A Look at Mathematical and Computational Issues in Manufacturing Inspection Using Coordinate Measuring Machines

January 31, 2006

Craig Shakarji

Manufacturing Engineering Laboratory

NIST

Overview

- Overview of Coordinate Measuring Machines
- Quick history of least squares testing
- ATEP-CMS program
- Other fit types
- Industrial Intercomparison:
  Alert to industrial need for new references
- Why are the other fit types hard?
- Solving the new, Cheybshev fit types
- Complex surface fitting

Introduction

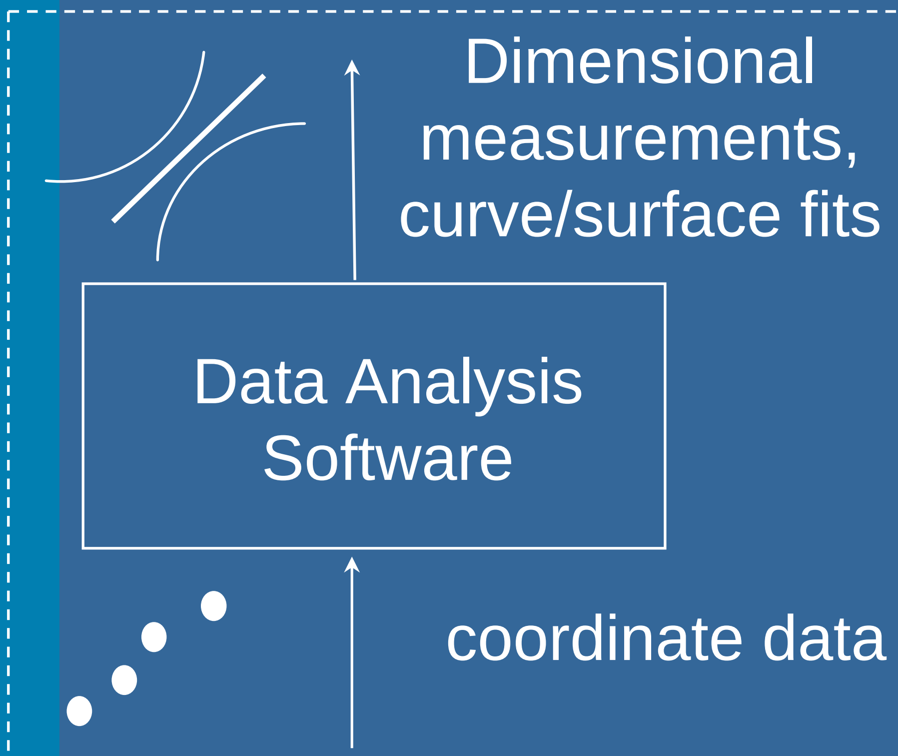
This talk involves fitting software embedded in coordinate measuring systems (CMMs and other systems that gather and process coordinate data, e. g., laser trackers, theodolites, photogrammetry, etc.)

Manufacturing Engineering

NIST

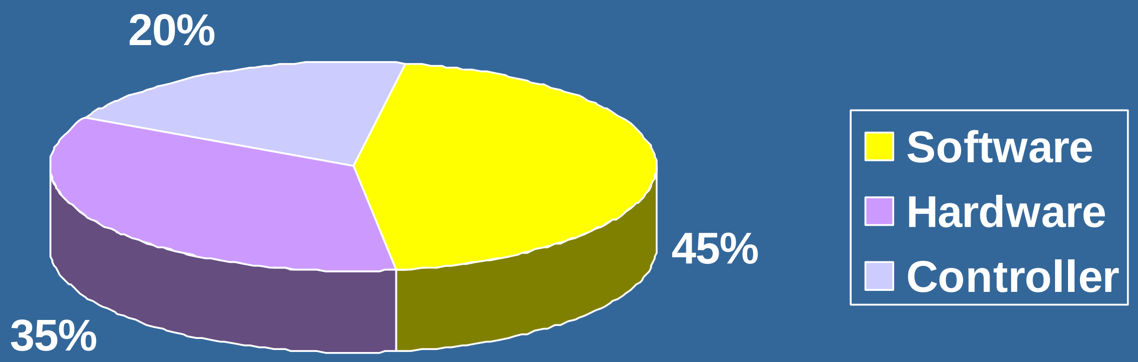
Mathematical Processing

There is measurement uncertainty associated with software embedded in coordinate measuring systems



Motivation and Background

- 1988 GIDEP alert
- Serious problems in least-squares fitting software



Least-Squares Testing

- **NIST and PTB offer least-squares algorithm testing testing for standard shapes (lines, planes, circles, spheres, cylinders, cones)**
- **Sample NIST ATEP-CMS test report:**

REPORT OF SPECIAL TEST

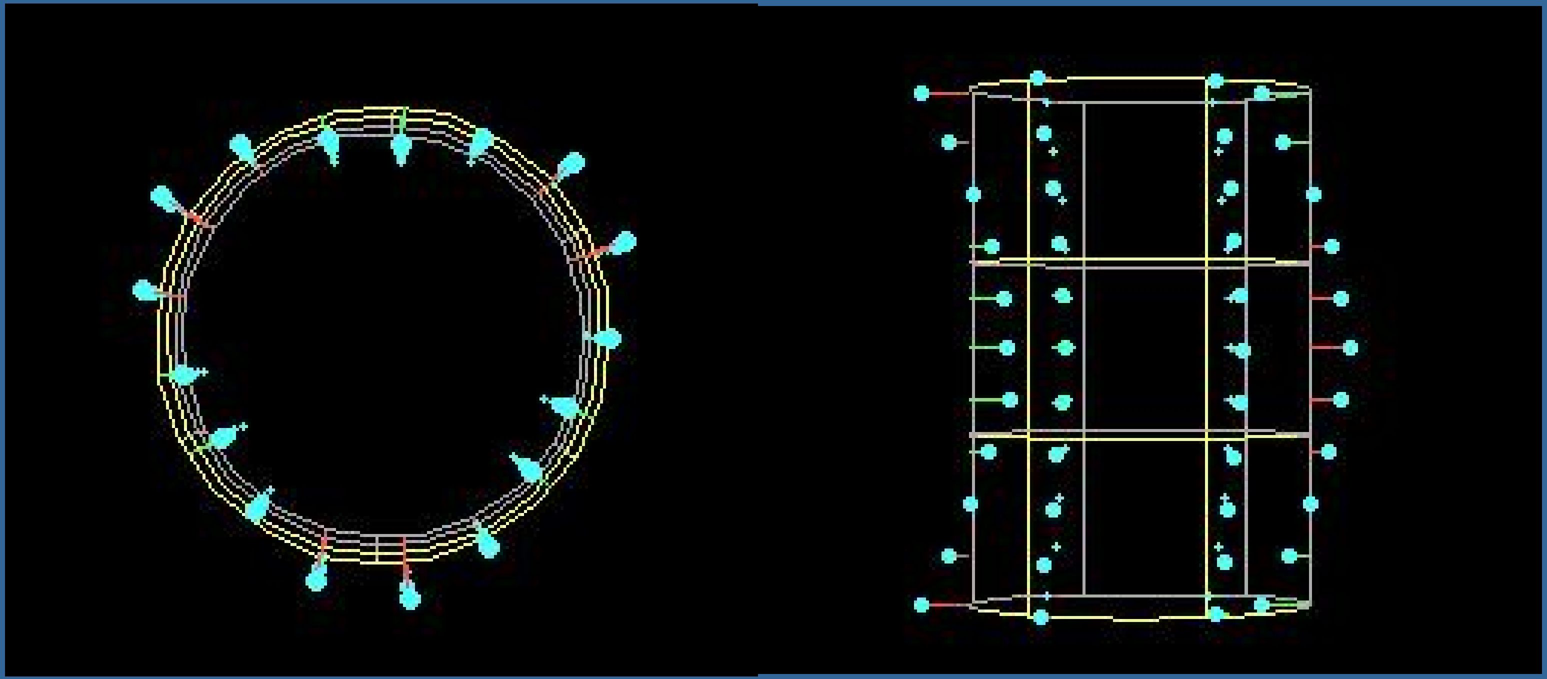
For

Submitted by: Good-Fit Inc.

This software package was tested on 180 data sets, representing the following geometry types: lines, circles, planes, spheres, cylinders, and cones and following the test procedures documented in NISTIR 5686. The results of the tests are as follows:

Deviation And Uncertainty	**Geometry Type**					
	lines	circles	planes	spheres	cylinders	cones
Separation (μm) Uncertainty (μm)	3.0×10^{-3} 3.5×10^{-3}	1.2×10^{-4} 1.0×10^{-4}	2.3×10^{-4} 5.4×10^{-4}	8.1×10^{-5} 5.3×10^{-5}	9.3×10^{-2} 1.8×10^{-3}	4.8×10^{-4} 5.7×10^{-4}
Tilt (arcseconds) Uncertainty	6.2×10^{-4} 3.5×10^{-4}	7.1×10^{-3} 5.5×10^{-3}	3.3×10^{-3} 3.0×10^{-3}	________	1.4×10^{-2} 2.3×10^{-2}	3.2×10^{-3} 4.3×10^{-3}
Radius (μm) Uncertainty (μm)	________	4.9×10^{-4} 6.2×10^{-4}	________	7.7×10^{-6} 8.4×10^{-6}	4.3×10^{-1} 3.4×10^{-1}	________
Distance (μm) Uncertainty (μm)	________	________	________	________	________	3.5×10^{-5} 5.3×10^{-5}
Angle(arcseconds)	________	________	________	________	________	4.9×10^{-4}

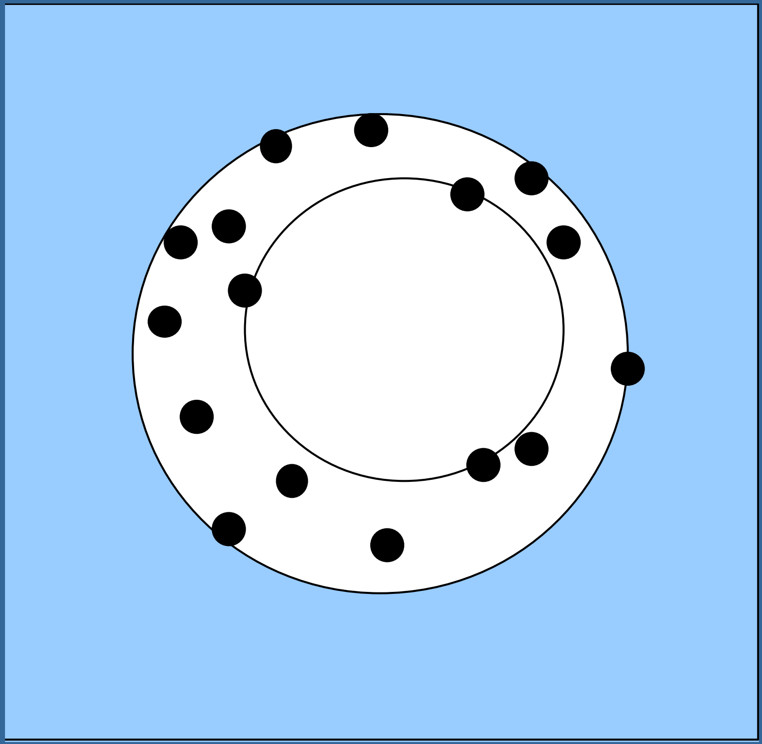
Imposed form error on data sets



- ASME B89.4.10
- ISO 10360-6

ATEP-CMS Program

- NIST Special Test Service: Least-squares algorithm testing for standard shapes (lines, planes, circles, spheres, cylinders, cones)
- Results … Better Algorithms? Yes!
- However … What about other fitting criteria? (Min-zone, max-inscribed, min-circumscribed) Improvements did not carry over



Manufacturing Engineering

NIST

Importance of Work

Recent work in testing and comparing maximum-inscribed, minimum-circumscribed, and minimum-zone (Chebyshev) fitting algorithms indicates that serious problems can exist in present commercial software packages

Applicability of Fit Objectives

	Minimum-zone	Max-inscribed	Min-circumscribed
Lines	X		
Planes	X		
Circles	X	X	X
Spheres	X	X	X
Cylinders	X	X	X
Cones	X		

Intercomparison Results

- Why only two packages? Is that enough?
- Can one identify which is the better fit when there is a difference from the reference fit
- Comparison classifications
  - “Good” < 10% of form error
  - “Poor” 10 - 50% of form error
  - “Failure” > 50% or other breakdown

Maximum-Inscribed Circles

	Industrial Software A	Industrial Software B
Good	●●●●●●●●●●	●●●●●●●●●
Poor		●
Failure		

Manufacturing Engineering

Maximum-Inscribed Spheres

	Industrial Software A	Industrial Software B
Good	●●●●●●	
Poor		
Failure	x	xxxxxxxx

NIST

Maximum-Inscribed Cylinders

	Industrial Software A	Industrial Software B
Good	●●●	●●●●●●●●●●
Poor	●	●
Failure	xxxxx	

Minimum-Circumscribed Circles

	Industrial Software A	Industrial Software B
Good	●●●●●●●●●●	●●●●●●●●●●
Poor		
Failure		

Minimum-Circumscribed Spheres

	Industrial Software A	Industrial Software B
Good	●●●●●●	●
Poor	●	
Failure		xxxxxxxxx

Minimum-Circumscribed Cylinders

	Industrial Software A	Industrial Software B
Good	●●●	●●●●●●●●●●
Poor	●	
Failure	xxxxx	

Minimum-Zone Lines

	Industrial Software A	Industrial Software B
Good	●●●●●●●	●●●●●
Poor	●	
Failure	xx	xxxxx

Manufacturing Engineering

NIST

Minimum-Zone Planes

	Industrial Software A	Industrial Software B
Good	●●●●●●●●●●	●●●●●●●●●
Poor		
Failure		x

Manufacturing Engineering

Minimum-Zone Circles

	Industrial Software A	Industrial Software B
Good	●●●●●●●●●●	●●●●●●●●●●
Poor		
Failure		

NIST

Minimum-Zone Spheres

	Industrial Software A	Industrial Software B
Good	●●●●●●●●●	●●●●●●●●●●
Poor		
Failure		

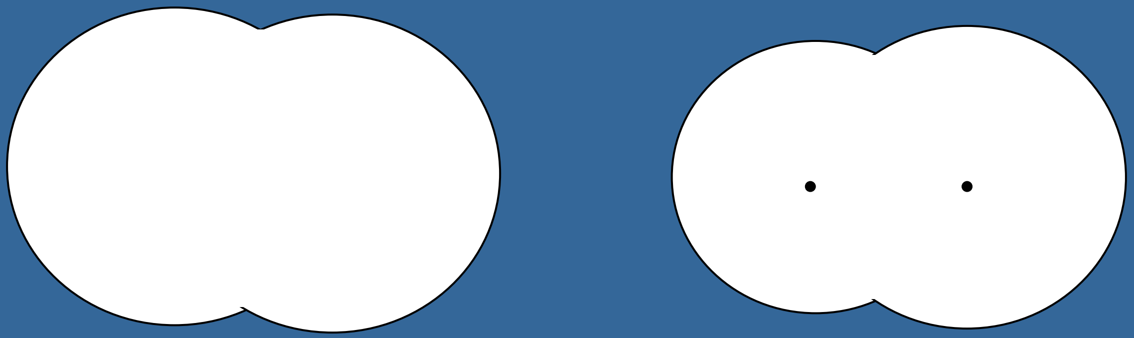
Minimum-Zone Cylinders

	Industrial Software A	Industrial Software B
Good	●●●●●●●●●	
Poor	●	
Failure		

Minimum-Zone Cones

	Industrial Software A	Industrial Software B
Good		
Poor	●●	
Failure	xxxxxxxx	

Why are these fits difficult?

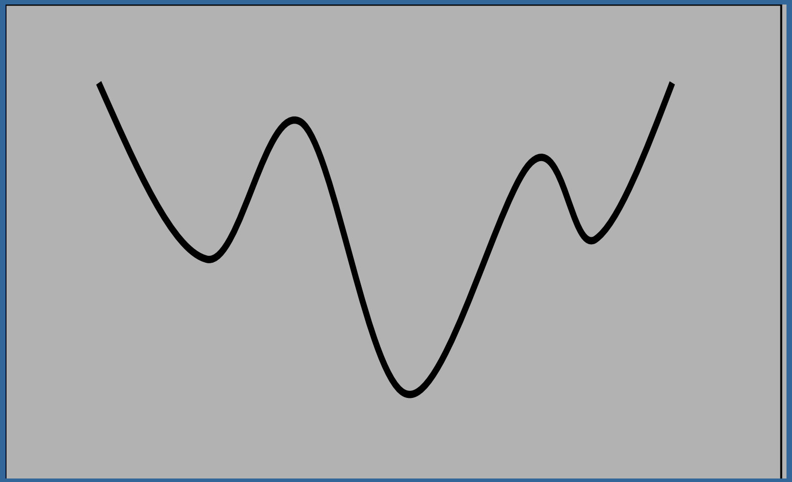


Maximum inscribed circles:

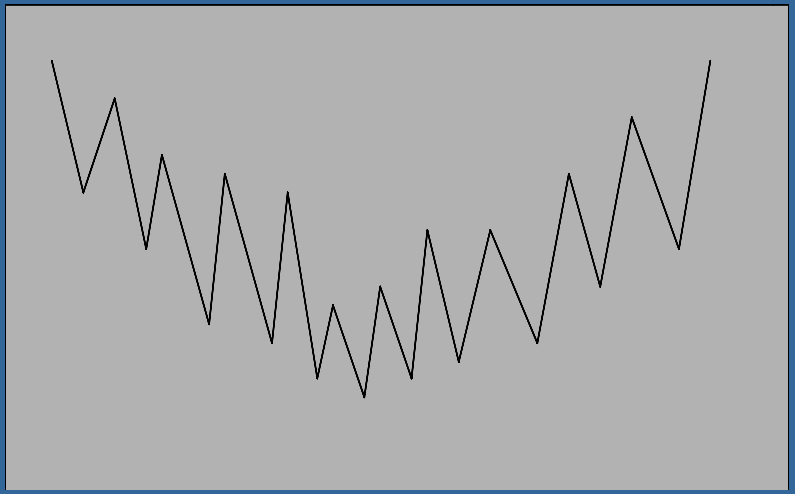
- **Multiple Solutions**
- **Hidden Solutions**

Fitting Objective Functions

- **Least-squares objective function is differentiable and has a wide range of convergence.**



- **Minimum-zone objective function is not smooth and has several local minima surrounding the optimal.**



NIST Reference Algorithms

- Correctness more important than speed
- Based on simulated annealing
- Known to find a global minimum in the presence of several nearby local minima
- “Temperature” parameter can be controlled to decrease slowly for better convergence
- Tested internally with constructed data sets

How does it work?

- **Compute least-squares fit (easy?)**
- **Rotate and translate the data based on the computed least-squares fit**
- **Define the geometry with fewer variables**
- **Search for the minimum (or maximum) using the simulated annealing technique.**
  - The parameters of the search are given in table
  - The transformed least-squares solution is used as the initial guess for the optimization search
- **Derive any additional parameters that define the geometry according to the table**

Table Information

	Location	Direction	Parameters used in optimization	Objective Function	Derived parameter after optimization
Min-Zone Cylinder	**(*x*, *y*, 0)**	**(*A*, *B*, 1)**	**(*x*, *y*, *A*, *B*)**	**max(*f*) – min(*f*)**	**r=[max(*f*) – min(*f*)] / 2**

Minimum-Zone Cylinder Example

Manufacturing Engineering

- Compute least-squares cylinder
- Rotate/Translate making cylinder axis = z-axis
- From Table: Define nearby cylinders by location of axis on *xy* plane and direction (*A*, *B*, 1). (Least squares cylinder is (0, 0, 0) and (0, 0, 1))
- Search over (*x*, *y*, *A*, *B*) starting with (0, 0, 0, 0) to find minimum of objective function, max(*f*) – min(*f*)
- Compute radius of min-zone cylinder:
  r=[max(*f*) – min(*f*)] / 2

NIST

View of Full Table

We also define

	Location	Direction	Parameters used in optimization	Objective function	Derived parameter after optimization
Min-zone line	$(x,y,0)$	$(A,B,1)$	(x,y,A,B)	$\max(f_i)$	
Min-zone plane	$(0,0,z)$	$(A,B,1)$	(A,B)	$\max(g_i)-\min(g_i)$	$z=[\max(g_i)+\min(g_i)]/(2c)$
Min-zone circle	$(x,y,0)$		(x,y)	$\max(h_i)-\min(h_i)$	$r=[\max(h_i)+\min(h_i)]/2$
Min-circ circle	$(x,y,0)$		(x,y)	$\max(h_i)$	$r=\max(h_i)$
Max-ins circle	$(x,y,0)$		(x,y)	$\min(h_i)$	$r=\min(h_i)$
Min-zone sphere	(x,y,z)		(x,y,z)	$\max(h_i)-\min(h_i)$	$r=[\max(h_i)+\min(h_i)]/2$
Min-circ sphere	(x,y,z)		(x,y,z)	$\max(h_i)$	$r=\max(h_i)$
Max-ins sphere	(x,y,z)		(x,y,z)	$\min(h_i)$	$r=\min(h_i)$
Min-zone cylinder	$(x,y,0)$	$(A,B,1)$	(x,y,A,B)	$\max(f_i)-\min(f_i)$	$r=[\max(f_i)+\min(f_i)]/2$
Min-circ cylinder	$(x,y,0)$	$(A,B,1)$	(x,y,A,B)	$\max(f_i)$	$r=\max(f_i)$
Max-ins cylinder	$(x,y,0)$	$(A,B,1)$	(x,y,A,B)	$\max(f_i)$	$r=\min(f_i)$
Min-zone cone	$(x,y,0)$	$(A,B,1)$	(x,y,A,B,ψ)	$\max(d_i)-\min(d_i)$	$s=[\max(d_i)+\min(d_i)]/2$

Maximum Inscribed Circle
Testing Versus Exhaustive Solutions
(Data Set Intentionally Created to Yield Multiple Solution)

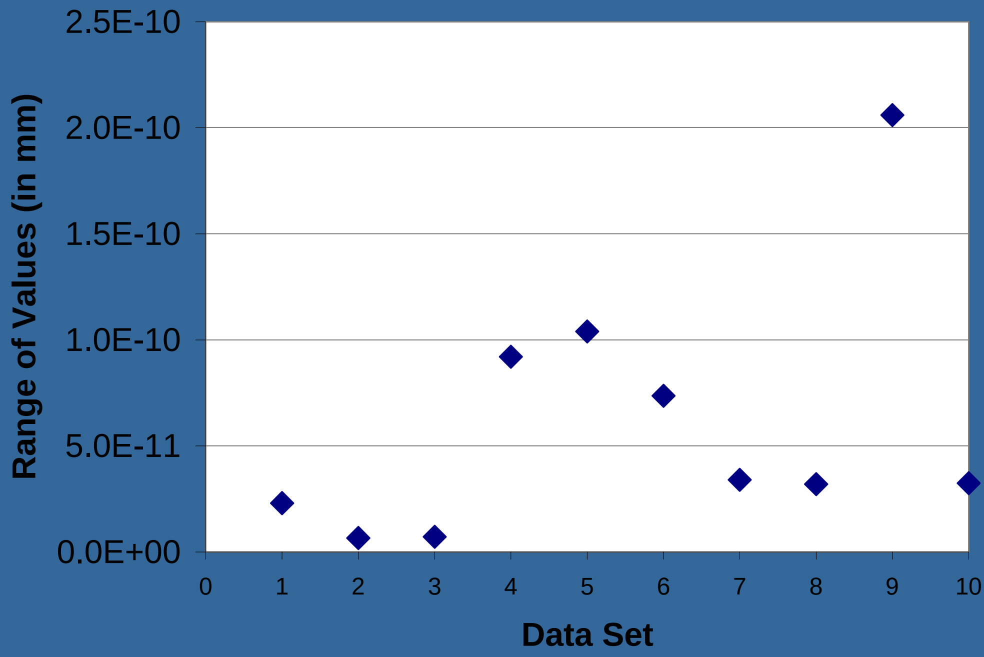
	Exhaustive Search	Simulated Annealing
x	-0. 00369371351261293	. 00369371351260858
y	-. 00784954077495501	. 00784954077494546
r	. 9726878093314897	. 9726878093314895

Manufacturing Engineering

NIST

Additional Testing

- Testing versus known solutions (data sets constructed with known solutions)
- Testing versus industrial results
- Testing by observing repeatability



General Surfaces: “Triples”

Goal: Provide industry with a collection of test cases, allowing for the comparison of industrial software with reference fits.

A “*Reference Triple*” consists of:

- **Dataset**
- **Defined Surface**
- **Correct Least-Squares Transformation**

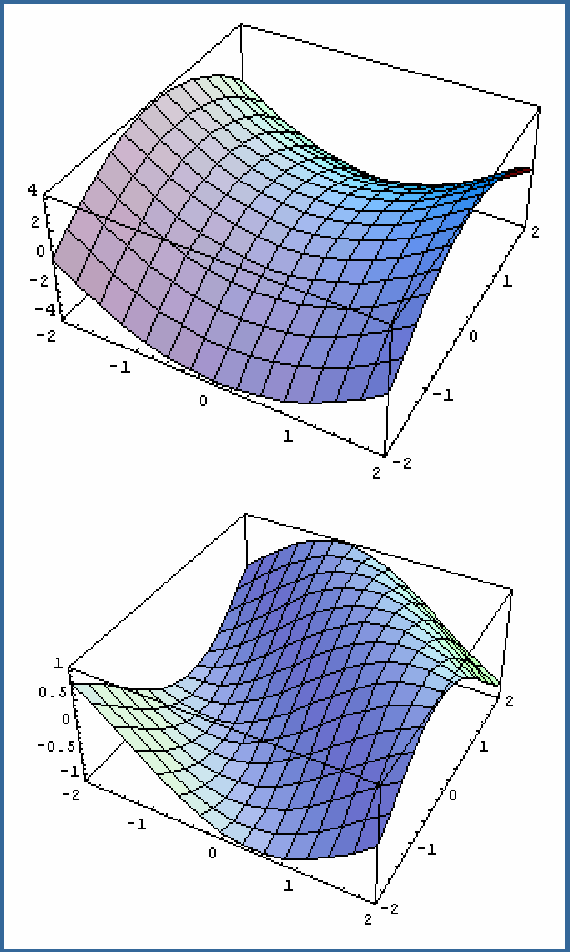
Manufacturing Engineering

NIST

Manufacturing
Engineering

Milestones

- Two reference algorithms exist to fit data rigidly to a general shape
- The two reference algorithms have been compared in many test cases; used standard shapes for verification (planes, cylinders, cones)
- Triples available for several shapes (paraboloids, ogives, saddles, etc.)
- Completed comparison work with industrial partner
- Mathematica arbitrary precision prevents roundoff effects in reference results



NIST

Conclusion

- **12 Chebyshev reference algorithms developed with various fit objectives and geometric shapes**
- **Fourfold method of testing**
  - Compare with exhaustive search
  - Compare with known solutions
  - Compare with industrial solutions
  - Compare with itself (repeatability)
- **Approach demonstrated to work well**
- **NIST making reference pairs available**
- **Future expansion of test service being considered at NIST and ASME**
- **Some application to complex surfaces**