Andromeda™

Software for the Global Optimization of Multiple Functions with Constraints

This presentation describes the functionality of the Andromeda software (currently implemented as an Excel application) and gives a basic overview of the new technology that it employs for global multi-objective optimization: the Particle Search Engine (PSE) algorithm.

What is Global Optimization?

Whenever you have a mathematical model, you typically want to find the input values to the model which will:

1) Maximize the output, or
2) Minimize the output, or
3) Target some particular output value.

Doing any one of those 3 things is called optimization.

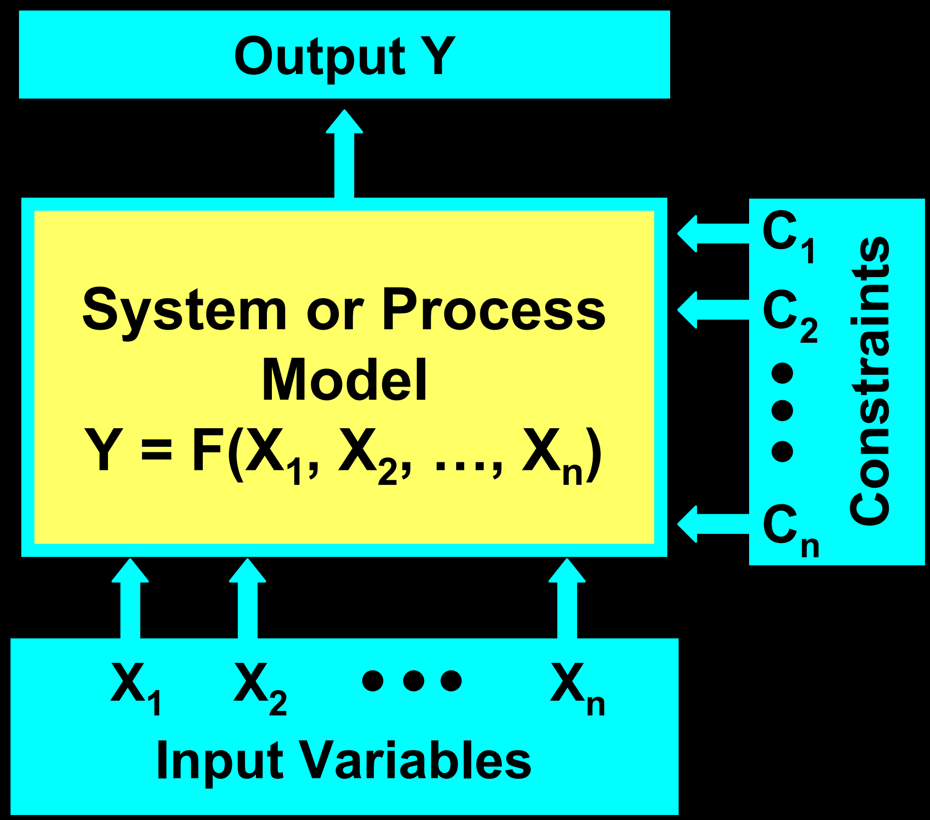
Finding the very best of all the locally optimal solutions (e.g., the very highest of all the peaks when maximizing a function) is called global optimization.

If there are two or more functions (model elements) that you wish to simultaneously and globally co-optimize, it's called global multiple objective optimization.

The PSE algorithm used in Andromeda offers a new way to perform single or multiple objective global optimization, and it works with ANY types of functions, constraints and input variables.

Uses of Global Optimization

In every technology and business, mathematical modeling and optimization can be used to make any system or process as effective or functional as possible.



Some Common Examples of Business Models and Processes to be Optimized

Finance and Investment

• Working capital management involves allocating cash to different purposes (accounts receivable, inventory, etc.) across multiple time periods, to maximize interest earnings.

• Capital budgeting involves allocating funds to projects that initially consume cash but later generate cash, to maximize a firm's return on capital.

• Portfolio optimization -- creating "efficient portfolios" -- involves allocating funds to stocks or bonds to maximize return for a given level of risk, or to minimize risk for a target rate of return.

Manufacturing

• Job shop scheduling involves allocating time for work orders on different types of production equipment, to minimize delivery time or maximize equipment utilization.

• Blending (of petroleum products, ores, animal feed, etc.) involves allocating and combining raw materials of different types and grades, to meet demand while minimizing costs.

• Cutting stock (for lumber, paper, etc.) involves allocating space on large sheets or timbers to be cut into smaller pieces, to meet demand while minimizing waste.

Distribution and Networks

• Routing (of goods, natural gas, electricity, digital data, etc.) involves allocating something to different paths through which it can move to various destinations, to minimize costs or maximize throughput.

• Loading (of trucks, rail cars, etc.) involves allocating space in vehicles to items of different sizes so as to minimize wasted or unused space.

• Scheduling of everything from workers to vehicles and meeting rooms involves allocating capacity to various tasks in order to meet demand while minimizing overall costs.

What types of global optimization problems can Andromeda solve?

Andromeda optimizes any type of Functions (F's): The global optimization problems can involve a single function that is to be optimized, or multiple functions that are to be simultaneously co-optimized (also known as multiple objective optimization). For example:

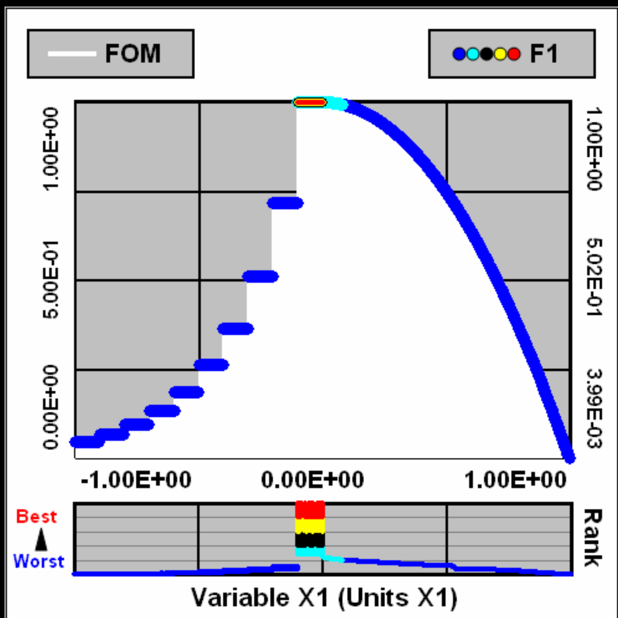
F1(X1,X2,…,Xn) = Maximum And F2 (X1,X2,…,Xn) = Minimum And F3 (X1,X2,…,Xn) = 0.0

The functions to be optimized can be continuous, or discontinuous, and even noisy. In its current implementation, the Andromeda software is an Excel application, and the functions to be optimized can be any valid expression in Excel (including logical functions, array formulas, user defined functions, and complex models that one links to either on another Excel worksheet or within another Excel workbook).

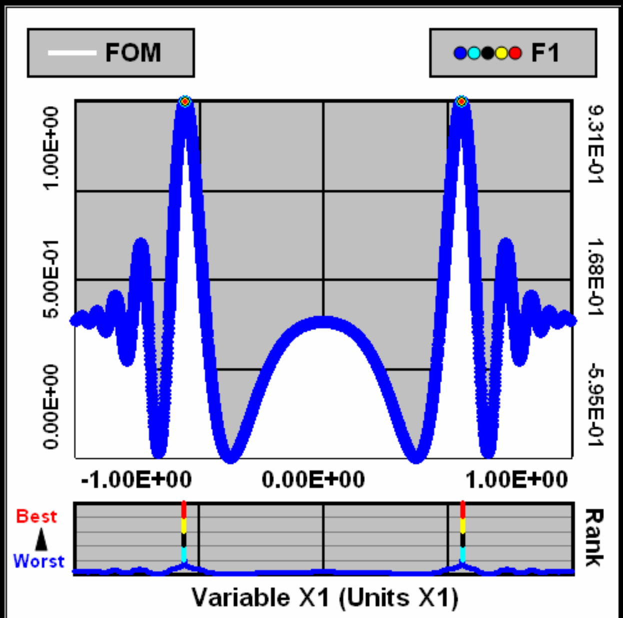
Andromeda handles any type of Constraints (C's): A constraint is any condition that a solution to an optimization problem must satisfy. The constraints specified by the user may be "Hard" (meaning that they must be exactly satisfied within the machine precision limits), or "Soft" (meaning that they are included as part of an objective function with some specified weight indicating relative importance), and they can also be in the form of either equality constraints or inequality constraints. Just as in the case of functions, any valid expression in Excel can be used to define each of the constraints.

Andromeda handles any Input Variables (X's): The independent variables (X's) which are being solved for may be continuous or integer, and the search interval for each individual X may be either bounded or unbounded (unbounded meaning that the solutions can evolve outside of the initial search region specified by the user).

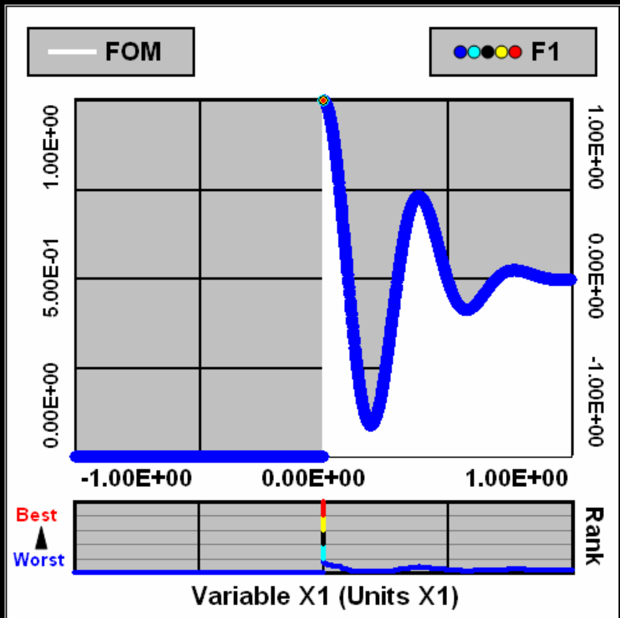
Some examples of global optimization problems that Andromeda can solve



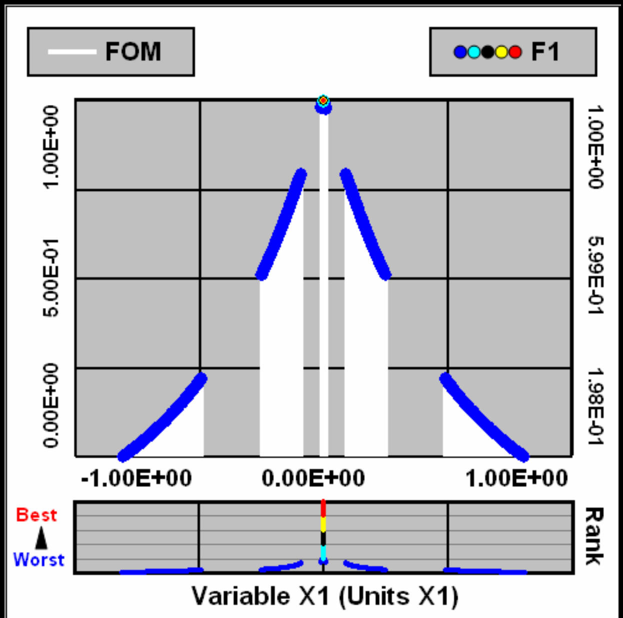
Any type of Functions (Smooth, or Discontinuous, and even Noisy)



Any number of Global Optima



- **Discontinuity at Solution values**
- **Solutions on Constraint Boundaries**
- **Flat Regions of Function Space**



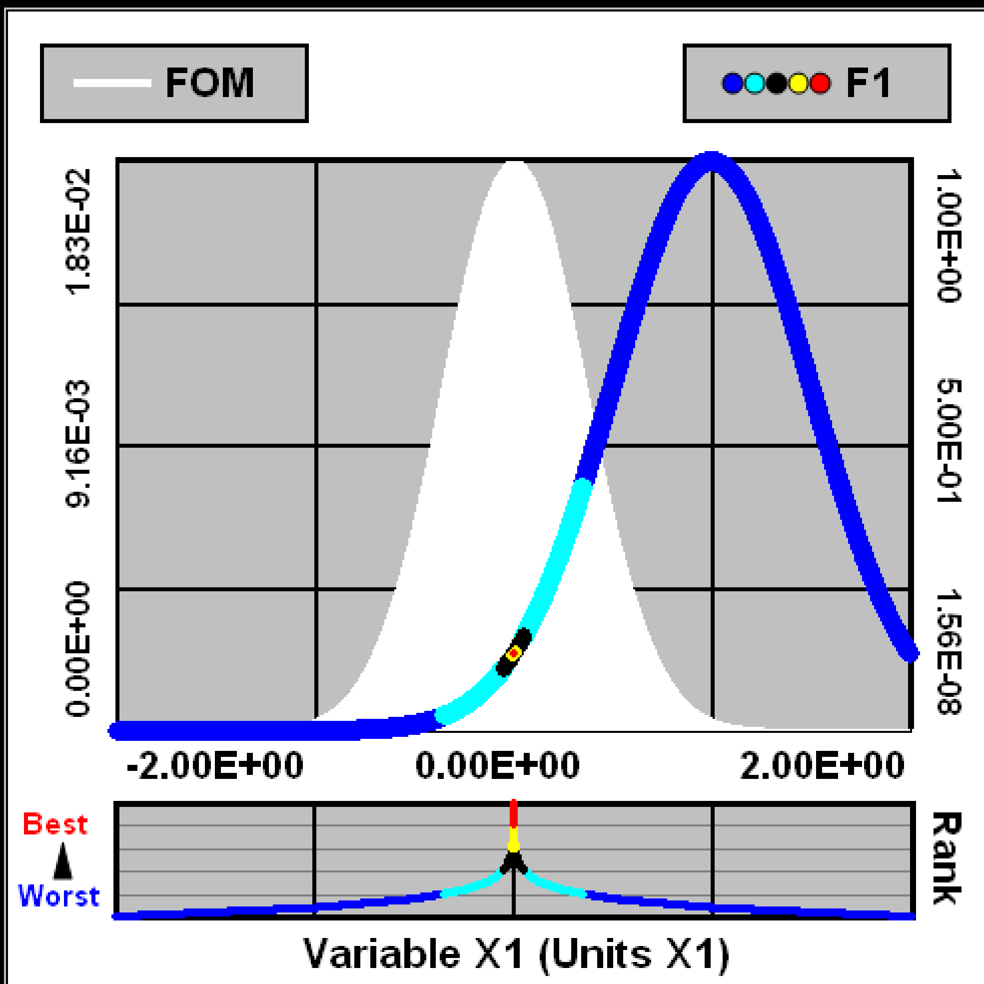
Any type of Constraints

More examples of global optimization problems Andromeda can solve

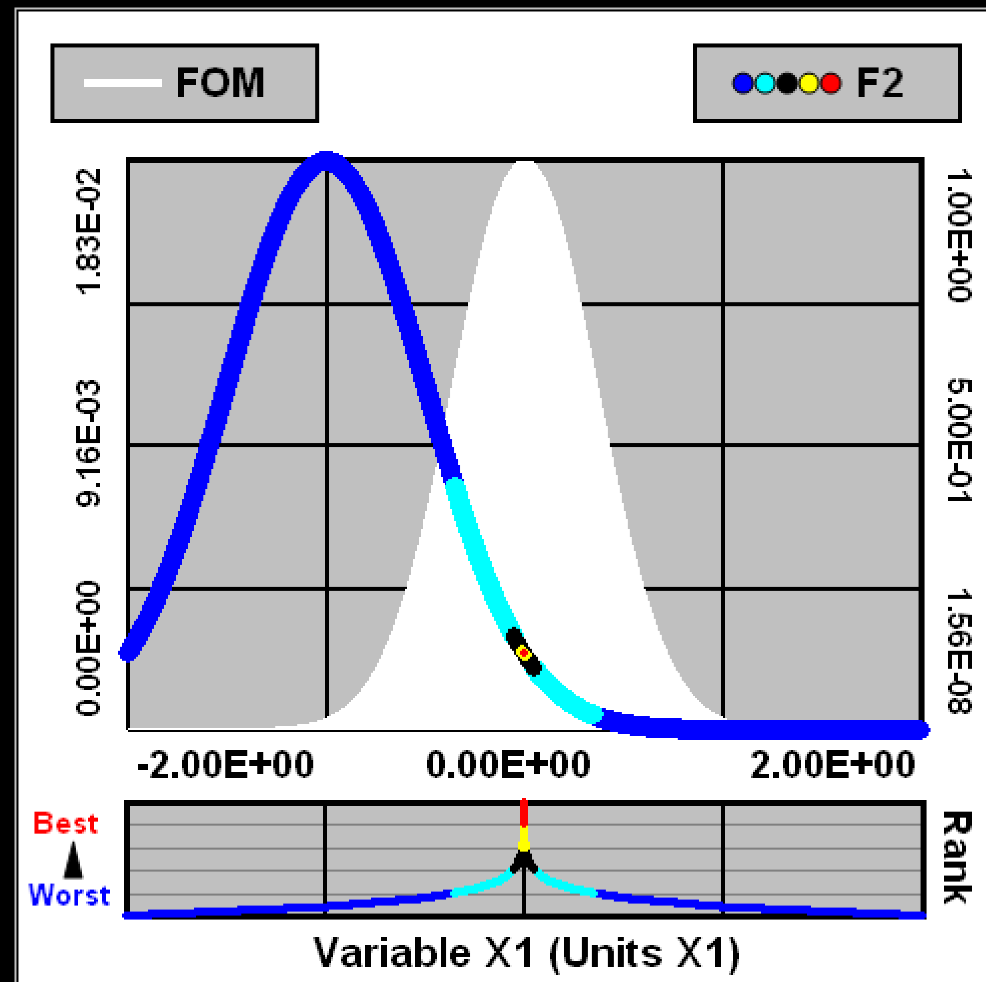
Multiple Objective Optimization

Example: Simultaneously Solve F1 = Maximum AND F2 = Maximum

F1 = Exp(-2*(X1-1)^2)

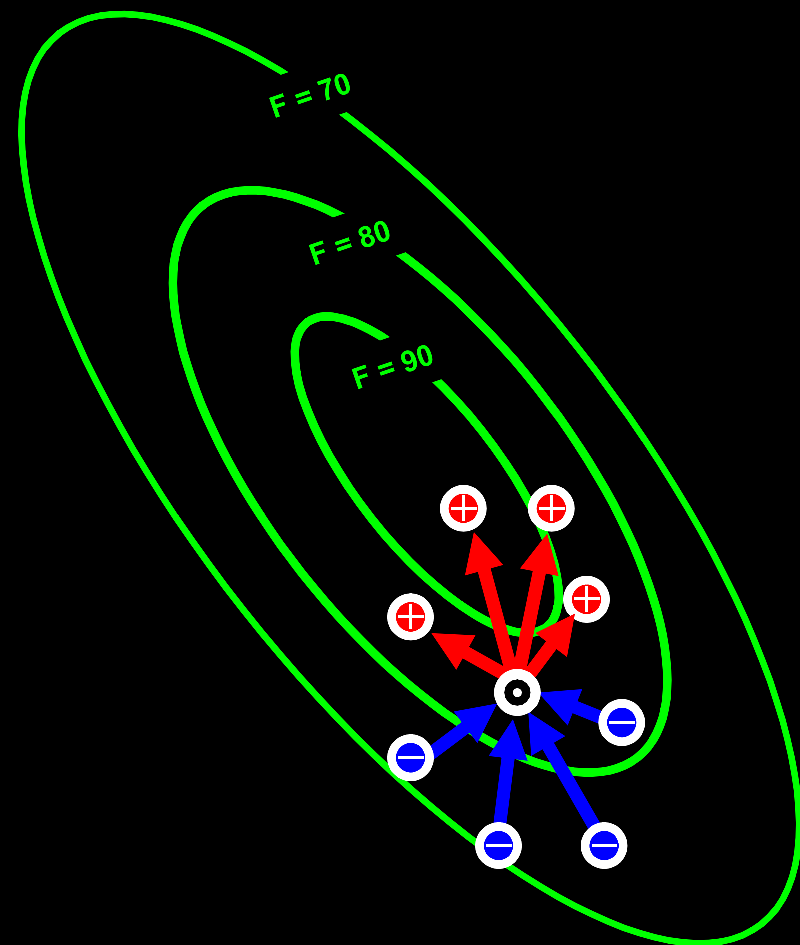


F2 = Exp(-2*(X1+1)^2)



How does the Particle Search Engine (PSE) algorithm work?

In the PSE algorithm, each sample of the function(s) at some particular coordinates within the search space is conceptually viewed as a "particle." The initial particle positions can be chosen either randomly, or as an array that uniformly spans the search space. The particle distribution then evolves as each particle moves towards increasingly better functional values with each iteration of the algorithm.



Particle Propagation Direction: The direction of propagation for each particle is governed by a Vector Force Field which is created by the nearest neighbor particles. Each particle is attracted to the nearest neighbor particles that have "better" functional values, and is repelled by the nearest neighbor particles having "worse" functional values.

Particle Propagation Distance: The particle propagation distance for each iteration is governed by simple rules involving factors such as the local particle density and some aspects of the general behavior of the function.

The Vector Force Field

Each propagating particle: ⊙

is attracted to particles with improved values: ⊕

and repelled by particles with worse values: ⊖

A new and universally applicable method for global optimization

The exact details of the PSE algorithm are proprietary, but in essence the vector force field governing particle motion causes each particle to move along a direction that is roughly the local path of steepest ascent for that particle, much like making an approximate gradient calculation. Unlike the case of an actual gradient calculation, however, the functions that are to be optimized do NOT need to be smooth and differentiable, or even continuous when using the PSE algorithm.

In addition, the particles are intelligent in the sense that they follow simple rules which determine the propagation distances chosen for each move. These rules make use of both local and global information that is shared by all of the particles.

The particles
move locally but "think" globally.

The PSE algorithm is an evolutionary algorithm in which the particle distribution continuously evolves towards the best solutions. Unlike other evolutionary techniques for finding the global optima of nonsmooth functions, the PSE algorithm can provide rapid and reliable convergence to EXACT solutions (within the precision limits of the machine, or to the precision specified by the user).

How many particles are required by the PSE algorithm?

The minimum number of particles that must be used in each iteration of the PSE algorithm will be referred to as the "Pod" size. For a given number of dimensions in the function space, N_x = # independent variables (X's) that we are solving for, the corresponding Pod sizes required in the PSE algorithm are:

N_x	Pod Size
1	2
2-10	10
≥ 10	N_x

The user specifies the particle population N_p (≥ the Pod size) representing the number of particles that will propagate in each iteration of the algorithm.

- If the problem has only a single optimum and no other local optima within the search space, the exact solution can be obtained by using the minimum number of particles (N_p = Pod size).

- If the problem has many local optima, then using a sufficiently large number of particles [N_p = (Pod size) x (# local optima)] will enable a massively parallel search of the function space, with the particles simultaneously converging on multiple local optima until the global optimum dominates.

Comparison to other methods of function optimization

Method of Steepest Ascent: A very broad class of algorithms use gradient calculations in order to search for the local optimum that is closest to the initial starting point specified by the user. These algorithms can often have extremely rapid convergence to local optima, especially in problems that they are specifically designed for (such as functions that are polynomials of some particular finite order). The key drawback to such methods when solving general optimization problems is that the functions to be optimized must be continuous and smooth (having first and sometimes also second derivatives) in order for these methods to be employed.

Evolutionary Algorithms: Global optimization requires the ability to explore all of the local optima within the designated search space, and this can be accomplished with the use of evolutionary algorithms such as the Genetic Algorithm or Particle Swarm Optimization. In addition, these techniques can also handle nonsmooth functions. The key drawback of previous evolutionary methods is that the local convergence is usually extremely slow, far too slow to make high precision solutions feasible.

The PSE algorithm has a convergence rate that is many orders of magnitude faster than other evolutionary methods. This is achieved by propagating the particles approximately along their local paths of steepest ascent without requiring an actual gradient calculation, and thereby combining the best aspects of steepest ascent and evolutionary techniques.

PSE Search Modes

The Andromeda application provides two different search modes for the user to choose from. In both cases, the initial population of Np particles is chosen either randomly, or as an array that uniformly spans the search space, whichever the user specifies. Both modes then proceed to search for global optima, and the only difference lies in the selection rules for choosing the new population of particles that propagate in the next iteration.

Search Mode 1: Most Rapid Convergence

In this mode, after each of the particles in the current population has propagated (thus creating Np new particles), the best Np of those 2Np particles are chosen to be the new population that propagates in the next iteration. This selection rule provides the most rapid convergence possible since only the best particles are always propagated in every iteration. If many roughly equivalent local optima are present, however, this search mode is not as thorough in finding global optima as the second search mode described below.

Search Mode 2: Most Thorough Search

In this mode, after each particle propagates and creates a new particle, the new particle is chosen for the next population if it has improved relative to the previous particle; otherwise, the previous particle is chosen and propagated again in the next population. The result of this selection rule is that the particles will never cease exploring any local optimum once convergence has begun, and therefore this mode provides the most thorough search of local optima that is possible. As one would expect, the convergence is generally slower than the search mode above since we are not always choosing the globally best particles for each new population that propagates.

User Interface: The Problem Definition Worksheet

Andromeda

Global Optimzation of Multiple Functions with Constraints

Functions

F's	Enter Equation	Select Function GOAL	Enter Target Value	Wt.	Name (Units)
F1	1.00E+00	=	1.00E+00	1	Function1 (Units 1)
F2	-7.13E-18	Max	0.00E+00	1	Function2 (Units 2)
F3	-5.74E-49	Min	0.00E+00	1	Function3 (Units 3)
F4				1	
F5				1	

Constraints

C's	Enter Equation	Select Constraint GOAL	Select Constraint Type	Wt.
C1	-4.51E-17	= 0	Soft	1
C2	-3.80E-17	> 0	Hard	1
C3	-8.31E-17	= 0	Hard	1
C4				1
C5				1
C6				1
C7				1
C8				1
C9				1
C10				1

Particle Search Engine Control

Execute | Help Menu

Active	Population (Np) =	10	Max # Runs =	100
3 F's	Initial Search =	Random	Max Run Time (sec) =	120
3 C's	Search Mode =	Thorough	<|ΔX|> / Range =	0.E+00
8 X's	Auto Plotting =	Yes	|ΔF| / Target =	0.E+00

Input Variables

X's	Enter Value	Initial Search Min	Initial Search Max	Bound	Name (Units)
X1	-4.51E-17	-1.00E+00	1.00E+00	Y	Variable X1 (Units X1)
X2	-3.80E-17	-1.00E+00	1.00E+00	Y	Variable X2 (Units X2)
X3	-6.50E-09	-1.00E+00	1.00E+00	Y	Variable X3 (Units X3)
X4	2.01E-08	-1.00E+00	1.00E+00	Y	Variable X4 (Units X4)
X5	-8.98E-10	-1.00E+00	1.00E+00	Y	Variable X5 (Units X5)
X6	1.83E-09	-1.00E+00	1.00E+00	Y	Variable X6 (Units X6)
X7	8.09E-09	-1.00E+00	1.00E+00	Y	Variable X7 (Units X7)
X8	-1.44E-09	-1.00E+00	1.00E+00	Y	Variable X8 (Units X8)
X9				Y	
X10				Y	

Additional Variables

Z's	Value or Equation	Description
Z1		
Z2		
Z3		
Z4		
Z5		
Z6		
Z7		
Z8		
Z9		
Z10		

The PSE algorithm can be applied to optimization problems having any number of Functions (F's), Constraints (C's), and Variables (X's).

Version 1.0 of the Andromeda software in Excel implements the following:
5 F's
10 C's
50 X's

The user interface is highly intuitive and easy to use.

The Functions, Constraints, and Variables are activated and deactivated by clicking on the corresponding symbols.

User Interface: The Results Worksheet

Menu

Andromeda

F1=EXP(-(X1^2)/2)

Functions		Goal	Target	Wt	Best Value	Error
Function1 (Units 1)	F1	Max	1.00E+00	1	1.00000000000000E+00	0.E+00

Input Variables		Min	Max	B	Best Value	± Est Error
Variable X1 (Units X1)	X1	-1.00E+00	1.00E+00	Y	-8.24999147471589E-09	2.E-08

Termination: Specified Function Error Tolerance (Zero Error) Achieved

Runs (Iterations) = 15
Particle Population (Np) = 2
Function Evals = 30
Hard Constraint Evals = 0

Execution Time Components (sec)
Function Evals = 0.03
PSE Algorithm = 0.02
Total RunTime = 0.05

Search Mode
Rapid Convergence
Memory Used
6.58 MB

A novel aspect of the PSE algorithm is that the particle density provides an accurate estimate of the Error in the solution values at every step of the particle evolution.

One of the options provided by the Menu button on the Results worksheet will create a new worksheet summarizing all of the function and corresponding variable values (rank ordered from best to worst) obtained as the particles converged to the best solution. The user can then analyze all of the data in any manner desired. Many graphical analysis tools are included in the Andromeda application, and are described on the following slides.

User Interface: The Graphical Analysis Worksheet

A key benefit of the PSE algorithm is that as the particles converge on solutions, they leave behind a trail, the graphical analysis of which can give the user a great deal of insight into the problem that they're solving. It allows the user to answer questions such as:

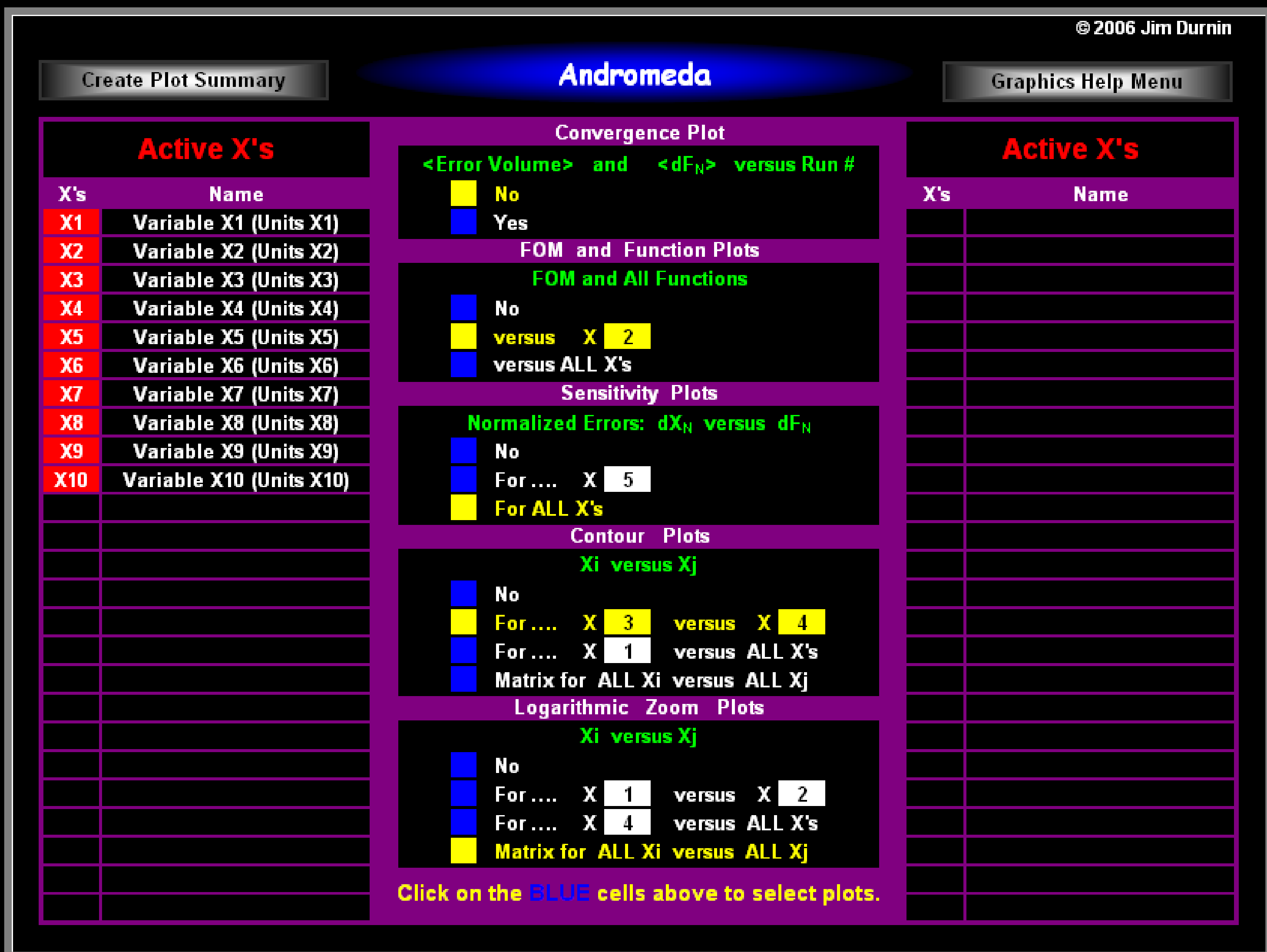
How many potential solutions exist?

How are they distributed throughout the solutions space?

Which variables are the most sensitive?

Andromeda provides a wide assortment of graphical analysis techniques to help the user distill and understand this information.

User Interface: The Graphical Analysis Worksheet



After the user selects all of the different types of plots that are of interest, clicking on the **"Create Plot Summary"** button will generate a new worksheet containing all of the plots requested.

Graphical Analysis: FOM and Function Plots

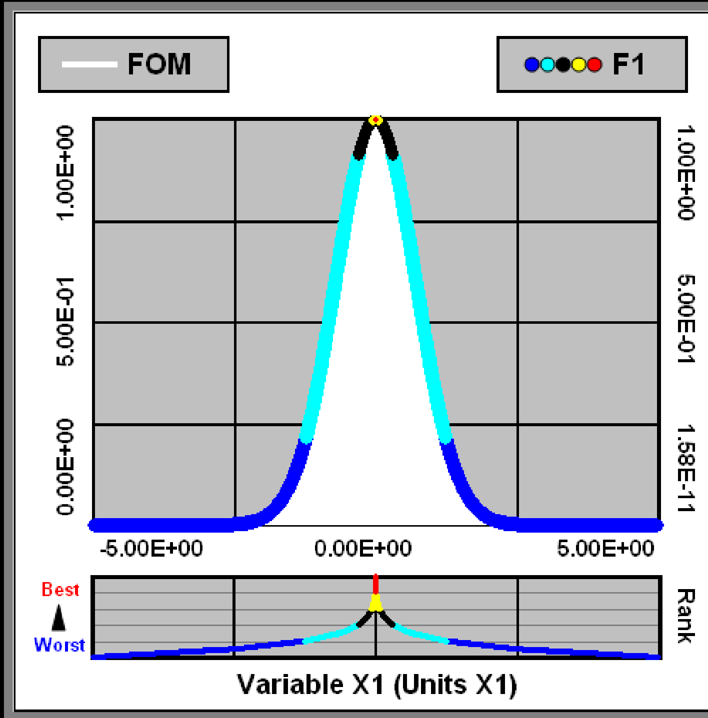
FOM = Figure Of Merit
= 1 at Best solution value
= 0 at Worst solution value
and is plotted in white.

The rank order of the functional values of the particles is indicated by a color code which is used throughout the application:

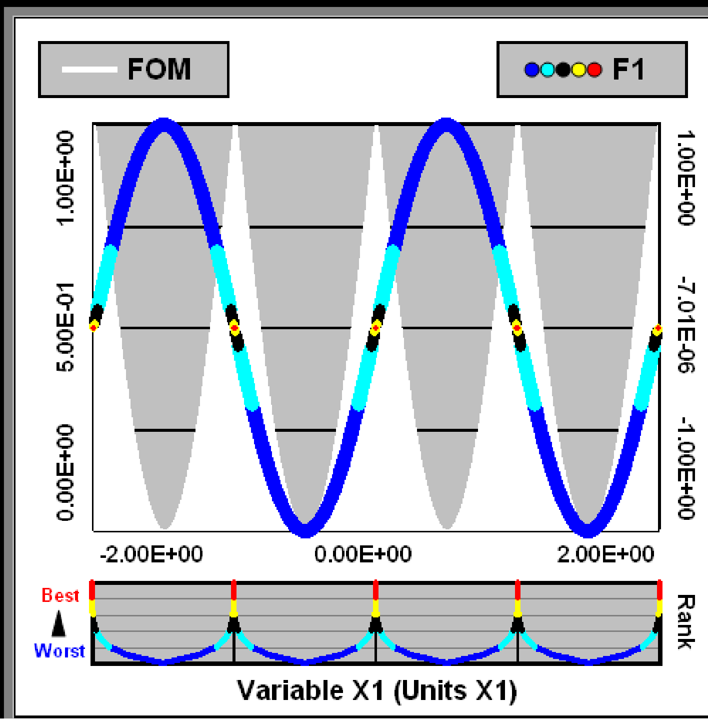
- Best 20% of Particles
- 60-80% range
- Median 20% of Particles
- 20-40% range
- Worst 20% of Particles

The X values are rank ordered from **Best** to **Worst** in the marginal plot located just below the X-axis.

Solve:
F1 = Exp(-(X1^2))
= Max



Solve:
F1 = Sin(πX1)
= 0



Each of the 5 color codes contains exactly the same number of particles (20% of the total), so the plots show that the particles have much higher densities in the solution areas where they are converging.

Graphical Analysis: Contour Plots

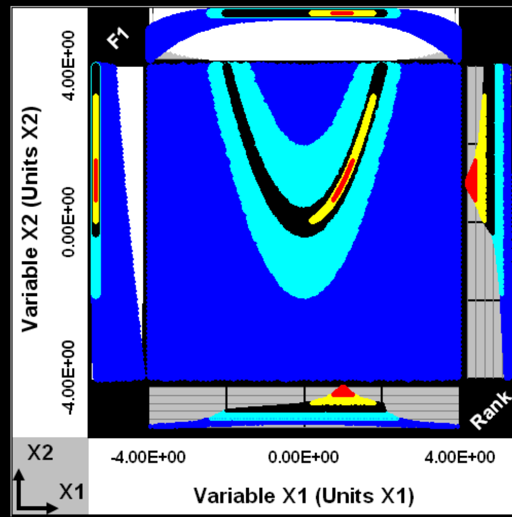
The large plot located at the center of each graph is a Contour plot showing the spatial coordinates of all the color coded particles (using the same color code defined on the previous slide).

The marginal plots along each side of the Contour plot are the corresponding Function and Rank plots for each X (also defined on the previous slide).

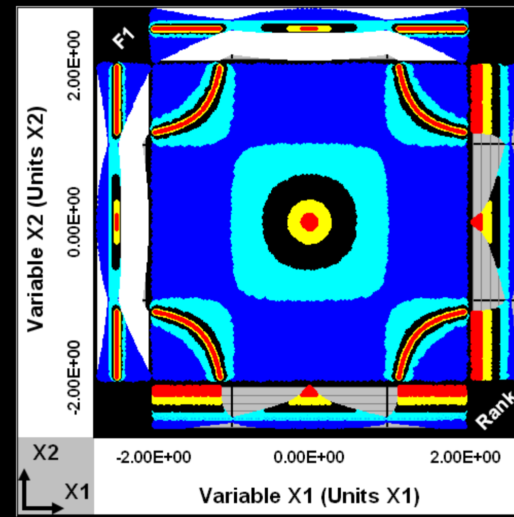
Since each color code contains exactly the same number of particles (20% of the total), the particle distributions exhibit much greater density in the solution areas where they are converging. Contour plotting the distribution of a large number of particles after a short number of runs is an excellent way to gain an understanding of complex functions in a multidimensional space!

(The name of the software – Andromeda – is derived from the fact that a contour plot of the particles converging on a solution once reminded me of a picture I had seen of that galaxy.)

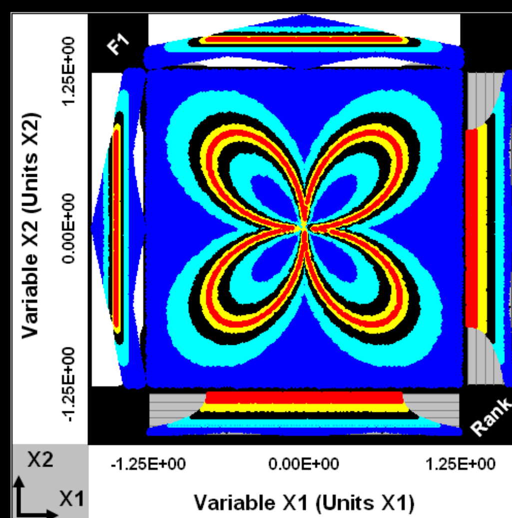
Solve: (Rosenbrock's Function)
F1 = 100(X2-X1^2)^2 + (1-X1)^2
= Min



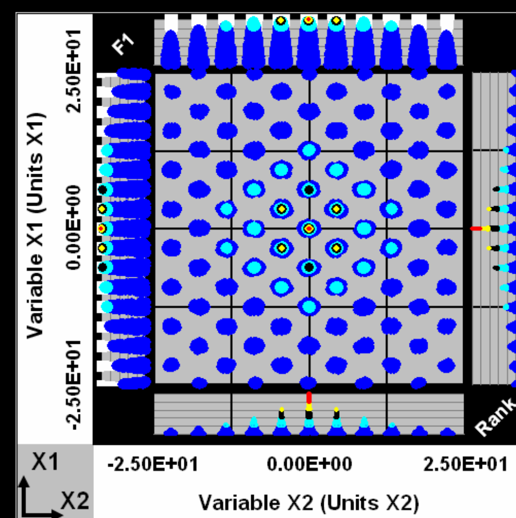
Solve:
F1 = Product(1-X1:X2^2)
= 1.0



Solve: (in Polar Coordinates)
F1 = R – Abs(Sin(2θ))
= 0.0



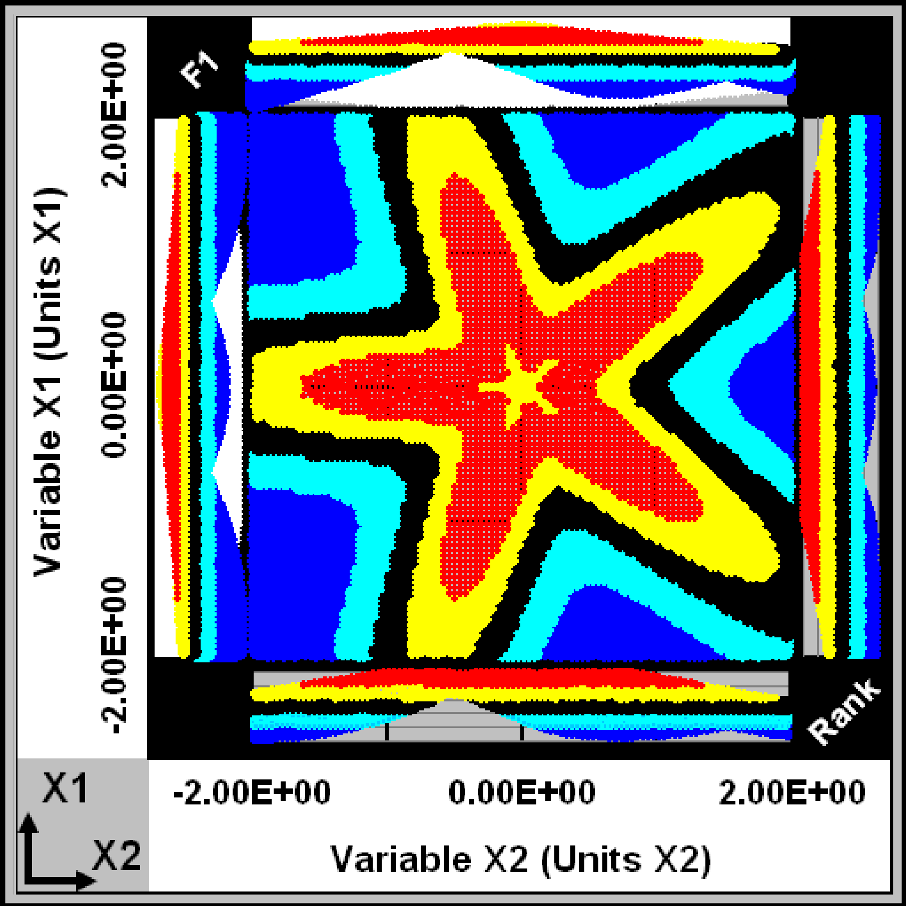
Solve:
F1 = Griewangk's Function (2D)
= Min



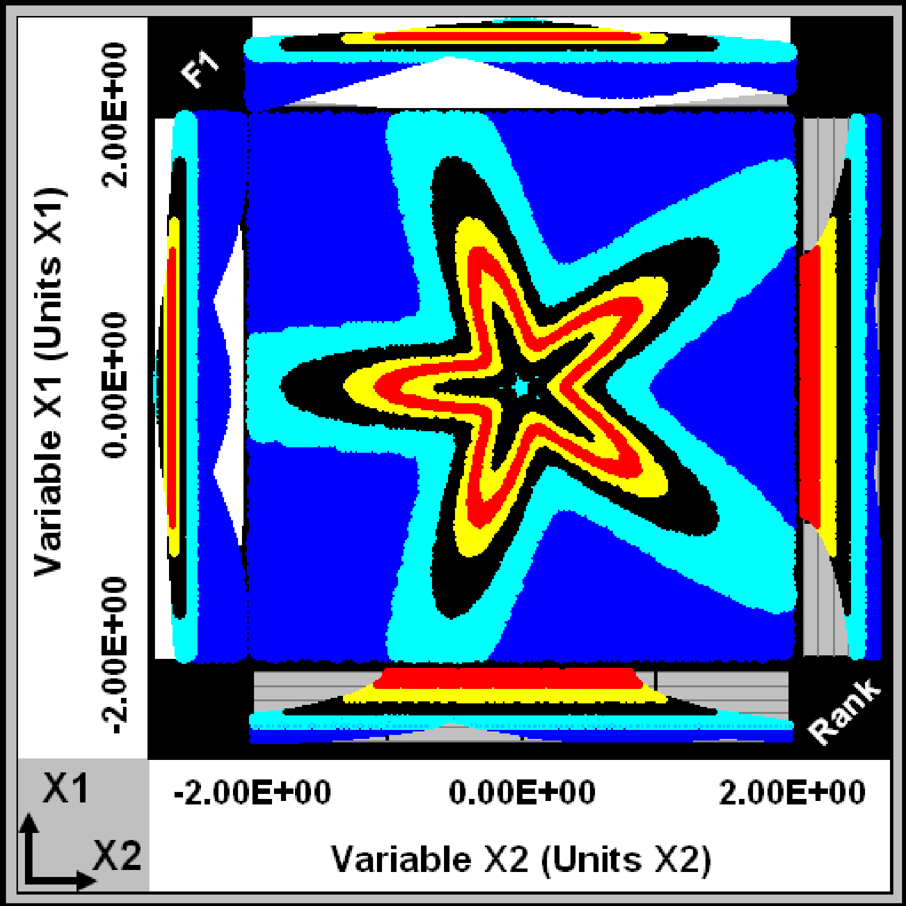
Graphical Analysis: Using Andromeda as a "smart" plotting tool

As the particles converge with each iteration, their density greatly increases in the areas that are of specific interest, and thus the contour plots can also be used as a "smart" plotting tool to gain an understanding of complex functions in multidimensional spaces.

Solve: (in Polar Coordinates)
F1 = R * [1 + 0.5 * Sin(5θ)] = 0.5



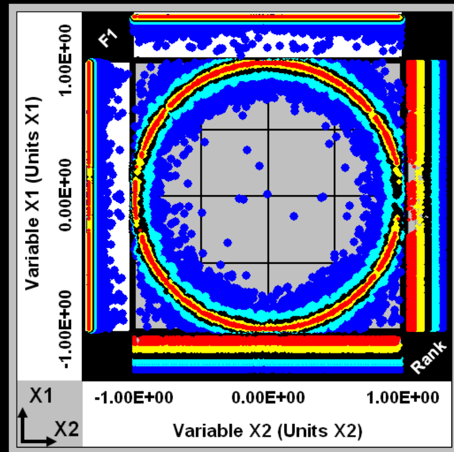
Runs = 1

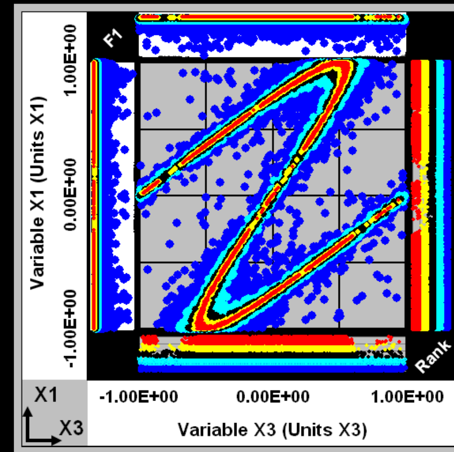


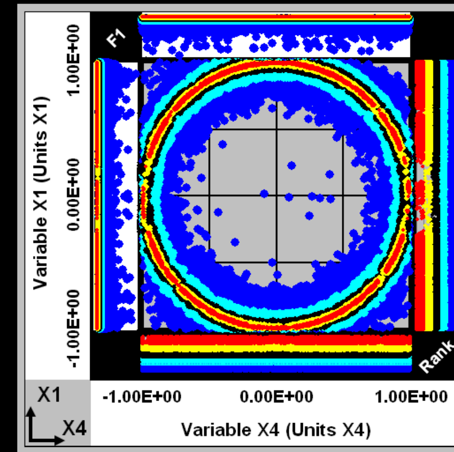
Runs = 5

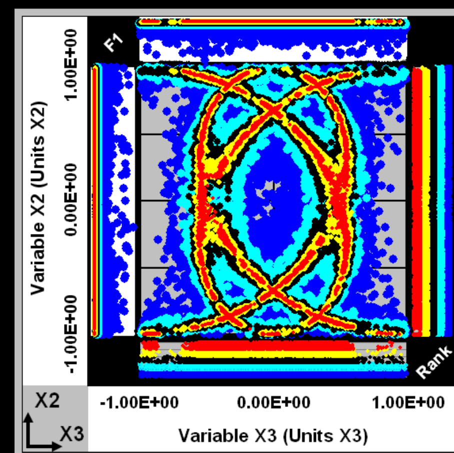
Matrix Plots

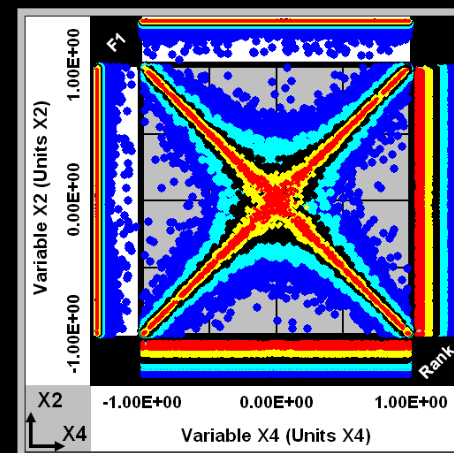
For determining the nature of the solution space.
This is a good first step in solving every optimization problem.



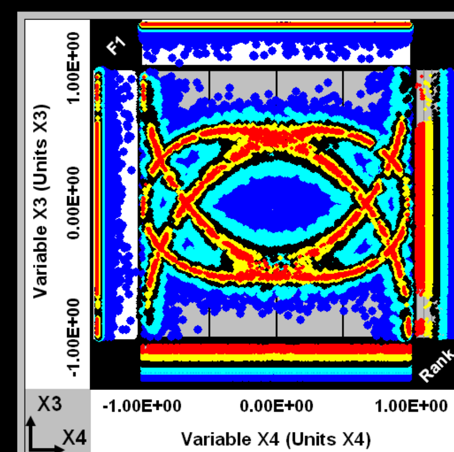








An example of matrix contour plotting the particle positions after only 10 iterations of the PSE algorithm to reveal the detailed structure of the solutions to the 4 dimensional optimization problem shown below.



Solve: F1(X1, X2, X3, X4)

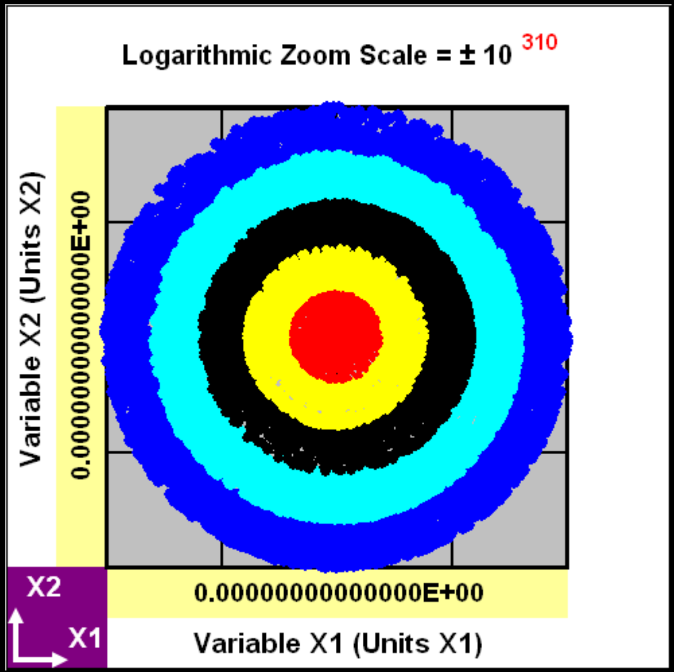
$$= |X1^2 + X2^2 - 1| + |X1 - \sin(\pi(X1-X3))| + |X2^2 - X4^2|$$

= Min

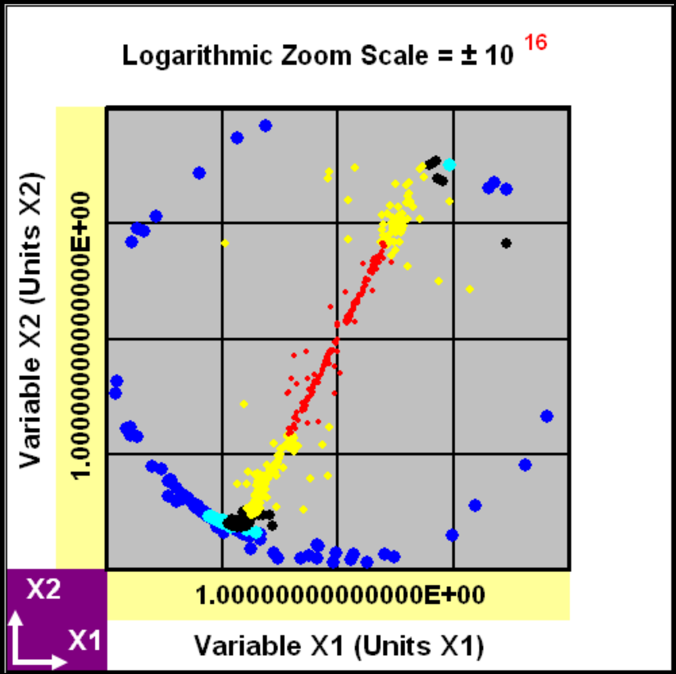
Graphical Analysis: Logarithmic Zoom Plots

The Logarithmic Zoom plot in Andromeda makes use of a logarithmic transformation of the X variables which preserves angle to allow the user to follow the particle distribution all the way down to the best solution through many orders of magnitude in spatial scale. The numerical values highlighted in yellow on each axis are the X values of the best solution, and they define the center of each plot.

Solve:
F1 = Abs(X1) + Abs(X2) = 0.0



Solve: (Rosenbrock's Function)
F1 = 100(X2-X1^2)^2 + (1-X1)^2 = 0.0



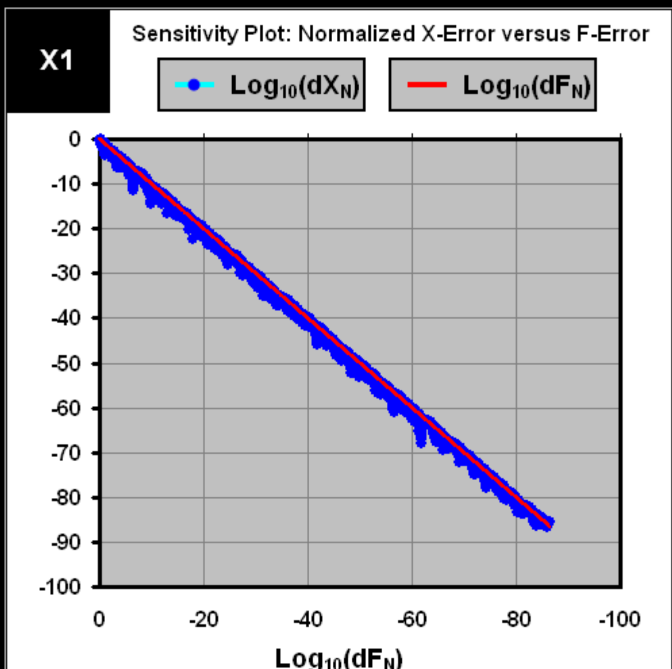
In this first example, the particles converge down through 310 orders of magnitude in spatial scale to the solution (representing the machine value of Zero in Excel).

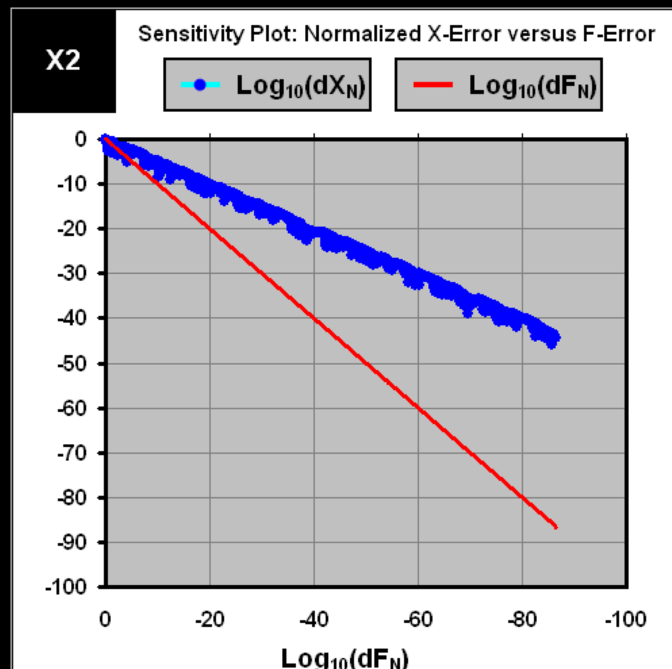
In this second example, the particles converge down through 16 orders of magnitude in spatial scale to the solution (having the 15 decimal place precision of Excel).

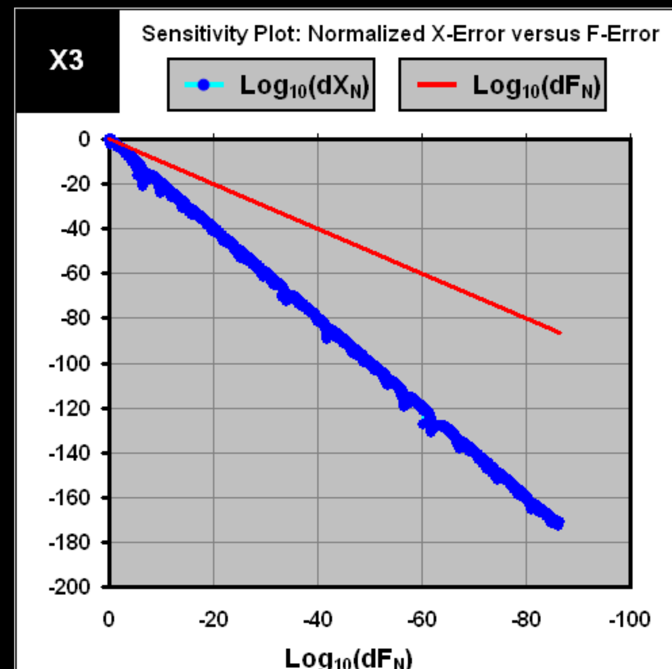
Graphical Analysis: Sensitivity Plots

Sensitivity plots make use of all the data generated during the course of particle convergence to determine the sensitivity of the function to each X variable as shown in the example below.

Solve: F1 = ABS(X1) + ABS(X2)^2 + ABS(X3)^0.5 = 0





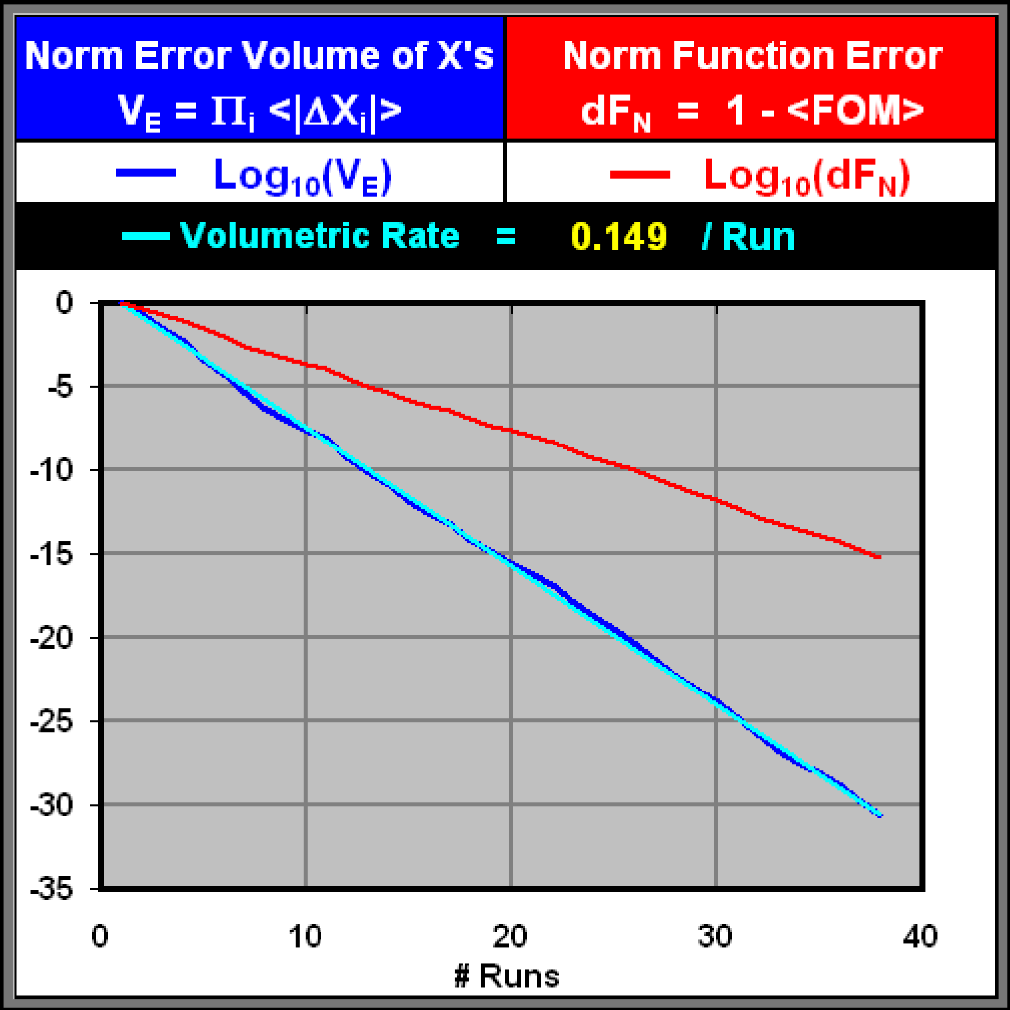


$F1 \propto |X1|$

$F1 \propto |X2|^2$

$F1 \propto |X3|^{1/2}$

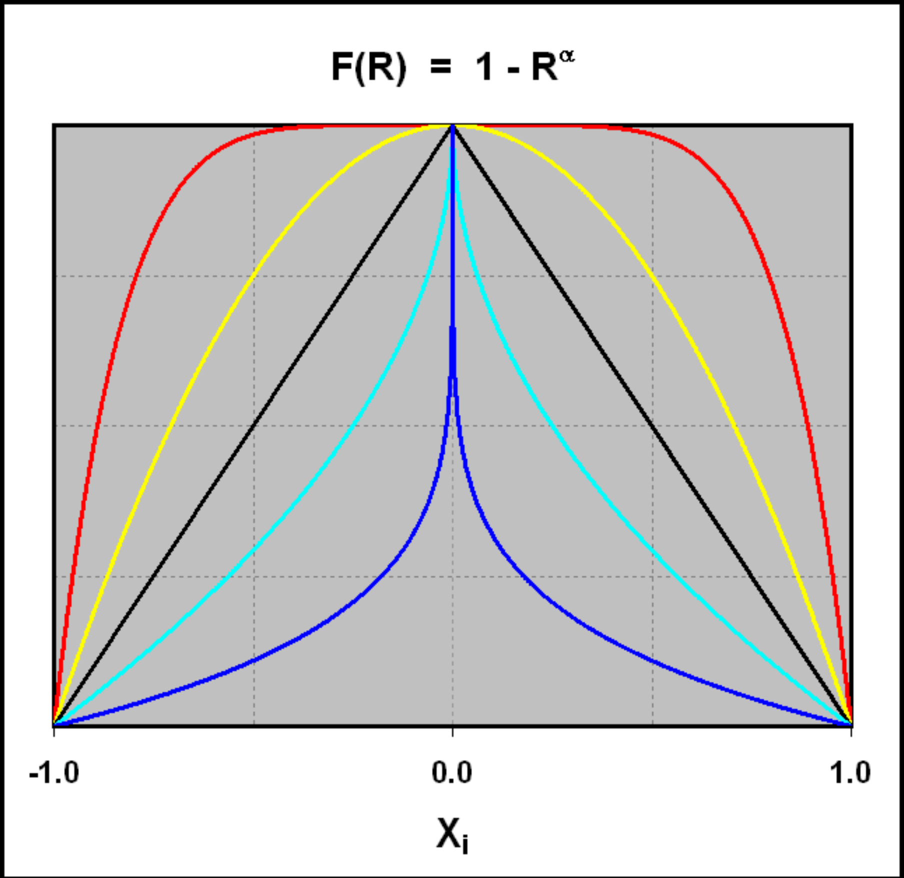
Graphical Analysis: The Convergence Plot



The convergence plot is provided as a tool for those who wish to study the convergence properties of the PSE algorithm. It shows the reduction in both the Error Volume of the X's and the Function Error versus the number of runs (iterations).

The "Volumetric Rate" shown on the plot header is the factor by which the Error Volume is reduced (on average) for each run.

How rapid is the "Rapid Convergence" Search Mode?



For an extremely broad range of problems with unique solutions such as maximizing any of the family of radially symmetric functions shown here,

the convergence properties of the PSE algorithm are easily summarized. The "Volumetric Rate" V_{RATE} (that is, the factor by which the Error Volume of the X's is reduced on average for each iteration) is typically

$$V_{RATE} \approx 0.25$$

and generally lies within the range 0.15 to 0.30, regardless of the number of solution variables N_x

This means that after only 10 iterations of the PSE algorithm, the initial search volume specified by the user will be reduced by $(.25)^{10}$ which is more than a factor of One Million.

The minimum number of function evaluations required in each iteration is the Pod size defined on slide <11>. If we have a function of N_x = 50 variables, for example, the Pod size is 50 and the X-error volume is reduced on average by 75% after every 50 function evaluations!

Relating Solution Precision and Volumetric Rate

If we define the precision in each X solution as

xP = |Error in X| / |Initial Search Range for X|

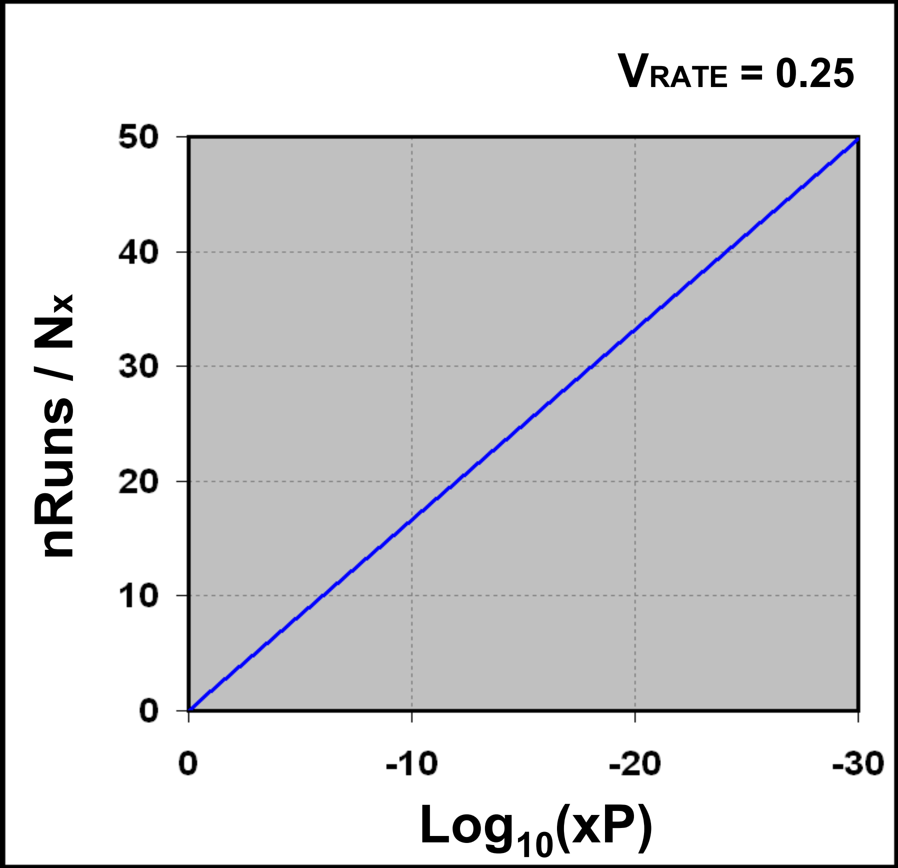
and we define nRuns as the number of iterations required to obtain that degree of precision for each X solution, then:

$$(V_{RATE})^{nRuns} = (xP)^{Nx}$$

The required number of iterations therefore varies linearly with the number of solution variables Nx:

$$nRuns = N_x\,[\,\mathrm{Log}\,(xP) \,/\, \mathrm{Log}\,(V_{RATE})\,]$$

The results for a Volumetric Rate = 0.25 are shown here.



For Nx ≥ 10, there are Nx function evaluations in each iteration, and thus the required number of function evaluations scales in proportion to $(N_x)^2$.

Convergence Rate for Multiple Solutions

In cases where the solution is not unique and we have many local optima, using a sufficiently large population of particles [Np = (# local optima) x (Pod size)] will permit simultaneous convergence at each solution with the same Volumetric Rate as in the case of a unique solution.

Summary

• The PSE algorithm in Andromeda can be used to efficiently solve an extremely broad range of problems in the global optimization of multiple functions with constraints.

- The particles enable the user to do a massively parallel search of the function space, simultaneously converging on multiple local optima until the global optimum dominates.
- Unlike gradient based algorithms for function optimization, the functions can be continuous, or discontinuous, and even noisy.
- Unlike other evolutionary algorithms for finding global optima, the PSE algorithm permits rapid and reliable convergence to EXACT global solutions (within the precision limits of the machine, or to the precision specified by the user).

• The trail left by the particles as they converge to a solution can give the user a great deal of information about the problem that they're solving, and Andromeda provides many novel graphical tools which allow the user to analyze and understand this information.

• The user interface is highly intuitive and easy to use. Anyone who is already familiar with Microsoft Excel can immediately begin solving optimization problems with Andromeda.

For further information, or to obtain an evaluation copy of the Andromeda software, contact:

Jim.Durnin@charter.net

Optimal Results
15822 Freedom Lane
Apple Valley, MN 55124
(952)953-3765