angle effects are very local and damp out in the fourth or fifth segments. The conclusion of this study is that if the end angles are within about 0.25 degree, the solid-based models can be used to design, inspect, and fabricate parts.

Summary and Conclusions

Three studies are presented in this report. The first study shows that the accuracy of the minimum-distance algorithm and the ability of both the WFS and PCS to represent analytical data sets (circle, ellipse, and parabola) are well within the inspection uncertainty. Of the 18 numerical analyses presented, the largest ratio of calculated deviation to inspection uncertainty is 0.1489. This ratio is associated with the parabola and is located at the point of highest curvature. The signs of the deviations are shown to be correct. The conclusion from this study is that both the WFS and PCS can be used to reproduce legacy data and to design new products and redesign old ones.

The second study evaluates three CAD systems (Pro/E, ICEM DDN, and CADDS), demonstrates their capabilities to model DOE legacy data, and determines that they can be utilized to develop future models. Three nonanalytical shapes (ellipse, lampshade, and a weird shape) are evaluated with the three CAD systems. Of the 18 numerical analyses presented above, the largest ratio of calculated deviation to inspection uncertainty is 0.6552. This ratio is associated with the weird shape curve and is located at the point of high curvature. This study shows that all three CAD systems can be used to design, inspect, and fabricate parts for both legacy data and new models.

The third study sets upper bounds on the variation of the end angles used to define the geometry of analytical shapes (circle, ellipse, and parabola) and still remain within the inspection uncertainty. These analyses show that, when the end angles deviate almost 0.25 degree from the nominal values, the deviations are still within the inspection uncertainly. End-angle effects are very local and damp out in the fourth or fifth segments of the spline data. The conclusion of this study is that if the end angles are within about 0.25 degree, the solid-based models still can be used to design, inspect, and fabricate parts.

The conclusions of these studies are that any CAD system that supports either PCS or B-spline can be used with confidence to reproduce DOE legacy data or to design, inspect and fabricate old and new parts. The NWC should move on to these modern systems, knowing that the legacy data generated by the WFS algorithm can be reproduced well within the inspection limits.

Appendix A—WFS Routines

```
      SUBROUTINE APTWF(NPTS,ANGIN,ANGOUT,IAFLG,WFSPL)
c-----------------------------------------------------------------------
c     SUBROUTINE APTWF(WFSPL,NBLPTS)
c
c This subroutine controls the generation of the baseline APT Wilson-
c Fowler spline.  The main features of this sub_routine are to put
c the IS standard data FORMAT into an APT FORMAT and to call two main
c APT routines (APT088 and APT089).  The reason that we convert to the
c APT data FORMAT is because we decided to not make any changes to the
c original APT program, thus perserving our purest baseline assumtions.
c
c Routine APT088 takes the initial through point data, computes slope
c segment angles and segment lengths and load the TAB array in the
c proper FORMAT.  After APT088 is done, routine APT089 can be called to
c the WF cubic coeficients.  The spline fitting method is:
c
c 1. Approximate the slopes at each interior point by assigning each
c point   the slope of a circle which passes through that point and the
c adjacent   point on each side (3 point circle).
c 2. At each interior point calculate the difference in curvature
c between   the cubic equation on one side of the point and the cubic
c equation on   the other side of a point (i.e., delta curvature) in
c terms of the   exterior angles.
c
c The final form of the data array is:
c   tab(1)    - Record number of external canonical form.
c   tab(2-10) - Nine elements of 3x3 rotation matrix used to transform
c               tabulated points into u,v,w-coordinate system.
c   tab(11)   - Total number of points, including the 2 extension
c               points.
c   tab(12)   - 14.0
c   tab(13)   - u-coordinate of extension point
c   tab(14)   - v-coordinate of extension point
c   tab(15)   - Coefficient of third degree cubic term for 1st interval
c   tab(16)   - Coefficient of second degree cubic term for 1st interval
c   tab(17)   - Length of 1st interval.
c   tab(18)   - Maximum value of cubic in 1st interval.
c   tab(19)   - Minimum value of cubic in 1st interval.
c     :                     :
c     :                     :
c   tab(13+m) - u-coordinate of mth point. m=1,..,n and n = # of through
c               pts
c   tab(14+m) - v-coordinate of mth point.
c   tab(15+m) - Coefficient of third degree cubic term for mth interval
c   tab(16+m) - Coefficient of second degree cubic term for mth interval
c   tab(17+m) - Length of mth interval.
c   tab(18+m) - Maximum value of cubic in mth interval.
c   tab(19+m) - Minimum value of cubic in mth interval.
c     :                     :
c     :                     :
c   tab(13+n) - u-coordinate of last extension point.
c   tab(14+n) - v-coordinate of last extension point.
c
c After the APT WF spline is computed, it is defined as a series of
c local(u,v) coordinate systems.  Each local coordinate system is
c defined by the cubic:
c             v(u) = c1*u**3 + c2*u**2 + c3*u + c4
c
c By applying boundary conditions on the local interval
c             v(0)  = 0,  v(L)  = 0
c             v'(0) = TA, v'(L) = TB,
```

```
c The local cubic reduces to
c             v(u) = c1*u**3 + c2*u**2 + c3*u,  where c3=TA
c
c since v(L) = 0, we get that c1*u**3 + c2*u**2 + c3*u = 0
c which implys:    c3 = -c1*u**2 - c2*u
c
c After the series of local cubics is defined, they will be transformed
c into global representation this will involve translation, rotation and
c a reparameterization.  The new parameterization is w.r.t. the spline's
c cumulative choord length.
c
c See chapter 5 of the following report for more inFORMATion.
c
c Dolin, R. M., "The Wilson-Fowler Spline in a Global IGES Coordinate
c                Frame," Los Alamos National Laboratory Report Number
c                LA-11024-MS, September, 1987, Los Alamos, NM.
c
c-----------------------------------------------------------------------
c
c
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
C
cwb   PARAMETER(NCOEF=13, MXKNOTS=201, DEGRAD=0.01745329)
      PARAMETER(NCOEF=13, MXKNOTS=201)
C
Cwb
      DIMENSION TAB(20*MXKNOTS),COEF(8,MXKNOTS),WFSPL(NCOEF,MXKNOTS)
c
c Determine if end angles are to be specified or computed.  KK and LL
c are end angle flags.  When they =1, end angles exist and =0 no end
c angles exist.  The KK flag is for the entry and the LL flag for exit
c of the spline.  When end angles (slopes) are given, they should be in
c radians.  APT defines the slope to be the tangent of the start or end
c angles.  Hence, what we want to give the APT routines is the Tangent
c of the end angles.
Cwb
      PI=4.0D0*DATAN(1.0D0)
      DEGRAD=PI/180.0D0
cwb
      DO 10 I = 1,18
        TAB(I)   = 0.0D0
   10 CONTINUE
c
      IF (IAFLG .EQ. 1) THEN
        TAB(1) = DTAN(ANGIN)
        TAB(2) = DTAN(ANGOUT)
        KK     = 1
        LL     = 1
      ELSE
        TAB(1) = 0.0D0
        TAB(2) = 0.0D0
        KK     = 0
        LL     = 0
      END IF
      NN     = NPTS*2 + 17
      J      = 1
      DO 100 I = 18,NN,2
        TAB(I)   = WFSPL(2,J)
        TAB(I+1) = WFSPL(6,J)
        J        = J + 1
  100 CONTINUE
c
c Define the necessary APT parameters.  MM is a coordinate definer, =3
```

```
c  => XY
c coordinate data.  JJ is a T&R flag, =0 => no data translation or
c rotation.
      MM     = 3
      JJ     = 0
cwb      RESULT = 0.D0
c
c Load the through points into the APT array and compute the initial
c set of local intervals
c
       CALL APT088 (MM, NPTS, KK, LL, JJ, TAB,ANGIN,ANGOUT)
c
c Compute the series of local cubics defined over each through point
c interval.
c
      CALL APT089(TAB)
c
c Convert the series of local cubics into a global spline definition
c with a cummulative choord length parameterization.  Begin by computing
c the linear term for each of the local cubic equations. The local cubic
c equation is
c                   v(u) = C1*u**3 + C2*u**2 + C3*u
c
c The two global cubic equations are given as:
c
c                   x(s) = Ax*s**3 + Bx*s**2 + Cx*s + Dx
c                   y(s) = Ay*s**3 + By*s**2 + Cy*s + Dy
c where
c      Ax = -C1*sin(gamma),              Ay =  C1*cos(gamma)
c      Bx = -C2*sin(gamma),              By =  C2*cos(gamma)
c      Cx =  C3*sin(gamma) + cos(gamma)  Cy = -C3*cos(gamma)+sin(gamma)
c      Dx =  u                           Dy =  v
c and
c         gamma = atan(Dv/Du)
c
c The coef array below is defined as Dx,Cx,Bx,Ax, Dy,Cy,By,Ay
c     i=1,2,..,8
      NSIZE = 20 + (NPTS-2)*7
      J     = 1
      DO 200 I = 20,NSIZE,7
        DXI   = TAB(I+7) - TAB(I)
        DYI   = TAB(I+8) - TAB(I+1)
        C1I   = TAB(I+2)
        C2I   = TAB(I+3)
        RLEN  = TAB(I+4)
        C3I   =-C1I*RLEN*RLEN - C2I*RLEN
        GAMMA = F5ATAN(DYI,DXI)
        GAMMA = GAMMA * DEGRAD
        COSG  = DCOS(GAMMA)
        SING  = DSIN(GAMMA)
C
C COMPUTE THE GLOBAL X-COEFFICIENTS
        COEF(4,J) =-C1I*SING            ! THESE GLOBAL COEFFICIENTS
        COEF(3,J) =-C2I*SING            ! ARE THE RESULT OF ROTATING
        COEF(2,J) =-C3I*SING + COSG     ! THE LOCAL CUBIC EQUATION
        COEF(1,J) = TAB(I)              ! INTO GLOBAL COORDINATES
C
C COMPUTE THE GLOBAL Y-COEFFICIENTS     !        [X Y] = [u v(u) 1] [R]
        COEF(8,J) = C1I*COSG            ! WHERE
        COEF(7,J) = C2I*COSG            !               -COSG   -SING
        COEF(6,J) = C3I*COSG + SING     !        [R] =  SING   -COSG
        COEF(5,J) = TAB(I+1)            !                XT      YT
        J         = J + 1
  200 CONTINUE
```

```
C
C LOAD X AND Y DATA POINTS OF LAST POINT INTO COEFFICIENT ARRAY
c Use the coordinates of the last point to define end interval
      COEF(1,J) = TAB(I)            ! THE NTH POINT HAS NO CUBIC
      COEF(5,J) = TAB(I+1)          ! BUT THE KNOT POINT WILL BE USED
c
c Define the WFSPL array w.r.t. the global coefficients
      DO 300 IG = 1,NPTS
        IF (IG .EQ. 1) THEN
          WFSPL(1,IG) = 0.0D0
        ELSE
          IT          = 24 + (IG-2)*7
          WFSPL(1,IG) = TAB(IT) + WFSPL(1,IG-1)
        END IF
C
        DO 350 JG = 1,8
          WFSPL(JG+1,IG) = COEF(JG,IG)
  350   CONTINUE
  300 CONTINUE
c
c Always compute the end angles after computing APT WF coefficients.
c Return angles in radians.
C      IF (IAFLG .EQ. 0) CALL GET_ANG(WFSPL,NPTS, ANGIN,ANGOUT)
c
c MAN O MAn O Man.................O man
c
      RETURN
      END
      SUBROUTINE APTWF(NPTS,ANGIN,ANGOUT,IAFLG,WFSPL)
c-------------------------------------------------------------------
c     SUBROUTINE APTWF(WFSPL,NBLPTS)
c
c This subroutine controls the generation of the baseline APT Wilson-
c Fowler spline.  The main features of this sub_routine are to put
c the IS standard data FORMAT into an APT FORMAT and to call two main
c APT routines (APT088 and APT089).  The reason that we convert to the
c APT data FORMAT is because we decided to not make any changes to the
c original APT program, thus perserving our purest baseline assumtions.
c
c Routine APT088 takes the initial through point data, computes slope
c segment angles and segment lengths and load the TAB array in the
c proper FORMAT.  After APT088 is done, routine APT089 can be called to
c the WF cubic coeficients.  The spline fitting method is:
c
c 1. Approximate the slopes at each interior point by assigning each
c point   the slope of a circle which passes through that point and the
c adjacent   point on each side (3 point circle).
c 2. At each interior point calculate the difference in curvature
c between   the cubic equation on one side of the point and the cubic
c equation on   the other side of a point (i.e., delta curvature) in
c terms of the   exterior angles.
c
c The final form of the data array is:
c   tab(1)    - Record number of external canonical form.
c   tab(2-10) - Nine elements of 3x3 rotation matrix used to transform
c               tabulated points into u,v,w-coordinate system.
c   tab(11)   - Total number of points, including the 2 extension
c               points.
c   tab(12)   - 14.0
c   tab(13)   - u-coordinate of extension point
c   tab(14)   - v-coordinate of extension point
c   tab(15)   - Coefficient of third degree cubic term for 1st interval
c   tab(16)   - Coefficient of second degree cubic term for 1st interval
c   tab(17)   - Length of 1st interval.
```

```
c   tab(18)   - Maximum value of cubic in 1st interval.
c   tab(19)   - Minimum value of cubic in 1st interval.
c     :                     :
c     :                     :
c   tab(13+m) - u-coordinate of mth point. m=1,..,n and n = # of through
c               pts
c   tab(14+m) - v-coordinate of mth point.
c   tab(15+m) - Coefficient of third degree cubic term for mth interval
c   tab(16+m) - Coefficient of second degree cubic term for mth interval
c   tab(17+m) - Length of mth interval.
c   tab(18+m) - Maximum value of cubic in mth interval.
c   tab(19+m) - Minimum value of cubic in mth interval.
c     :                     :
c     :                     :
c   tab(13+n) - u-coordinate of last extension point.
c   tab(14+n) - v-coordinate of last extension point.
c
c After the APT WF spline is computed, it is defined as a series of
c local(u,v) coordinate systems.  Each local coordinate system is
c defined by the cubic:
c             v(u) = c1*u**3 + c2*u**2 + c3*u + c4
c
c By applying boundary conditions on the local interval
c             v(0)  = 0,  v(L)  = 0
c             v'(0) = TA, v'(L) = TB,
c The local cubic reduces to
c             v(u) = c1*u**3 + c2*u**2 + c3*u,  where c3=TA
c
c since v(L) = 0, we get that c1*u**3 + c2*u**2 + c3*u = 0
c which implys:    c3 = -c1*u**2 - c2*u
c
c After the series of local cubics is defined, they will be transformed
c into global representation this will involve translation, rotation and
c a reparameterization.  The new parameterization is w.r.t. the spline's
c cumulative choord length.
c
c See chapter 5 of the following report for more inFORMATion.
c
c Dolin, R. M., "The Wilson-Fowler Spline in a Global IGES Coordinate
c                Frame," Los Alamos National Laboratory Report Number
c                LA-11024-MS, September, 1987, Los Alamos, NM.
c
c-----------------------------------------------------------------------
c
c
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
C
cwb   PARAMETER(NCOEF=13, MXKNOTS=201, DEGRAD=0.01745329)
      PARAMETER(NCOEF=13, MXKNOTS=201)
C
Cwb
      DIMENSION TAB(20*MXKNOTS),COEF(8,MXKNOTS),WFSPL(NCOEF,MXKNOTS)
c
c Determine if end angles are to be specified or computed.  KK and LL
c are end angle flags.  When they =1, end angles exist and =0 no end
c angles exist.  The KK flag is for the entry and the LL flag for exit
c of the spline.  When end angles (slopes) are given, they should be in
c radians.  APT defines the slope to be the tangent of the start or end
c angles.  Hence, what we want to give the APT routines is the Tangent
c of the end angles.
Cwb
      PI=4.0D0*DATAN(1.0D0)
      DEGRAD=PI/180.0D0
```

```
cwb
      DO 10 I = 1,18
        TAB(I)   = 0.0D0
   10 CONTINUE
c
      IF (IAFLG .EQ. 1) THEN
        TAB(1) = DTAN(ANGIN)
        TAB(2) = DTAN(ANGOUT)
        KK     = 1
        LL     = 1
      ELSE
        TAB(1) = 0.0D0
        TAB(2) = 0.0D0
        KK     = 0
        LL     = 0
      END IF
      NN     = NPTS*2 + 17
      J      = 1
      DO 100 I = 18,NN,2
        TAB(I)   = WFSPL(2,J)
        TAB(I+1) = WFSPL(6,J)
        J        = J + 1
  100 CONTINUE
c
c Define the necessary APT parameters.  MM is a coordinate definer, =3
c  => XY
c coordinate data.  JJ is a T&R flag, =0 => no data translation or
c rotation.
      MM     = 3
      JJ     = 0
cwb      RESULT = 0.D0
c
c Load the through points into the APT array and compute the initial
c set of local intervals
c
       CALL APT088 (MM, NPTS, KK, LL, JJ, TAB,ANGIN,ANGOUT)
c
c Compute the series of local cubics defined over each through point
c interval.
c
      CALL APT089(TAB)
c
c Convert the series of local cubics into a global spline definition
c with a cummulative choord length parameterization.  Begin by computing
c the linear term for each of the local cubic equations. The local cubic
c equation is
c                    v(u) = C1*u**3 + C2*u**2 + C3*u
c
c The two global cubic equations are given as:
c
c                    x(s) = Ax*s**3 + Bx*s**2 + Cx*s + Dx
c                    y(s) = Ay*s**3 + By*s**2 + Cy*s + Dy
c where
c      Ax = -C1*sin(gamma),              Ay =  C1*cos(gamma)
c      Bx = -C2*sin(gamma),              By =  C2*cos(gamma)
c      Cx =  C3*sin(gamma) + cos(gamma)  Cy = -C3*cos(gamma)+sin(gamma)
c      Dx =  u                           Dy =  v
c and
c         gamma = atan(Dv/Du)
c
c The coef array below is defined as Dx,Cx,Bx,Ax, Dy,Cy,By,Ay
c     i=1,2,..,8
      NSIZE = 20 + (NPTS-2)*7
      J     = 1
```

```
      DO 200 I = 20,NSIZE,7
        DXI   = TAB(I+7) - TAB(I)
        DYI   = TAB(I+8) - TAB(I+1)
        C1I   = TAB(I+2)
        C2I   = TAB(I+3)
        RLEN  = TAB(I+4)
        C3I   =-C1I*RLEN*RLEN - C2I*RLEN
        GAMMA = F5ATAN(DYI,DXI)
        GAMMA = GAMMA * DEGRAD
        COSG  = DCOS(GAMMA)
        SING  = DSIN(GAMMA)
C
C COMPUTE THE GLOBAL X-COEFFICIENTS
        COEF(4,J) =-C1I*SING            ! THESE GLOBAL COEFFICIENTS
        COEF(3,J) =-C2I*SING            ! ARE THE RESULT OF ROTATING
        COEF(2,J) =-C3I*SING + COSG     ! THE LOCAL CUBIC EQUATION
        COEF(1,J) = TAB(I)              ! INTO GLOBAL COORDINATES
C
C COMPUTE THE GLOBAL Y-COEFFICIENTS     !         [X Y] = [u v(u) 1] [R]
        COEF(8,J) = C1I*COSG            ! WHERE
        COEF(7,J) = C2I*COSG            !               -COSG   -SING
        COEF(6,J) = C3I*COSG + SING     !         [R] =  SING   -COSG
        COEF(5,J) = TAB(I+1)            !                XT      YT
        J         = J + 1
  200 CONTINUE
C
C LOAD X AND Y DATA POINTS OF LAST POINT INTO COEFFICIENT ARRAY
c Use the coordinates of the last point to define end interval
      COEF(1,J) = TAB(I)             ! THE NTH POINT HAS NO CUBIC
      COEF(5,J) = TAB(I+1)           ! BUT THE KNOT POINT WILL BE USED
c
c Define the WFSPL array w.r.t. the global coefficients
      DO 300 IG = 1,NPTS
        IF (IG .EQ. 1) THEN
          WFSPL(1,IG) = 0.0D0
        ELSE
          IT          = 24 + (IG-2)*7
          WFSPL(1,IG) = TAB(IT) + WFSPL(1,IG-1)
        END IF
C
        DO 350 JG = 1,8
          WFSPL(JG+1,IG) = COEF(JG,IG)
  350   CONTINUE
  300 CONTINUE
c
c Always compute the end angles after computing APT WF coefficients.
c Return angles in radians.
C      IF (IAFLG .EQ. 0) CALL GET_ANG(WFSPL,NPTS, ANGIN,ANGOUT)
c
c MAN O MAn O Man.................O man
c
      RETURN
      END
      SUBROUTINE APT089(TAB)
crmd
crmd-----------------------------------------------------------------
c**** SOURCE FILE : M0002233.V12   ***
c*
C.....FORTRAN SUBROUTINE  ....APT089         8/68                 HG,RN
C.....FORTRAN SUBROUTINE             APT089...             3/1/68   GK
C          PART 2 OF APT088
C              FORTRAN SUBROUTINE APT089
C
C PURPOSE      TO GENERATE THE CANONICAL FORM OF A TABULATED
```

```
C              CYLINDER DEFINED BY THE POINTS THROUGH WHICH IT
C              MUST PASS BY THE FOLLOWING APT STATEMENT
C                TABCYL/*, V, TRFORM, MI, P1, **, K1, P2,
C                   P3, ..., PN, ***, KN
C                * = NOX, NOY, NOZ, XYZ, RTHETA, OR THETAR
C                ** = SLOPE OR NORMAL
C                 *** = SLOPE OR NORMAL
C
C LINKAGE      CALL APT089 (A)
C
C
C ARGUMENTS    A       ARRAY CONTAINING THE INFORMATION NECESSARY
C                      TO PLACE THE TABCYL CANONICAL FORM ON TAPE
C                      AND LATER RETRIEVE IT FROM TAPE
C
C SUBSIDIARIES TYPE                ENTRY
C              SUBROUTINE          APT040
C              SUBROUTINE          APT087
C              SUBROUTINE          APT094
C              REAL FUNCTION       ATAN
C              REAL FUNCTION       ATAN2
C              LOGICAL FUNCTION    CKDEF
C              REAL FUNCTION       COS
C              SUBROUTINE          DOTF
C              SUBROUTINE          ERROR
C              REAL FUNCTION       MINO
C              REAL FUNCTION       SIN
C              REAL FUNCTION       SQRT
C              SUBROUTINE          TABTAP
C              LOGICAL FUNCTION    ZVECT
C
C        ADDITIONS FOR PRINT /TABPRT,ON OR OFF
C        TABPRT FLAG IS CHECKED  EACH TIME BEFORE PRINTING
C        FLAG IS SET IN PRINT ROUTINE AND INITIALIZED IN APT227
c input:
c     TAB          ARRAY CONTAINING THE INFORMATION NECESSARY TO
c                  GENERATE THE TABCYL CANONICAL FORM
c     TAB(1)     - Number of data locations (i.e., size of TAB array)
c     TAB(2-10)  - Rotation matrix (set to [I] for IS applications).
c     TAB(11)    - Number of through points including extension points.
c     TAB(12)    - 14.0  Yes, its that simple
c     TAB(13-19) - Space for first extension interval (zero initially)
c     TAB(20)    - u1
c     TAB(21)    - v1
c     TAB(22)    - Slope of first segment
c     TAB(23)    - Segment angle of first segment
c     TAB(24)    - Segment length of first segment
c     TAB(25)    - None
c     TAB(26)    - None
c       :            :
c       :            :
c     TAB(13+7i) - ui
c     TAB(14+7i) - vi
c     TAB(15+7i) - Slope of first segment
c     TAB(16+7i) - Segment angle of first segment
c     TAB(17+7i) - Segment length of first segment
c     TAB(18+7i) - None
c     TAB(19+7i) - None
c
c Comment lines with an rmd or cd monicker where added by Ron Dolin and
c comment lines with an RJG CG monicker where added by Ralph Gladfelter.
crmd-----------------------------------------------------------------
c
crmd      SUBROUTINE APT089(A)
```

```
C96       SUBROUTINE APT089(TAB, ANGIN,ANGOUT)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
C
CG    INCLUDE 'BLANKCOM.INC'
C
C     UNLABELLED COMMON
C
C---      SIZE OF BLANK COMMON IS ALLOCATED RIGHT HERE.
C
crmd      INTEGER COMSIZ,SSIZ
crmd      PARAMETER (COMSIZ=36000)
crmd      PARAMETER (SSIZ=20000)
crmd      DOUBLE PRECISION COM
crmd      DIMENSION COM(COMSIZ)
C
crmd      COMMON COM
C
crmd      DOUBLE PRECISION CANON,SSCAN,BCANON,CAN
crmd      DIMENSION CANON(COMSIZ),SSCAN(SSIZ),BCANON(SSIZ),CAN(SSIZ)
CG    EQUIVALENCE (COM(1),CANON(1))
CG    EQUIVALENCE (COM(41),SSCAN(1))
CG    EQUIVALENCE (COM(41),BCANON(1))
CG    EQUIVALENCE (COM(41),CAN(1))
C
C     END OF BLANKCOM.INC
crmd      LOGICAL CKDEF,ZVECT
crmd      DOUBLE PRECISION TAB(SSIZ)
CG    EQUIVALENCE (COM(41),TAB(1))
crmd      DIMENSION A(2),VV(3),DD(12),ROTM(9),TABLE(8)
crmd      DIMENSION TEM(1000)
c ADD RJG
CG    INCLUDE 'DARRAY.INC'
C
C   ***  20.  DARRAY BLOCK  ***
C
C   PRINT BUFFER
C
CG    INTEGER CPL
CG    PARAMETER (CPL=120)
CG    CHARACTER DARRAY*(CPL)
C
CG    COMMON/DARRAY/DARRAY
C
C     END OF DARRAY.INC
crmd      DOUBLE PRECISION DY1,DY2
crmd      DOUBLE PRECISION A1,B1,A2,B2,SL1,SL2
C
CG    INCLUDE 'TOTAL.INC'
C
C    THE ORIGINAL COMMON-DIMENSION-EQUIVALENCE (CDE) PACKAGE TOTAL
C    HAS BEEN REPLACED BY INDIVIDUAL COMMON BLOCKS, AND EQUIVALENCE
C    STATEMENTS HAVE BEEN ELIMINATED AS FAR AS POSSIBLE.
C
CG    INCLUDE 'DSHAR6.INC'
C
C   ***  6.  DSHARE BLOCK 6  ***
C
CG    DOUBLE PRECISION B,C,D(12),DX1,DX2,SB,TEM(3),TEMP,V(3),X2,Y2
C
CG    COMMON/DSHAR6/B,C,D,DX1,DX2,SB,TEM,TEMP,V,X2,Y2
C
CG    SAVE /DSHAR6/
```

```
C
CG    INCLUDE 'ZNUMBR.INC'
crmd -The IS program implementation's arrays and parameters
      PARAMETER( MXKNOTS=201)
      DIMENSION TAB(20*MXKNOTS), TEM(MXKNOTS)
      DIMENSION A(2)
crmd96 DIMENSION CANON(COMSIZ), SSCAN(SSIZ), BCANON(SSIZ), CAN(SSIZ),
crmd96 D(12)
crmd96 DIMENSION VV(3),         DD(12),      V(3),         ROTM(9),
crmd96 TABLE(8)
C
C   ***  10.  ZNUMBR BLOCK  ***
C
C   REAL LITERALS
C
CG    DOUBLE PRECISION  Z0,    Z1,    Z2,    Z3,    Z5,
CG   1                 Z10,   Z90,  Z1E6, Z1E38, Z5EM1,
CG   2               Z6EM1, Z9EM1,Z11EM1,Z12EM1, Z1EM2,
CG   3               Z1EM3, Z1EM5, Z5EM6, Z1EM6, Z1EM7,
CG   4               Z1EM9, Z1EM1,   ZM1,DEGRAD,    PI
C
CG    COMMON/ZNUMBR/    Z0,    Z1,    Z2,    Z3,    Z5,
CG   1                 Z10,   Z90,  Z1E6, Z1E38, Z5EM1,
CG   2               Z6EM1, Z9EM1,Z11EM1,Z12EM1, Z1EM2,
CG   3               Z1EM3, Z1EM5, Z5EM6, Z1EM6, Z1EM7,
CG   4               Z1EM9, Z1EM1,   ZM1,DEGRAD,    PI
C
CG    INCLUDE 'LDEF.INC'
       PARAMETER (Z1EM9 = 1.0D-9)
C
C   ***  11.  LDEF BLOCK  ***
C
C   LOGICAL VARIABLES WHICH MUST REMAIN INVIOLATE
C
CG    LOGICAL JCS,PRNTON,REFFLG,SUBFLG,UNFLAG,ZFLAG,JDS,
CG   1            BOUNDF,PCHLST,CANFLG,BNDERR,TABPRT,REFMOT,ALTMLT
C
CG    COMMON/LDEF/JCS,PRNTON,REFFLG,SUBFLG,UNFLAG,ZFLAG,JDS,
CG   1            BOUNDF,PCHLST,CANFLG,BNDERR,TABPRT,REFMOT,ALTMLT
C
CG    INCLUDE 'ISHR17.INC'
C
C   *** 17.  ISHARE17 BLOCK  ***
C
C   TABCYL SHARED INTEGER VARIABLES
C   SOME OF THESE MAY BE ONLY USED AS LOCAL VARIABLES
C
      INTEGER       I,INC,I1,J,J1,K,L,LIM,L1,M,N,NM1
      COMMON/ISHR17/I,INC,I1,J,J1,K,L,LIM,L1,M,N,NM1
C
CG    SAVE /ISHR17/
C
CG    INCLUDE 'KNUMBR.INC'
C
C   ***  19.  KNUMBR BLOCK  ***
C
C      INTEGER LITERALS
C
CG    INTEGER          K0,   K1,   K2,   K3,   K4,   K5,   K6,   K7,
CG   1                 K8,   K9,  K10,  K12,  K13,  K15,  K16,  K18,
CG   2                K19,  K23,  K24,  K25,  K26,  K27,  K29,  K30,
CG   3                K31,  K32,  K33,  K34,  K44,  K45,  K46,  K47,
CG   4                K48,  K50,  K51,  K52,K1013, K1E4, K1E6,  KM1
C
```

```
CG    COMMON/KNUMBR/   K0,   K1,   K2,   K3,   K4,   K5,   K6,   K7,
CG   1                 K8,   K9,  K10,  K12,  K13,  K15,  K16,  K18,
CG   2                K19,  K23,  K24,  K25,  K26,  K27,  K29,  K30,
CG   3                K31,  K32,  K33,  K34,  K44,  K45,  K46,  K47,
CG   4                K48,  K50,  K51,  K52,K1013, K1E4, K1E6,  KM1
C
C
      DOUBLE PRECISION A1, B1, A2,B2,SL1,SL2
crmd
crmd -The following two data statements were added to define the numbers
crmd  used by this sub_routine.
crmd
      DATA K1,   K4,   K7, K15
     1    /1,     4,    7,  15/
crmd
      DATA  Z0,     Z1,     Z2,
     1         ZM1 , Z1EM3 , Z1EM7
     2     /0.D0,   1.D0,   2.D0,
     3      1.0D-1, 1.0D-3, 1.0D-7/
cwb  1      Z1EM3,  Z1EM7,  ZM1,    DEGRAD,     PI
cwb  3      1.0E-3, 1.0E-7, 1.0E-1, 0.01745329, 3.141592653589793/
crmd96
c96     DATA K0,   K1,   K2,   K3,   K4,   K5,   K6,   K7,
c96     1     K8,   K9,  K10,  K12,  K13,  K15,  K16,  K18,
c96     2     K19,  K23,  K24,  K25,  K26,  K27,  K29,  K30,
c96     3     K31,  K32,  K33,  K34,  K44,  K45,  K46,  K47,
c96     4     K48,  K50,  K51,  K52,K1013, K1E4, K1E6,  KM1
c96     5    /0,1,2,3,4,5,6,7,8,9,10,12,13,15,16,18,19,23,24,25,26,
c96     6     27,29,30,31,32,33,34,44,45,46,47,48,50,51,52,
c96     7     1013,1E4,1E6,-1 /
crmd
c96     DATA  Z0,    Z1,    Z2,    Z3,     Z5,
c96     1      Z10,   Z90,  Z1E6,   Z1E38,  Z5EM1,
c96     2      Z6EM1, Z9EM1,Z11EM1, Z12EM1, Z1EM2,
c96     3      Z1EM3, Z1EM5, Z5EM6, Z1EM6,  Z1EM7,
c96     4      Z1EM9, Z1EM1, ZM1,   DEGRAD, PI
c96     5     /0.D0, 1.D0, 2.D0, 3.D0, 5.D0, 10.D0, 90.D0, 1.0E6,
c96     6      1.0E38,   5.0E-1,   6.0E-1,   9.0E-1,
c96     7     11.0E-1,  12.0E-1,   1.0E-2,   1.0E-3,
c96     8      1.0E-5,   5.0E-6,   1.0E-6,   1.0E-7,
c96     9      1.0E-9,   1.0E-1,  -1.0,      0.01745329,
c96     &      3.141592653589793/
crmd
CG    INCLUDE 'XUNITS.INC'
C
CG    DOUBLE PRECISION TABEXT,SSEXT
CG    INTEGER IOLD
CG    CHARACTER*6 OLDMOD
C
CG    COMMON/XUNITS/TABEXT,SSEXT,IOLD
CG    COMMON/XUNITC/OLDMOD
C
C     END OF XUNITS.INC
C
CG    EQUIVALENCE (TAB(2),ROTM(1))
C
CG    CHARACTER FORM1*112,FORM2*92,FORM3*120,FORM4*28
CG    CHARACTER FORM5*4,FORM7*16,FORM9*100
C
CG    DATA FORM1/
CG   1' NUM        THETA        RADIUS         X-CORD         Y-CORD
CG   2      SEG LENGTH       SEG ANGLE      EXT ANGLE'/
CG    DATA FORM2/
CG   1' NUM     SLOPE        NORMAL      ALPHA   TANGENT A   TANGENT B
```

```
CG   2   CURVA    DELTA CURV     '/
CG    DATA FORM3/
CG   1'    CURVATURE    .+......................+....................
CG   2...+......................+......................+.'/
CG    DATA FORM4/' EXTENSION INTERSECTION U=  '/
CG    DATA FORM5/' V= '/
CG    DATA FORM7/' ROTATION MATRIX'/
CG    DATA FORM9/
CG   1'0      U              V              A             B            L
CG   2ENGTH         MAX           MIN    '/
crmd
cmrd - Need these two data statements for IS applications
crmd96 DATA TABLE /0.05D0, 0.1D0, 0.2D0, 0.5D0, 1.D0, 2.D0, 5.D0, 10.D0/
crmd
crmd
crmd - Let the games begin.  The first section defines local functions
crmd  that are used by this sub_routine.  They represent equations from
crmd  the original Fowler and Wilson report.
C ARITHMETIC STATEMENT FUNCTIONS
c
c96      DOUBLE PRECISION DX1, DX2, DY1,DY2
crmd96  DATA ZLIT2, ZLIT3, ZLIT4, ZLIT6 /1.D10, 50.D0, 52.5001D0, 5.D-5/
       DATA ZLIT2 /1.0D10/
C
crmd96  DATA K21, K14, ZLIT1 /21, 14, .707D0/
crmd
crmd - This function returns the smaller of either the input number or
crmd   1e-9.
crmd  The reason for this function is to insure that we never get into a
crmd  numerical divide by zero situation.
crmd
      SMAL(Z1)=DSIGN(DMAX1(DABS(Z1),Z1EM9),Z1)
c
crmd - The following functions compute the slope, tangent and curvature
crmd   of the input variables.  The function assumes knowledge of the
crmd   delta X's and Y's for the interval represented by Z1 and Z2.
CRJG - THE EQUATION USED BY THIS FUNCTION IS EQUIVALENT TO EQUATION 9 OF
CRJG  THE Y-1400 "CUBIC SPLINE, A CURVE FITTING ROUTINE REPORT BY FOWLER
CRJG  AND WILSON"
c
C96  These following functions were moved into their own function
C96  statements outside of this subroutine. This subroutine can then
c96  call them.
C96   SLOP(Z1)      = (DY2+DY1*Z1) / SMAL(DX2+DX1*Z1)
C96   TAN(Z1)       = DSIN(Z1)    / SMAL(DCOS(Z1))
C96   CRVA(Z1,Z2,Z3)=-(4.*TAN(Z1)+2.*TAN(Z2))*DABS(DCOS(Z1))**3 /Z3
C
crmd -The following function computes the difference in curvature at a
crmd  given through point.  There are two measures of curvature at each
crmd  through point, the interval to the left and the interval to the
crmd  right both have curvatures that need to be compared so that they
crmd  can be checked for convergence.  It calls function CRVA from
crmd  above.
CRJG -THE FOLLOWING EQUATION IS EQUIVALENT TO EQUATION 18 OF THE Y-1400
CRJG  CUBIC SPLINE, A CURVE FITTING ROUTINE REPORT BY FOWLER AND WILSON
C
      CURV(SL1,SL2) = CRVA(A2,B2,SL2) + CRVA(B1,A1,SL1)
c
crmd -A majority  of the sub_routine is not necessary for the particual
crmd  application of the APT WF-spline representation that we have and
crmd  is therefore commented out.  The only functionality that we really
crmd  need, the data manipulation and initial data calculation necessary
crmd  to run the WF algorithm.
crmd  We decided not to move lines around, so read carefully to see what
```

```
crmd  has been commented out and what has not.  The next hundred or so
crmd  lines have executable lines intermixed.
Cwb
      PI=4.0D0*DATAN(1.0D0)
      DEGRAD=PI/180.0D0
Cwb
      L1=0
  560 DCMAX =Z0
crmd
crmd -Find the maximum value of the delta curvature at each of the 2nd to
crmd  n-1 through point intervals.
crmd
      DO 570 I1=14,NM1,7
      IF (DABS(TAB(I1+18)).LT.DCMAX) GO TO 570
      DCMAX = DABS(TAB(I1+18))
      J1=I1
  570 CONTINUE
crmd
crmd -If all the interval delta curvatures where less than 1e-3, we can
crmd  skip the curvature minimization stuff.  However, if even one
crmd  through point interval did not have continous curvatue, we must
crmd  minimize it.  If several intervals had discontinous curvatures,
crmd  minimize the interval that was the most outta whack.
crmd
C MINIMIZE MAXIMUM CURVATURE, USING NEWTON'S METHOD
cwb   IF (DCMAX.LE.Z1EM3 .OR. L1.GE.K4*I) GO TO 640
      IF (DCMAX.LE.Z1EM9 .OR. L1.GE.K4*I) GO TO 640
      A1  = TAB(J1+8)  - TAB(J1+9)
      A2  = TAB(J1+15) - TAB(J1+16)
      B2  = TAB(J1+22) - TAB(J1+16)
      DCP = -4.D0*(TAB(J1+10)+TAB(J1+17)) / TAB(J1+10)/TAB(J1+17)
cwb   DCP = -4.*(TAB(J1+10)+TAB(J1+17)) / TAB(J1+10)/TAB(J1+17)
crmd
C OBTAIN NEW APPROXIMATION FOR SLOPE AT P(J1), and a NEW CURVATURE
cwb   DO 580 I1 = 1,4
      DO 580 I1 = 1,25
        A2=DATAN(TANGENT(A2)-TAB(J1+18)/DCP)
        TAB(J1+15) = A2 +TAB(J1+16)
        B1         = TAB(J1+15) - TAB(J1+9)
        TAB(J1+18) = CURV(TAB(J1+10),TAB(J1+17))
        IF (DABS(TAB(J1+18)) .LE. Z1EM9) GO TO 590
  580 CONTINUE
crmd
crmd -Changes to the slope and curvature at the ith interval impacts the
crmd  computed slope and curvature at the i-1st and i+1st intervals,
crmd  which in turn impacts their j-1st and j+1st intervals.  In other
crmd  words, changing the slope and curvature in one interval can impact
crmd  all others.
C CHANGE IN A2 - CHANGE IN CURVATURE AT P(J1-1), P(J1+1)
C AT START OR END OF TABCYL, REFLECT ANGLE
  590 X2 = A2
      Y2 = B2
      L1 = L1 + K1
crmd
crmd -The first computed goto directs work depending on whether the
crmd  start angle has been specified (e.g., 610 => no start angle).  The
crmd  second computed goto directs flow depending on whether the end
crmd  angle has been specified.
      IF (J1+K-K15)620,630,600
  600 IF(J1+L-NM1) 610,601,630
  601 B2        = PI - A2
      TAB(I-6)  = TAB(I-12) + B2
      TAB(I-10) = CURV(TAB(I-18),TAB(I-11))
  610 A2         = A1
```

```
      B2         = B1
      A1         = TAB(J1+1) - TAB(J1+2)
      B1         = TAB(J1+8) - TAB(J1+2)
      TAB(J1+11) = CURV(TAB(J1+3),TAB(J1+10))
      GO TO 630
crmd
  620 A1      = PI - B1
      TAB(22) = TAB(23) + A1
      TAB(32) = CURV(TAB(24),TAB(31))
crmd
  630 IF (J1 .EQ. NM1) GO TO 560
      A1         = X2
      B1         = Y2
      A2         = TAB(J1+22)-TAB(J1+23)
      B2         = TAB(J1+29)-TAB(J1+23)
      TAB(J1+25) = CURV(TAB(J1+17),TAB(J1+24))
      GO TO 560
crmd
C SAVE END SLOPES AND WRITE OUT DATA --- not!!
  640 IF(L .NE. 0) GO TO 642
      A1        =TAB(I-20) -TAB(I-19)
      B1        =TAB(I-13) -TAB(I-19)
      A2        =TAB(I-13) -TAB(I-12)
      B2        = PI - A2
      TAB(I-6)  = TAB(I-12) + B2
      TAB(I-10) = CURV(TAB(I-18),TAB(I-11))
crmd
crmd -We are now done computing the WF spline wrt a series of peicewise
crmd  local cubics.  The task before us now is compute the cubic equation
crmd  that can define the spline.
crmd Since end angles exist, use them.  SB is the entry splope and SE2
crmd is the exit slope.
  642 SB    = TANGENT(TAB(22))
      SE2   = TANGENT(TAB(I-6))
crmd      ANGIN = SB
crmd      ANGOUT= SE2
crmd
C CHECK TABPRT FLAG
CG    IF(TABPRT) GO TO 643
C...      CALL PRINT TO OUTPUT ISN AND TABCYL IDENTIFICATION INFORMATION
CG    CALL PRINT(15,A,1)
C
CG    CALL CFORM(FORM1,DARRAY,1,112)
CG    CALL CPRINT(DARRAY)
crmd
crmd The following do-loop computes the polar coordinates of each set of
crmd through points.  Perform the conversion because the polar angle is
crmd used in the next set of executables
C96  The below call used to be "CALL APT0897(TEM,TAB(J1+13))"
  643 DO 672 I1=1,N
        J1 = K7 * I1
        CALL APT087(TEM(1),TAB(J1+13))
        TAB(J1+19) = TEM(2)
crmd
C CHECK TABPRT FLAG
CG        IF(TABPRT) GO TO 672
C GA IS SEGMENT ANGLE, XA EXTERIOR ANGLE
crmd        GA1 = GA
crmd        IF (I1 .EQ. N) GO TO 650
crmd        GA = TAB(J1+16) / DEGRAD
crmd        IF (I1 .EQ. K1) GO TO 660
crmd        XA = GA - GA1
crmd        IF(DABS(XA) .GT. Z2*Z90) XA = XA - DSIGN(360.0D0,XA)
CG        GO TO 670
```

```
C NO SEGMENT ANGLE FOR LAST POINT
crmd  650   GA = Z0
crmd  660   XA = Z0
CR670     CALL ICONV(I1,DARRAY,1,4)
CG        CALL FCONV(TEM(2),DARRAY,5,15,4)
CG        CALL FCONV(TEM(1),DARRAY,20,15,6)
CG        CALL FCONV(TAB(J1+13),DARRAY,35,15,6)
CG        CALL FCONV(TAB(J1+14),DARRAY,50,15,6)
CG        CALL FCONV(TAB(J1+17),DARRAY,65,15,6)
CG        CALL FCONV(GA,DARRAY,80,15,4)
CG        CALL FCONV(XA,DARRAY,95,15,4)
CG        CALL CPRINT(DARRAY)
  672 CONTINUE
crmd
crmd -Next set of instructions.  Find the maximum and minimum curvatures
C        WRITE MATCHED CURVATURES
C        CHECK TABPRT FLAG
CG    IF(TABPRT) GO TO 674
CG    CALL CFORM('0',DARRAY,1,1)
CG    CALL CFORM(FORM2,DARRAY,2,92)
CG    CALL CPRINT(DARRAY)
crmd
  674 CMIN = ZLIT2
      CMAX =-ZLIT2
      DO 690 I1=1,N
        J1 = K7*I1
        IF (I1 .EQ. N) GO TO 675
        TA     = TAB(J1+15) - TAB(J1+16)
        TB     = TAB(J1+22) - TAB(J1+16)
        TEM(2) = CRVA(TA,TB,TAB(J1+17))
        GO TO 680
  675   TA         = Z0
        TB         = Z0
        TAB(25)    = Z0
        TEM(2)     = 0.0D0
  680   TA         = TANGENT(TA)
        TB         = TANGENT(TB)
        TAB(J1+15) = TANGENT(TAB(J1+15))
        PHI        = DATAN(ZM1/SMAL(TAB(J1+15))) / DEGRAD
        AL         = PHI -TAB(J1+19)
        TAB(J1+19) = TEM(2)
C CHECK TABPRT FLAG
CG        IF(TABPRT) GO TO 685
CG        CALL ICONV(I1,DARRAY,1,4)
CG        CALL FCONV(TAB(J1+15),DARRAY,5,12,5)
CG        CALL FCONV(PHI,DARRAY,17,12,4)
CG        CALL FCONV(AL,DARRAY,29,12,4)
CG        CALL FCONV(TA,DARRAY,41,12,7)
CG        CALL FCONV(TB,DARRAY,53,12,7)
CG        CALL FCONV(TAB(J1+19),DARRAY,65,10,4)
CG        CALL FCONV(TAB(J1+18),DARRAY,75,13,4)
CG        CALL CPRINT(DARRAY)
C PLOT CURVATURES
  685   TAB(J1+15) = TA
        TAB(J1+16) = TB
        CMIN       = DMIN1(CMIN,TAB(J1+19))
  690 CMAX         = DMAX1(CMAX,TAB(J1+19))
crmd
crmd -Next set of instructions.
crmd      CEN2 = (CMAX-CMIN) / Z2
crmd      DO 700 J1=1,7
crmd        I1 = J1
crmd        IF (CEN2 .LE. TABLE(J1)) GO TO 710
crmd  700 CONTINUE
```

```
crmd      I1     = 8
crmd  710 CURVRG = TABLE(I1)
crmd      CEN1   = 50.0 / CURVRG
crmd      CENTER = CEN2 + CMIN
crmd      IDUMY  = CENTER*CEN1 + DSIGN(Z5EM1,CENTER)
crmd      CENTER = IDUMY
crmd      CENTER = CENTER / CEN1
crmd      CURTI1 = CENTER -CURVRG
crmd      CURTI2 = CENTER - Z5EM1*CURVRG
crmd      TEMP   = CENTER + Z5EM1*CURVRG
crmd      TEM(1) = CENTER + CURVRG
C CHECK TABPRT FLAG
CG      IF(TABPRT) GO TO 732
CG      CALL CFORM('0',DARRAY,1,1)
CG      CALL FCONV(CURTI1,DARRAY,17,7,4)
CG      CALL FCONV(CURTI2,DARRAY,42,7,4)
CG      CALL FCONV(CENTER,DARRAY,67,7,4)
CG      CALL FCONV(TEMP,DARRAY,92,7,4)
CG      CALL FCONV(TEM(1),DARRAY,113,7,4)
CG      CALL CPRINT(DARRAY)
C
CG      CALL CFORM(FORM3,DARRAY,1,119)
CG      CALL CPRINT(DARRAY)
crmd      KP=K7*N
CG      DO 730 I1=7,KP,7
CG      CALL CFORM('.',DARRAY,18,1)
CG      CALL CFORM('.',DARRAY,119,1)
CG      J1=ZLIT4-ZLIT3*CENTER/CURVRG
CG      J1=MIN0(MAX0(J1,2),102)+17
CG      CALL CFORM('.',DARRAY,J1,1)
CG      J1=ZLIT3*(TAB(I1+19)-CENTER)/CURVRG+ZLIT4
CG      J1=MIN0(MAX0(J1,2),102)+17
CG      CALL CFORM('*',DARRAY,J1,1)
CG      L1=I1/7
CG      CALL ICONV(L1,DARRAY,1,3)
CG      CALL FCONV(TAB(I1+19),DARRAY,4,12,6)
CG  730 CALL CPRINT(DARRAY)
CG      CALL CFORM(FORM3(17:),DARRAY,17,103)
CG      CALL CPRINT(DARRAY)
crmd
C FIT CUBICS TO GIVEN SLOPES - TRANSLATE AND ROTATE TO ELIMINATE
C CONSTANT TERM
  732 DO 771  I1 = 7,NM1,7
        TLENGT = TAB(I1+17)
        S1     = TAB(I1+15)
        T1     = TAB(I1+16)
C COMPUTE COEFFICIENTS OF CUBIC, STORE IN TAB ARRAY, A, B, and C
        TAB(I1+15) = (T1+S1) / TAB(I1+17)**2
        TAB(I1+16) = (-Z2*S1-T1) / TAB(I1+17)
        TAB(I1+17) = S1
C COMPUTE MAXIMUM AND MINIMUM VALUES ON EACH CURVE
        IF (DABS(TAB(I1+15)) .GT. Z1EM9) GO TO 750
        TAB(I1+19) = Z0
        TAB(I1+15) = Z0
        IF(DABS(TAB(I1+16)) .GT. Z1EM9) GO TO 740
C EQUATION IS LINEAR - MUST BE  Y = 0
        TAB(I1+16) = Z0
        TAB(I1+17) = Z0
        TAB(I1+18) = Z0
        GO TO 770
C EQUATION IS QUADRATIC - EXTREMUM AT  -C/2B
  740   TAB(I1+18) = -TAB(I1+17)**2 / (4.0D0*TAB(I1+16))
cwb740   TAB(I1+18) = -TAB(I1+17)**2 / (4.*TAB(I1+16))
        GO TO 760
```

```
C EQUATION IS CUBIC - SOLVE FOR FIRST DERIVATIVE ZERO
  750   TEMP = TAB(I1+16)**2
        TEM1 = 3.0D0 * TAB(I1+15) * S1
cwb     TEM3 = (TEMP-TEM1)* DSQRT(TEMP-TEM1) * 2.0
        TEM3 = (TEMP-TEM1)* DSQRT(TEMP-TEM1) * 2.0D0
        TEM2 = TAB(I1+16) * (2.0D0*TEMP-3.0D0*TEM1)
cwb     TEM2 = TAB(I1+16) * (2.*TEMP-3.*TEM1)
        TEM4 = TEM2 + DSIGN(TEM3,TEM2)
        TEM3 = S1**2 *((4.D0/3.D0)*TEM1-TEMP)
cwb     TEM3 = S1**2 * (1.3333333*TEM1-TEMP)
        TEMP = TEM4 / ((TAB(I1+15)**2)*27.0D0)
cwb     TEMP = TEM4 / ((TAB(I1+15)**2)*27.)
        IF(Z1) 760,760,755
  755   CONTINUE
        TEM1       = TEM3 / TEM4
        TAB(I1+18) = TEMP
        TAB(I1+19) = TEM1
C TEST FOR MAX GREATER THAN MIN
  760   IF(TAB(I1+18) .GE. TAB(I1+19)) GO TO 769
        TEMP       = TAB(I1+18)
        TAB(I1+18) = TAB(I1+19)
        TAB(I1+19) = TEMP
C MAX OR MIN MUST BE WITHIN INTERVAL
  769   IF(S1.LE.Z0  .AND. T1.GE.Z0) TAB(I1+18) = Z0
        IF (S1.GE.Z0 .AND. T1.LE.Z0) TAB(I1+19) = Z0
  770   TAB(I1+18) = TAB(I1+18) / TLENGT
        TAB(I1+19) = TAB(I1+19) / TLENGT
        TAB(I1+17) = TLENGT
  771 CONTINUE
crmd
C COMPUTE EXTENSION INTERVALS
C EXTENSION EQUIVALENT TO  10 INCH. REGARDLESS OF UNITS
crmd      DST   = TABEXT
crmd      DELTA = DST / DSQRT(Z1+SB**2)
crmd      IF ((TAB(21)-TAB(28))*SB+TAB(20)-TAB(27).LT.Z0) DELTA = -DELTA
crmd      TAB(13) = TAB(20) + DELTA
crmd      TAB(14) = TAB(21) + DELTA*SB
crmd      DELTA   = DST / DSQRT(Z1+SE2**2)
crmd      IF ((TAB(I-7)-TAB(I-14))*SE2+TAB(I-8)-TAB(I-15).LT.Z0)DELTA=-crmd          DELTA
crmd      TAB(I-1) = TAB(I-8) + DELTA
crmd      TAB(I)   = TAB(I-7) + DELTA*SE2
crmd      DO 780 I1=15,19
crmd       TAB(I1) = Z0
crmd       J1      = K7*N + I1
crmd  780 TAB(J1)  = Z0
C REDUCE EXTENSION IF NECESSARY
crmd      IF (DABS(SB-SE2) .LT. ZLIT6) GO TO 790
crmd      X  = (TAB(I-7)-TAB(14)+SB*TAB(13)-SE2*TAB(I-8)) / (SB-SE2)
crmd      A1 = X - TAB(13)
crmd      B1 = SB * A1
crmd      IF ( A1**2+B1**2 .GT. DST**2 ) GO TO 790
crmd      Y = B1 + TAB(14)
crmd      IF ( (X-TAB(I-8))**2+(Y-TAB(I-7))**2.GT.DST**2 ) GO TO 790
crmd      IF ((X-TAB(20))*(TAB(27)-TAB(20))
crmd     1  + (Y-TAB(21))*(TAB(28)-TAB(21)).GT.Z0) GO TO 790
crmd      IF ((X-TAB(I-8))*(TAB(I-15)-TAB(I-8))
crmd     1  + (Y-TAB(I-7))*(TAB(I-14)-TAB(I-7)).GT.Z0) GO TO 790
crmd      TAB(13)  = X-Z1EM2 * (X-TAB(20))
crmd      TAB(14)  = Y-Z1EM2 * (Y-TAB(21))
crmd      TAB(I-1) = X-Z1EM2 * (X-TAB(I-8))
crmd      TAB(I)   = Y -Z1EM2* (Y-TAB(I-7))
CG      CALL CFORM(FORM4,DARRAY,1,26)
CG      CALL FCONV(X,DARRAY,27,15,8)
CG      CALL CFORM(FORM5,DARRAY,46,3)
```

```
CG      CALL FCONV(Y,DARRAY,49,15,8)
CG      CALL CPRINT(DARRAY)
crmd 790 TAB(17)  = DSQRT((TAB(20)-TAB(13))**2 + (TAB(21)-TAB(14))**2)
crmd     TAB(I-4) = DSQRT((TAB(I)-TAB(I-7))**2 + (TAB(I-1)-TAB(I-8))**2)
C CHECK TABPRT FLAG
CG      IF(TABPRT) GO TO 796
CG      CALL CFORM(FORM7,DARRAY,1,16)
CG      CALL CPRINT(DARRAY)
crmd      L=1
CG      DO 791 I1=2,10
CG      CALL FCONV(TAB(I1),DARRAY,L,13,6)
CG  791 L=L+13
CG      CALL CPRINT(DARRAY)
CG      CALL CFORM(FORM9,DARRAY,1,100)
CG      CALL CPRINT(DARRAY)
CG      DO 9095 I1=13,I,7
CG      L=1
CG      DO 9096 J1=1,7
CG      L1=I1+J1-1
CG      IF(L1.GT.I) GO TO 9094
CG      CALL FCONV(TAB(L1),DARRAY,L,15,8)
C  9096 L=L+15
CG    9094 CALL CPRINT(DARRAY)
 9095 CONTINUE
  796 A(2)=TAB(1)
CG      CALL APT094(1,A(1),TAB(1))
      RETURN
      END
      LOGICAL FUNCTION CKDEF(ARG)
c
c**********************************************************************
c*** SOURCE FILE : CKDEF000.V01   ***
c
c
c  * CKDEF *
c
c  LOGICAL FUNCTION CKDEF
c
c  PURPOSE  TO DETERMINE THAT THE ARGUMENT IS PROPERLY DEFINED
c           THE VALUE .FALSE. IS RETURNED IF DEFINED,.TRUE. OTHERWISE
c
c
c Modified for FORTRAN 90 by Ron Dolin on 12/96.....goal was to not
c change source code or programming at all.
c**********************************************************************
C
CG    INCLUDE 'SDP.INC'
C
      INTEGER*4 ARG(2),STR,DTR,ASH,I3,I2
      LOGICAL FIRST
      SAVE STR,DTR,ASH,FIRST
C
      DATA FIRST/.TRUE./
      DATA NBCHAR /0/
C
      IF (FIRST) THEN
        I3=3*NBCHAR
        I2=2*NBCHAR
        STR=ISHFT(ICHAR('*'),I3)+ISHFT(ICHAR('T'),NBCHAR)+ICHAR('R')
        DTR=ISHFT(ICHAR('$'),I3)+ISHFT(ICHAR('T'),NBCHAR)+ICHAR('R')
        ASH=ISHFT(ICHAR('A'),I3)+ISHFT(ICHAR('S'),I2)
     +     +ISHFT(ICHAR('H'),NBCHAR)
        FIRST=.FALSE.
      ENDIF
```

```
C
      IF ((ARG(1).EQ.STR).AND.(ARG(2).EQ.ASH)) THEN
        CKDEF=.TRUE.
CG      CALL ERROR(1,'CKDEF   ')
      ELSE IF ((ARG(1).EQ.DTR).AND.(ARG(2).EQ.ASH)) THEN
        CKDEF=.TRUE.
      ELSE
        CKDEF=.FALSE.
      END IF
      RETURN
      END
      SUBROUTINE DOTF (RESULT,ARG1,ARG2)
C
C-----------------------------------------------------------------------
c*** SOURCE FILE : M0002836.V02   ***
c
C.....FORTRAN SUBROUTINE             DOTF.....             5/1/68   GK
C                       THE FIRST INPUT VECTOR
C              ARG2    ARRAY CONTAINING THE CANONICAL FORM OF
C                       THE SECOND INPUT VECTOR
C
C SUBSIDIARIES TYPE                ENTRY
C              LOGICAL FUNCTION    CKDEF
C              SUBROUTINE          ERROR
C-----------------------------------------------------------------------
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
C
      DIMENSION ARG1(3),ARG2(3),DS(6),IARG1(2),IARG2(2)
cwb   DOUBLE PRECISION DS
      LOGICAL CKDEF
C
CG    INCLUDE 'TOTAL.INC'
CG    INCLUDE 'ZNUMBR.INC'
CG    INCLUDE 'KNUMBR.INC'
C
C
C96  Need to have the input variable for CKDEF to be an integer so
C96  convert...
        IARG1(1) = ARG1(1)
        IARG1(2) = ARG1(2)
        IARG2(1) = ARG2(1)
        IARG2(2) = ARG2(2)
      IF (CKDEF(IARG1).OR.CKDEF(IARG2)) GO TO 20
C
C...      MOVE ARGUMENTS TO DOUBLE PRECISION SCRATCH LOCATIONS
C
      DO 10 I=1,3
        DS(I)  = ARG1(I)
        DS(I+3)= ARG2(I)
   10 CONTINUE
C
C...      COMPUTE DOT PRODUCT
C
      RESULT = DS(1)*DS(4) +DS(2)*DS(5) +DS(3)*DS(6)
      GO TO 9
C
C...      ISSUE DIAGNOSTIC, INPUT UNDEFINED, RESULT=0
C
CG 20 CALL ERROR (10,'DOTF   ')
   20 CONTINUE
      RESULT = 0.0D0
```

```
C
    9 RETURN
       END
       FUNCTION SMAL(Z1)
c
C96-----------------------------------------------------------------
C96  FUNCTION SMAL(Z1) - This function returns the smaller of either the
C96 input number or 1e-9.  The reason for this function is to insure
C96 that we never get into a numerical divide by zero situation.
C96
C96 Yanked out of the main software body and put into its own funciton
C96 by Ron Dolin as part of the FORTRAN 90 upgrade on 12/9/96
C96 This was not a full logic upgrade, just enough to get the program
C96 running.
C96-----------------------------------------------------------------
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        PARAMETER (Z1EM9 = 1.0D-9)
cwb     PARAMETER (Z1EM9 = 1.0E-9)
C
      SMAL = DSIGN(DMAX1(DABS(Z1),Z1EM9),Z1)
c
        RETURN
      END
      FUNCTION SLOP088(Z1,Z2,DX1,DX2,DY1,DY2)
c
C96-----------------------------------------------------------------
C96  FUNCTION SLOP088 (Z1,Z2,DX1,DX2,DY1,DY2) - This function computes
C96 the slope of the Z1 interval.  The DX and DY variables are local
C96 coord lengths on either side of the interval being evaluated.  This
C96 is the slope function that is used in subroutine APT088.
C96
C96 Created by Ron Dolin as part of the FORTRAN 90 upgrade on 12/9/96
C96 This was not a full logic upgrade, just enough to get the program
C96 running.
C96-----------------------------------------------------------------
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
       SLOP088 = (Z1*DY2+Z2*DY1) / SMAL(Z1*DX2+Z2*DX1)
C
        RETURN
       END
      FUNCTION TANGENT(Z1)
c
C96-----------------------------------------------------------------
C96  FUNCTION TANGENT (Z1) - This function computes the tangent of
C96 the Z1 interval.  The function was renamed from TAN to TANGENT
C96 because the new compiler did not like overwriting a prexisting
C96 function name.
C96
C96 Created by Ron Dolin as part of the FORTRAN 90 upgrade on 12/9/96
C96 This was not a full logic upgrade, just enough to get the program
C96 running.
C96-----------------------------------------------------------------
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
       TANGENT = DSIN(Z1) / SMAL(DCOS(Z1))
C
       RETURN
       END
       FUNCTION CRVA(Z1,Z2,Z3)
C
```

```
C96-----------------------------------------------------------------------------
C96  FUNCTION CRVA (Z1,Z2,Z3) - This function computes the curvature of
C96  and interval.
C96
C96 Created by Ron Dolin as part of the FORTRAN 90 upgrade on 12/9/96
C96 This was not a full logic upgrade, just enough to get the program
C96 running.
C96-------------------------------------------------------------------
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      CRVA= -(4.D0*TANGENT(Z1)+2.D0*TANGENT(Z2))*DABS(DCOS(Z1))**3 /Z3
cwb   CRVA = -(4.*TANGENT(Z1)+2.*TANGENT(Z2))*DABS(DCOS(Z1))**3 /Z3
C
      RETURN
      END
      SUBROUTINE GET_SPLPT(SPLINE,IKNOT,S,X,Y)
C---------------------------------------------------------------------
C     GET_SPLPT(SPLINE,IKNOT,S, X,Y) -  Computes the coordinates for an
c input spline. The variables are:
c
c INPUT:  SPLINE - Spline from which parametric definition is specified
c         IKNOT  - Knot point interval that computed point lies in.
c         S      - Parametric distance from start of interval that pt
c                    lies.
c OUTPUT: X      - X-coordinate of computed point.
c         Y      - Y-coordaínte of computed point.
c
c Written by Ron Dolin begining 7/21/92
c---------------------------------------------------------------------
c
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      IMPLICIT INTEGER*4 (I-N)
      PARAMETER (NCOEF=13,  MXKNOTS=201)
c
cwb
      COMMON /WBSPNRLT/ XOUT,YOUT,DXTNVT,DYTNVT,CRDLNG
      DOUBLE PRECISION  XOUT,YOUT,DXTNVT,DYTNVT,CRDLNG
c
      COMMON /WBSPNFSD/ DX,DY,DDX,DDY
      DOUBLE PRECISION  DX,DY,DDX,DDY
cwb
      DIMENSION SPLINE(NCOEF,MXKNOTS)
C
C Compute the coordinates of a point in the IKNOTth interval.
C
      X = SPLINE(2,IKNOT) + S*SPLINE(3,IKNOT)
     &                   + S*S*SPLINE(4,IKNOT)
     &                   + S*S*S*SPLINE(5,IKNOT)
C
      Y = SPLINE(6,IKNOT) + S*SPLINE(7,IKNOT)
     &                   + S*S*SPLINE(8,IKNOT)
     &                   + S*S*S*SPLINE(9,IKNOT)
CWB
      DX = SPLINE(3,IKNOT)
     &                   +2.0D0*S*SPLINE(4,IKNOT)
     &                   +3.0D0*S*S*SPLINE(5,IKNOT)
C
      DY = SPLINE(7,IKNOT)
     &                   +2.0D0*S*SPLINE(8,IKNOT)
     &                   +3.0D0*S*S*SPLINE(9,IKNOT)
C
      DDX= 2.0D0*SPLINE(4,IKNOT)+6.0D0*S*SPLINE(5,IKNOT)
C
```

```
      DDY= 2.0D0*SPLINE(8,IKNOT)+6.0D0*S*SPLINE(9,IKNOT)
CWB
      SGMLNG=SPLINE(1,IKNOT)+S
c
      XOUT=X
      YOUT=Y
      DXTNVT=DX/DSQRT(DX**2+DY**2)
      DYTNVT=DY/DSQRT(DX**2+DY**2)
      CRDLNG=SGMLNG
c
C We be done
c
      RETURN
      END
      SUBROUTINE HORNERS(A,OLDS, S)
c
c---------------------------------------------------------------------
c      HORNERS(A,OLDS, S)
c  Computes the roots of a cubic equation given an initial value for the
c  cubic's parameter (OLDS).  The root will be returned in the variable
c  'S'.  Horner's method can be used to solve for the roots of nth
c  ordered polynomials but it is coded here for strictly cubics.
c  Horner's method is a variation of Newton's method.
c
c  The equation for the cubic is:
c                       P(S) = A(1)*S**3 + A(2)*S**2 + A(3)*S + A(4)
c  The equation for the derivative of the cubic is:
c                      P'(S) = 3*A(1)**2 + 2*A(2)*S + A(3)
c  For an initial parameter (OLDS) the root of the cubic is found by:
c                          s = OLDS - [P(olds) / P'(olds)]
c
c  Written by Ron Dolin begining on 8/20/92
c---------------------------------------------------------------------
c
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION A(4)
c
c  The cubic equation to be solved is of the form:
c            P  = A(1)*S**3 + A(2)*S**2 + A(3)*S + A(4)
c  The derivative of the cubic equation is:
c            DP = 3*A(1)**2 + 2*A(2)*S + A(3)
c  The first time we compute a root may not render the best value.  We
c  will therefore iterate potentially ten times.  Since this method is a
c  variation of Newton's method, it follows the Newton method's for
c  convergence.
      S  = OLDS
      DS = 0.0D0
      DO 100 I = 1,50
cwb   DO 100 I = 1,10
        P  = ( ( A(1)*S + A(2) )*S + A(3) )*S + A(4)
        DP = ( 3.0D0*A(1)*S + 2.0D0*A(2) )*S + A(3)
        IF (DP .EQ. 0.0D0) RETURN
        S    = OLDS - (P / DP)
        DS   = DABS(S - OLDS)
        OLDS = S
        IF (DS .LT. 1.0D-9) RETURN
cwb     IF (DS .LT. 1E-7) RETURN
  100 CONTINUE
c
      RETURN
      END
      SUBROUTINE APT087 (DRESULT,RECT)
C
```

```
C-------------------------------------------------------------------
C.....FORTRAN SUBROUTINE             APT087...              5/1/68   GK
C
C              FORTRAN SUBROUTINE APT087
C
C PURPOSE      TO GENERATE THE POLAR COORDINATES OF A GIVEN
C              POINT.
C
C LINKAGE      CALL APT087 (RESULT, RECT)
C
C ARGUMENTS    RESULT (1)          DISTANCE FROM ORIGIN TO INPUT POINT
C              RESULT (2)          ANGLE IN DEGREES BETWEEN INPUT POINT
C                                  AND POSITIVE X-AXIS
C              RECT                ARRAY CONTAINING THE CANONICAL FORM
C                                  OF INPUT POINT IN RECTANGULAR
C                                  COORDINATES
C
C This subroutine is so short and focused on the simple task of
c computing
c the polar coordinates of a rectangular coordinate system that the
c original APT sub_routine has been dramatically edited.  The executable
c statements have not been altered but the unnecessary bagage has been
c removed..... By order of Ron Dolin on 10/15/92
C-------------------------------------------------------------------
C
c
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
cwb   PARAMETER (DEGRAD = 0.017453293)
c
      DIMENSION DRESULT(2), RECT(2)
      DIMENSION SC(2)
C
C CHANGE RECTANGULAR COORDINATES TO POLAR -  DRESULT(1) = RAD,
C                                            DRESULT(2) = ANGLE
Cwb
      PI=4.0D0*DATAN(1.0D0)
      DEGRAD=PI/180.0D0
Cwb
      SC(1) = RECT(1)
      SC(2) = RECT(2)
      Z0    = 0.0D0
crmd
crmd      UNFLAG = CKDEF(SC)
crmd
      DRESULT(1) = DSQRT(SC(1)**2 + SC(2)**2)
      DRESULT(2) = Z0
C
      IF (DRESULT(1) .NE. Z0) DRESULT(2) = DATAN2(SC(2),SC(1))/DEGRAD
C
      RETURN
      END
      FUNCTION F5ATAN(AY,AX)
c
c -------------------------------------------------------------------
c     F5ATAN(AY,AX)
c Computes the arctangent of AY / AX. The function takes care of
c infinite
c arctangent problems.  It also takes care of improper quadrant problems
c that can arise when using various compilers.  The value of the
c arctangent is returned in degress.  This is a double precision verion
c of the original IDEAL library function.
c
c Written by Rob Oakes or Dwight Jaeger in the 1980's
```

```
c Modified for SE module by Ron Dolin begining on 8/25/92
c ------------------------------------------------------------------
c
c
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT INTEGER*4 (I-N)
cwb
      PI=4.0D0*DATAN(1.0D0)
      RADDEG=180.0D0/PI
cwb
C
cwb   ENTRY ATANDL(AY,AX)
      X =DABS(AX)
      Y =DABS(AY)
cwb   X = ABS(AX)
cwd   Y = ABS(AY)
      IF(Y .GT. X) GO TO 1050
      IF(X. EQ. 0.0D0) THEN
        F5ATAN = 45.0D0
        RETURN
      END IF
c
cwb
      F5ATAN =RADDEG*DATAN(Y/X)
      GO TO 1075
 1050 F5ATAN = 90.0D0 - RADDEG*DATAN(X/Y)
c
cwb   F5ATAN = 57.2957795131D0 * ATAN(Y/X)
cwb   GO TO 1075
c1050 F5ATAN = 90.D0 - 57.2957795131D0*ATAN(X/Y)
c
 1075 IF(AX .LT. 0.0D0) F5ATAN = 180.0D0 - F5ATAN
      IF(AY .LT. 0.0D0) F5ATAN = 360.0D0 - F5ATAN
c
c We're just a couple of happy campers lost in the proper quadrant
      RETURN
      END
```

Appendix B—PCS Routines

```
      subroutine NRFCS (npt1,angin,angout,iaflg,wfspl)
c
c       Wilbur D. Birchler, Ph.D.
c       Engineering Analysis
c       Los Alamos National Laboratory
c       Los Alamos, New Mexico
c       (505) 667-9361
c
      implicit none
c
c     Begin WILSON-FOWLER Information Block
c
      PARAMETER (ncoef=13, mxknots=201, mxpts=1000, ndim=3)
c
      double precision wfspl
      dimension wfspl(ncoef,mxknots)
c
c     E n d WILSON-FOWLER Information Block
c
      integer*4 i, j, npt1, iaflg, ncoef, mxknots, mxpts, ndim, one
c
      double precision t, x, y, dxt2, dyt2, angin, angout
      double precision anginx, angoutx, anginy, angouty, pi
      dimension t(mxpts), x(mxpts), y(mxpts), dxt2(mxpts), dyt2(mxpts)
c
      pi=4.0d0*datan(1.0d0)
      one=1
c
      x(1)=wfspl(2,1)
      y(1)=wfspl(6,1)
      t(1)=0.0d0
      wfspl(1,1)=0.0d0
c
      do 10 i=2,npt1
      x(i)=wfspl(2,i)
      y(i)=wfspl(6,i)
      t(i)=t(i-1)+dsqrt((x(i)-x(i-1))**2+(y(i)-y(i-1))**2)
      wfspl(1,i)=t(i)
   10 continue
c
c     values of end slopes
c
      anginx=dcos(angin)
      anginy=dsin(angin)
      angoutx=dcos(angout)
      angouty=dsin(angout)
c
      call spline (t,x,npt1,one,anginx,one,angoutx,dxt2)
      call spline (t,y,npt1,one,anginy,one,angouty,dyt2)
c
      do 20 i=1,npt1-1
      wfspl(5,i)=(dxt2(i+1)-dxt2(i))/(6.0d0*(t(i+1)-t(i)))
      wfspl(9,i)=(dyt2(i+1)-dyt2(i))/(6.0d0*(t(i+1)-t(i)))
      wfspl(4,i)=dxt2(i)/2.0d0
      wfspl(8,i)=dyt2(i)/2.0d0
      wfspl(3,i)=(x(i+1)-x(i))/(t(i+1)-t(i))-(t(i+1)-t(i))*(dxt2(i)/3.d0
     1 +dxt2(i+1)/6.0d0)
      wfspl(7,i)=(y(i+1)-y(i))/(t(i+1)-t(i))-(t(i+1)-t(i))*(dyt2(i)/3.d0
     1 +dyt2(i+1)/6.0d0)
   20 continue
c
```

```
      do 30 i=3,5
      wfspl(i,npt1)=0.0d0
      wfspl(i+4,npt1)=0.0d0
   30 continue
c
      return
      end
      subroutine spline (x,y,n,iflg1,yp1,iflgn,ypn,y2)
c
c       Wilbur D. Birchler, Ph.D.
c       Engineering Analysis
c       Los Alamos National Laboratory
c       Los Alamos, New Mexico
c       (505) 667-9361
c
      implicit none
c
      integer*4 n, iflg1, iflgn, ncoef, mxknots, mxpts, ndim
      PARAMETER (ncoef=13, mxknots=201, mxpts=1000, ndim=3)
      double precision yp1, ypn, x, y, y2, u
      dimension x(mxpts), y(mxpts), y2(mxpts), u(mxpts)
c
c     iflg1 = end condition for end 1
c           = 0 - natural boundary
c           = 1 - specified end angle
c     iflgn = end condition for end 1
c           = 0 - natural boundary
c           = 1 - specified end angle
c
      integer*4 i, k
      double precision p, qn, sig, un
c
      if (iflg1.eq.0) then
        y2(1)=0.0d0
        u(1)=0.0d0
      else
        y2(1)=-0.5d0
        u(1)=(3.0d0/(x(2)-x(1)))*((y(2)-y(1))/(x(2)-x(1))-yp1)
      endif
c
      do 10 i=2,n-1
      sig=(x(i)-x(i-1))/(x(i+1)-x(i-1))
      p=sig*y2(i-1)+2.0d0
      y2(i)=(sig-1.0d0)/p
      u(i)=(6.0d0*((y(i+1)-y(i))/(x(i+1)-x(i))-(y(i)-y(i-1))/(x(i)-x(i-1
     1 )))/(x(i+1)-x(i-1))-sig*u(i-1))/p
   10 continue
c
      if (iflgn.eq.0) then
        qn=0.0d0
        un=0.0d0
      else
        qn=0.5d0
        un=(3.0d0/(x(n)-x(n-1)))*(ypn-(y(n)-y(n-1))/(x(n)-x(n-1)))
      endif
c
      y2(n)=(un-qn*u(n-1))/(qn*y2(n-1)+1.0d0)
c
      do 20 k=n-1,1,-1
      y2(k)=y2(k)*y2(k+1)+u(k)
   20 continue
c
      return
      end
```

Appendix C—Minimum-Distance Routines

```
      subroutine analyze_datawb (bl_spl,nblpts,spltyp,pt_data,ndpts
     1 ,pt_data1)
c
c       Wilbur D. Birchler, Ph.D.
c       Engineering Analysis
c       Los Alamos National Laboratory
c       Los Alamos, New Mexico
c       (505) 667-9361
c
      implicit double precision (a-h,o-z)
      implicit integer*4 (i-n)
c
      parameter (ncoef=13, mxknots=201, mxpts=1000, ndim=3)
      parameter (tolz=1.0d-20, tols=1.0d-10)
c
      character*10 spltyp
      dimension bl_spl(ncoef,mxknots), pt_data(3,mxpts)
      dimension sbgnd(2,2*mxknots), jbgnd(2*mxknots)
c
      common /wbspnrlt/ xout, yout, dxtnvt, dytnvt, crdlng
      double precision xout, yout, dxtnvt, dytnvt, crdlng
c
      dimension pt_data1(5,mxpts)
c
c     Build Subsegments - Inflection Points
c
      nsbsgm=0
c
      do 10 i=1,nblpts-1
      nsbsgm=nsbsgm+1
      sbgnd(1,nsbsgm)=0.0d0
      jbgnd(nsbsgm)=i
      a=6.0d0*(bl_spl(4,i)*bl_spl(9,i)-bl_spl(8,i)*bl_spl(5,i))
      b=6.0d0*(bl_spl(3,i)*bl_spl(9,i)-bl_spl(7,i)*bl_spl(5,i))
      c=2.0d0*(bl_spl(3,i)*bl_spl(8,i)-bl_spl(7,i)*bl_spl(4,i))
c
      sl=0.0d0
      sh=bl_spl(1,i+1)-bl_spl(1,i)
      call gdsqrt (a,b,c,sl,sh,n,s1,s2)
c
      if (n.eq.0) then
        sbgnd(2,nsbsgm)=bl_spl(1,i+1)-bl_spl(1,i)
        go to 10
      endif
c
      if (n.eq.1) then
        sbgnd(2,nsbsgm)=s1
        nsbsgm=nsbsgm+1
        jbgnd(nsbsgm)=i
        sbgnd(1,nsbsgm)=s1
        sbgnd(2,nsbsgm)=bl_spl(1,i+1)-bl_spl(1,i)
        go to 10
      endif
c
      if (n.eq.2) then
        sbgnd(2,nsbsgm)=s1
        nsbsgm=nsbsgm+1
        jbgnd(nsbsgm)=i
        sbgnd(1,nsbsgm)=s1
        sbgnd(2,nsbsgm)=s2
        nsbsgm=nsbsgm+1
```

```
          jbgnd(nsbsgm)=i
          sbgnd(1,nsbsgm)=s2
          sbgnd(2,nsbsgm)=bl_spl(1,i+1)-bl_spl(1,i)
          go to 10
        endif
c
   10 continue
c
        do 70 i=1,ndpts
c
        x1=pt_data(1,i)
        y1=pt_data(2,i)
c
        jseg=0
        dmin=1.0d30
        smin=1.0d30
c
        do 20 j=1,nsbsgm
        indx=jbgnd(j)
        ss=sbgnd(1,j)
        call GET_SPLPT (bl_spl,indx,ss,xs,ys)
        dist=(x1-xs)**2+(y1-ys)**2
        if (dist.lt.dmin) then
          dmin=dist
          jseg=j
          smin=ss
c
          if (dmin.lt.tolz) then
            indx=jbgnd(jseg)
            go to 50
          endif
c
        endif
        ss=(sbgnd(1,j)+sbgnd(2,j))*0.1d0
        call GET_SPLPT (bl_spl,indx,ss,xs,ys)
        dist=(x1-xs)**2+(y1-ys)**2
        if (dist.lt.dmin) then
          dmin=dist
          jseg=j
          smin=ss
c
          if (dmin.lt.tolz) then
            indx=jbgnd(jseg)
            go to 50
          endif
c
        endif
        ss=(sbgnd(1,j)+sbgnd(2,j))*0.5d0
        call GET_SPLPT (bl_spl,indx,ss,xs,ys)
        dist=(x1-xs)**2+(y1-ys)**2
        if (dist.lt.dmin) then
          dmin=dist
          jseg=j
          smin=ss
c
          if (dmin.lt.tolz) then
            indx=jbgnd(jseg)
            go to 50
          endif
c
        endif
        ss=(sbgnd(1,j)+sbgnd(2,j))*0.9d0
        call GET_SPLPT (bl_spl,indx,ss,xs,ys)
        dist=(x1-xs)**2+(y1-ys)**2
```

```
      if (dist.lt.dmin) then
        dmin=dist
        jseg=j
        smin=ss
c
        if (dmin.lt.tolz) then
          indx=jbgnd(jseg)
          go to 50
        endif
c
      endif
      ss=sbgnd(2,j)
      call GET_SPLPT (bl_spl,indx,ss,xs,ys)
      dist=(x1-xs)**2+(y1-ys)**2
      if (dist.lt.dmin) then
        dmin=dist
        jseg=j
        smin=ss
c
        if (dmin.lt.tolz) then
          indx=jbgnd(jseg)
          go to 50
        endif
c
      endif
c
   20 continue
c
   30 continue
c
      if (jseg.gt.nsbsgm) then
        jseg=nsbsgm-1
        ss=sbgnd(2,jseg)
      endif
      if (jseg.le.0) then
        jseg=1
        ss=sbgnd(1,jseg)
      endif
c
      indx=jbgnd(jseg)
c
   40 continue
      call GET_SPLPT (bl_spl,indx,ss,xs,ys)
      ds=(x1-xs)*dxtnvt+(y1-ys)*dytnvt
      ss=ss+ds
      if (dabs(ds).lt.tols) go to 50
      go to 40
c
   50 continue
c
      if (jseg.eq.1.and.ss.lt.0.0d0) then
        write (6,80)
        write (9,80)
        write (6,90) i,x1,y1
        write (9,90) i,x1,y1
        ss=0.0d0
        indx=1
        go to 60
      endif
c
      if (jseg.eq.nsbsgm.and.ss.gt.sbgnd(2,nsbsgm)) then
        write (6,80)
        write (9,80)
        write (6,90) i,x1,y1
```

```
        write (9,90) i,x1,y1
        ss=sbgnd(2,nsbsgm)
        indx=nblpts-1
        go to 60
      endif
c
      if (ss.lt.sbgnd(1,jseg)) then
        ss=sbgnd(1,jseg)+ss
        jseg=jseg-1
        go to 30
      endif
c
      if (ss.gt.sbgnd(2,jseg)) then
        ss=ss-sbgnd(2,jseg)
        jseg=jseg+1
        go to 30
      endif
c
   60 continue
c
      call GET_SPLPT (bl_spl,indx,ss,xs,ys)
c
      pt_data1(1,i)=xout
      pt_data1(2,i)=yout
      pt_data1(3,i)=dxtnvt
      pt_data1(4,i)=dytnvt
      pt_data1(5,i)=crdlng
c
   70 continue
c
      return
c
   80 format ('*,')
   90 format ('Warning...Point ',i3,' is off Spline ',' x= ',f12.6,' y=
     1',f12.6,'.........PLTSPLN')
      end
      subroutine gdsqrt (a,b,c,sl,sh,n,s1,s2)
c
c       Wilbur D. Birchler, Ph.D.
c       Engineering Analysis
c       Los Alamos National Laboratory
c       Los Alamos, New Mexico
c       (505) 667-9361
c
      implicit none
c
      real*8 a, b, c, s1, s2, s3, rdl, tolz, sl, sh
      integer*4 n
c
      data tolz /1.0d-30/
c
      if (dabs(a).lt.tolz) then
        if (dabs(b).lt.tolz) then
          s1=0.0d0
          s2=0.0d0
          n=0
          return
        else
          s1=-c/b
          n=1
          s2=0.0d0
          if (s1.le.sl) then
            s1=0.0d0
            n=0
```

```
          return
        endif
        if (s1.ge.sh) then
          s1=0.0d0
          n=0
          return
        endif
        return
      endif
    else
      rdl=b**2-4.0d0*a*c
      if (rdl.eq.0.0d0) then
        s1=-b/(2.0d0*a)
        s2=0.0d0
        n=1
        if (s1.le.sl) then
          s1=0.0d0
          n=0
          return
        endif
        if (s1.ge.sh) then
          s1=0.0d0
          n=0
          return
        endif
        return
      endif
      if (rdl.lt.0.0d0) then
        s1=0.0d0
        s2=0.0d0
        n=0
        return
      endif
      if (rdl.gt.0.0d0) then
        s1=(-b-dsqrt(rdl))/(2.0d0*a)
        s2=(-b+dsqrt(rdl))/(2.0d0*a)
        n=2
        s3=s1
        if (s1.gt.s2) then
          s1=s2
          s2=s3
        endif
        if (s1.le.sl) then
          s1=0.0d0
          n=n-1
          go to 10
        endif
        if (s1.ge.sh) then
          s1=0.0d0
          n=n-1
          go to 10
        endif
10      continue
        if (s2.le.sl) then
          s2=0.0d0
          n=n-1
          go to 20
        endif
        if (s2.ge.sh) then
          s2=0.0d0
          n=n-1
          go to 20
        endif
20      continue
```

```
          if (n.eq.1.and.s1.eq.0.0d0) then
            s1=s2
            s2=0.0d0
          endif
          return
        endif
      endif
c
      return
      end
```

Appendix D—Analytical Spline-Point Data

The three analytical spline data-point files are listed in this appendix.

Circle Spline-Point File

```
!
!    WF Start Angle (Deg)  ..  90.00
!    WF End Angle (Deg)  .... 180.00
!
!    Number of Points ... 46
!
          0         100         100           0
          2         100   99.939083     3.48995
          4         100   99.756405    6.975647
          6         100    99.45219   10.452846
          8         100   99.026807    13.91731
         10         100   98.480775   17.364818
         12         100    97.81476   20.791169
         14         100   97.029573    24.19219
         16         100    96.12617   27.563736
         18         100   95.105652   30.901699
         20         100   93.969262   34.202014
         22         100   92.718385   37.460659
         24         100   91.354546   40.673664
         26         100   89.879405   43.837115
         28         100   88.294759   46.947156
         30         100    86.60254          50
         32         100    84.80481   52.991926
         34         100   82.903757    55.91929
         36         100   80.901699   58.778525
         38         100   78.801075   61.566148
         40         100   76.604444   64.278761
         42         100   74.314483   66.913061
         44         100    71.93398   69.465837
         46         100   69.465837    71.93398
         48         100   66.913061   74.314483
         50         100   64.278761   76.604444
         52         100   61.566148   78.801075
         54         100   58.778525   80.901699
         56         100    55.91929   82.903757
         58         100   52.991926    84.80481
         60         100          50    86.60254
         62         100   46.947156   88.294759
         64         100   43.837115   89.879405
         66         100   40.673664   91.354546
         68         100   37.460659   92.718385
         70         100   34.202014   93.969262
         72         100   30.901699   95.105652
         74         100   27.563736    96.12617
         76         100    24.19219   97.029573
         78         100   20.791169    97.81476
         80         100   17.364818   98.480775
         82         100    13.91731   99.026807
         84         100   10.452846    99.45219
         86         100    6.975647   99.756405
         88         100     3.48995   99.939083
         90         100           0         100
```

Ellipse Spline-Point File

```
!
!    WF Start Angle (Deg)  ..  90.00
!    WF End Angle (Deg)  .... 180.00
!
!    Number of Points ... 46
!
          0         100         100           0
          2   99.965762   99.904866    3.488755
          4   99.863425   99.620163     6.96612
          6    99.69411   99.147975   10.420872
          8   99.459654   98.491719   13.842108
         10   99.162566   97.656063   17.219399
         12   98.805964   96.646817   20.542915
         14   98.393509   95.470801   23.803544
         16   97.929315     94.1357   26.992978
         18    97.41787     92.6499   30.103777
         20   96.863943   91.022332    33.12942
         22   96.272494   89.262302   36.064311
         24   95.648594   87.379339   38.903788
         26   94.997344   85.383047   41.644094
         28     94.3238   83.282972   44.282342
         30   93.632918   81.088485   46.816459
         32   92.929494   78.808681   49.245129
         34   92.218128   76.452293   51.567723
         36   91.503182    74.02763   53.784221
         38   90.788764   71.542523   55.895145
         40   90.078706   69.004292   57.901476
         42   89.376556   66.419725   59.804589
         44   88.685578   63.795066   61.606179
         46   88.008752   61.136017   63.308198
         48   87.348784   58.447744   64.912796
         50   86.708111   55.734899   66.422266
         52   86.088921   53.001632   67.838995
         54   85.493164   50.251621   69.165423
         56   84.922571   47.488099   70.404002
         58   84.378666    44.71388   71.557167
         60   83.862787   41.931393   72.627304
         62   83.376103    39.14271    73.61673
         64   82.919629   36.349573   74.527669
         66   82.494241   33.553431   75.362239
         68   82.100691    30.75546   76.122435
         70   81.739621   27.956597   76.810119
         72   81.411578   25.157561   77.427012
         74   81.117021   22.358881   77.974685
         76   80.856333   19.560917   78.454555
         78   80.629833   16.763885   78.867878
         80   80.437781   13.967874    79.21575
         82   80.280383    11.17287     79.4991
         84   80.157803    8.378772    79.71869
         86   80.070162    5.585412   79.875115
         88   80.017545    2.792572     79.9688
         90          80           0          80
```

Parabola Spline-Point File

```
!
!    WF Start Angle (Deg)  ..  90.0000
!    WF End Angle (Deg)  .... 163.3008
!
!    Number of Points ... 46
!
          0         100         100           0
```

2	99.724286	99.663537	3.480327
4	98.918384	98.677424	6.900198
6	97.64132	97.106431	10.206297
8	95.977836	95.043786	13.357533
10	94.02365	92.59522	16.327036
12	91.872561	89.864925	19.101379
14	89.607846	86.94611	21.6781
16	87.298161	83.916379	24.062634
18	84.996849	80.836807	26.265471
20	82.743289	77.753258	28.299871
22	80.565148	74.698705	30.180236
24	78.480811	71.695788	31.921022
26	76.501597	68.75918	33.536093
28	74.633627	65.897582	35.038366
30	72.879303	63.115328	36.439651
32	71.238442	60.413625	37.750623
34	69.709126	57.791485	38.980849
36	68.288327	55.246417	40.138871
38	66.972353	52.774934	41.232298
40	65.757172	50.372917	42.267896
42	64.638637	48.035869	43.25169
44	63.612639	45.759103	44.189052
46	62.675213	43.537862	45.084775
48	61.822613	41.367403	45.943155
50	61.051349	39.243051	46.768047
52	60.358217	37.160229	47.562924
54	59.740316	35.114476	48.330931
56	59.195049	33.101451	49.07492
58	58.720131	31.116929	49.797496
60	58.313583	29.156792	50.501045
62	57.97373	27.217018	51.187766
64	57.699199	25.293664	51.859696
66	57.488912	23.382847	52.518735
68	57.342091	21.480726	53.166661
70	57.258252	19.583475	53.805157
72	57.237206	17.687269	54.435818
74	57.27907	15.788251	55.060176
76	57.384264	13.88251	55.679706
78	57.553528	11.966051	56.295845
80	57.78793	10.034769	56.910002
82	58.088888	8.084411	57.523571
84	58.458183	6.110544	58.137943
86	58.897991	4.108516	58.754518
88	59.41091	2.073411	59.374719
90	60	0	60

Appendix E—Nonanalytical Spline-Point Data

The three nonanalytical spline data-point files are listed in this appendix.

Ellipse Spline-Point File

```
!
!   CAD Start slope (Deg) ..  -90.0000
!   CAD End slope (Deg) ....   -0.0000
!
!   WF Start Angle (Deg)  ..   90.0000
!   WF End Angle (Deg)   ...  180.0000
!
!   Number of Points ... 46
!
     0.0000  93.500000  93.500000   0.000000
     2.0000  93.198700  93.141926   3.252588
     4.0000  92.313400  92.088529   6.439457
     6.0000  90.896800  90.398858   9.501303
     8.0000  89.027700  88.161289  12.390261
    10.0000  86.799900  85.481214  15.072644
    12.0000  84.311000  82.468602  17.529243
    14.0000  81.653700  79.228236  19.753818
    16.0000  78.909500  75.852680  21.750406
    18.0000  76.145600  72.418769  23.530284
    20.0000  73.414900  68.987440  25.109375
    22.0000  70.756400  65.604192  26.505814
    24.0000  68.197700  62.301699  27.738504
    26.0000  65.756500  59.101551  28.825752
    28.0000  63.443100  56.016932  29.784731
    30.0000  61.262500  53.054881  30.631250
    32.0000  59.215200  50.217338  31.379275
    34.0000  57.299100  47.503107  32.041250
    36.0000  55.510300  44.908776  32.628136
    38.0000  53.843400  42.429178  33.149307
    40.0000  52.292500  40.058379  33.612971
    42.0000  50.851400  37.789955  34.026228
    44.0000  49.513900  35.617319  34.395245
    46.0000  48.273700  33.533730  34.725194
    48.0000  47.124900  31.532713  35.020626
    50.0000  46.062100  29.608147  35.285616
    52.0000  45.079800  27.753896  35.523367
    54.0000  44.173200  25.964356  35.736869
    56.0000  43.337900  24.234246  35.928747
    58.0000  42.569600  22.558451  36.101068
    60.0000  41.864600  20.932300  36.255807
    62.0000  41.219400  19.351336  36.394570
    64.0000  40.630900  17.811414  36.518811
    66.0000  40.096300  16.308634  36.629793
    68.0000  39.613000  14.839291  36.728534
    70.0000  39.178800  13.399939  36.816029
    72.0000  38.791600  11.987264  36.893004
    74.0000  38.449700  10.598174  36.960224
    76.0000  38.151600   9.229707  37.018334
    78.0000  37.895800   7.878980  37.067686
    80.0000  37.681300   6.543289  37.108836
```

```
    82.0000  37.507000   5.219965  37.141984
    84.0000  37.372200   3.906459  37.167471
    86.0000  37.276400   2.600270  37.185597
    88.0000  37.219100   1.298928  37.196427
    90.0000  37.200000   0.000000  37.200000
```

Lampshade Spline-Point File

```
!
!   CAD Start slope (Deg) ..   -80.7233
!   CAD End slope (Deg) ....    -8.4766
!
!   WF Start Angle (Deg)  ..   180.0000 - 80.7233 =  99.2767
!   WF End Angle (Deg)  ....   180.0000 -  8.4766 = 171.5234
!
!    Number of Point ... 28
!
     0.0000  56.700000  56.700000   0.000000
     3.5700  56.196049  56.087000   3.499200
     6.5780  55.882886  55.515000   6.401700
     9.9364  55.677563  54.842400   9.607400
    12.9638  55.650525  54.232100  12.484400
    16.4533  55.851777  53.564700  15.819100
    19.4683  56.250573  53.034500  18.747500
    22.2533  56.694383  52.471700  21.470300
    24.0755  56.822700  51.879600  23.180300
    25.6628  56.598902  51.015900  24.511500
    27.0216  55.956783  49.848300  25.422600
    27.9623  55.115696  48.681300  25.843200
    28.9510  53.848536  47.119300  26.066000
    30.0986  52.337354  45.280400  26.246600
    31.6816  50.765148  43.200100  26.661800
    33.2417  49.620034  41.500500  27.200300
    34.8287  48.667661  39.949500  27.795300
    40.1855  46.174994  35.275800  29.795100
    45.9210  44.140762  30.706500  31.709900
    51.1727  42.567917  26.689000  33.162100
    57.5074  40.972975  22.010300  34.559100
    64.4569  39.813978  17.167400  35.922600
    70.8328  39.650009  13.018100  37.452000
    76.2420  40.473634   9.625500  39.312400
    80.0399  41.727403   7.217300  41.098500
    83.3522  43.121427   4.992000  42.831500
    86.6486  44.340431   2.592100  44.264600
    90.0000  45.000000   0.000000  45.000000
```

Weird Shape Spline-Point File

```
!
!   CAD Start slope (Deg) ..    -5.3125
!   CAD End slope (Deg) ....    -2.4783
!
!   WF Start Angle (Deg)  ..   180.0000 - 5.3125 = 174.6875
!   WF End Angle (Deg)  ....   180.0000 - 2.4783 = 177.5217
!
!   Number of Points ... 24
!
    10.6325 107.297800 105.455588  19.797396
```

11.4126	102.431700	100.406388	20.268458
12.3071	97.399100	95.160788	20.760760
13.5209	91.686100	89.144992	21.436214
14.9906	85.836300	82.915143	22.202466
16.2770	81.302000	78.043244	22.787438
17.7706	76.781800	73.118243	23.434320
20.5084	70.726300	66.243726	24.778584
24.3245	65.636600	59.809857	27.035981
29.3600	62.194400	54.205922	30.493629
34.1821	61.005100	50.466843	34.274188
39.6041	61.441800	47.338918	39.167865
44.1534	62.945900	45.162260	43.846968
48.5757	65.118700	43.084482	48.827989
53.8197	68.143900	40.227264	55.003257
58.6382	70.973000	36.937219	60.603701
63.7764	73.585800	32.515754	66.012087
67.6314	74.981100	28.535080	69.339127
72.0364	75.792000	23.375217	72.097340
77.7110	75.784000	16.130079	74.047520
83.0883	75.288700	9.060209	74.741561
87.9188	75.014500	2.724212	74.965018
91.2798	75.133600	-1.678099	75.114858
94.7630	75.560400	-6.274106	75.299466

Appendix F—Keyword Graphics Builder Program—Command Files

The three Keyword Graphics Builder Program command files, which were used to calculate and plot the deviation results, are listed in this appendix.

Analytical Shapes—Accuracy Study

```
*, Analytical Shapes - Offset Study
*,
*, Circle Data
splndata,Circle_n.spn,132,132,Circle_n_spn,3,4,d,90.,d,180.
proedata,Circle_n.pts,132,132,Circle_n_pts,3,4
proedata,Circle_p.pts,132,132,Circle_p_pts,3,4
proedata,Circle_m.pts,132,132,Circle_m_pts,3,4
*, Ellipse Data
splndata,Ellipse_n.spn,132,132,Ellipse_n_spn,3,4,d,90.,d,180.
proedata,Ellipse_n.pts,132,132,Ellipse_n_pts,3,4
proedata,Ellipse_p.pts,132,132,Ellipse_p_pts,3,4
proedata,Ellipse_m.pts,132,132,Ellipse_m_pts,3,4
*, Parabola Data
splndata,Parabola_n.spn,132,132,Parabola_n_spn,3,4,d,90.,d,163.3008
proedata,Parabola_n.pts,132,132,Parabola_n_pts,3,4
proedata,Parabola_p.pts,132,132,Parabola_p_pts,3,4
proedata,Parabola_m.pts,132,132,Parabola_m_pts,3,4
*,
on,tek,p,.9
on,grid
on,geom
*,
*, Open plot file
open,Circle.wf.ps,p
*,
*, Plot Analytical shapes - Spline Data
plot,,,Circle_n.spn,Ellipse_n.spn,Parabola_n.spn,
*,
*, Circle - Calculate and plot deviation results - Wilson-Fowler
pltptwf,Circle_n_wf,Circle_n.spn,Circle_n.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptwf,Circle_p_wf,Circle_n.spn,Circle_p.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptwf,Circle_m_wf,Circle_n.spn,Circle_m.pts,132,132,ur,1,1000.,.0254,.00254,.0254
*,
*, Circle - Calculate maximum/minumm deviations - Wilson-Fowler
maxmin,Circle_n_wf.lg
maxmin,Circle_p_wf.lg
maxmin,Circle_m_wf.lg
*,
close
*,
*, Open plot file
open,Ellipse.wf.ps,p
*,
*, Ellipse - Calculate and plot deviation results - Wilson-Fowler
pltptwf,Ellipse_n_wf,Ellipse_n.spn,Ellipse_n.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptwf,Ellipse_p_wf,Ellipse_n.spn,Ellipse_p.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptwf,Ellipse_m_wf,Ellipse_n.spn,Ellipse_m.pts,132,132,ur,1,1000.,.0254,.00254,.0254
*,
*, Ellipse - Calculate maximum/minumm deviations - Wilson-Fowler
maxmin,Ellipse_n_wf.lg
maxmin,Ellipse_p_wf.lg
maxmin,Ellipse_m_wf.lg
```

```
*,
close
*,
*, Open plot file
open,Parabola.wf.ps,p
*,
*, Parabola - Calculate and plot deviation results - Wilson-Fowler
pltptwf,Parabola_n_wf,Parabola_n.spn,Parabola_n.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptwf,Parabola_p_wf,Parabola_n.spn,Parabola_p.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptwf,Parabola_m_wf,Parabola_n.spn,Parabola_m.pts,132,132,ur,1,1000.,.0254,.00254,.0254
*, Parabola - Calculate maximum/minumm deviations - Wilson-Fowler
maxmin,Parabola_n_wf.lg
maxmin,Parabola_p_wf.lg
maxmin,Parabola_m_wf.lg
*,
close
*,
*, Open plot file
open,Circle.cs.ps,p
*,
*, Circle - Calculate maximum/minumm deviations - Parametric Cubic
pltptcs,Circle_n_cs,Circle_n.spn,Circle_n.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptcs,Circle_p_cs,Circle_n.spn,Circle_p.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptcs,Circle_m_cs,Circle_n.spn,Circle_m.pts,132,132,ur,1,1000.,.0254,.00254,.0254
*,
*, Circle - Calculate maximum/minumm deviations - Parametric Cubic
maxmin,Circle_n_cs.lg
maxmin,Circle_p_cs.lg
maxmin,Circle_m_cs.lg
*,
close
*,
*, Open plot file
open,Ellipse.cs.ps,p
*,
*, Ellipse - Calculate maximum/minumm deviations - Parametric Cubic
pltptcs,Ellipse_n_cs,Ellipse_n.spn,Ellipse_n.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptcs,Ellipse_p_cs,Ellipse_n.spn,Ellipse_p.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptcs,Ellipse_m_cs,Ellipse_n.spn,Ellipse_m.pts,132,132,ur,1,1000.,.0254,.00254,.0254
*,
*, Ellipse - Calculate maximum/minumm deviations - Parametric Cubic
maxmin,Ellipse_n_cs.lg
maxmin,Ellipse_p_cs.lg
maxmin,Ellipse_m_cs.lg
off,geom
*
close
*, Open plot file
open,Parabola.cs.ps,p
*,
*, Parabola - Calculate maximum/minumm deviations - Parametric Cubic
pltptcs,Parabola_n_cs,Parabola_n.spn,Parabola_n.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptcs,Parabola_p_cs,Parabola_n.spn,Parabola_p.pts,132,132,ur,1,1000.,.0254,.00254,.0254
pltptcs,Parabola_m_cs,Parabola_n.spn,Parabola_m.pts,132,132,ur,1,1000.,.0254,.00254,.0254
*,
*, Parabola - Calculate maximum/minumm deviations - Parametric Cubic
maxmin,Parabola_n_cs.lg
maxmin,Parabola_p_cs.lg
maxmin,Parabola_m_cs.lg
*,
close
tty
```

Nonanalytical Shapes—Deviation Study

```
*, Non-Analytical Shapes CAD Systems Evaluations
*, Spline data
splndata,c01762.spn,132,132,c01762.spn,3,4,d, 90.0000,d,180.0000
splndata,c01763.spn,132,132,c01763.spn,3,4,d, 99.2767,d,171.5234
splndata,c01764.spn,132,132,c01764.spn,3,4,d,174.6875,d,177.5217
*,
*, Evaluation data
proedata,c01762_CAD.pts,132,132,c01762_CAD.pts,3,4
proedata,c01763_CAD.pts,132,132,c01763_CAD.pts,3,4
proedata,c01764_CAD.pts,132,132,c01764_CAD.pts,3,4
proedata,c01762_PRO.pts,132,132,c01762_PRO.pts,1,2
proedata,c01763_PRO.pts,132,132,c01763_PRO.pts,1,2
proedata,c01764_PRO.pts,132,132,c01764_PRO.pts,1,2
proedata,c01762_ICM.pts,132,132,c01762_ICM.pts,4,5
proedata,c01763_ICM.pts,132,132,c01763_ICM.pts,4,5
proedata,c01764_ICM.pts,132,132,c01764_ICM.pts,4,5
*,
on,tek,p,.9
on,grid
on,geom
*,
 Open plot file
open,AWE.wf.ps,p
*,
*, Plot Non-Analytical Shapes – Spline Data
*,
plot,,,c01762.spn,c01763.spn,c01764.spn
*,
*, Calculate and plot deviation results – Wilson-Fowler
pltptwf,c01762_CAD,c01762.spn,c01762_CAD.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01762_CAD.lg
pltptwf,c01763_CAD,c01763.spn,c01763_CAD.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01763_CAD.lg
pltptwf,c01764_CAD,c01764.spn,c01764_CAD.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01764_CAD.lg
pltptwf,c01762_PRO,c01762.spn,c01762_PRO.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01762_PRO.lg
pltptwf,c01763_PRO,c01763.spn,c01763_PRO.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01763_PRO.lg
pltptwf,c01764_PRO,c01764.spn,c01764_PRO.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01764_PRO.lg
pltptwf,c01762_ICM,c01762.spn,c01762_ICM.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01762_ICM.lg
pltptwf,c01763_ICM,c01763.spn,c01763_ICM.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01763_ICM.lg
pltptwf,c01764_ICM,c01764.spn,c01764_ICM.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01764_ICM.lg
*,
close
*,
*, Open plot file
open,AWE.cs.ps,p
*,
*, Calculate and plot deviation results – Parametric Cubic
pltptcs,c01762_CAD,c01762.spn,c01762_CAD.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01762_CAD.lg
pltptcs,c01763_CAD,c01763.spn,c01763_CAD.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01763_CAD.lg
pltptcs,c01764_CAD,c01764.spn,c01764_CAD.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01764_CAD.lg
pltptcs,c01762_PRO,c01762.spn,c01762_PRO.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01762_PRO.lg
```

```
pltptcs,c01763_PRO,c01763.spn,c01763_PRO.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01763_PRO.lg
pltptcs,c01764_PRO,c01764.spn,c01764_PRO.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01764_PRO.lg
pltptcs,c01762_ICM,c01762.spn,c01762_ICM.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01762_ICM.lg
pltptcs,c01763_ICM,c01763.spn,c01763_ICM.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01763_ICM.lg
pltptcs,c01764_ICM,c01764.spn,c01764_ICM.pts,132,132,ur,1,5000.,.00254,.00254
maxmin,c01764_ICM.lg
*,
close
tty
```

Analytical Shapes—End-Angle Effects

```
*, Analytical Shapes – End Angle Study
*,
*, Circle – Spline and evaluation data
splndata,Circle_n.spn,132,132,Circle_n_spn,3,4,d,90.00,d,180.00
splndata,Circle_p.spn,132,132,Circle_n_spn,3,4,d,90.25,d,180.25
splndata,Circle_m.spn,132,132,Circle_n_spn,3,4,d,89.75,d,179.75
proedata,Circle_n.pts,132,132,Circle_n_pts,3,4
*,
*, Ellipse – Spline and evaluatuin data
splndata,Ellipse_n.spn,132,132,Ellipse_n_spn,3,4,d,90.00,d,180.00
splndata,Ellipse_p.spn,132,132,Ellipse_n_spn,3,4,d,90.25,d,180.25
splndata,Ellipse_m.spn,132,132,Ellipse_n_spn,3,4,d,89.75,d,179.75
proedata,Ellipse_n.pts,132,132,Ellipse_n_pts,3,4
*,
*, Parabola – Spline and evaluation data
splndata,Parabola_n.spn,132,132,Parabola_n_spn,3,4,d,90.00,d,163.3008
splndata,Parabola_p.spn,132,132,Parabola_n_spn,3,4,d,90.25,d,163.5508
splndata,Parabola_m.spn,132,132,Parabola_n_spn,3,4,d,89.75,d,163.0508
proedata,Parabola_n.pts,132,132,Parabola_n_pts,3,4
*,
on,tek,p,.9
on,geom
*, Ope plt file
open,Circle.end.ps,p
*,
*, Circle – Calculate and plot deviations – Wilson-Fowler
pltptwf,Circle_p_wf,Circle_p.spn,Circle_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
pltptwf,Circle_m_wf,Circle_m.spn,Circle_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
maxmin,Circle_p_wf.lg
maxmin,Circle_m_wf.lg
*,
*, Circle – Calculate and plot deviations – Parametric Cubic
pltptcs,Circle_p_cs,Circle_p.spn,Circle_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
pltptcs,Circle_m_cs,Circle_m.spn,Circle_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
maxmin,Circle_p_cs.lg
maxmin,Circle_m_cs.lg
*,
close
*,
*, Open plot file
open,Ellipse.end.ps,p
*,
*, Ellipse – Calculate and plot deviations – Wilson-Fowler
pltptwf,Ellipse_p_wf,Ellipse_p.spn,Ellipse_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
pltptwf,Ellipse_m_wf,Ellipse_m.spn,Ellipse_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
maxmin,Ellipse_p_wf.lg
maxmin,Ellipse_m_wf.lg
*,
```

```
*, Ellipse – Calculate and plot deviations – Parametric Cubic
pltptcs,Ellipse_p_cs,Ellipse_p.spn,Ellipse_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
pltptcs,Ellipse_m_cs,Ellipse_m.spn,Ellipse_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
maxmin,Ellipse_p_cs.lg
maxmin,Ellipse_m_cs.lg
*,
close
*,
*, Open plot file
open,Parabola.end.ps,p
*,
*, Parabola – Calculate and plot deviations – Wilson-Fowler
pltptwf,Parabola_p_wf,Parabola_p.spn,Parabola_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
pltptwf,Parabola_m_wf,Parabola_m.spn,Parabola_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
maxmin,Parabola_p_wf.lg
maxmin,Parabola_m_wf.lg
*,
*, Parabola – Calculate and plot deviations – Parametric Cubic
pltptcs,Parabola_p_cs,Parabola_p.spn,Parabola_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
pltptcs,Parabola_m_cs,Parabola_m.spn,Parabola_n.pts,132,132,ur,1,5000.,.00254,.00254,.0254
maxmin,Parabola_p_cs.lg
maxmin,Parabola_m_cs.lg
*,
close
tty
```

References

1. A.H. Fowler and C.W. Wilson, "Cubic Spline, A Curve Fitting Routine," Union Carbide Nuclear Company report (September 1962).

2. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in FORTRAN*, *The Art of Scientific Computing*, second ed. (Cambridge University Press, New York, 1992), pp. 107-110.

3. Ronald M. Dolin, "The Wilson-Fowler Spline in a Global IGES Coordinate Frame," Los Alamos National Laboratory report LA-11024-MS (September 1987).

4. W.R. Melvin, "Error Analysis and Uniqueness Properties of the Wilsion-Fowler Spline," Los Alamos National Laboratory report LA-9178 (August 1982).

5. "The CAD/CAM Hall of Fame," *Machine Design*, October 1998, p. 104.

6. Ronald M. Dolin and The WX-Division Integrated Software Tools Team (Jose Arellano, Ralph Gladfelter, John Hopkins, Robert Montoya, Kent Musgrave, Scott Parkinson, Bob Reinhart, Mike Smith, Royce Tyler), "Interchangeable Spline Reference Guide," Los Alamos National Laboratory report LA-12746-M (May 1994).

7. Billie M. Wheat and Wilbur D. Birchler, "TDAP User's Guide," Los Alamos National Laboratory WX-4 Technical Engineering Support Group internal manual (April 1984).

8. *Mathcad, User's Guide, Mathcad Plus 6, Macintosh Version*, professional ed. (MathSoft, Inc., Cambridge, Massachusetts, 1996) pp. 1-630.

9. Andy Oughton, computer-aided design manager, AWE Hunting-Brae, Aldermaston, Reading, Berkshire, United Kingdom, personal communication, November 1999.

This report has been reproduced directly from the best available copy. It is available electronically on the Web (http://www.doe.gov/bridge).

Copies are available for sale to U.S. Department of Energy employees and contractors from—

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831
(423) 576-8401

Copies are available for sale to the public from—

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22616
(800) 553-6847

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545