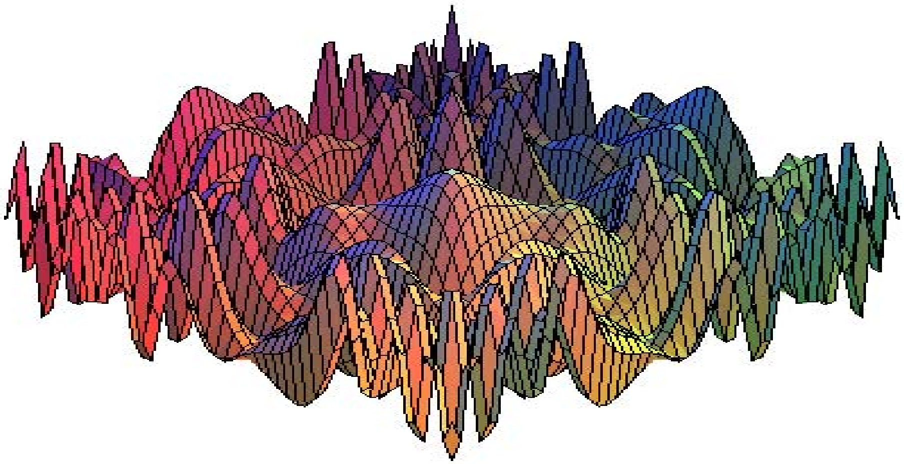
Global Optimization

Models, Algorithms, Software, and Applications



János D. Pintér
PCS Inc., Canada, and
Dept. of Ind. Eng., Bilkent University, Ankara, Turkey

Presented at the eVITA Winter School 2009, Geilo, Norway

Presentation Topics

- The Relevance of Nonlinear and Global Optimization
- General (Continuous) GO Model, and Some Examples
- Review of Exact and Heuristic GO Algorithms
- GO Software Implementations
- Illustrative Applications and Case Studies
- References
- Software Demonstrations (as time allows, or after talk)

Acknowledgements

To all developer partners, clients, and many other interested colleagues for cooperation and feedback

Decisions, Models and Optimization

- Decision making under resource constraints is a key paradigm in strategic planning, design and operations by government and private organizations
- Examples: environmental management; healthcare; industrial design and production; inventory planning; scheduling, transportation and distribution, and many others
- Quantitative decision support systems (DSS) tools – specifically, optimization models and solvers – can effectively assist decision makers and analysts in finding better solutions

A KISS* Model Classification

Convex Deterministic Models

Linear Programming, Convex Nonlinear Programming (including special cases)

Non-Convex Deterministic Models

Continuous Global Optimization, Combinatorial Optimization, Mixed Integer/Continuous Optimization (including special cases)

Stochastic Models

Generic Stochastic Optimization model; special cases that lead to LP, CP, and general NLP equivalents; and to "black box" models

Formally, both the convex and stochastic model-classes can be considered as subsets of the non-convex model class

Combinatorial models can also be formulated as continuous GO models; however, added specifications and insight are helpful

* Keep it Simple, Stupid…

Nonlinear Systems Modeling & Optimization

- As the previous slide already indicates, nonlinear systems are arguably more the norm than the exception…
- Why? Because nonlinearity is found literally everywhere: in processes leading to natural objects, formations, organisms, and in their interactions
- This fact is reflected by descriptive models in applied mathematics, physics, chemistry, biology, engineering, econometrics and finances, and in the social sciences
- Some of the most frequently used elementary nonlinear function forms: polynomials, power functions, exponential, logarithm, and trigonometric functions

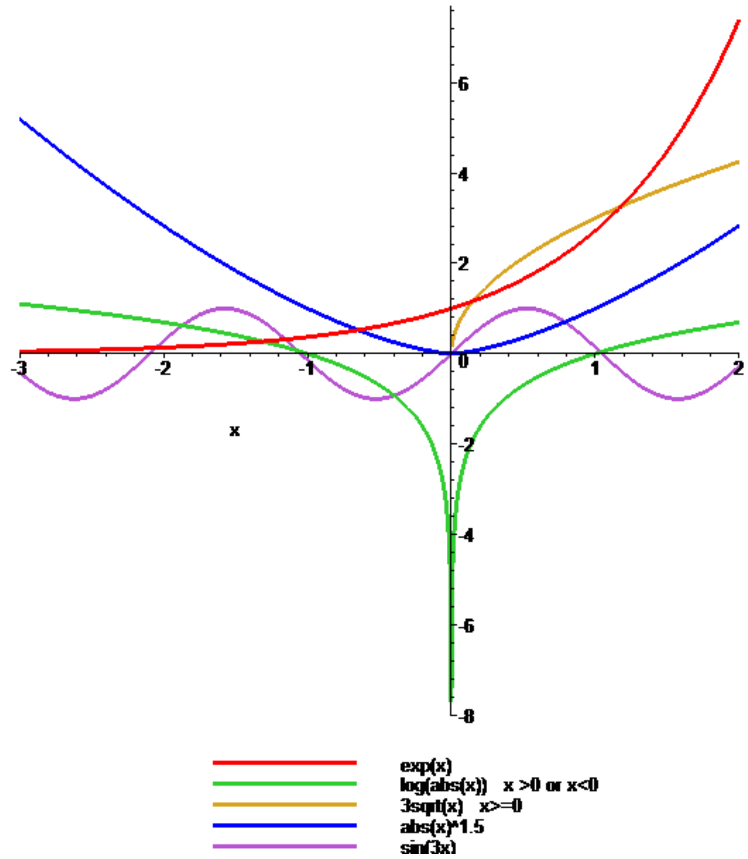
Nonlinear Systems Modeling & Optimization

(continued)

- Composite and more complicated nonlinear functions: special functions, integral equations, linear system of ordinary differential equations, partial differential equations, and so on
- Statistical models: probability distributions, stochastic processes
- “Black box” deterministic or stochastic simulation models, closed (e.g. confidential) models, models with computationally expensive functions,…
- Need for suitable descriptive system models, used in combination with control (optimization) methods

Examples of Basic Nonlinear Functions

A large variety of such functions exists: many of these are used to describe objects, and processes of practical relevance



Nonlinearity in Nature

[A small collection of great photos from the Web]

Nature is clearly the most successful of all artists.

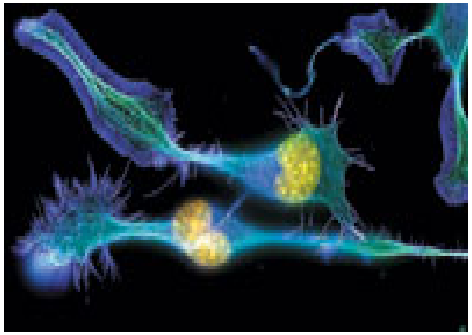
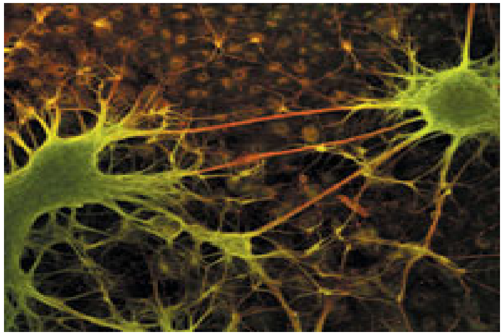
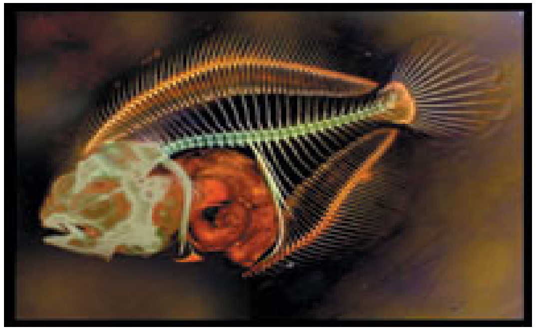
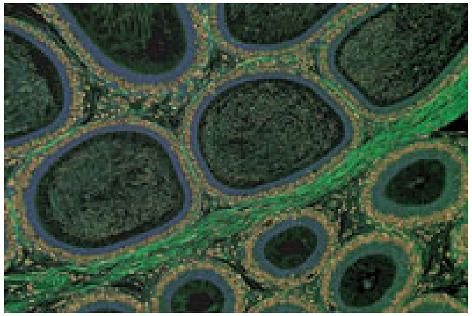
Alvar Aalto, Finnish architect and designer (1898-1976)

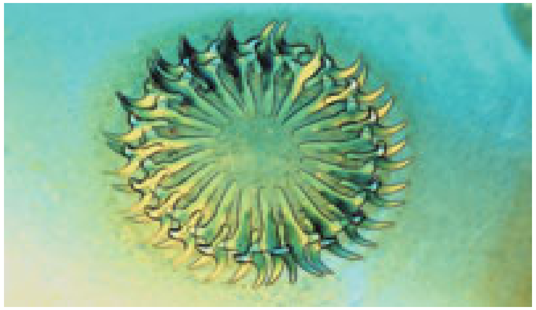
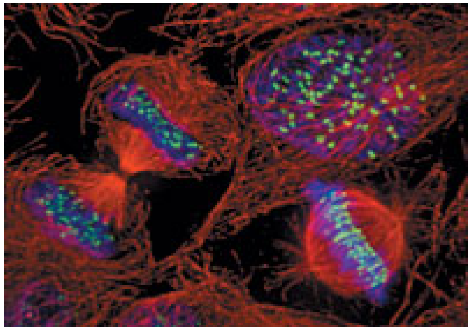
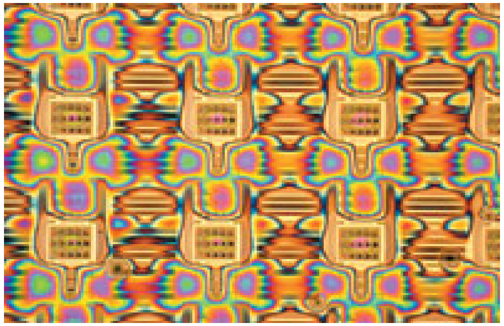
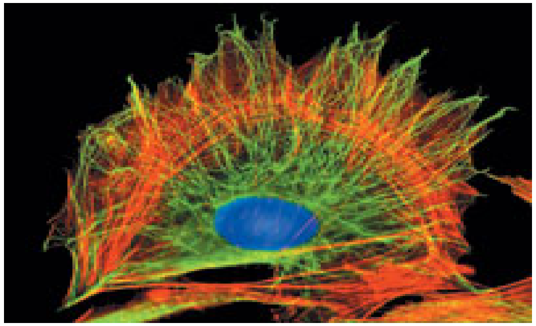
Nonlinearity in Nature

Twisted Vines – Nature's Art, Malaysia, 2006

Nonlinear Universe: Further Examples

Credits: Scientific Computing & Instrumentation, 2004

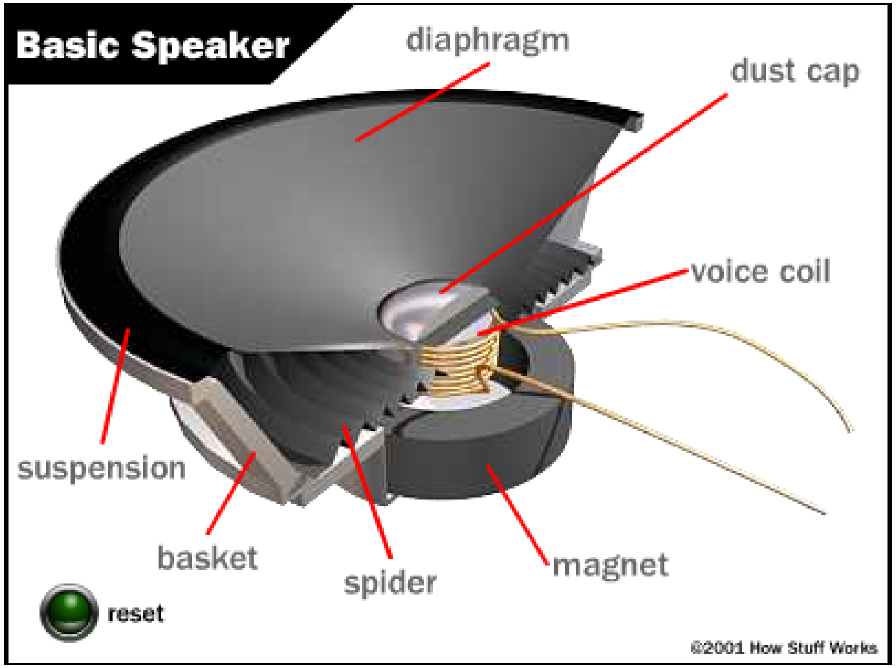





Nonlinearity in Man-Made Systems

Example: Audio Speaker Design

Credits: "How Stuff Works" Website, 2005



Nonlinearity in Man-Made Systems

Example: Automotive Engine Design

Credits: “How Stuff Works” Website & Daimler-Chrysler, 2005

2003 Jeep®
Grand Cherokee

Discovery Spaceship

A Man-Made System with Many Nonlinear Components

Credits: Robert Sullivan, New York Times, 2006

Numerical Algorithms Group

Survey of Technical Computing Users and Environments

The Changing Landscape of Technical Computing

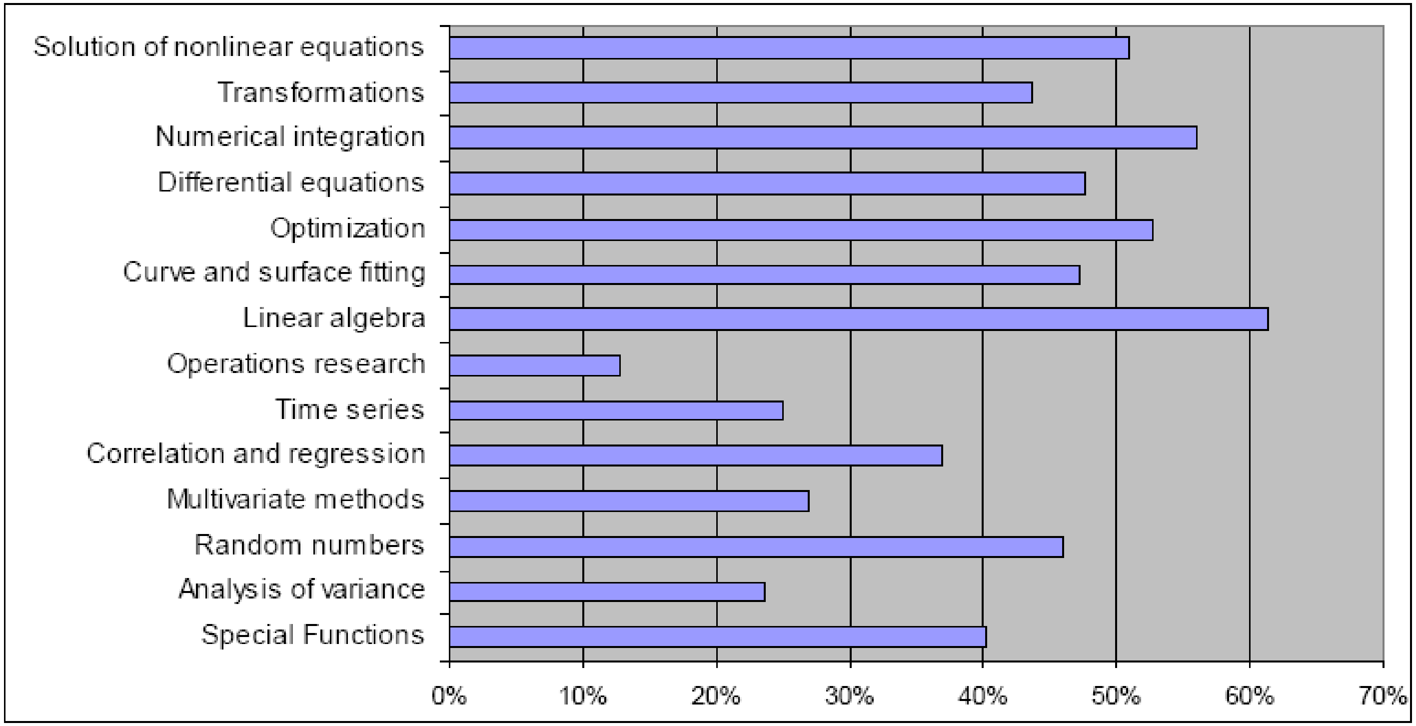
Rob Meyer
Sue Pearson
Katie O'Hare

August 1, 2006

NAG Survey on Technical Computing Needs

Functionality use

Survey participants were asked to report on the use of functionality found in numerical libraries. The following shows the usage of each of the listed functional areas as a percentage of all survey participants. Percentages exceed 100% since most users specified more than one functional area.



Notice that many of these application areas need NLP/GO (software)

The Global Optimization Challenge: Theoretical Motivation

"The great watershed in optimization isn't between linearity and nonlinearity, but between convexity and nonconvexity."

R. Tyrell Rockafellar

Lagrange multipliers and optimality,
***SIAM Review* 35 (1993) 2, 183-238.**

The Relevance of Global Optimization: Practical Motivation

"Theorists interested in optimization have been too willing to accept the legacy of the great eighteenth and nineteenth century mathematicians who painted a clean world of [linear, or convex] quadratic objective functions, ideal constraints and ever present derivatives.

The real world of search is fraught with discontinuities, and vast multi-modal, noisy search spaces..."

D. E. Goldberg
(A well-known genetic algorithms pioneer)

The Relevance of Global Optimization

- Optimization is often based on highly nonlinear descriptive models
- Several important and general model-classes:
  - Provably non-convex models
  - “Black box” systems design and operations
  - Decision-making under uncertainty
  - Dynamic optimization models
- Nonlinear models frequently possess multiple optima: hence, finding their “very best” solution requires a suitable global scope search approach
- The objective of global optimization is to find the absolutely best solution, in the possible presence of a multitude of local sub-optima

Continuous Global Optimization Model

$\min f(x)$ $\quad$ $f\colon R^n \to R^1$

$g(x) \le 0$ $\quad$ $g\colon R^n \to R^m$

$l \le x \le u$ $\quad$ $l, x, u, (l < u)$ are real n-vectors

Key ("minimalist") analytical assumptions:

- l, u are finite vectors; $l \le x \le u$ is interpreted component-wise
- the feasible set $D=\{x_l \le x \le x_u\colon g(x) \le 0\}$ is non-empty
- f and the components of g are continuous functions

Continuous Global Optimization Model

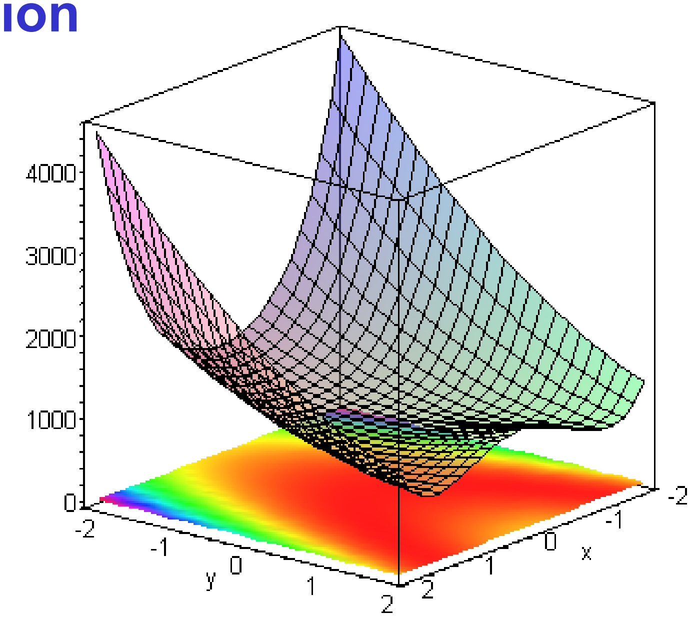
- The structural assumptions stated on the previous slide are sufficient to guarantee the existence of the global solution set X^*; for any x^* in X^*, define $z^* = f(x^*)$
- They also support the application of theoretically exact, globally convergent search methods
- In practice, we wish to find numerical estimates of x^* or X^*, and z^*, using efficient global scope search methods
- The CGO prototype model covers many special cases
- Several examples follow on the next slides that hint at the potential difficulty of GO models

A Mildly Non-Convex Function (a Classic NLP Test)

$$\min_x f(x) = 100(x_2 - x_1^2) + (1 - x_1)^2 + 90(x_4 - x_3^2) + (1 - x_3)^2$$
$$+ 10.1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19.8(x_2 - 1)(x_4 - 1)$$

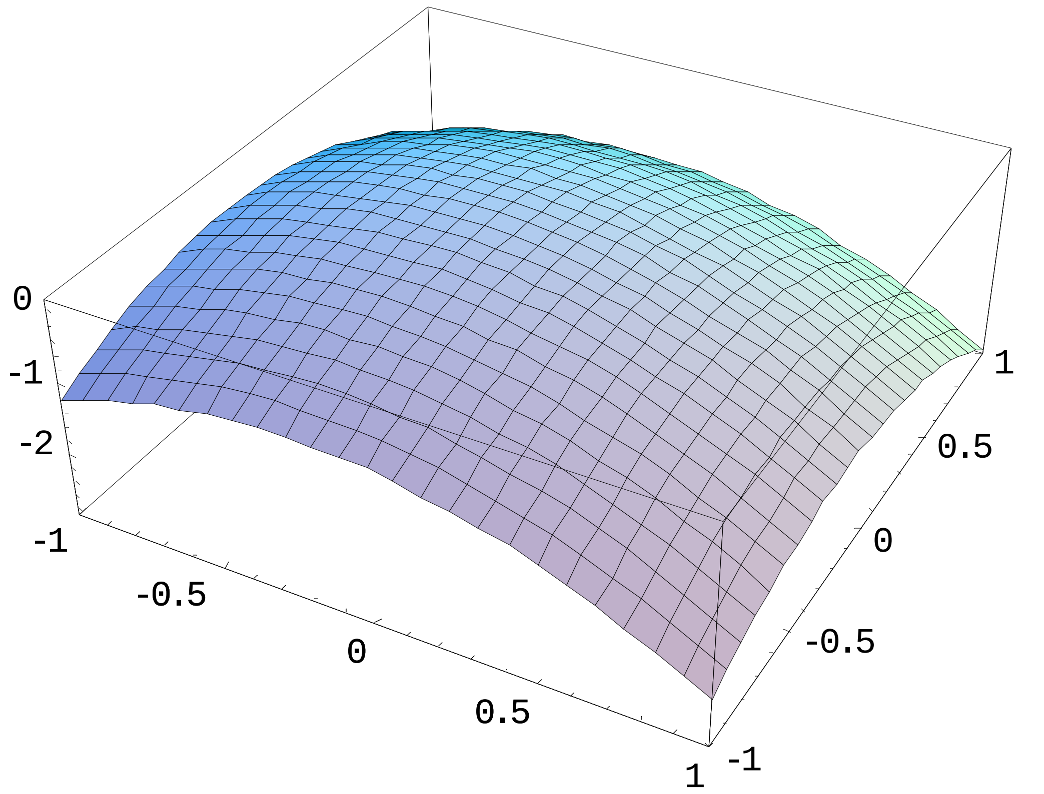
Wood's 4-variable polynomial test function

This function is projected into a 2-dimensional subspace, the other two coordinates are set at their optimal solution value; see figure



Notice that a local search method may end up in either one of two different "valleys", depending on its starting point

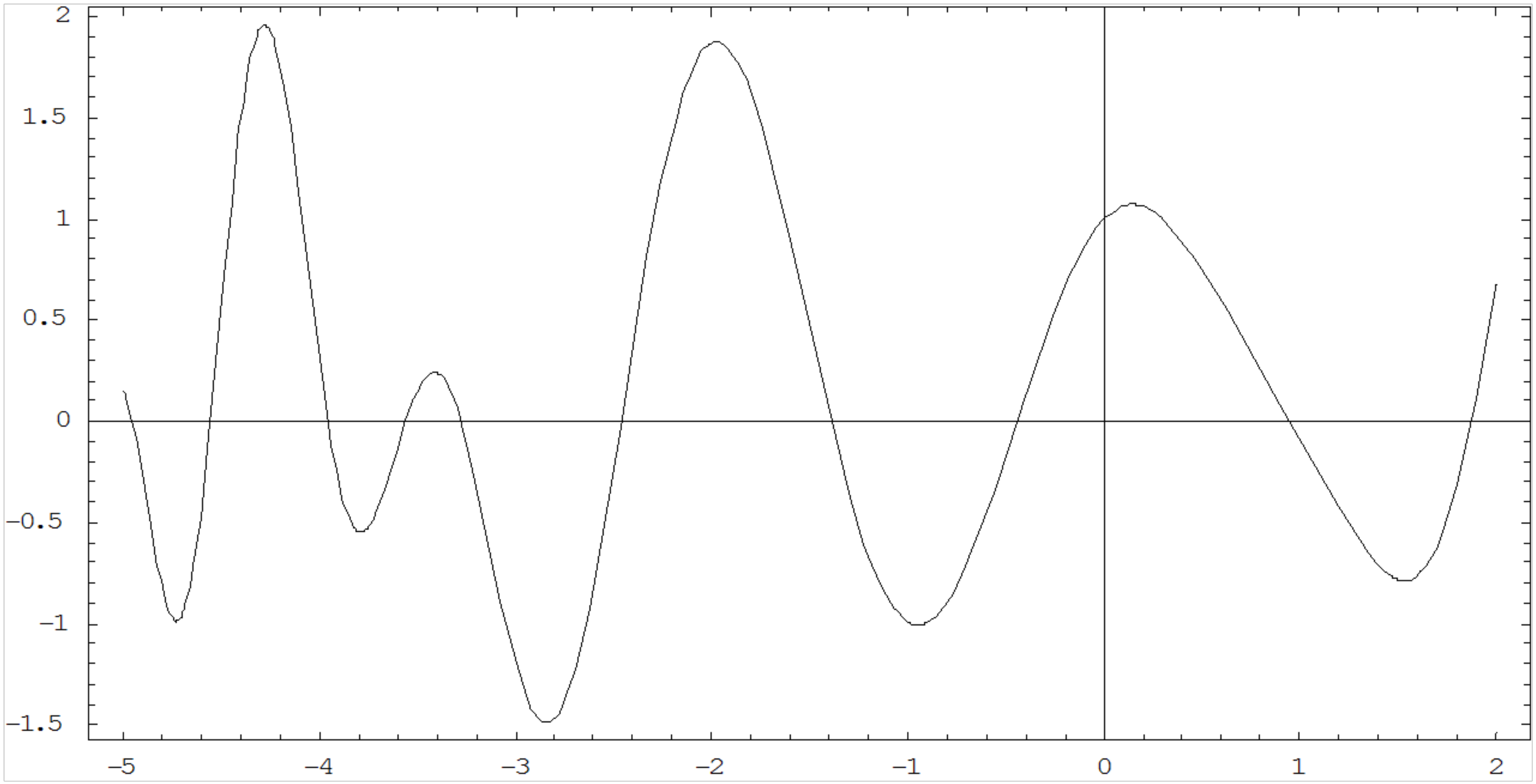
A Concave Minimization Problem



***min f*(*x*)** $x \in D$; $f = -x_1^2 - 0.5 \cdot x_1 - x_2^2 - 0.3 \cdot x_2$ **is concave;** $D = [-1,1]^2$

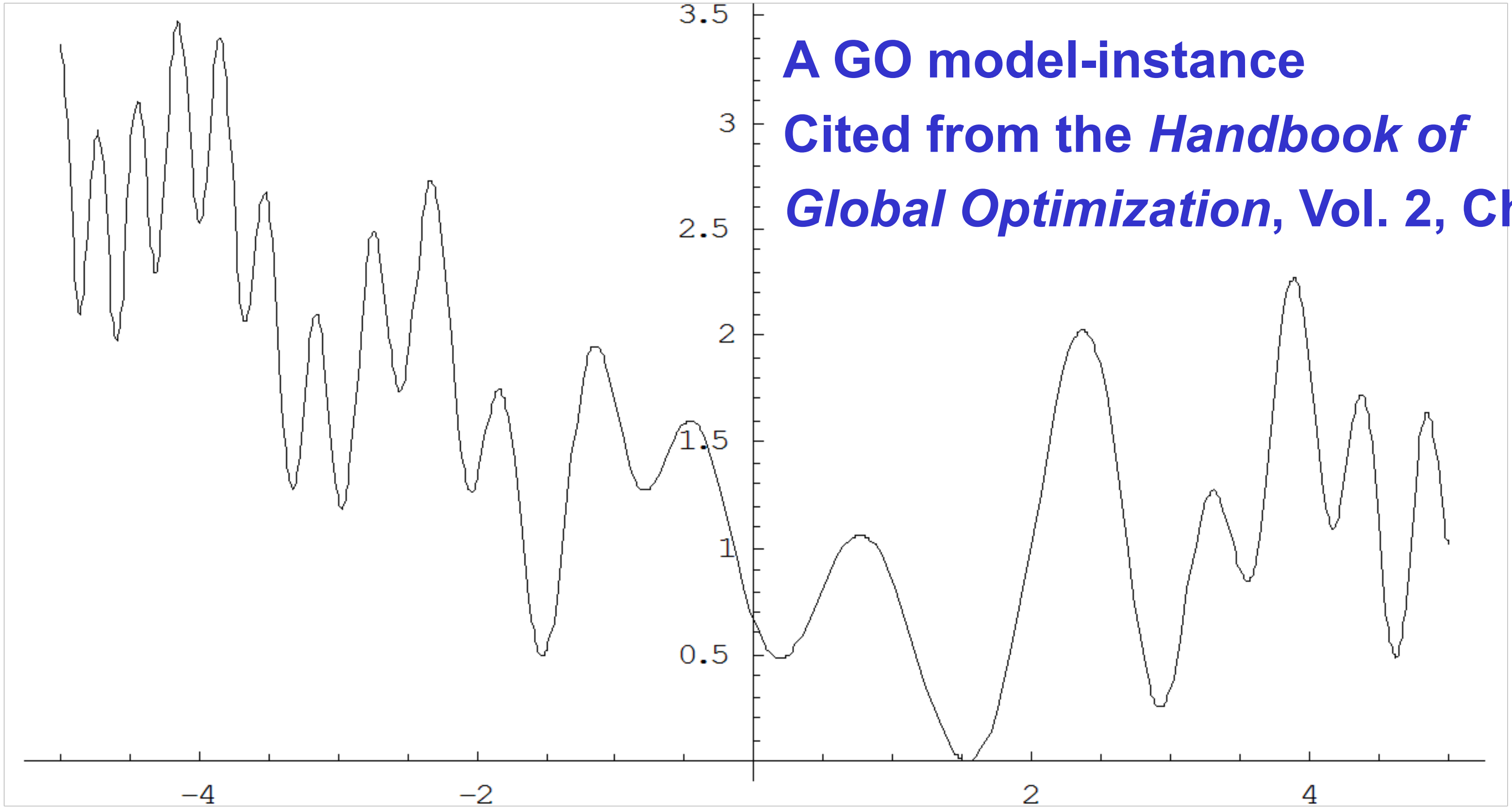
***f* attains its minimum at (1,1), and all vertices of *D* are local minima**

GO Models Can Pose Difficult Challenges For Local Scope Search...



Example: minimize $sin(x^2+x)+cos(3x)$ for $-5 \le x \le 2$
Local search can fail (local information is not sufficient)

GO Models Can Be Even More Difficult (In Principle, Arbitrarily Difficult...)

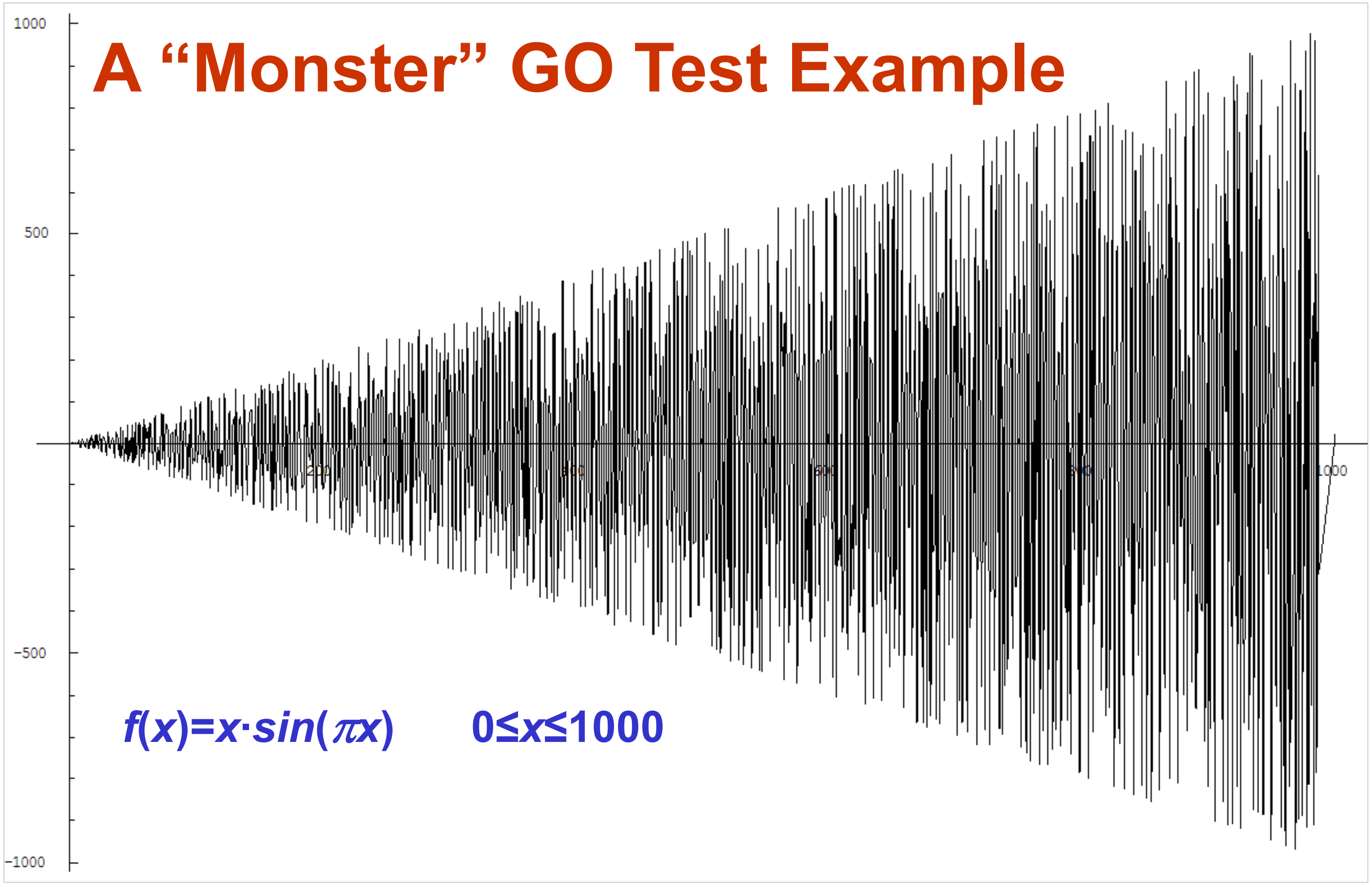


A GO model-instance

Cited from the *Handbook of Global Optimization*, Vol. 2, Ch. 15

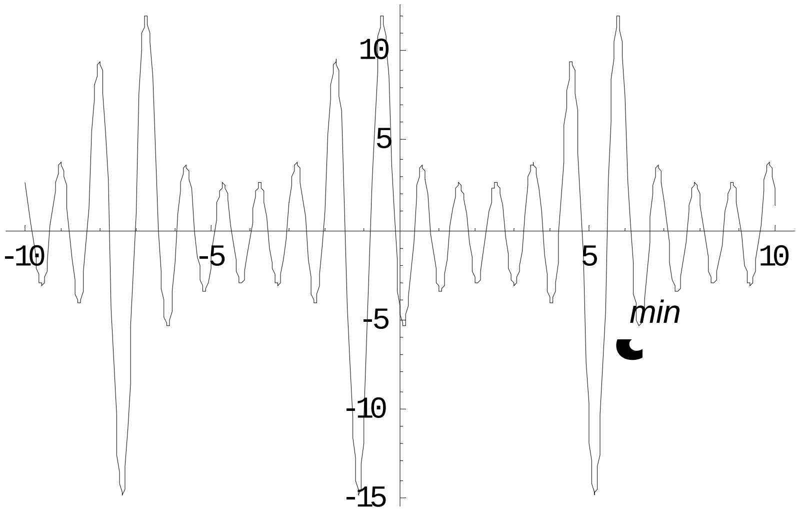
Obviously, a local view of such a function is not sufficient: instead, global scope search is needed

A “Monster” GO Test Example



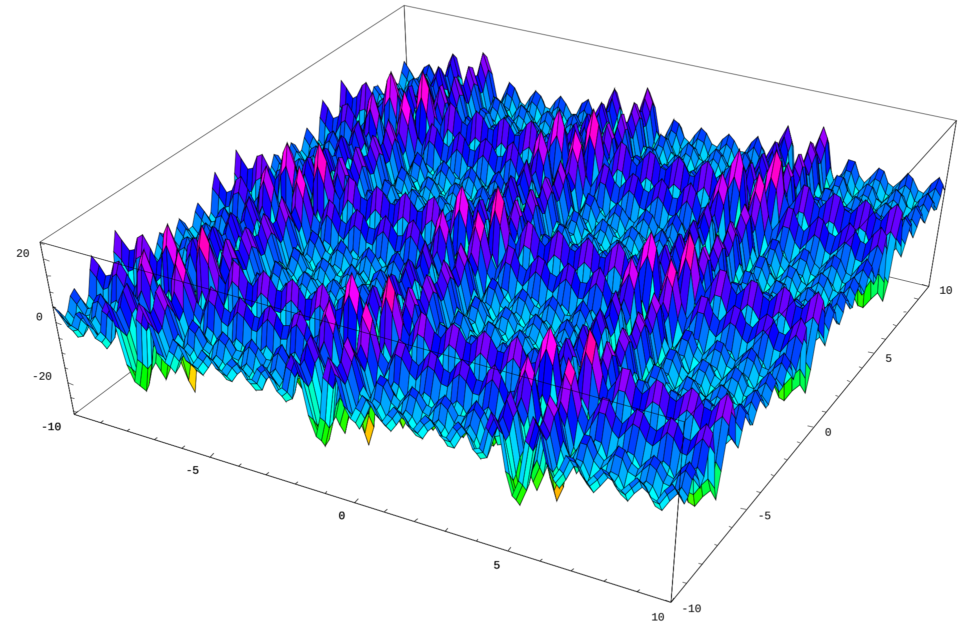
GO models can be extremely difficult to solve, even in (very) low-dimensions, if the search effort is limited… as in prefixed (default) GO solver settings

Another Inherent Issue in GO: "Curse of Dimensionality"



$\sum_{k=1,\ldots,5} k \sin(k+(k+1)x)$

$-10 \le x \le 10.$



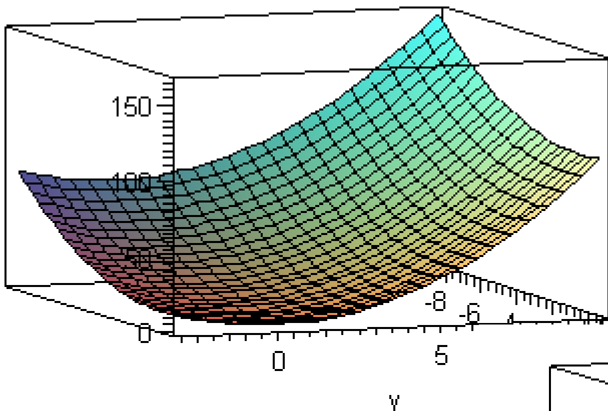
$\sum_{k=1,\ldots,5} k \sin(k+(k+1)x) + \sum_{k=1,\ldots,5} k \sin(k+(k+1)y)$

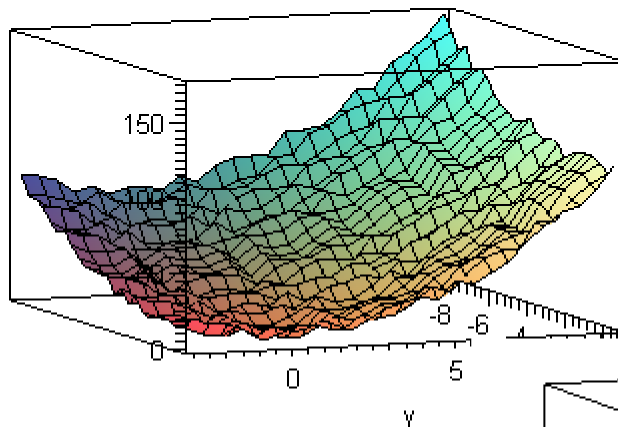
$-10 \le x \le 10,\ -10 \le y \le 10$..

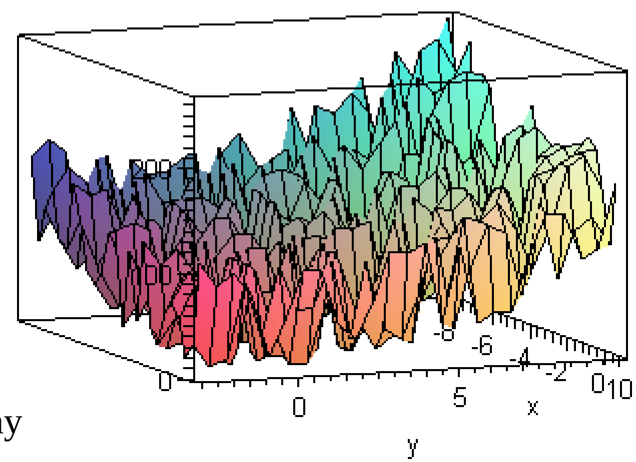
Shubert's one-dimensional box-constrained optimization model, and its simplest two-dimensional extension

Computational complexity increases exponentially, when the model size (n, m) grows

Further Examples: Increasingly More Difficult (Parameterized) Test Functions







Example:

$x^2+y^2+c\cdot\sin^2(x^2+x+y^2-y)$

$x=-8..1$, $y=-3..10$; $c=1, 10, 100$

Note: easy to modify, in order to generate randomized solution points of model instances

A General Class of GO Models Formulated with DC Functions

$f = f_1 – f_2$ if f_1 and f_2 are convex, then f is a DC function (the difference of convex functions)

A similar component-wise structure is postulated for g

DC structure supports the general B&B algorithmic framework (to be discussed later on)

However, a general DC structure is difficult to exploit (in terms of implementable algorithms), except for the case of general quadratic optimization under linear and quadratic constraints

Lipschitz(-Continuous) GO Models

Function f is Lipschitz-continuous on D, if there exists a suitable Lipschitz-constant $L=L(D,f)\geq 0$ such that

$|f(x_1)-f(x_2)| \leq L||x_1-x_2||$ holds for all pairs $x_1,x_2 \in D$

Similar conditions can be postulated for all functions in g

The Lipschitz model-structure allows to generate lower bound estimates of the optimum value, based on an arbitrarily given finite sample set (next slide)

Based upon (mere) continuity and Lipschitz properties, a broad class of globally convergent algorithms can be axiomatically defined, designed and implemented, to solve general GO models – including all examples listed above (except "black boxes", at least in theory)

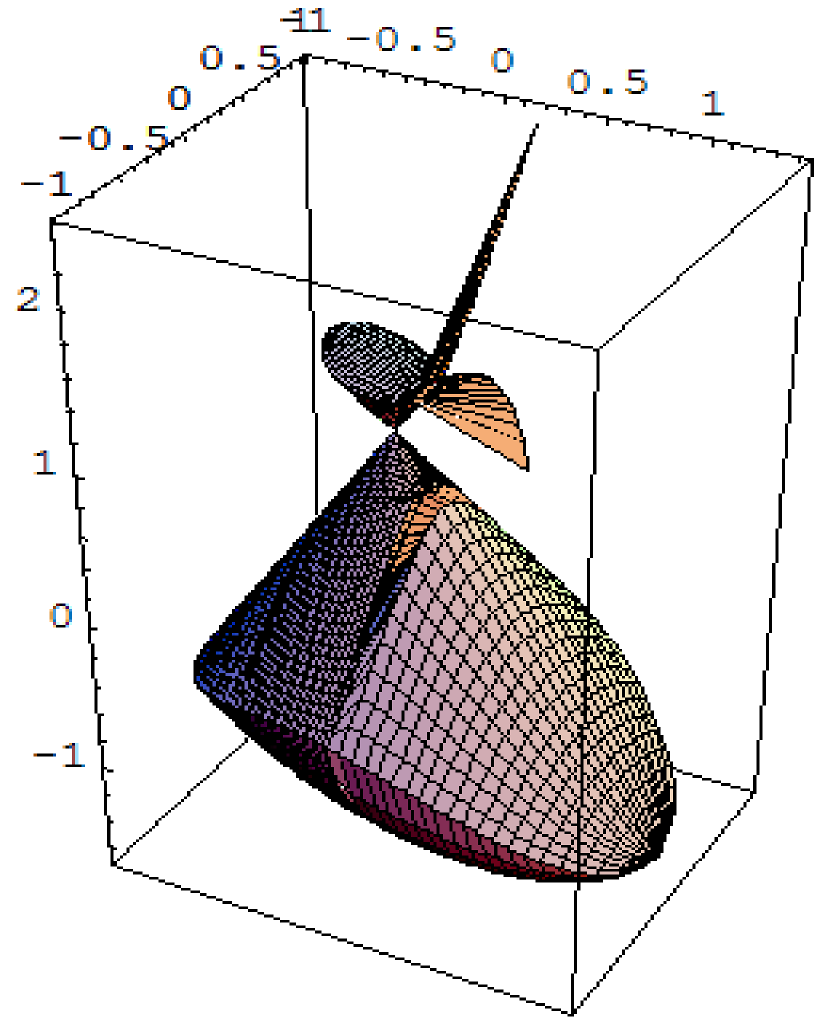
Tricky Feasible Sets Can Be Another Major Source of Difficulty…

Example: Feasible set in R^3 defined by the constraints

$x \cdot y \cdot z \le 1$

$x^2+2y^2+z^2+x \cdot z \le 2$

$3x^2+2y^2-(1-z)^2 \le 0$



Convex programming methods (direction and line searches) may fail on difficult non-convex feasible sets

The Formal Equivalence of Combinatorial and Corresponding Continuous GO Models 1

Each finitely bounded integer variable can be represented by a suitable set of binary variables

Example 1: all integer $0 \leq x \leq 15$ values can be exactly represented by 4 binary variables, since $2^4 -1=15$

For instance, $13 = 1*2^3+1*2^2+0*2^1+1*2^0 = 1101_2$

Example 2: all integer $0 \leq x \leq 10^6$ values can be described by at most 20 binary variables, since $2^{20}>10^6$

Therefore it suffices to use binary variables instead of a given set of finitely bounded integer variables

The Formal Equivalence of Combinatorial and Corresponding Continuous GO Models 2

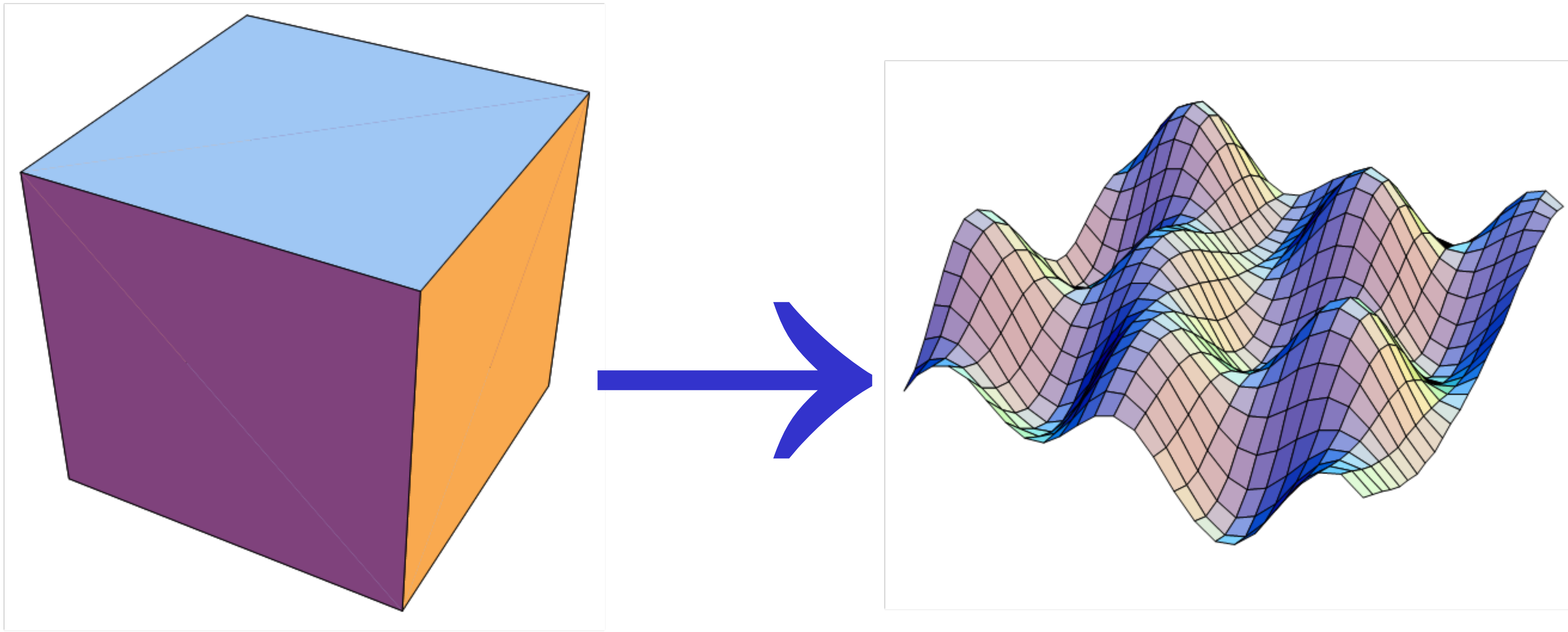
Next, each binary variable can be represented jointly by its continuous extension, and a single non-convex constraint

Example: $x \in \{0,1\}$ (i.e., x is binary) is equivalent to the pair of relations $0 \le x \le 1$ and $x(1-x) \le 0$; other formulations also exist

Therefore formally it suffices to use continuous variables instead of binary ones, and hence also instead of finitely bounded integers; consequently, the same applies also to the most general optimization problems defined with mixed integer-continuous variables and continuous functions

This simple technical note shows a close connection between combinatorial and global optimization, both in terms of their overall complexity, and also regarding the classes of suitable solution strategies for such models

The Mixed Integer Global Optimization Challenge



Each binary variable selection (combination) induces a CGO sub-model

The overall numerical complexity is characterized by the combined complexity of combinatorial optimization and continuous global optimization… Hence, it is massively exponential as the model size characterized by $n=n_B+n_C$ and m grows

Global Optimization Models: Summary

Key model types reviewed:

- **Concave Optimization (Minimization Over a Convex Set)**
- **DC Optimization**
- **Lipschitz Optimization**
- **Continuous Optimization**

These general model-classes cover all GO models of relevance, including also further specific cases

The following chain of set inclusions is valid:

{Concave GO}$\subset${DC GO}$\subset${Lipschitz GO}$\subset${Continuous GO}

Recall also that CGO models cover mixed integer-continuous models

Global Optimization: Historical Perspective

An approximate timeline

Theory beginnings: 1950's, foundations: 1970's

↓↑

Methods beginnings: 1960's, key results: 1980's

↓↑

Software beginnings: 1980's, professional: ~2000+

↓↑

Applications GO needed for a long time, but only recently tackled by suitable GO tools

Ideally, all key components of knowledge are (should be) developed in close interaction

Global Optimization Strategies

- **A significant repertoire of GO methods, including both exact and heuristic approaches, have been suggested since ~1950; systematic studies conducted since ~1970**
- **These solution methods differ with respect to the key analytical conditions of their applicability; and their proof of global convergence properties — or lack of it…**
- **A brief review of some of the GO approaches is provided in the following slides**

Global Optimization Strategies

- **Related general in-depth references are offered by the *Handbook of Global Optimization, Vol. 1* (Horst and Pardalos, Eds., 1995) and *Vol. 2* (Pardalos and Romeijn, Eds., 2002); and in other volumes of the topical Kluwer (now Springer) book series; see also Neumaier’s reviews (2001, 2004)**
- **For simplicity, we shall consider here the box-constrained GO model**

$\min f(x)$

$l \leq x \leq u$

- **Note that the presence of constraints could cause considerable grief to many of the GO approaches discussed below…**

GO Solution Approaches: Exact Methods

Adaptive Stochastic Search Methods

These are procedures based (at least partially) on randomized sampling in the feasible set *D*

Search strategy (parameter) adjustments, sample clustering, deterministic solution refinement options, statistical stopping rules can be added as enhancements to the basic (pure random) sampling scheme

Applicable to both discrete and continuous global optimization problems under general conditions

See e.g., Zhigljavsky (1991), Boender and Romeijn (1995), Pintér (1996), Zabinsky (2003)

Bayesian Search Algorithms

These methods are based on some *a priori* postulated stochastic model of the objective function f

The subsequent adaptive estimation of the problem-instance characteristics is based upon the search (sample points and function values) results, towards building a posterior function (problem) model

Typically, “myopic” (one-step optimal) approximate decisions govern the search procedure, since only these can be implemented

Applicable to continuous GO models, w/o added Lipschitz or other structural assumptions

Consult, e.g., Mockus, Eddy, Mockus, Mockus and Reklaitis (1996), Sergeyev and Strongin (2000)

Branch and Bound Algorithms

Adaptive partition, sampling, and bounding procedures (within subsets of the feasible set *D*) can be applied to continuous GO models, analogously to the well-known integer linear programming methodology

This general approach subsumes many specific cases, and allows for significant flexibility in implementations

Applicable to diverse structured GOPs such as concave minimization, DC programming, and Lipschitz problems

Consult, e.g., Neumaier (1990), Hansen (1992), Ratschek and Rokne (1995), Horst and Tuy (1996), Kearfott (1996), Pintér (1996), Tawarmalani and Sahinidis (2002)

Enumeration Strategies

These are based upon a complete (streamlined) enumeration of all possible global or local solutions

Applicable to combinatorial optimization problems, and to certain structured continuous GOPs such as e.g., concave minimization models

Consult, e.g., Horst and Tuy (1996)

Homotopy and Trajectory Methods

These strategies have the ambitious objective of visiting all stationary points of the objective function *f*, within the set *D*; then checking for minima, maxima, saddle points

This search effort then leads to the list of all - global as well as local - optima (the latter being a subset of the stationary points)

In principle, applicable to smooth GO problems, but the numerical demands can be very substantial

Consult, for instance, Diener (1995) and Forster (1995)

Integral Methods

These methods are aimed at the determination of the essential supremum of the objective function *f* over *D*, by numerically approximating the level sets of *f*

Consult, e.g., Zheng and Zhuang (1995), or Hichert, Hoffmann and Phú (1997)

Naïve Approaches

These include both passive (simultaneous) grid search and passive (pure) random search

Note that these basic (and similar) methods are obviously convergent under mild analytical assumptions, they are truly “hopeless” in solving higher-dimensional problems (already for *n* = 3 or more)

For more details, see for instance Zhigljavsky (1991) or Pintér (1996), with further references therein

Relaxation (Outer Approximation) Strategies

In this general approach, the GOP is replaced by a sequence of relaxed sub-problems that are easier to solve

Successive refinement of sub-problems to approximate the initial problem is applied: cutting planes and more general cuts, diverse minorant function constructions, and other customizations are possible

Applicable to diverse structured GO models such as concave minimization, or DC programming

See, e.g., Horst and Tuy (1996), or Benson (1995)

GO Solution Approaches: Heuristic Methods

These often offer a "plausible" approach to handle difficult models, but without any theoretical justification (global convergence guarantee)

Approximate Convex Underestimation

This strategy attempts to estimate the (possible large-scale, overall) convexity characteristics of the objective function based on directed sampling in *D*

Applicable to smooth GO problems

See Dill, Phillips and Rosen (1997), and some related studies in classical (local) optimization studies

Continuation Methods

These approaches first transform the objective function into some more smooth, simpler function with fewer local minimizers, and then use a local minimization procedure to (hopefully) trace all minimizers back to the original function

Applicable to smooth GO problems

Genetic Algorithms, Evolution Strategies

These "adaptive population" based heuristic approaches mimic biological and social evolution models (including e.g. ant colonies, memetic and other algorithmic approaches)

Various deterministic and stochastic algorithms can be constructed, based on diverse "evolutionary" rules

These strategies are applicable to both discrete and continuous GO problems under mild structural requirements; typically, customization is needed

Consult, e.g., Michalewicz (1996), Osman and Kelly (1996), Glover and Laguna (1997), or Voss, Martello, Osman and Roucairol (1999)

A General Framework for Population-Based Strategies

Initial population of sample points

Iteration cycle steps:

- **Competitive selection, drop the poorest solutions**
- **The remaining pool of points with higher fitness value can be recombined with other solutions, by swapping components with another**
- **The active points can also be mutated by making some (stochastic) change to a current point**
- **Recombination and mutation moves are applied sequentially, in each major iteration cycle**
- **Check algorithm stopping criteria: stop, or return to execute next major iteration cycle**

Sequential Improvement of Local Optima

These approaches — including tunneling, deflation, and filled function methods — operate on adaptively defined auxiliary functions, to assist the search for improving optima

Applicable to smooth GO problems

Consult, for instance, Levy and Gomez (1985), and their many followers (Ge Renpu and others)

Simple Globalized Extensions of Local Search Methods

These "pragmatic" strategies are often based on a rather quick global search (e.g. a limited passive grid or random search) phase, followed by local scope search

Applicable to smooth GO problems: differentiability is typically postulated (only), to guarantee the convergence of the local search component

However, global convergence is guaranteed only by the global scope search phase (which could be inefficient in a rudimentary implementation)

Consult, for instance, Zhigljavsky (1991) or Pintér (1996)

Simulated Annealing

SA is based upon the physical analogy of cooling crystal structures that spontaneously arrive at a stabilized configuration, characterized by (globally or locally) minimal potential energy

Applicable to both discrete and continuous GOPs under mild structural requirements

See, for instance, Osman and Kelly (1996), or Glover and Laguna (1997)

Tabu Search

The essential idea of this meta-heuristics is to forbid search moves towards points already visited in the (usually discrete) search space, within the next few steps, as governed by the algorithm

Tabu search has been mainly used so far to solve combinatorial optimization problems, but it can also be extended to handle continuous GOPs

Consult, e.g., Osman and Kelly (1996), Glover and Laguna (1997), or Voss, Martello, Osman and Roucairol (1999)

GO Solution Approaches: Concluding Notes

Observe that overlaps may (in fact, do) exist among the algorithm categories listed above

Both exact and heuristic methods could suffer from drawbacks: “overly sophisticated for practice” vs. “simplistic” approaches and their implementations

Search strategy combinations are often both desirable and possible: this, however, leads to non-trivial issues in algorithm design

Global Optimization Software Development

"Those who say it cannot be done should not interrupt those who are busy doing it."

Chinese proverb

"It does not matter whether a cat is black or white, as long as it catches mice."

Deng Xiaoping

"I don't want it perfect, I want it Tuesday."

J.P. Morgan

GO Software Development Environments

- General purpose, “low level” programming languages: C, Fortran, Pascal, … and their modern extensions
- Business analysis and modeling: Excel and its various extensions and add-ons (Excel PSP, @RISK,…)
- Specialized algebraic modeling languages with a focus on optimization: AIMMS, AMPL, GAMS, LINGO, LPL, MPL,…
- Integrated scientific and technical computing systems: Maple, Mathematica, MATLAB,…
- Relative pros and cons: instead of a “dogmatic” approach, one should choose the most appropriate platform considering user needs and requirements

GO Software: State-of-Art in a Nutshell 1

- Websites (e.g., by Fourer, Mittelmann and Spellucci, Neumaier, NEOS, and others) list discuss research and commercial codes: examples of the latter listed below

- Excel Premium Solver Platform: Evolutionary, Interval, MS-GRG, MS-KNITRO, MS-SQP, OptQuest solver engines

- Modeling languages and related solver options

AIMMS: BARON, LGO

AMPL: LGO

GAMS: BARON, DICOPT, LGO, OQNLP

LINGO: built-in global solver by the developers; also in What'sBest! for spreadsheets

MPL: LGO

GO Software: State-of-Art in a Nutshell 2

- Integrated scientific-technical computing environments

Maple: Global Optimization Toolbox (LGO for Maple)

Mathematica: Global Optimization (package),

MathOptimizer, MathOptimizer Professional (LGO for Mathematica), NMinimize

Matlab: GADS Toolbox

TOMLAB solvers for MATLAB: CGO, LGO, OQNLP

Detailed information and references:

- Developer websites
- Handbook of GO, Vol. 2, Chapter 15
- Neumaier's GO website

LGO (Lipschitz Global Optimizer) Solver Suite: Summary of Key Features

- LGO is introduced here as an example of GO software
- LGO offers a suite of global and local nonlinear optimization algorithms, in an integrated framework
- Globally search methods (solver options):

  continuous branch-and-bound

  adaptive random search (single-start)

  adaptive random search (multi-start)

  exact penalty function applied in global search phase
- Local optimization follows from the best global search based point(s), or from a user-supplied initial point, by the generalized reduced gradient method

LGO: Summary of Key Features (continued)

- LGO can analyze and solve complex nonlinear models, under minimal analytical assumptions
- Computable values of continuous or Lipschitz model functions are needed only, without higher order information
- Hence, LGO can be applied also to completely “black box” system models, defined by continuous functions
- Tractable model sizes depend only on hardware and time… however, the inherent massive complexity of GO problems remains a challenge (for all GO software products)

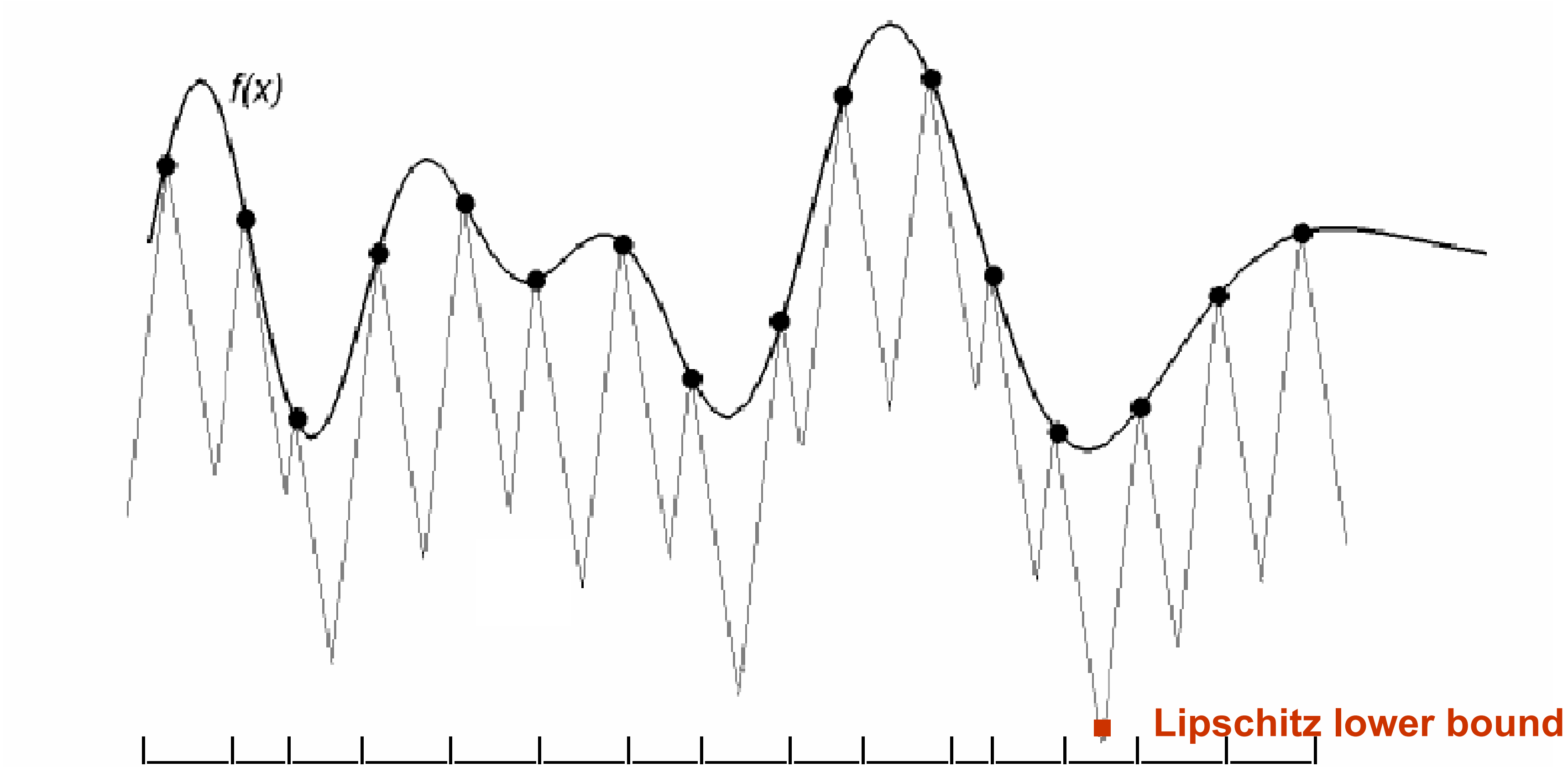
LGO: Summary of Key Features (continued)

- LGO reviews in *ORMS Today*, *Optimization Methods and Software*; various other LGO implementations reviewed in *ORMS Today*, *Scientific Computing*, *Scientific Computing World*, *IEEE Control Systems Magazine*, *Int. J. of Modeling, Identification and Control*, and in *AlgOR*
- LGO is currently available to use with C/C++/C# and Fortran compilers; with links to AIMMS, AMPL, GAMS, Excel and MPL; and with links to Maple, Mathematica, and Matlab
- MPL/LGO demo accompanies Hillier & Lieberman OR textbook (from 8th edition, 2005)
- LGO demos for C/C#/Excel/Fortran,… are available upon request

LGO Solver Suite: Technical Background Notes

- LGO offers a suite of global and local nonlinear optimization algorithms, in an integrated framework
- This approach is dictated by the demands of (many, although not all) GO software users who need to solve their optimization problems relatively quickly
- The global search components are (theoretically) globally convergent, either deterministically, or stochastically (with probability 1)
- The local search component aims at finding KKT points that satisfy the necessary local optimality conditions
- This flexible combination of strategies leads to global and local search based numerical solutions

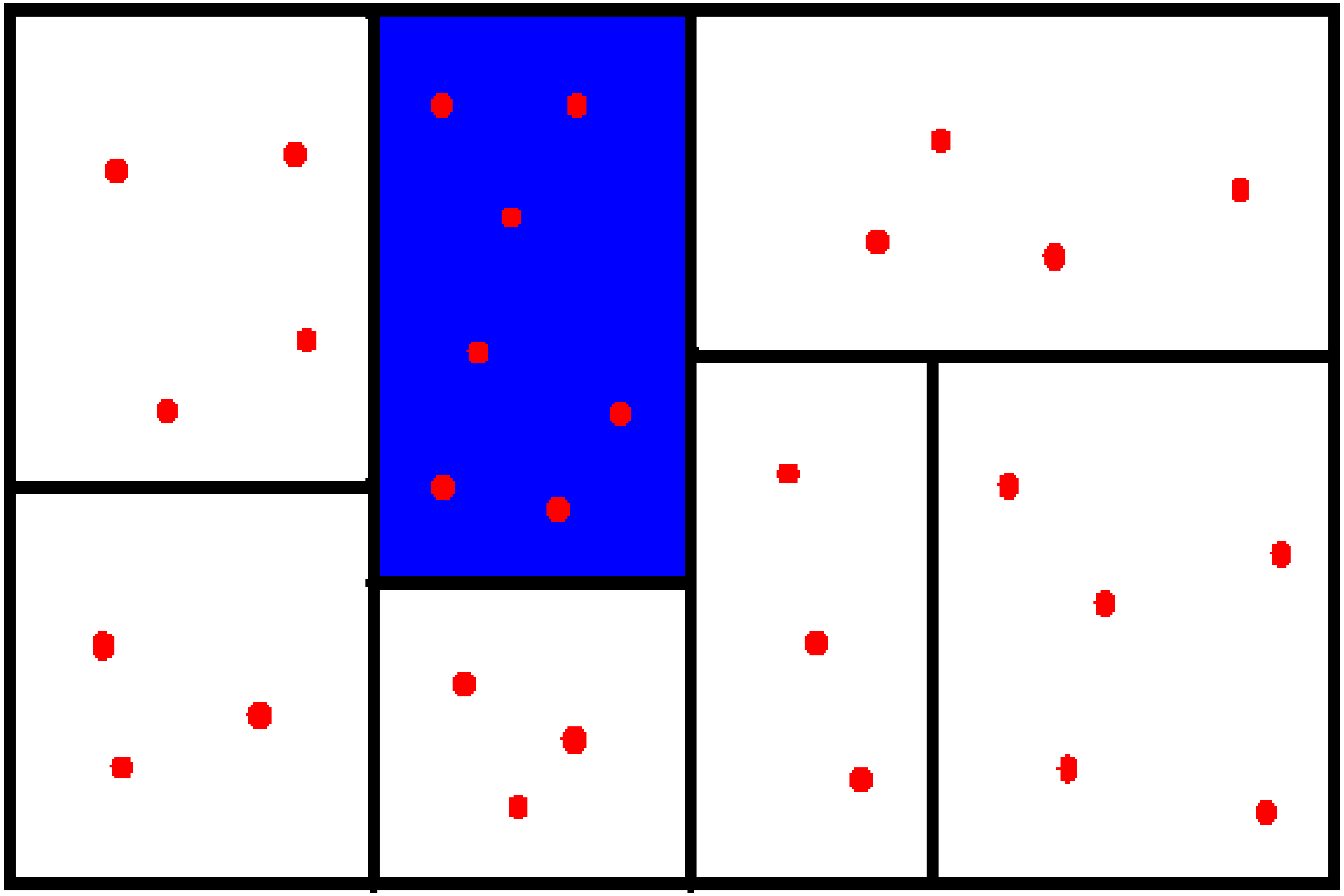
Lipschitz Function and its Minorant: An Example



Search interval: the function values at the sample points | are shown above by dots

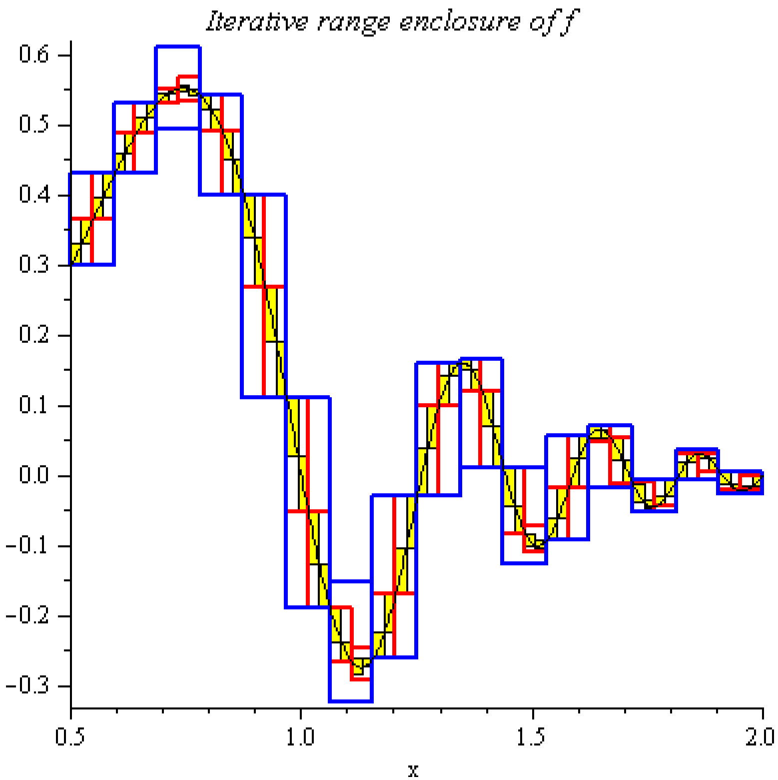
Lipschitzian minorant construction, based on a given sample point sequence and the Lipschitz-constant (overestimate); the basis for a B&B algorithm

Example 2: Adaptive Partition and Sampling in R^2

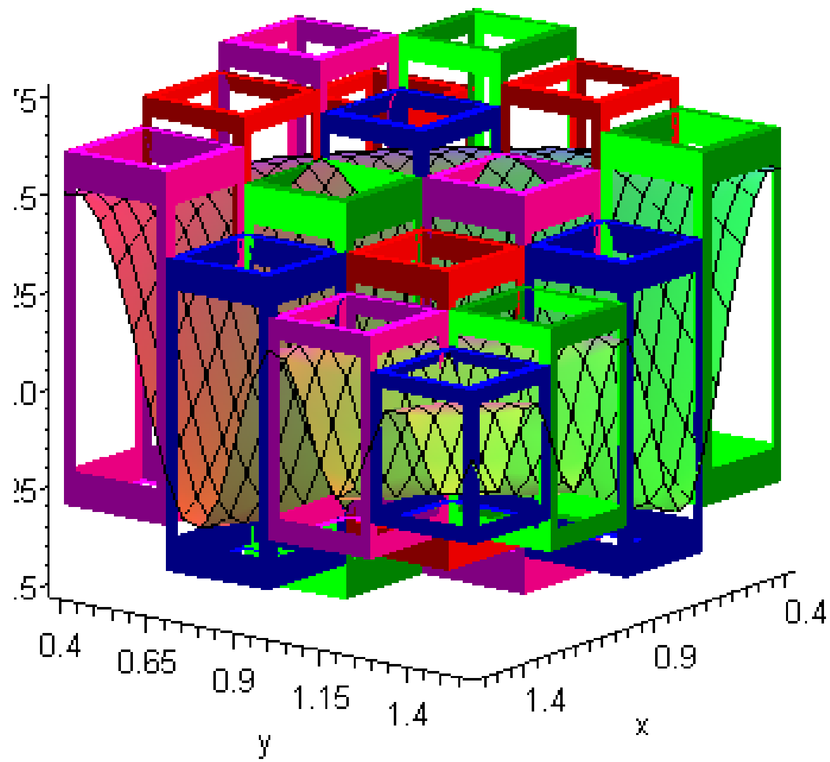


Partition sets, sample points, and selected subset

Example 3: Interval Arithmetic Package (in Maple)



$f3 := x \to e^{-x^2} \sin(\pi x^3)$



$g := (x, y) \to e^{-y\,x} \sin(\pi x^2 y^2)$

Credits: intpakX v1 - User's Guide, by Markus Grimmer, University of Wuppertal, Germany

Deterministic vs. Stochastic GO Methods

- **Exact deterministic methods have the advantage of guaranteed quality of the solution found. Examples: branch-and-bound strategies, including interval methods**
- **However, the computational demand of such methods in the worst case is exponential in *n* and *m*. Essentially, no method is better for the worst possible function(s) *f* than passive grid search…**
- **In practice, to verify optimality and to sufficiently reduce the gap between the incumbent solution and the guaranteed lower bound can be very demanding: this may not be acceptable in certain (as a matter of fact, in numerous) practical applications**

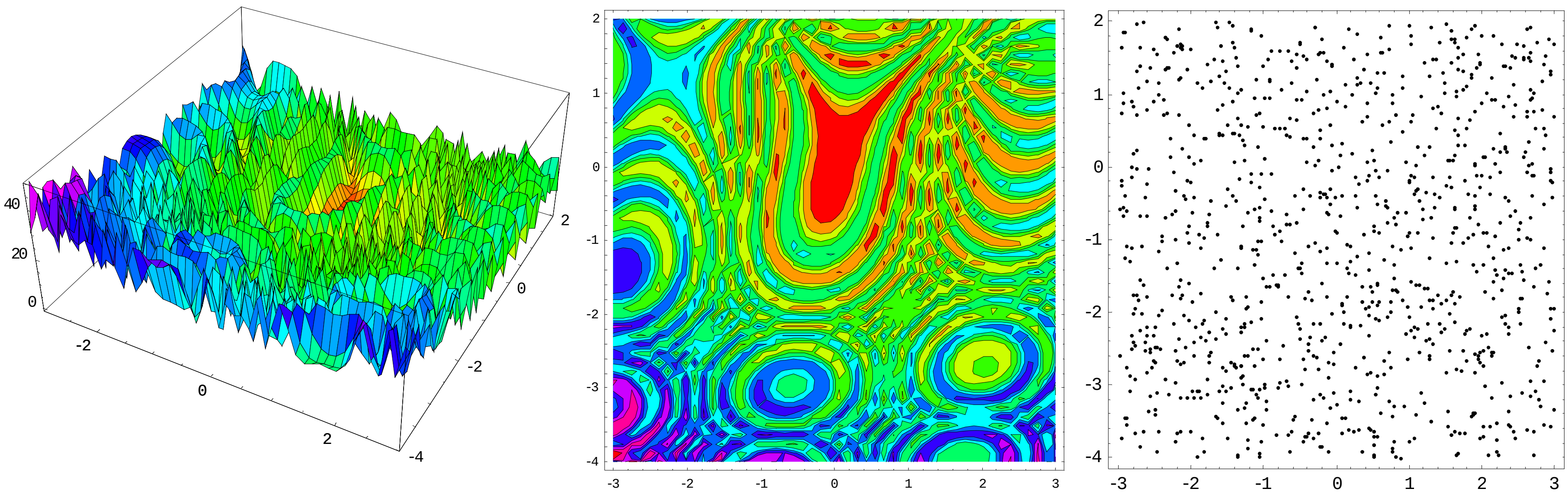
Deterministic vs. Stochastic GO Methods

God is subtle but he is not malicious…

Albert Einstein

- **Models to solve typically come from a "random source" (typically with unknown statistical features)**
- **This is a key motivation to look for alternative solution approaches, including stochastic search algorithms**
- **We will highlight the basics, and then some more advanced uses of stochastic search strategies**

The Power of Stochastic Search: An Illustrative Example



- A complicated multi-modal function, and 1,000 random sample points
- Pure (passive) random search will find points in an arbitrarily small neighborhood of the global solution x*, if the sampling effort tends to infinity
- Adaptive search strategies and statistical modeling tools become essential in higher dimensions, to improve search efficiency

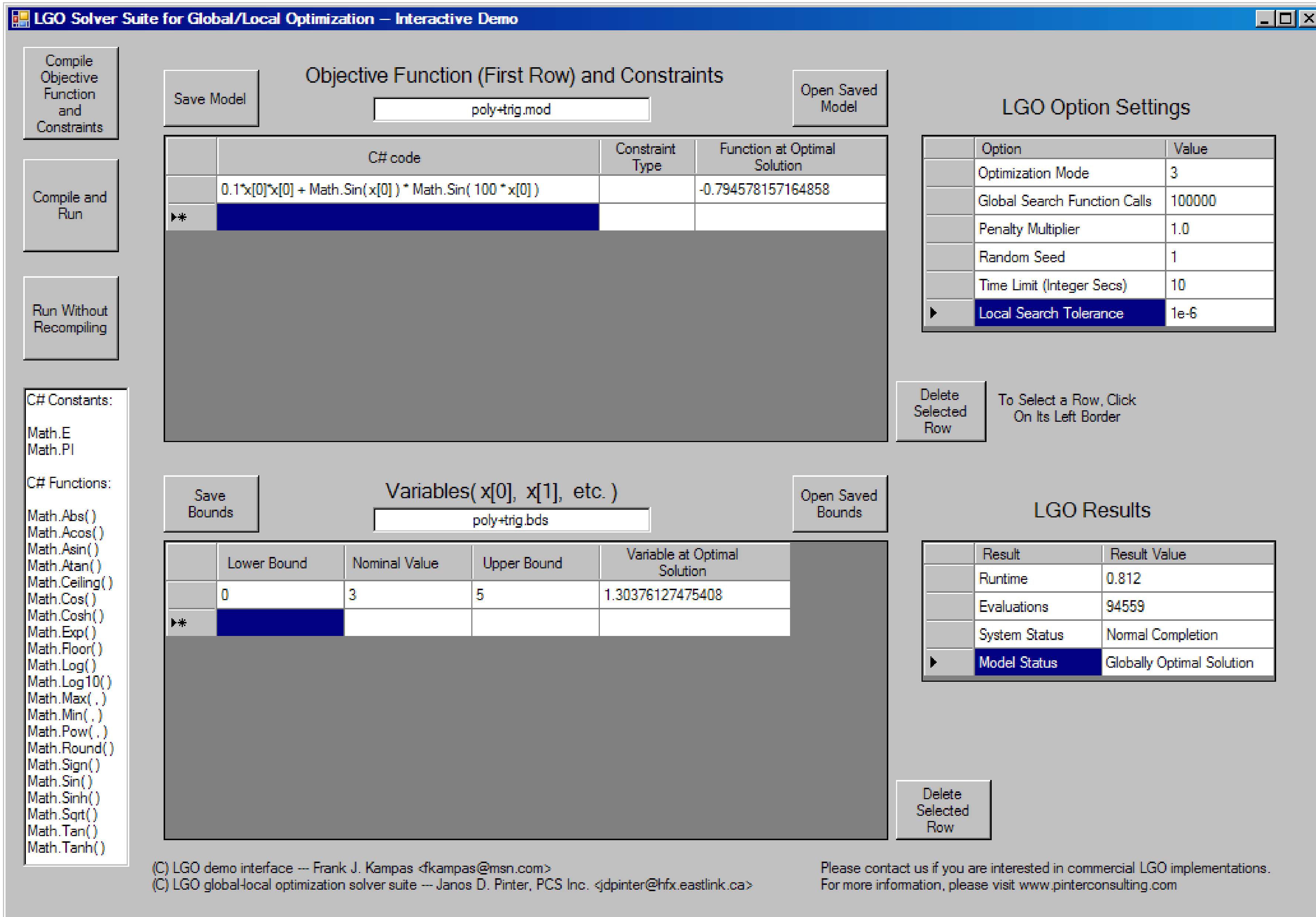
Stochastic Search Methods:
Some Key Theoretical Results

- **Global convergence of pure random search (w.p. 1) over *D* (assuming that *f* is continuous, and *D* has a suitable, but still very general topological structure)**
- **Global convergence of adaptive random search**
- **Global convergence of stochastically combined (sub)algorithms, assuming the "sufficiently frequent" usage of a globally convergent algorithm component**

GO Software Implementations: Illustrative Examples

- The following set of slides serves to illustrate various features of (LGO) software implementations
- Some of the key features apply also to other GO software implementations, *mutatis mutandis*
- The examples also hint at the capabilites and the limitations of GO software (as of today)

A Simple-to-Use LGO Demo (C, C#)



LGO Demo: Example poly+trig

Refer to previous slide where this model is solved

Model formulation and bounds given in *.mod and *.bds text files

Example 1

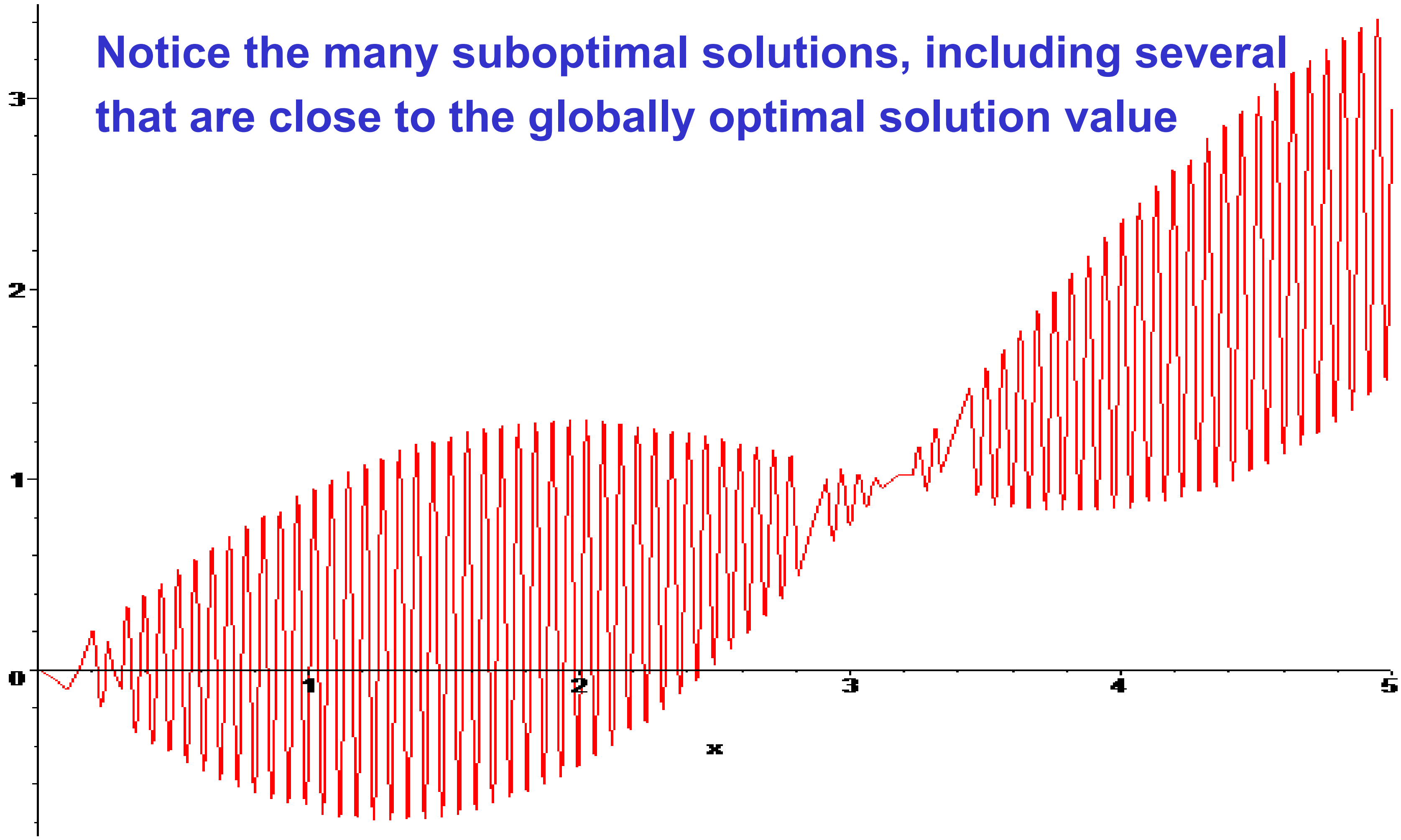
Model: cited from poly+trig.mod

0.1*x[0]*x[0] + Math.Sin(x[0]) * Math.Sin(100*x[0]) objective fct

Bounds: cited from poly+trig.bds

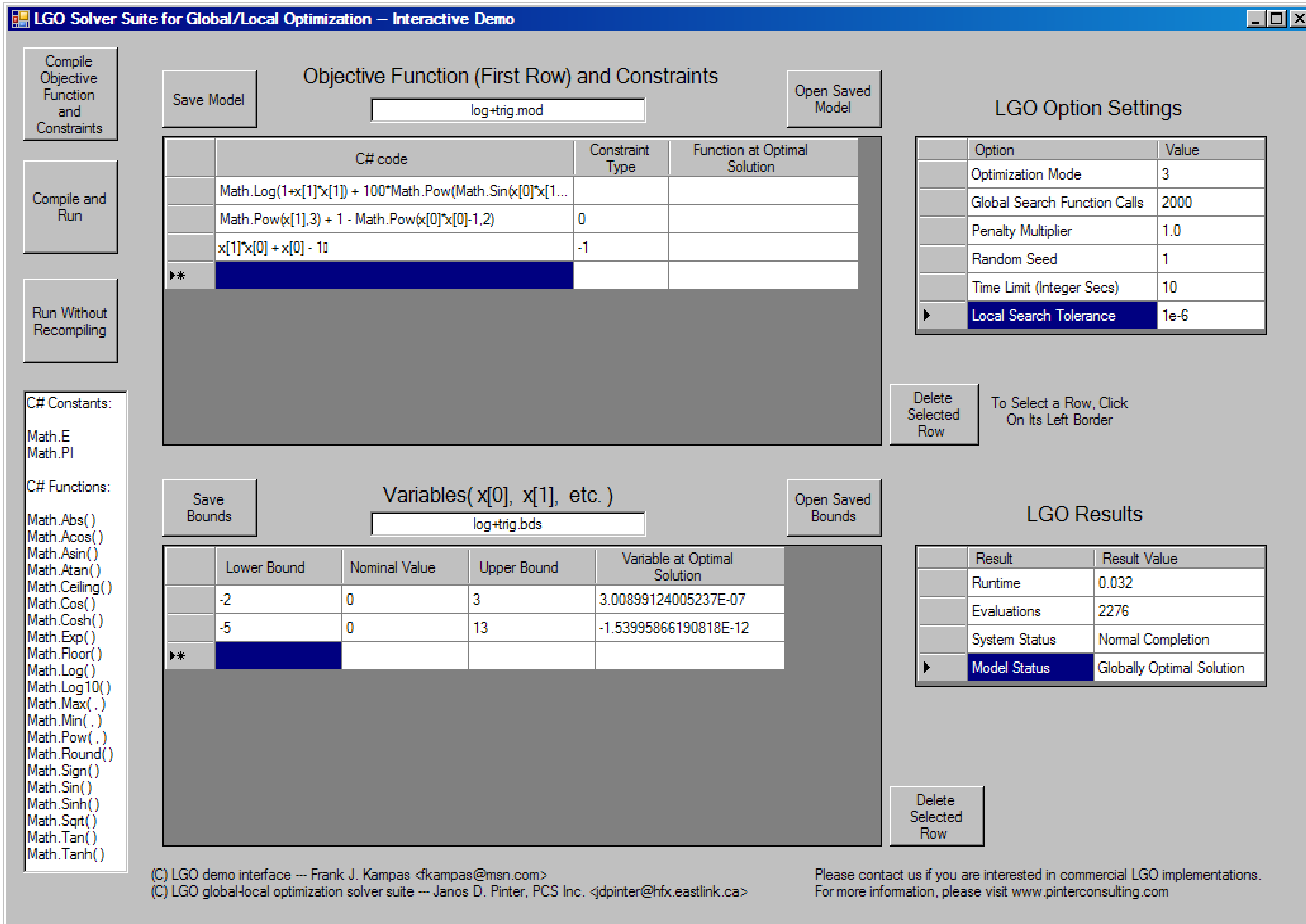
0	**lower bound**
3	**nominal value**
5	**upper bound**

LGO Demo: Example poly+trig



Global solution argument found ~1.30376; optimum value ~-0.79458

LGO Demo: Example 2



LGO Demo: Example 2

Model: log+trig.mod

Math.Log(1+x[1]*x[1]) + 100*Math.Pow(Math.Sin(x[0]*x[1]),2) objective

Math.Pow(x[1],3) + 1 - Math.Pow(x[0]*x[0]-1,2) constraint1

0 equality

x[1]*x[0] + x[0] – 1 constraint2

-1 inequality

Bounds: log+trig.bds

-2 lower bound

0 nominal value

3 upper bound

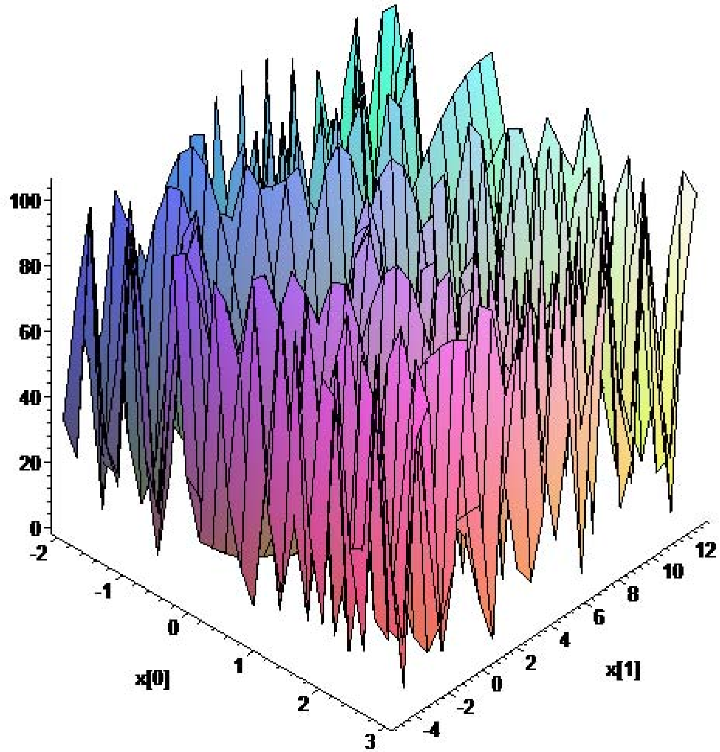
-5 lower bound

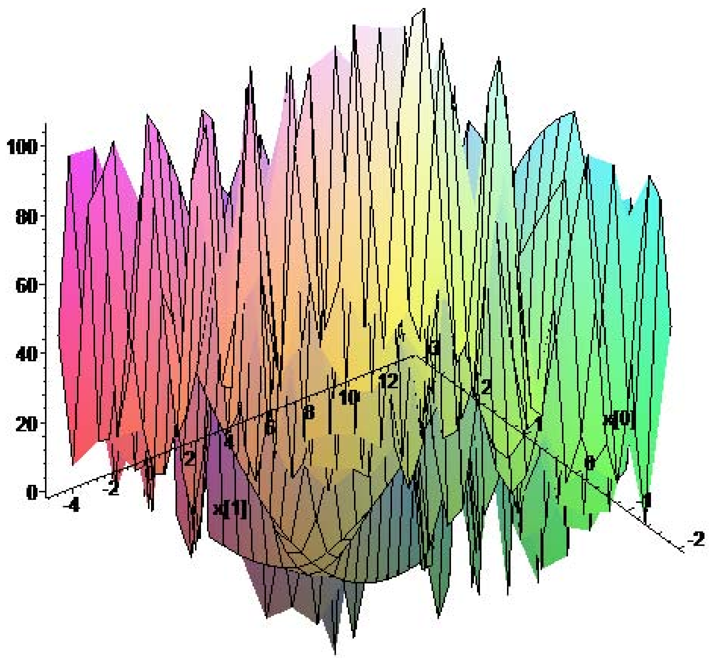
0 nominal value

13 upper bound

LGO Demo: Example 2

Two views of the objective function in log+trig.mod (from previous slide)





The unique global solution is *x*[0]=*x*[1]=0, *f=0**

LGO Integrated Development Environment

LGO IDE works with C and Fortran compilers

Microsoft Excel - Circuit Design.xls

LGO Link to Excel

Model Formulation Example: Circuit Design Problem

Circuit design model: Objective function value (error) is ~0 at the unique global solution

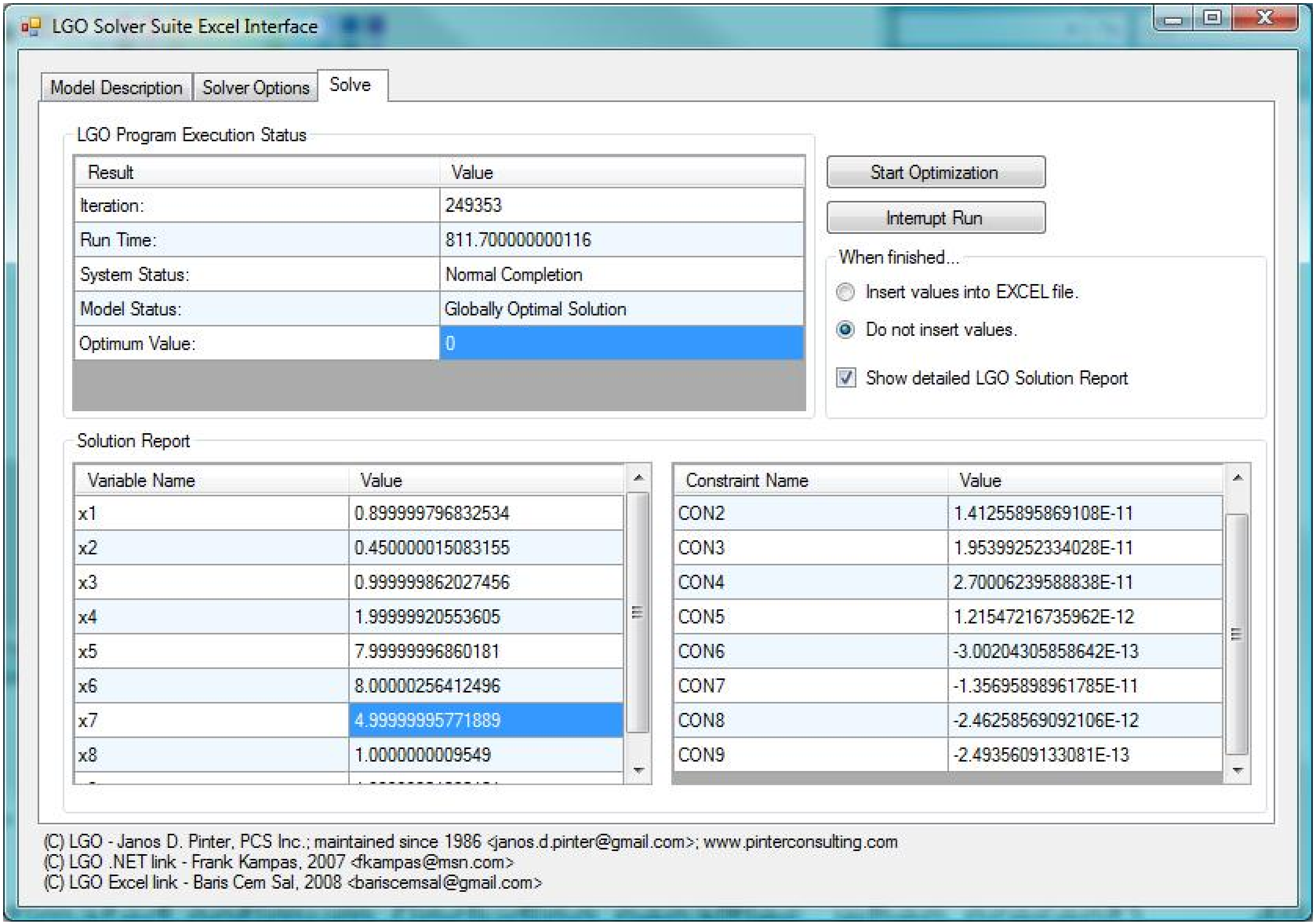
VARIABLE NAME	VARIABLE VALUE	LOWER BOUND	NOMINAL VALUE	UPPER BOUND
x1		0	1	10
x2		0	2	10
x3		0	3	10
x4		0	4	10
x5		0	5	10
x6		0	6	10
x7		0	7	10
x8		0	8	10
x9		0	9	10

CONSTRAINT NAME	CONSTRAINT TYPE	CONSTRAINT VALUE
CON1	EQ	-28.51323389
CON2	EQ	-111.8468514
CON3	EQ	-134.3885201
CON4	EQ	-211.4819315
CON5	EQ	-12.84483733
CON6	EQ	-44.68362513
CON7	EQ	-59.64595919
CON8	EQ	-85.13722788
CON9	EQ	0

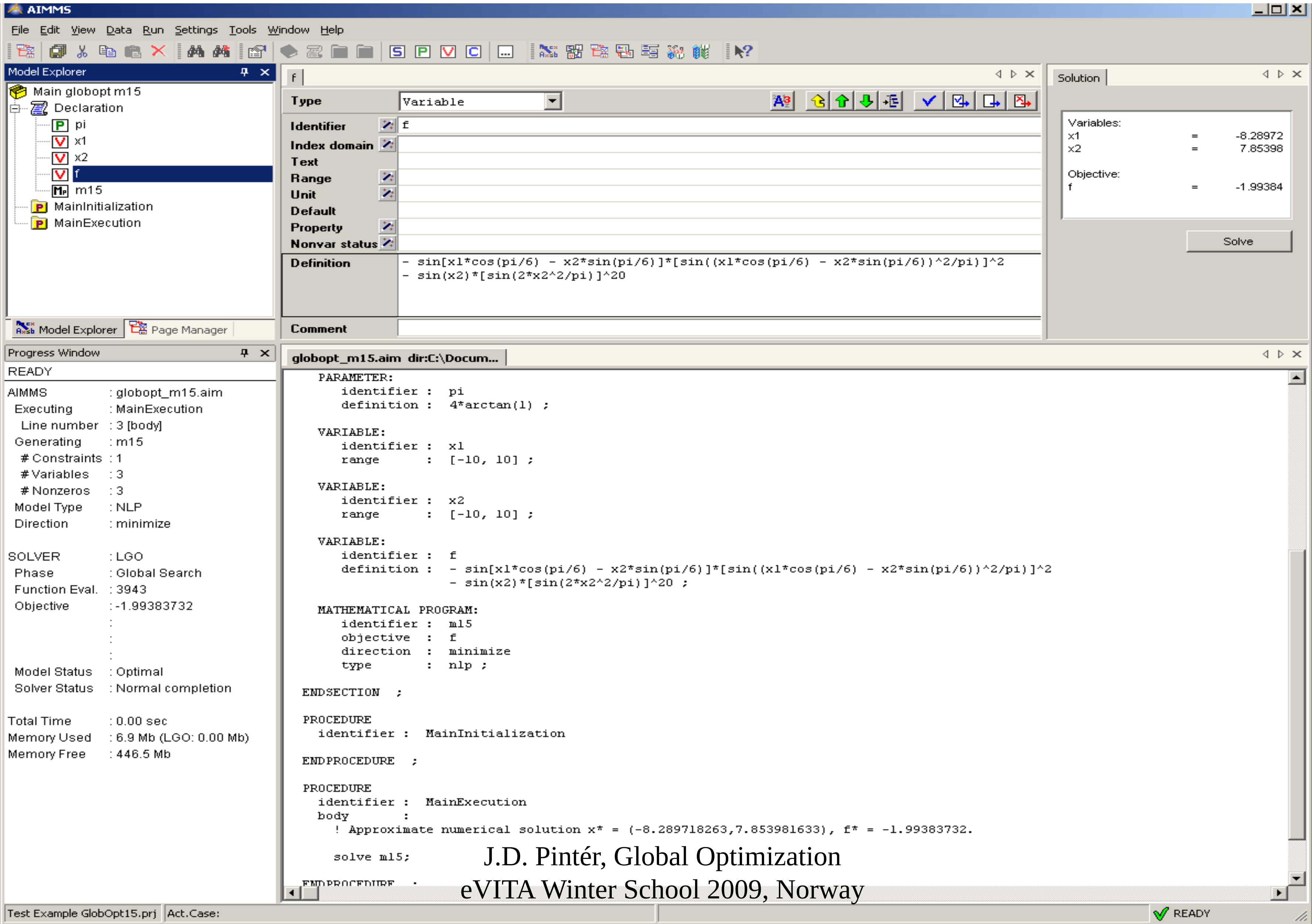
OBJECTIVE NAME	OBJECTIVE VALUE
Cost	0

Note: This development work is independent of the Excel Solver development by Frontline Systems

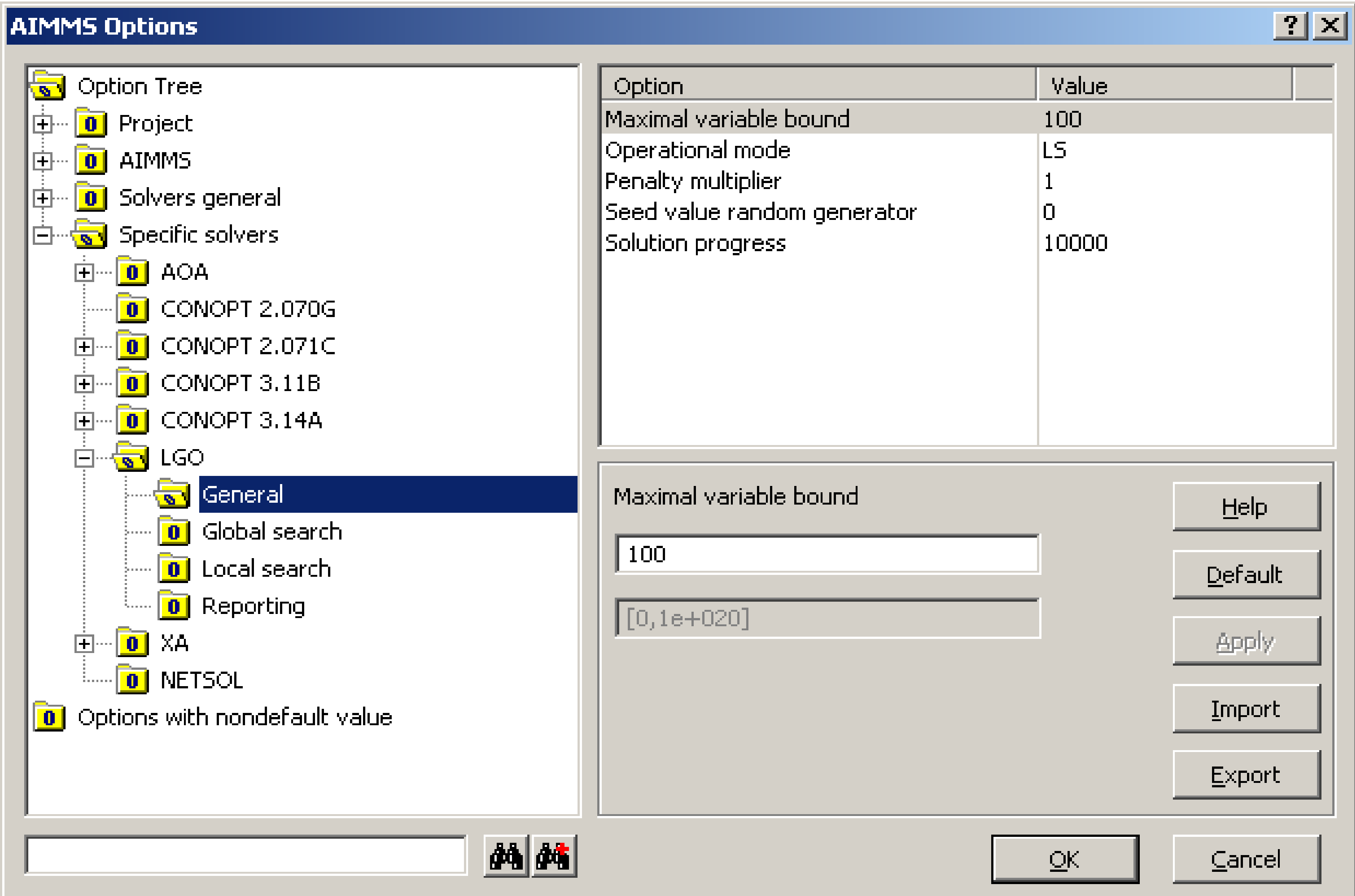
LGO Link to Excel



Model Development and Solution by AIMMS/LGO



AIMMS/LGO Solver Link Options



An AMPL Model Solved by LGO

AMPLWin 19990818

File Edit View Send Window Help

New Model Data Save Solve Display Prev Next Find: C M D

Command interface

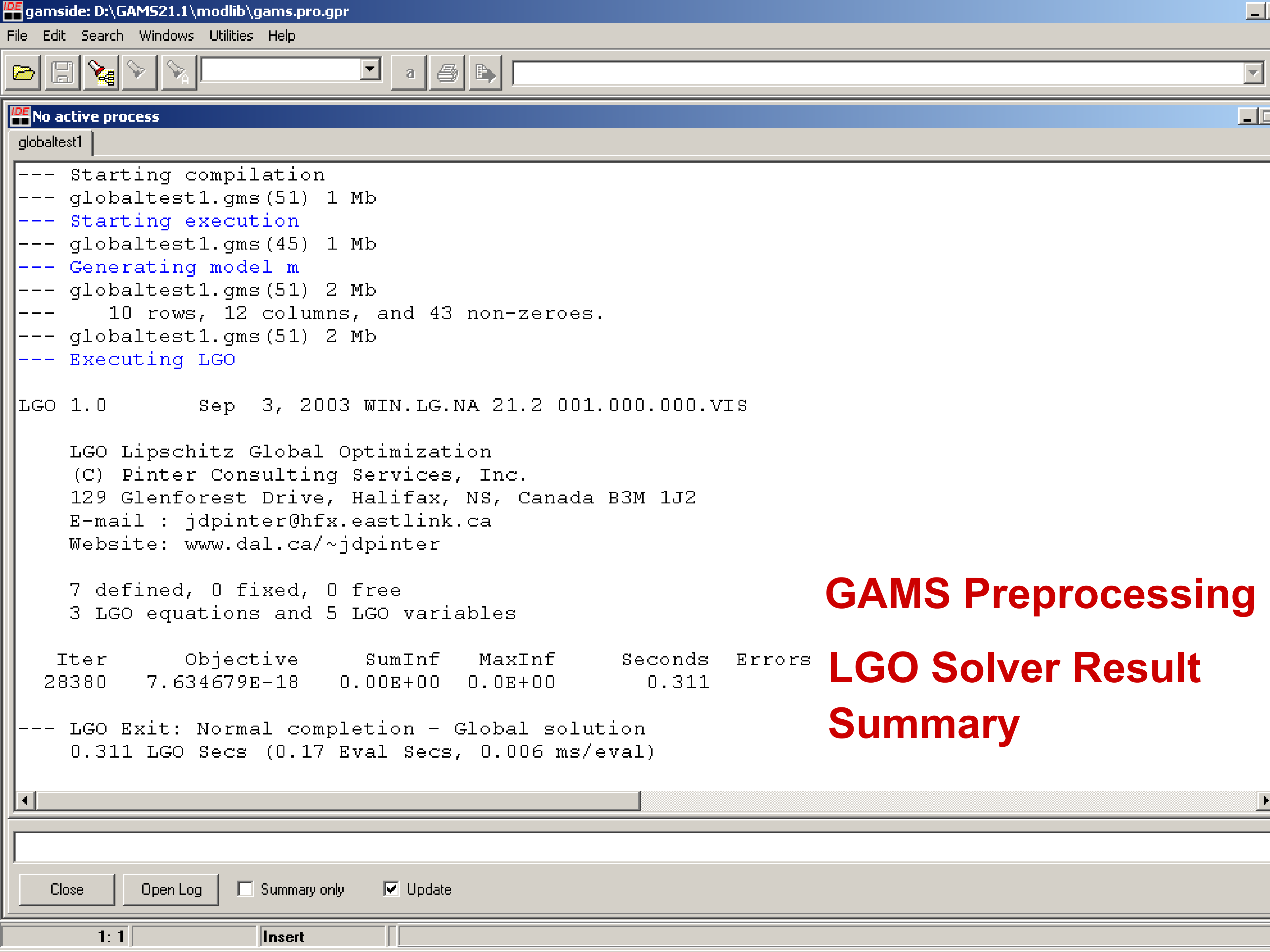
ampl: ample.out;||data;||# Type data here or open a data file...||sol

reset;||# !NLPexample.mod||# A parametric NLP model examp

```
== 1 ================
option show_stats 1; option display_eps .000001; print $version;
AMPL Version 20061130 (MS VC++ 6.0)
Licensed to Janos D. Pinter <jdpinter@hfx.eastlink.ca>.
Trial license expires 20070804.

== 2 ================
reset;
# !NLPexample.mod
# A parametric NLP model example
# J.D. Pinter, 2007
# Model description, background, references etc. can be included here
# Model structure summary
# Number of variables: 2
# Number of bound constraints: 2 (theoretically needed in global optimization,
# Number of general constraints: 2
# Objective: nonconvex, multimodal
# The global minimum value in this example depends on the parameter scale (bel
# and it is bounded from below by 0
# Parameters (as needed to define model)
# Increasing the parameter scale leads to more difficult global optimization t
param scale := 100;
# Model variables
var x{1..2};
# Objective function
minimize Obj: (x[2] - x[1])^2 + scale*(sin(x[1] + x[2]))^2;
subject to
# Bound constraints
Box1l: x[1] >= -8;
Box1u: x[1] <= 10;
Box2l: x[2] >= -17;
Box2u: x[2] <= 4;
# General constraints (in addition to bounds)
Con1: cos(x[1]^2 - x[2]^2) = 0.3;
Con2: x[1] - sin(x[2] - x[1]) <= 2;
# Initial values (typically used by local solvers) can be given here
# data;
# var x :=
#       1  3
#       2 -1
# You can try various (available) solver options
# option solver knitro;
option solver lgo;
# option solver minos;
# Set display precision
option display_round 10;
option display_eps 1e-10;
option display_precisison 10;
# Solve model stated above
solve;
# Display results (in command window)
display Obj;
display _varname, _var;
display _conname, _con;
```

Model I:\AMPL Pro\!NLPexample.mod

```
# !NLPexample.mod
# A parametric NLP model example
# J.D. Pinter, 2007

# Model description, background, references etc. can be included here

# Model structure summary
# Number of variables: 2
# Number of bound constraints: 2 (theoretically needed in global optimization, for all variables
# Number of general constraints: 2
# Objective: nonconvex, multimodal
# The global minimum value in this example depends on the parameter scale (below),
# and it is bounded from below by 0

# Parameters (as needed to define model)
# Increasing the parameter scale leads to more difficult global optimization test models
param scale := 100;

# Model variables
var x{1..2};

# Objective function
minimize Obj: (x[2] - x[1])^2 + scale*(sin(x[1] + x[2]))^2;

subject to

# Bound constraints
Box1l: x[1] >= -8;
Box1u: x[1] <= 10;
Box2l: x[2] >= -17;
Box2u: x[2] <= 4;

# General constraints (in addition to bounds)
Con1: cos(x[1]^2 - x[2]^2) = 0.3;
Con2: x[1] - sin(x[2] - x[1]) <= 2;

# Initial values (typically used by local solvers) can be given here
# data;
# var x :=
#       1  3
#       2 -1

# You can try various (available) solver options
# option solver knitro;
option solver lgo;
# option solver minos;

# Set display precision
option display_round 10;
option display_eps 1e-10;
option display_precisison 10;

# Solve model stated above
solve;

# Display results (in command window)
display Obj;
display _varname, _var;
```


gamside: D:\GAMS21.1\modlib\gams.pro.gpr
File Edit Search Windows Utilities Help
No active process
globaltest1
--- Starting compilation
--- globaltest1.gms(51) 1 Mb
--- Starting execution
--- globaltest1.gms(45) 1 Mb
--- Generating model m
--- globaltest1.gms(51) 2 Mb
--- 10 rows, 12 columns, and 43 non-zeroes.
--- globaltest1.gms(51) 2 Mb
--- Executing LGO
LGO 1.0 Sep 3, 2003 WIN.LG.NA 21.2 001.000.000.VIS
LGO Lipschitz Global Optimization
(C) Pinter Consulting Services, Inc.
129 Glenforest Drive, Halifax, NS, Canada B3M 1J2
E-mail : jdpinter@hfx.eastlink.ca
Website: www.dal.ca/~jdpinter
7 defined, 0 fixed, 0 free
3 LGO equations and 5 LGO variables
Iter Objective SumInf MaxInf Seconds Errors
28380 7.634679E-18 0.00E+00 0.0E+00 0.311
--- LGO Exit: Normal completion - Global solution
0.311 LGO Secs (0.17 Eval Secs, 0.006 ms/eval)
GAMS Preprocessing
LGO Solver Result
Summary
Close
Open Log
Summary only
Update
1: 1
Insert

An MPL/LGO Model and its Solution

File Edit Search Project Run View Graph Options Window Help

J:\!JDP\Software Development\Implementations\MPL_LGO\Software Development\MPL_LGO tests\Box Design\...

```
{   BoxDesign.mpl   }
{   Source/author: LINGO Model Library   }
{   Adapted by Bjarni Kristjansson and Janos D. Pinter   }
{   Design a minimum-cost computer box that meets engrg and aesthetics constraints  }
{   A 3-variable, 5-constraint global optimization test problem solved by the global
{   Global solution: obj=50.9650752407, d=23.0309622444, h=6.8656565766, w=9.56219578

TITLE
   BoxDesign;

OPTIONS
   ModelType=Nonlinear
   ParserType=Extended

   LGO.opmode=0
   LGO.g_maxfct=10000
   LGO.max_nosuc=10000
   LGO.penmult=1
   LGO.acc_tr=-1000000
   LGO.fct_trg=-1000000

VARIABLES
   d  INIT 50;
   h  INIT 50;
   w  INIT 50;

MODEL

   MIN obj = 0.2*h*w + 0.1*(d*h + d*w) ;

SUBJECT TO

   con1: 888. - 2.*(d*h + d*w + h*w) <= 0;
   con2: 1512. - d*h*w <= 0;
   con3:  h - 0.718*w <= 0;
   con4: -h + 0.518*w <= 0;
   con5: -252. + d*w <= 0;
```

Model Definitions

- TITLE BoxDesign
  - VARIABLE
    - d (1)
    - h (1)
    - w (1)
  - CONSTRAINT
    - con1 (1)
    - con2 (1)
    - con3 (1)
    - con4 (1)
    - con5 (1)

View File: BoxDesign.sol

```
  MIN obj      =         50.9651

DECISION VARIABLES

PLAIN VARIABLES

  Variable Name              Activity      Reduced Cost
 -----------------------------------------------------
  d                           23.0310            0.0000
  h                            6.8657            0.0000
  w                            9.5622            0.0000
 -----------------------------------------------------

CONSTRAINTS

PLAIN CONSTRAINTS
  Constraint Name               Slack      Shadow Price
 -----------------------------------------------------
  con1                         0.0000            0.0000
  con2                         0.0000            0.0000
  con3                         0.0000            0.0000
  con4                        -1.9124            0.0000
  con5                       -31.7734            0.0000
 -----------------------------------------------------
```

Main model file: BoxDesign.mpl | 37:1 | Solved

GO in Integrated Scientific and Technical Computing Systems

- Maple, Mathematica, Matlab (and some others that are more specific to certain engineering or scientific fields)
- Model prototyping and development: simple and advanced calculations, programming, documentation, visualization,… supported in "live" interactive documents
- Data I/O and management features
- Links to external software products
- Portability across hardware and OS platforms
- "One-stop" tools for interdisciplinary development
- ISTCs are particularly suitable for developing complex, advanced nonlinear models; obvious GO relevance
- Several articles discuss our implementations (refs later)

Getting Started ppt form.nb *

MathOptimizer Model

- ## Getting Started

- ### Model Formulation

```
vars = {x1, x2};                                 (* decision variables *)
varnom = {8., -14.};                             (* nominal values *)
varlb = {-10., -15.};                            (* lower bounds *)
varub = {20., 10.};                              (* upper bounds *)
objf = 10.*(x1^2 - x2)^2 + (x1 - 1)^2;           (* objective function*)
eqs = {x1 - x1*x2};                              (* equality constraints *)
ineqs = {3.*x1 + 4.*x2 - 25.};                   (* inequality constraints, ≤0 form *)
```

- ### Numerical Solution

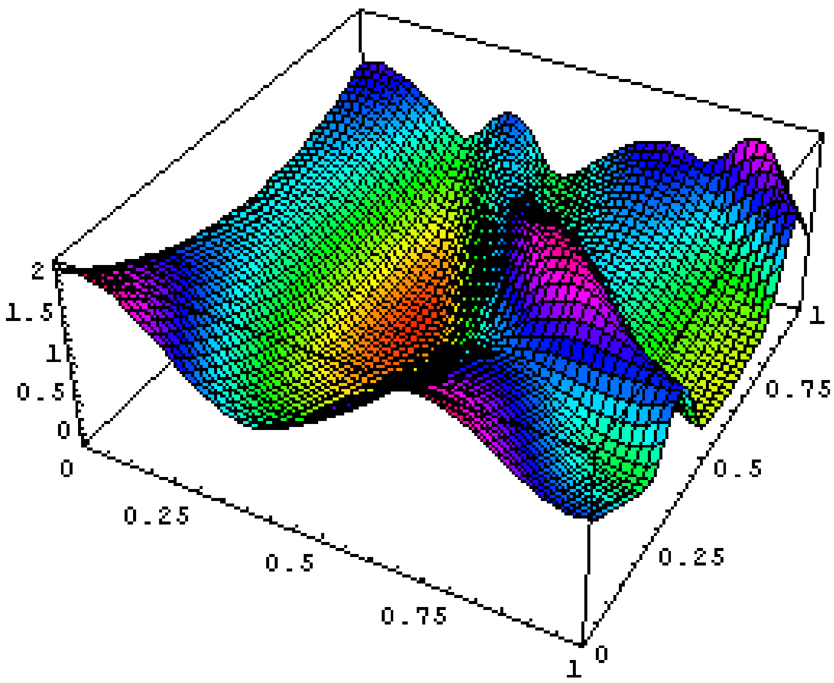
```
Optimize[objf, eqs, ineqs, vars, varnom, varlb, varub,
GlobalSolverMode -> 1, LocalSolverMode -> 1, ReportLevel -> 1]
```

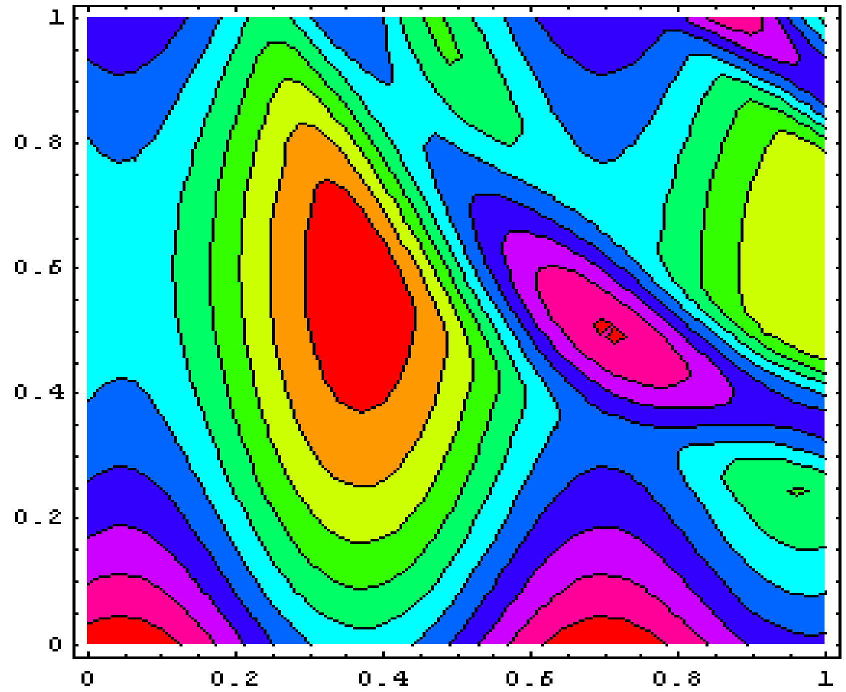
Note that dense nonlinear models (including many GO models) are similarly formulated across platforms: relatively easy model conversions, converters available in several cases (example: GAMS CONVERT utility)

150%

Start | 451.5 MB - F... | Mathemati... | D:\Mathema... | Mathematica... | Eudora - [to... | 2 Internet ... | Microsoft Po...

Advanced Visualization Tools in ISTCs





An example from the MathOptimizer User Guide:
Surface and contour plot of a randomly generated test function

Note: MO is a native Mathematica solver product, as opposed to the LGO implementations reviewed here

Mathematica 5.1 - [UserGuide.nb *]

File Edit Cell Format Input Kernel Find Window Help

UserGuide.nb *

MathOptimizer Professional

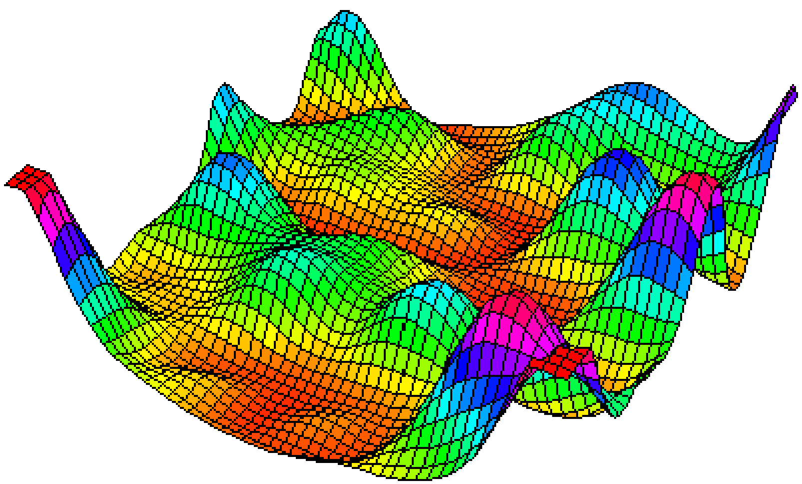
An Advanced Modeling and Optimization System for *Mathematica*,

Using the LGO Solver Engine

User Guide

User Guide can be invoked from Mathematica's online Help menu

The same applies to MathOptimizer



This feature supports efficient prototyping and modular development

Mathematica 5.0 - [MOP Getting Started Example.nb *]

File Edit Cell Format Input Kernel Find Window Help

MOP Getting Started Example.nb *

Getting Started with MathOptimizer Professional: Illustrative Examples

■ *Mathematica* Platform and Date

■ Activate MathOptimizer Professional

```
In[1]:= Needs["MathOptimizerPro`callLGO`"];

In[2]:= ? callLGO
```

■ A Simple One-line Example

It is straightforward to define a (small) optimization model, as illustrated by the following example.

```
In[24]:= callLGO[2*x^2 + y^2, {x + y - 1 == 0, x^2 + 3*y ≤ 2},
           {{x, -2, 1, 3}, {y, -3, 2, 2}}]

Out[24]= {0.673762, {x → 0.381966, y → 0.618034}, 2.79532 × 10^-9}
```

100%

Start | 342.1 MB - FreeM... | Notebooks | MiER article and r... | Mathematica 5... | 7:18 PM

File Edit View Web Window Help

Current Directory: D:\Matlab6.5\work

```
Problem: lgo2_prob -   2: M2                 f_k      -6.400184647478858400

Solver: LGO.  EXIT=0.  INFORM=4.
MEX Interface to LGO solver (Multis
Terminated by solver : Globally opt

FuncEv 1649 GradEv    0
CPU time: 1.613000 sec. Elapsed tim
Optimal vector x:
x_k:  -0.199384  -2.942209
User given stationary point x_* (1)
x_*:   3.340980  -0.199384
```

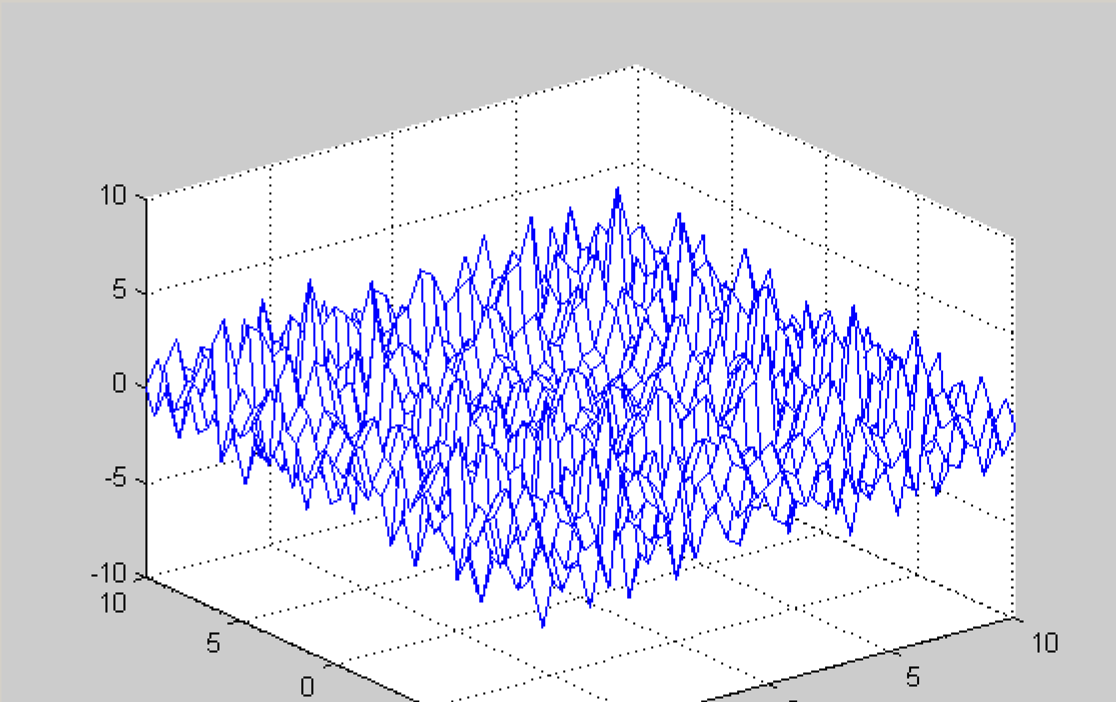
D:\TOMLAB\testprob\lgo2_f.m - metapad

File Edit Favourites Options Help

```
% function f = lgo2_f(x, Prob)
%
% Test functions for global optimization.
% Two or more dimensions
%
% Reference:
% Pintér, J.D., Bagirov, A., and Zhang, J. (2003) An Illustrated Collection of
% Global Optimization Test Problems. Research Report, Pintér Consulting Services,
% Inc. Halifax, NS, Canada; and CIAO-ITMS, University of Ballarat, Ballarat, Vic.,
% Australia.
nc., E-mail: medvall@tomlab.biz.
timization Inc., Sweden. $Release: 4.2.0$
d Feb 1, 2004.

(x2)*(cos(x2)-sin(x2))^2;

in(11*x1)+sin(3*x2)+sin(5*x2)+sin(7*x2)+sin(11*x2
```

Figure No. 1

File Edit View Insert Tools Window Help



Model setup, solution and visualization in Matlab

TOMLAB /LGO solver

*Maple 10 - I:\!My Documents\Software Development\Implementations\Maple Optimization and GOT\GOT Demo Apps\Teaching ORMS with ICSs --- demo examples.mw - [Server 2]

File Edit View Insert Format Table Plot Spreadsheet Sketch Tools Window Help

*GOT_Introduction_and_Examples.mw | *Teaching ORMS with ICSs --- demo examples.mw

```
> with(GlobalOptimization):
> objective := (sin(1+2*x)*sin(3*x+y))^2:
> constraints := {x^2-3*x*y>= 0, sin(y)*y-sin(x)<= 0.3, (x-y)^2<=0.1}:
> bounds := x=-1..3, y=-2..3:
> objective, constraints, bounds;
```

$$\sin(1+2x)^2 \sin(3x+y)^2, \{0 \le x^2 - 3xy, \sin(y)\,y - \sin(x) \le 0.3, (x-y)^2 \le 0.1\}, x = -1..3, y = -2..3 \quad (1)$$

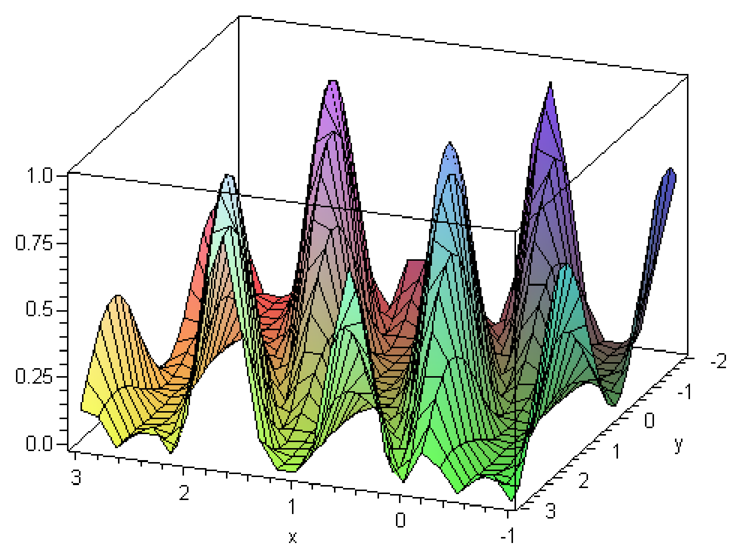
```
> solution := GlobalSolve(objective, constraints, bounds);
```

$$solution := [1.94826655347192452\ 10^{-31}, [x = -0.00908113196697686414, y = 0.0272433959009311215]] \quad (2)$$

```
> eval(constraints, solution[2]);
```

$$\{0 \le 0.0008246695780, 0.009823117965 \le 0.3, 0.001319471325 \le 0.1\} \quad (3)$$

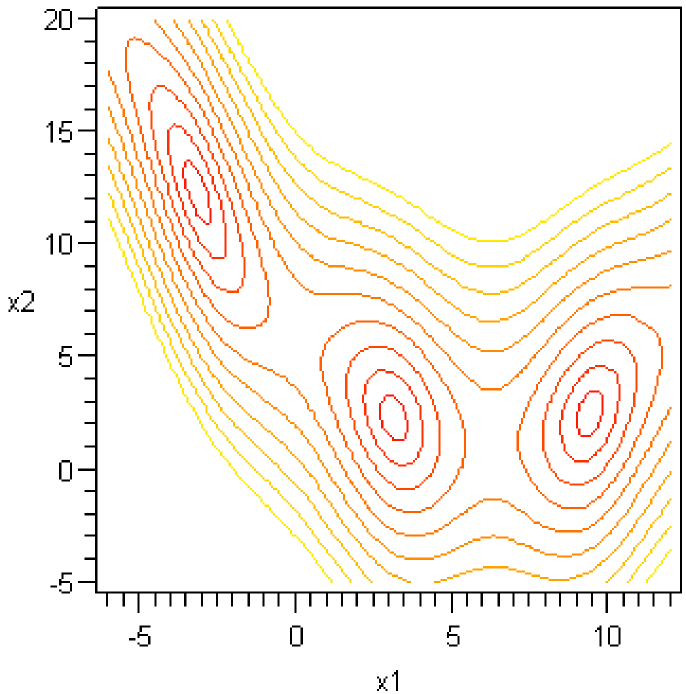
```
> plot3d(objective, bounds, axes=boxed);
```

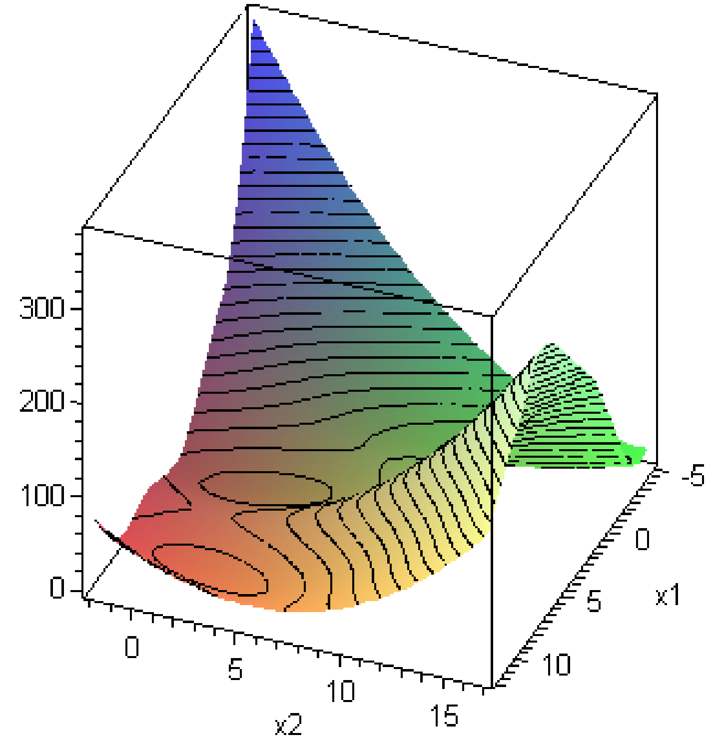


Model formulation, numerical solution, and visualization, using the GO Toolbox for Maple

Branin’s Test Problem with Multiple (3) Global Solutions

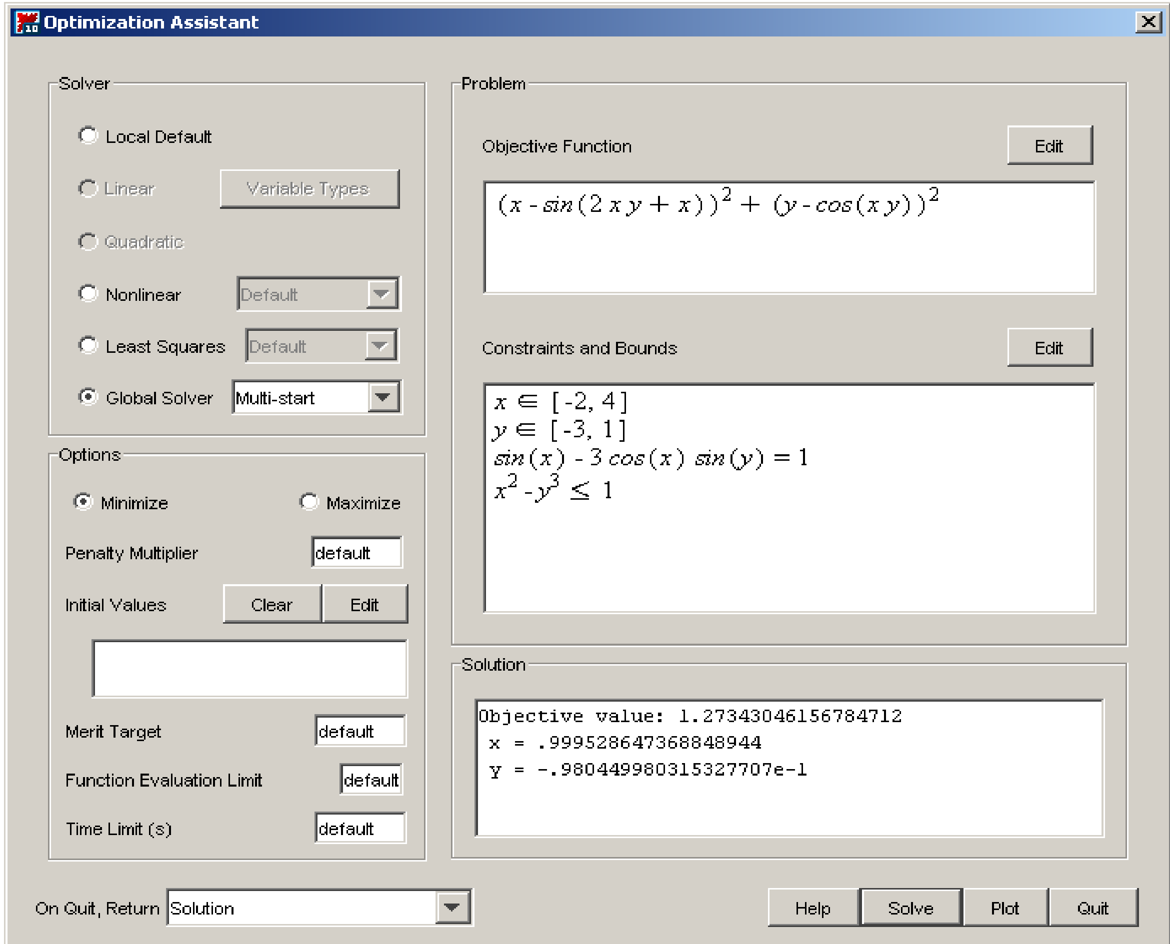
$$f := \left(x2 - \frac{1.275000000\, x1^2}{\pi^2} + \frac{5\, x1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\,\pi}\right)\cos(x1) + 10$$





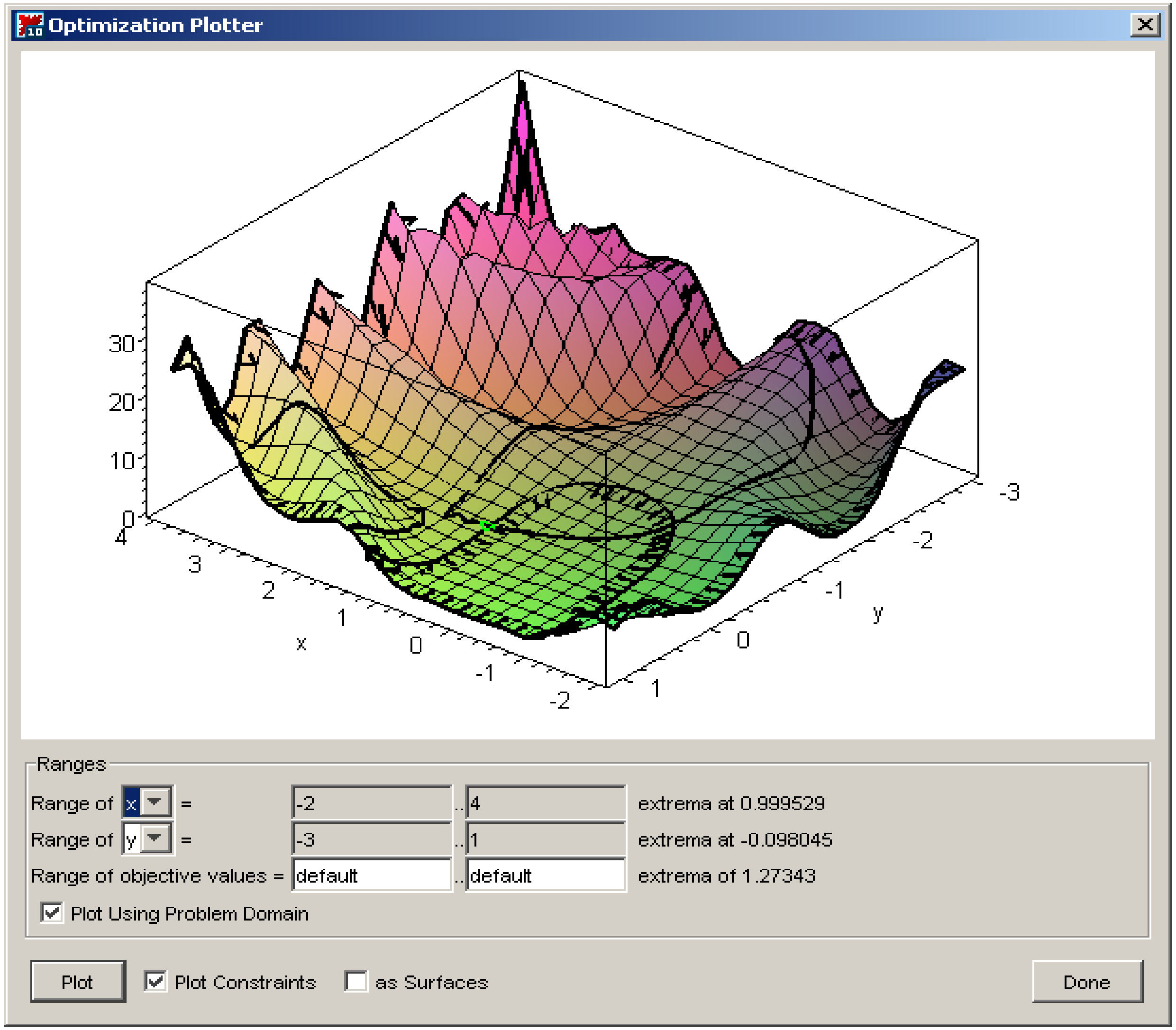
The GOT can also be used to find a sequence of global solutions

Maple GO Toolbox: Optimization Assistant



See e.g. *Optimization Methods and Software* (2006)

Maple GO Toolbox: Optimization Plotter



Illustrative Case Studies

A Concise Summary

- Many of the actual client case studies reviewed here are based on multi-disciplinary research, in addition to the global optimization component
- All detailed case studies can (could) be presented in full detail, each in a separate lecture… instead, we shall briefly review a selection of these
- References, demo software examples, publications, and additional details are all available upon request

Illustrative Case Studies reviewed in this talk (as time allows)

- An illustrative “black box” client model
- Trefethen’s HDHD Challenge, Problem 4
- Systems of nonlinear equations
- Optimization problems featuring numerical procedures
- Nonlinear model fitting examples
- Experimental design
- Non-uniform circle packings, and other packings
- Computational chemistry: potential energy models
- Portfolio selection, with a non-convex purchase cost
- Solving differential equations by the shooting method
- Data classification and visualization
- Circuit design model
- Rocket trajectory optimization

Illustrative Case Studies reviewed in this talk (as time allows)

- Industrial design model examples
- Collision (trajectory) analysis
- Design optimization in robotics
- Laser design
- Cancer therapy planning
- Sonar equipment design
- Oil field production optimization
- Automotive suspension system design

and other areas

- In addition, many standard NLP/GO and other test problems have been used to evaluate solver performance across the various modeling environments reviewed here
- Experiments conducted by developer partners and clients, in addition to the author's own work

“Black Box” Model Received from Client:
“Can your software handle this problem?…”

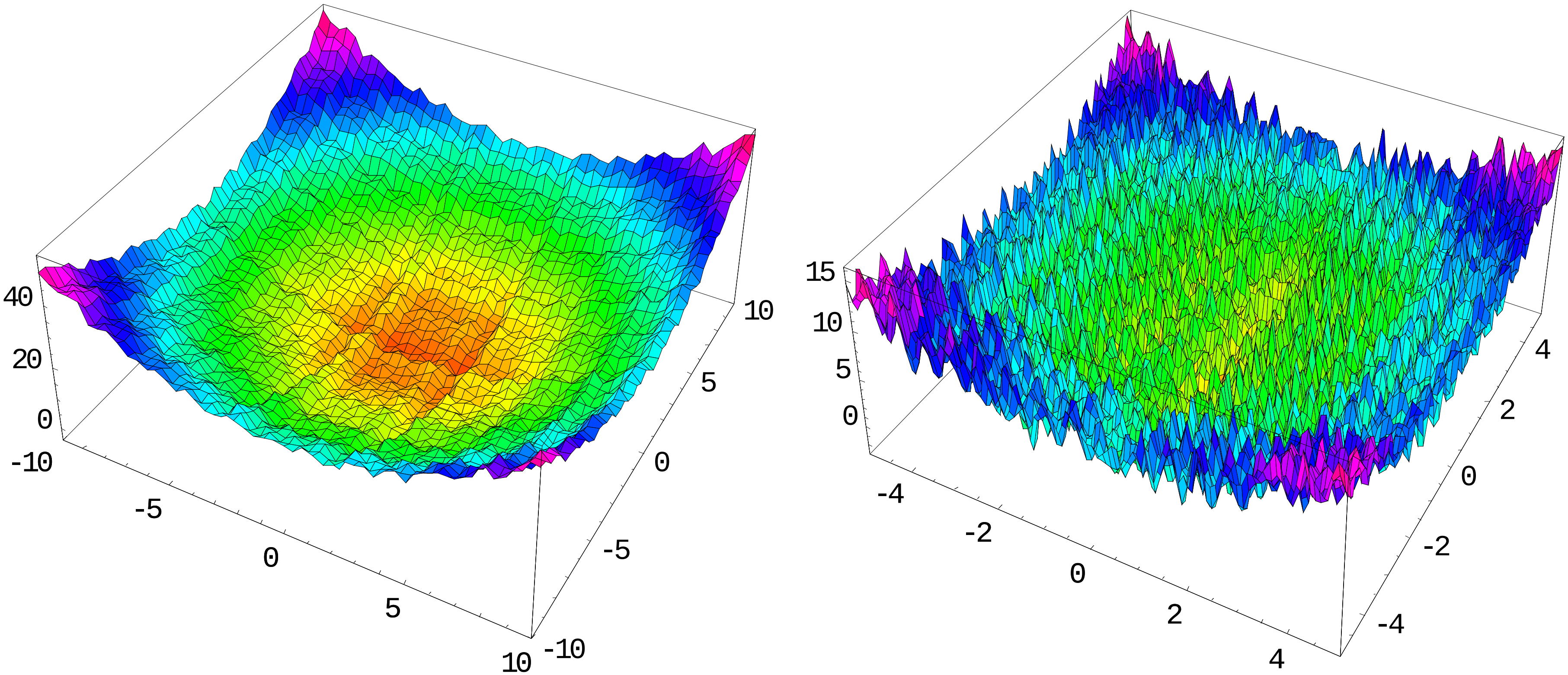
C:\Documents and Settings\Janos\Desktop\messages\USER_FCT.C - metapad

File Edit Favourites Options Help

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int __declspec(dllexport) _stdcall USER_FCT( double x[], double fox[1], double gox[])
{
fox[0] = pow(-52.2814830080429 + 0.291083080677605*(sin(x[0])*(1.*(-1.*sin(1.813087*(0.000122738408770829 -
7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(-57.7094469266987*cos(x[0\
]) + 0.000122738408462659*cos(x[0])*cos(x[2])*x[8] + 0.999999992467641*cos(x[0])*sin(x[2])*x[8]) + sin(x[0])*(-
2.30934895009823*sin(x[0]) + cos(x[0])*(0.999999992467641*(48.5067855663717 + cos(x[2])*x[8]) + 0.0001227\
38408462659*(57.7154010070919 - 1.*sin(x[2])*x[8]))) + cos(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] -
3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(-2.30934895009823 - 1.*sin(x[0])*(-2.309\
34895009823*sin(x[0]) + cos(x[0])*(0.999999992467641*(48.5067855663717 + cos(x[2])*x[8]) + 0.000122738408462659*
(57.7154010070919 - 1.*sin(x[2])*x[8]))))) - 1.*(sin(x[0])*(2.2328632*pow(cos(x[0]),2.) + sin(x[0])*(2.2\
328632*sin(x[0]) + 0.000122738408770829*x[1] + 7.53235849379756e-9*x[6] + x[9])) + cos(1.813087*(0.000122738408770829 -
7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(2.2328632*si\
n(x[0]) + 0.000122738408770829*x[1] + 7.53235849379756e-9*x[6] + x[9] - 1.*sin(x[0])*(2.2328632*pow(cos(x[0]),2.) + sin(x
[0])*(2.2328632*sin(x[0]) + 0.000122738408770829*x[1] + 7.53235849379756e-9*x[6] + x[9]))) - 1.\
*sin(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(-
1.*cos(x[0])*x[4] - 0.000122738408770829*cos(x[0])*x[10]))) + cos(x[0])*(1.*(-1.*sin(1.813087\
*(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*
(57.7094469266987*sin(x[0]) - 0.000122738408462659*cos(x[2])*sin(x[0])*x[8] - 0.999999992467641*sin(x[0])*si\
n(x[2])*x[8]) + cos(x[0])*(-2.30934895009823*sin(x[0]) + cos(x[0])*(0.999999992467641*(48.5067855663717 + cos(x[2])*x[8]) +
0.000122738408462659*(57.7154010070919 - 1.*sin(x[2])*x[8]))) + cos(1.813087*(0.000122738408\
770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(0.999999992467641*
(48.5067855663717 + cos(x[2])*x[8]) + 0.000122738408462659*(57.7154010070919 - 1.*sin(x[2])*x[8]) - 1.*co\
s(x[0])*(-2.30934895009823*sin(x[0]) + cos(x[0])*(0.999999992467641*(48.5067855663717 + cos(x[2])*x[8]) +
0.000122738408462659*(57.7154010070919 - 1.*sin(x[2])*x[8]))))) - 1.*(cos(x[0])*(2.2328632*pow(cos(x[0]),2.) +\
 sin(x[0])*(2.2328632*sin(x[0]) + 0.000122738408770829*x[1] + 7.53235849379756e-9*x[6] + x[9])) + cos(1.813087*
(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7])\
)*(2.2328632*cos(x[0]) - 1.*cos(x[0])*(2.2328632*pow(cos(x[0]),2.) + sin(x[0])*(2.2328632*sin(x[0]) + 0.000122738408770829*x
[1] + 7.53235849379756e-9*x[6] + x[9]))) - 1.*sin(1.813087*(0.000122738408770829 - 7.5323584\
9379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(sin(x[0])*x[4] + 0.000122738408770829*sin(x[0])*x
[10])))) - 0.70817848041004*(0.999991873452302*(-1.*sin(1.813087*(0.000122738408770829 - 7.5\
3235849379756e-9*x[3] - 3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(2.30934895009823*cos(x[0]) +
48.5138690974646*sin(x[0]) + 0.999999992467641*cos(x[2])*sin(x[0])*x[8] - 0.000122738408462659*sin(x[0])*s\
in(x[2])*x[8]) + cos(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] -
9.45607074651404e-18*x[7]))*(0.000122738408462659*(48.5067855663717 + cos(x[2])*x[8]) + 0.999999992467641*(\
-57.7154010070919 + sin(x[2])*x[8])) + sin(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13
*x[5] - 9.45607074651404e-18*x[7]))*(-0.000122738408770829*cos(x[0])*x[1] - 7.53235849379756e\
-9*cos(x[0])*x[6] - 1.*cos(x[0])*x[9]) - 1.*cos(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] -
3.08169898606717e-13*x[5] - 9.45607074651404e-18*x[7]))*(-1.*x[4] - 0.000122738408770829*x[10])) + 0.0040315\
0460189537*(-1.*cos(x[0])*(1.*(-1.*sin(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5]
- 9.45607074651404e-18*x[7]))*(-57.7094469266987*cos(x[0]) + 0.000122738408462659*cos(x[0])\
*cos(x[2])*x[8] + 0.999999992467641*cos(x[0])*sin(x[2])*x[8]) + sin(x[0])*(-2.30934895009823*sin(x[0]) + cos(x[0])*
(0.999999992467641*(48.5067855663717 + cos(x[2])*x[8]) + 0.000122738408462659*(57.7154010070919 - 1.*\
sin(x[2])*x[8]))) + cos(1.813087*(0.000122738408770829 - 7.53235849379756e-9*x[3] - 3.08169898606717e-13*x[5] -
9.45607074651404e-18*x[7]))*(-2.30934895009823 - 1.*sin(x[0])*(-2.30934895009823*sin(x[0]) + cos(x[0])*(\
0.999999992467641*(48.5067855663717 + cos(x[2])*x[8]) + 0.000122738408462659*(57.7154010070919 - 1.*sin(x[2])*x[8]))))) - 1.
*(sin(x[0])*(2.2328632*pow(cos(x[0]),2.) + sin(x[0])*(2.2328632*sin(x[0]) + 0.00012273840877\
```

DOS | INS | Line: 1/19428 | Col: 1

Trefethen's HDHD Challenge, Problem 4 (SIAM News, 2002)



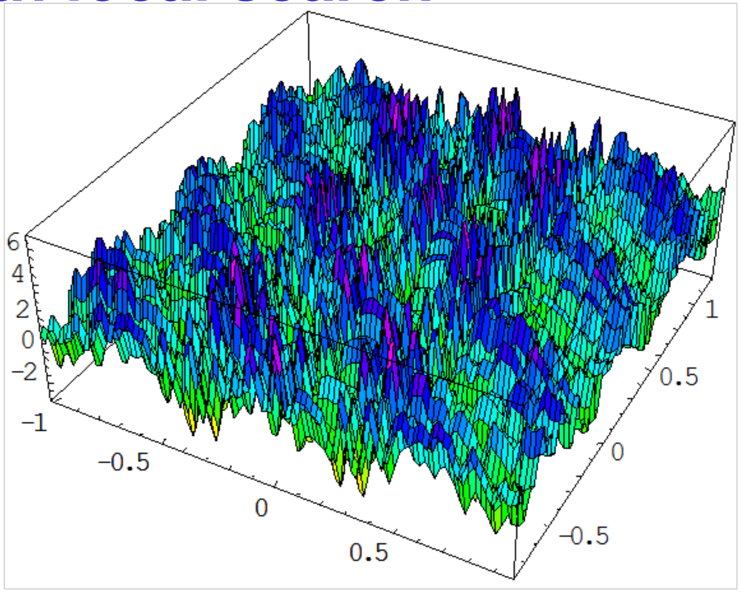
Looks easy from far away, and very difficult when more details are seen

HDHD Challenge, Problem 4 continued

- This model has been used as a test for LGO, as well as with MathOptimizer, MathOptimizer Pro, TOMLAB /LGO, and the Maple GOT
- The solution found by all listed implementations is identical to more than 10 decimals to the announced solution; the latter was originally based on an enormous grid sampling effort combined with local search

x*~ (-0.024627…, 0.211789…)
f*~ -3.30687…

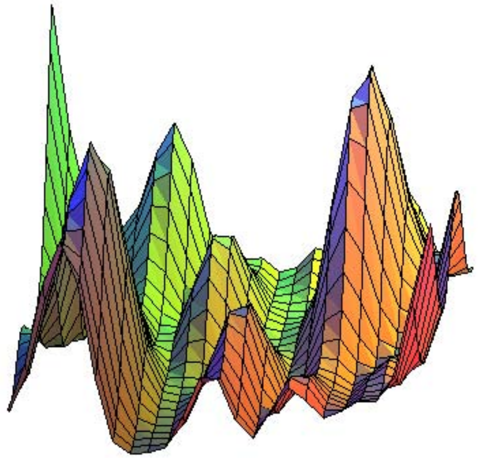
- Close-up picture near to global solution: still looks quite difficult...



Solving Systems of Nonlinear Equations

Equivalent GO model formulation assuming that solution exists; else minimal norm solution sought

$F(x)=0 \leftrightarrow \min \|F(x)\|$



Error function plot

Example solved by
Maple GO Toolbox

x-y+sin(2*x)-cos(y)=0
4*x-exp(-y)+5*sin(6*x-y)+3*cos(3*y)=0

A numerical solution found is

$x \sim 0.0147589760525313926$, $y \sim -0.712474169476650099$

l_2-norm error ~ $1.2213673543564359 < 10^{-16}$

Note: there could be other solutions; systematic search is possible

***Optimization Methods and Software* (2006)**

*Maple 10 - I:\!My Documents\Software Development\Implementations\Maple Optimization and GOT\GOT De

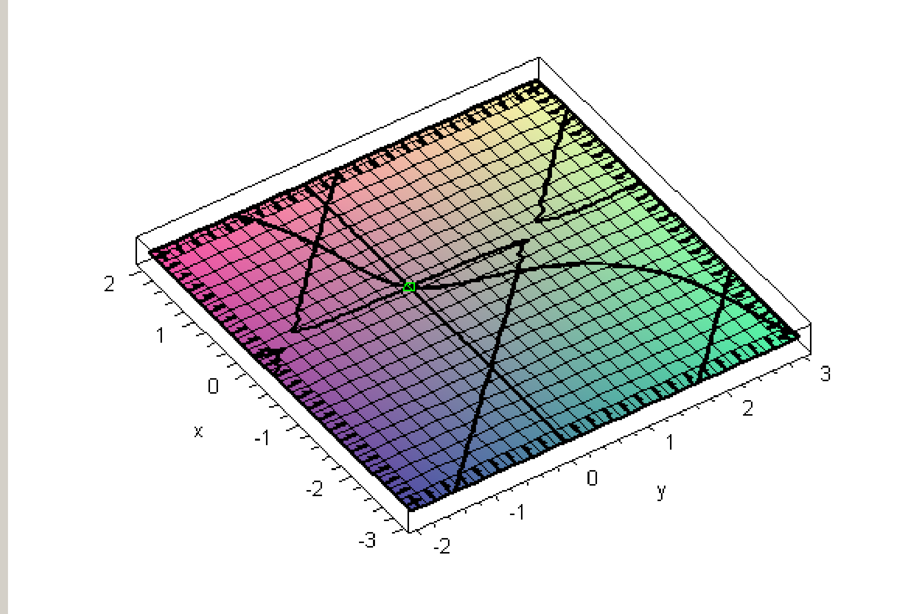
File Edit View Insert Format Table Plot Spreadsheet Sketch Tools Window Help

*GOT_Introduction_and_Examples.mw | *Teaching ORMS with ICSs --- demo examples.mw

Nonlinear Equations

Observe the equations (lines in *x-y* subspace projection), and the solution point found as indicated by the green dot

Global Optimization Plotter



Ranges

Range of x = -3 .. 2 extrema at 2.15276e-10

Range of y = -2 .. 3 extrema at -5.10338e-10

Range of objective values = default .. default extrema of 0

☑ Plot Using Problem Domain

Plot ☑ Plot Constraints ☐ as Surfaces Done

Systems of Nonlinear Equations

```
> eqn1:= x*cos(x-y-z)+y*z=0:
> eqn2:= y-(x+y)*z=0:
> eqn3:= exp(x+y+z)+cos(x-y)-2=0:
> constraints := {eqn1,eqn2,eqn3};
```

$$constraints := \{y-(x+y)\,z=0,\ e^{x+y+z}+\cos(x-y)-2=0,\ x\cos(x-y-z)+y\,z=0\} \quad (7)$$

```
> objective := 0:
> bounds := x=-3..2, y=-2..3, z=-1..4:
> solution := GlobalSolve(objective, constraints, bounds);
```

$$solution := [0.,\ [x=2.15276144907379962\ 10^{-10},\ y=-5.10337879464516135\ 10^{-10},\ z=1.06478223075657034\ 10^{-10}]] \quad (8)$$

```
> eval(constraints, solution[2]);
```

$$\{-5.103378795\ 10^{-10}=0,\ 0.=0,\ 2.152761448\ 10^{-10}=0\} \quad (9)$$

*Maple 10 - I:\!My Documents\Software Development\Implementations\Maple Optimization and GOT\GOT Demo Apps\Teaching ORMS with ICSs --- demo examples.mw - [Server 2]

File Edit View Insert Format Table Plot Spreadsheet Sketch Tools Window Help

Text Math 2D Input Times New Roman 12

*GOT_Introduction_and_Examples.mw *Teaching ORMS with ICSs --- demo examples.mw

The example presented here illustrates the point that the Global Optimization Toolbox can handle a broad range of Maple functions as part of the model formulation. In the example, we shall use the built-in gamma function denoted by GAMMA(x).

The gamma function is defined for Re(z)>0 by

$$\mathrm{GAMMA}(z) = \int_0^{\infty} e^{-t} t^{z-1} dt$$

It is extended to the rest of the complex plane, except the non-positive integers, by analytic continuation. (Consult Maple's Help system for more information.)

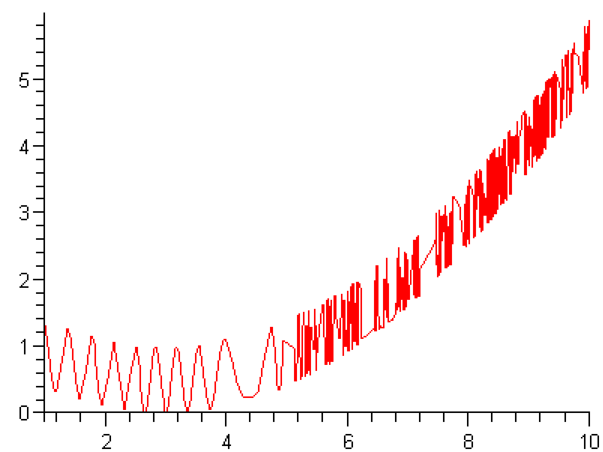
```
> objective := 0.1*(x-3)^2+sin(x^2+5*x-GAMMA(x))^2;
```

$$objective := 0.1\,(x-3)^2 + \sin\left(x^2 + 5\,x - \Gamma(x)\right)^2 \quad (4)$$

```
> bounds:= x=1..10:
> GlobalSolve(objective, bounds);
```

$$[9.33734073493614977\ 10^{-8}, [x = 2.99903427678795830]] \quad (5)$$

```
> plot(objective, bounds);
```



Optimization with Arbitrary Computable Model Functions

A unique - and practically important - feature

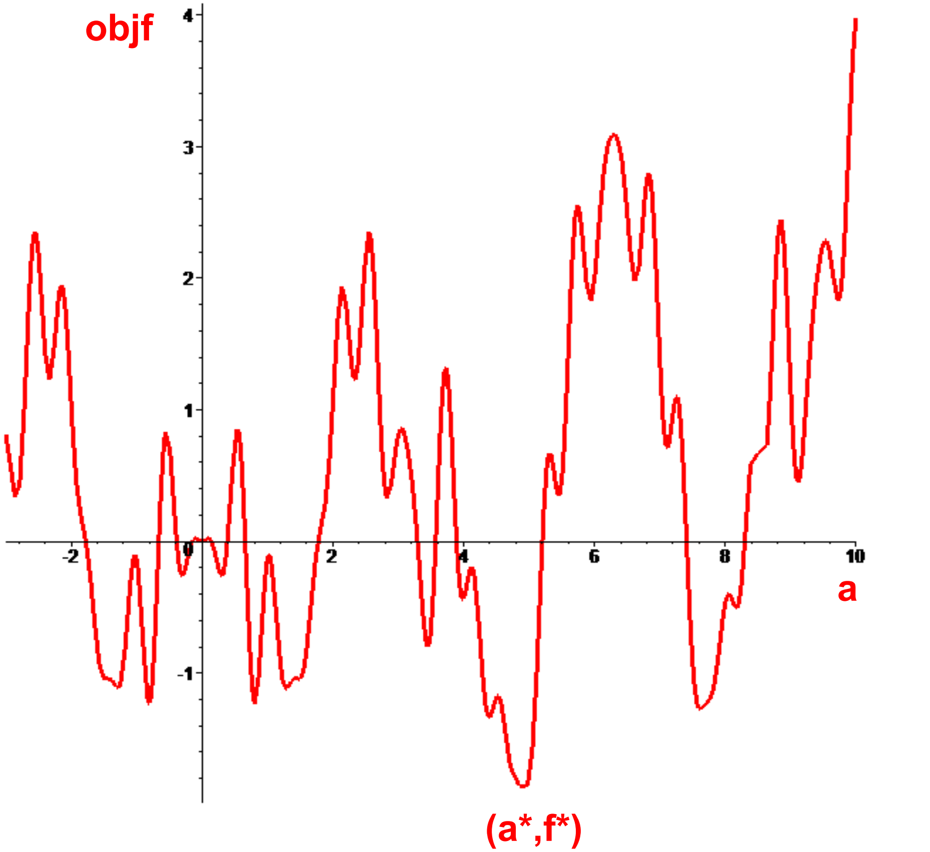
Optimization of A Parametric Integral Expression

```
> objf := a->2*evalf(Int(0.01*x+sin(5*x)*cos(3*x)-sin(3*x)*cos(2*x)+sin(x)*cos(7*x), x=-1.5*a..2*a));
> bounds := -3..10;
> GlobalSolve(objf, bounds)
```

Result found by the GOT:
a*~ 4.92574563473295957
f*~-1.86463327469610008; here f*=objf(a*)
Total number of function evaluations: 1351
Runtime in external LGO solver: 14 seconds
(integration takes time for each input value a)

```
> plot(objf, bounds);
```

The plot shows the high multi-modality of the parametric integral value (as a function of a)
Notice the location of the global solution (a*,f*)



$$objf := a \to 2\ \text{evalf}\left(\int_{-1.5\,a}^{2\,a} 0.01\,x + \sin(5\,x)\cos(3\,x) - \sin(3\,x)\cos(2\,x) + \sin(x)\cos(7\,x)\,dx\right)$$

Nonlinear Model Calibration in Presence of Noise

An example model (in *Mathematica* notation), inspired by a client's (medication dosage effect) study:

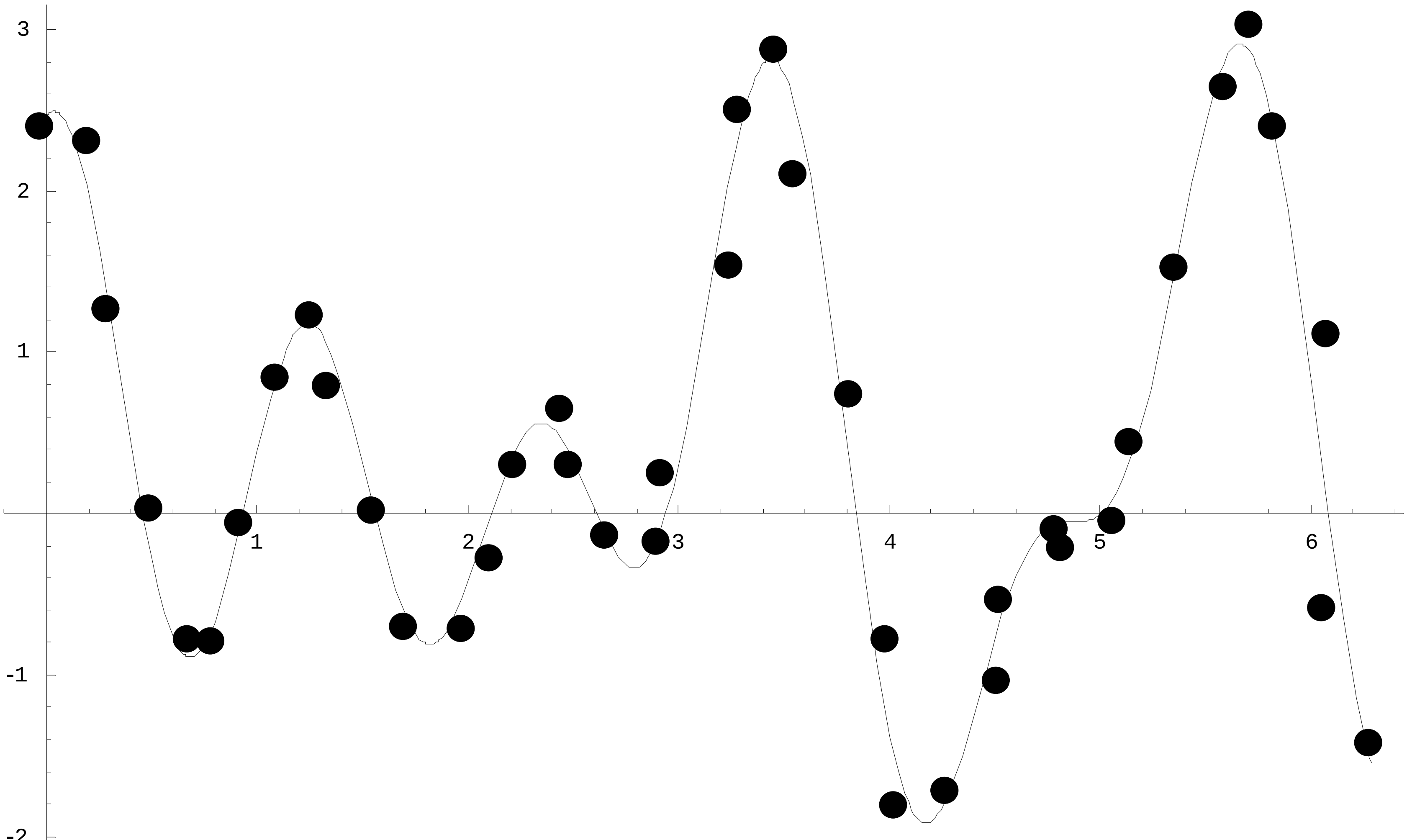
a+Sin[b*(Pi*t)+c]+Cos[d*(3Pi*t)+e]+Sin[f*(5Pi*t)+g]+ξ

The parameters a,b,c,d,e,f,g are randomly generated from interval [0,1]; ξ is a stochastic noise term from *U*[-0.1,0.1]

Subsequently, the optimal parameterization is recovered numerically by MathOptimizer: this gives superior results, in comparison with Mathematica's corresponding built-in local solver functionality (NonlinearFit)

***Optimization Methods and Software* (2003)**

Calibration of Nonlinear Model in Presence of Noise (cont.)

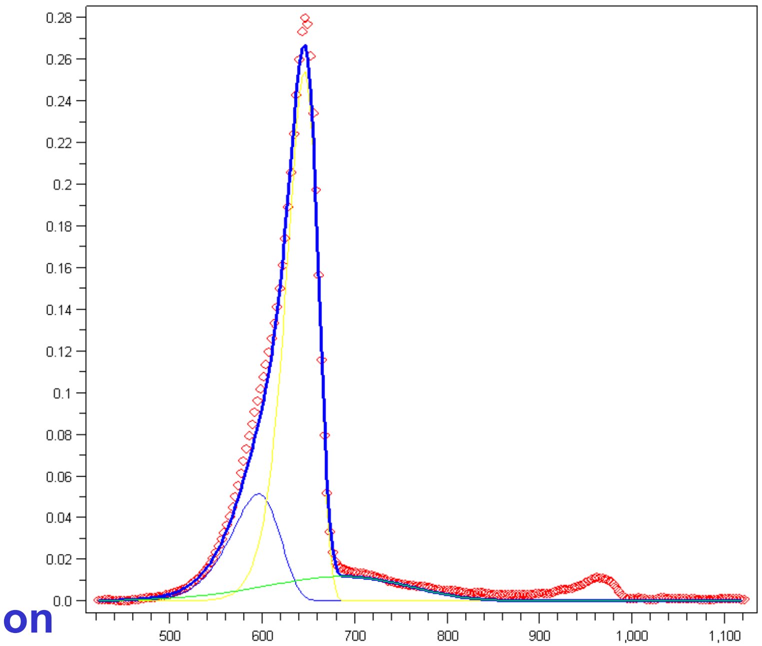


a+Sin[b*(Pi*t)+c]+Cos[d*(3Pi*t)+e]+Sin[f*(5Pi*t)+g] +ξ

Global search based fit, obtained by MathOptimizer

Arrhenius Probe Model Calibration

Credits: Grigoris Pantoleontos et al., Chemical Engineering Department, Aristotle University of Thessaloniki, Greece



ln(y) = A-Ea /RT **Arrhenius formula**

Describes temperature dependence of reaction rate coefficient y

A multi-component version of the formula:

$$y_i = c_i A_i \exp\left(-E_i / \left(R\,Temp[j]\right)\right) \cdot \left(1 - R_i[j]\right)$$

Here $R_i[j]$ is calculated from another (rather complicated) expression

The study by GP *et al.* is aimed at the determination of the parameters c_i, A_i and E_i i=1,2,3 by comparing the computed model output values to the experimental ones

The figure shows the initially given data points (red circles), the component curves (green, blue, yellow), and the resulting curve (bold blue); a fairly good fit

The solution of this computationally intensive example (9 variables to calibrate, very large search region, hundreds of data points, difficult model functions to compute) took about an hour on a desktop PC (in 2007); GP used the Maple GOT

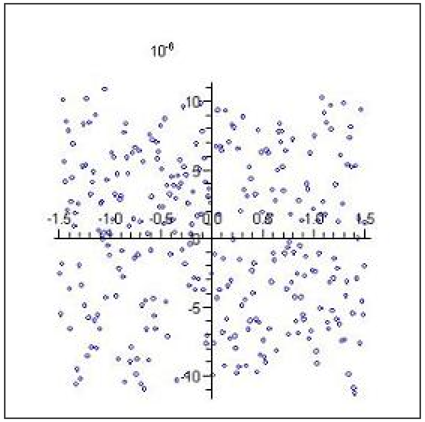
Maple Application Center
- Research PowerTools
- Education PowerTools
- Browse Applications
- Advanced Search
- Top Rated Applications
- New Applications
- Most Downloaded Applications
- Tips & Techniques
- Submit Your Work

Other Resources
- Maple 10 Training
- Maplesoft Books
- Maple Reporter

More Information
- Contact Maplesoft

Top Rated Applications | New Applications | Most Downloaded | maple APPLICATION CENTER
Conduct a search: simple : advanced

Aspherical Lens Surface Identification - Non-Linear Fitting with the Global Optimization Toolbox



Member Rating:	(rate this application)
Author:	Maplesoft Cybernet Systems Co., Ltd .
Application Type:	Maple Worksheet
Publish date:	August, 2006
Related Products:	Maple 10 Global Optimization
Language:	English
Related Link:	http://www.maplesoft.com/products/toolboxes/glo...
Related Book:	Applied Nonlinear Optimization in Modeling Environments Global Optimization : Scientific and Engineering Case Studies Global Optimization in Action : Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications

Options:
- View HTML version
- Download Maple Code *(.zip, 67.9kb)*
- E-mail to a colleague

Abstract:
In this Application Demonstration, we investigate Aspherical Lenses and apply non-linear fitting to obtain an accurate representation of the given data in the form of a function, using the GlobalOptimization Toolbox for Maple.

Related Application Categories
Mathematics : Operations Research
Operations Research : Regression
Science : Physics
Statistics & Data Analysis : Regression

Member rating: not rated *(min. 5 ratings required)*
You have not rated this application.
Rate it: - Rate - SUBMIT
View top rated applications

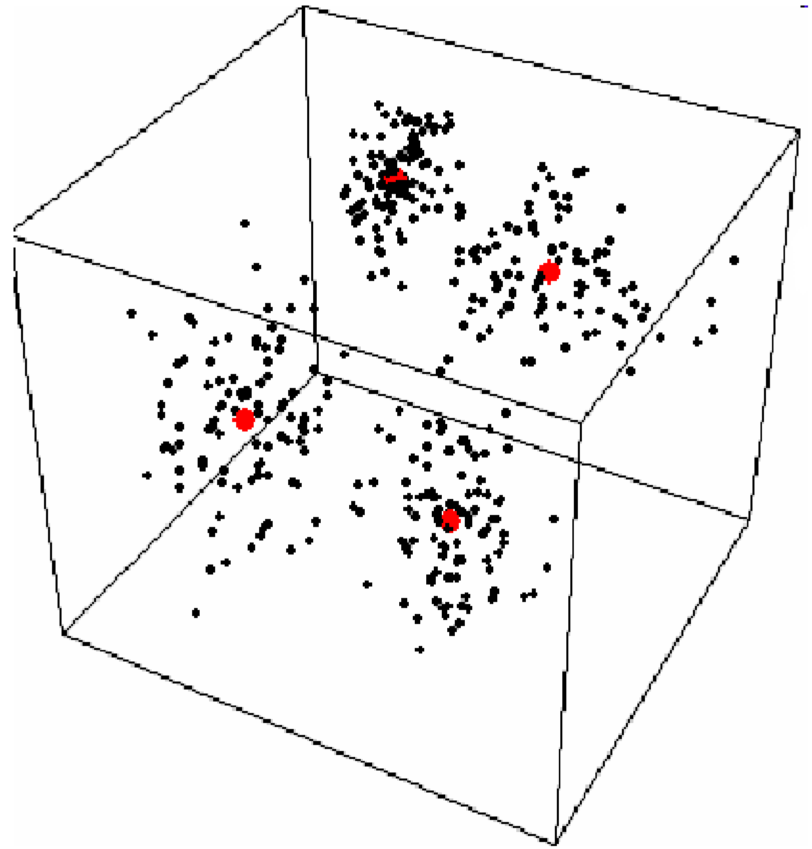
Data Classification (Clustering) by Global Optimization

Details: *Global Optimization in Action*, Ch. 4.5

Classification objective:
Find the "most homogeneous" or "most discriminative" grouping of a given set of entities (see black dots shown in rhs figure)

This can be done numerically, by globally optimizing the position of cluster centres (medoids, see red dots)

For any given (candidate) medoid configuration, one can use e.g. the "nearest neighbor" rule to associate the points x_j with the cluster centres c_k

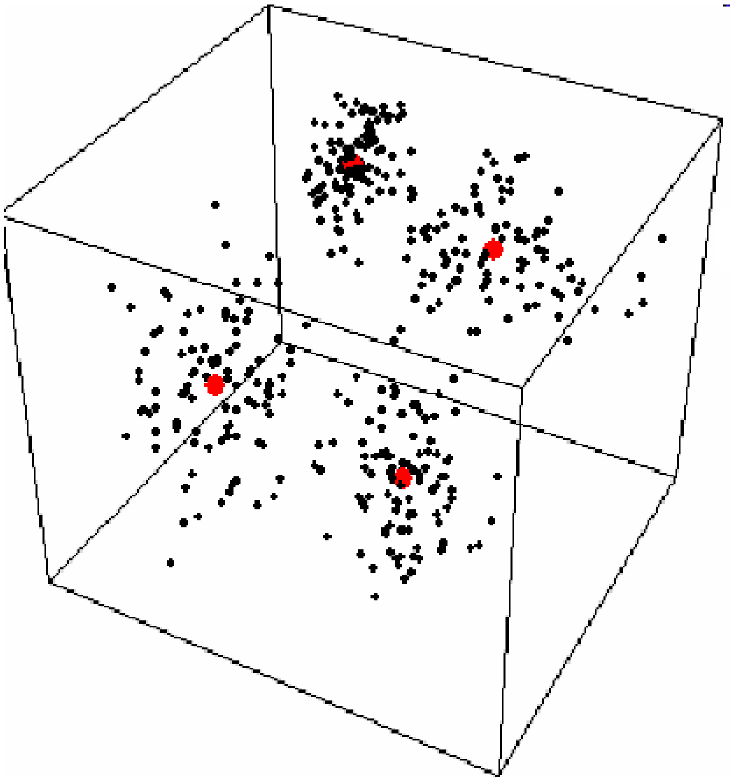


Example:

400 3-dimensional points classified into 4 clusters

Data Classification (Clustering) by Global Optimization

For a given (prefixed) number of clusters, one can use the following model to identify the cluster medoids $\{c_k\}$:



$$\min \Sigma_k \Sigma_j ||c_k - x_j|| \quad \text{s.t.} \quad cl_k \le c_k \le cu_k$$

For any given set of $\{c_k\}$ and each x_j, the index $k = k(x_j)$ is chosen following the "nearest neighbor" rule

In general, this is a GO problem

Model description and detailed discussion with numerical examples in *Global Optimization in Action*, Ch. 4.2

Key advantage of the GO model formulation compared to the usual combinatorial optimization based approach: model dimension is *ndim*nclusters*=12... *vs. nentities*=400... The example is solved in seconds, the approach also scales up well

Maxi-Min and Related Point Arrangements

In a large variety of applications, one is interested in the "best possible covering" arrangement of points in a set

- numerical approximation methods
- design of experiments for expensive "black box" models
- potential energy models (in physics, chemistry, biology)
- crystallography, viral morphology, and other areas

For illustration, consider a maxi-min model instance

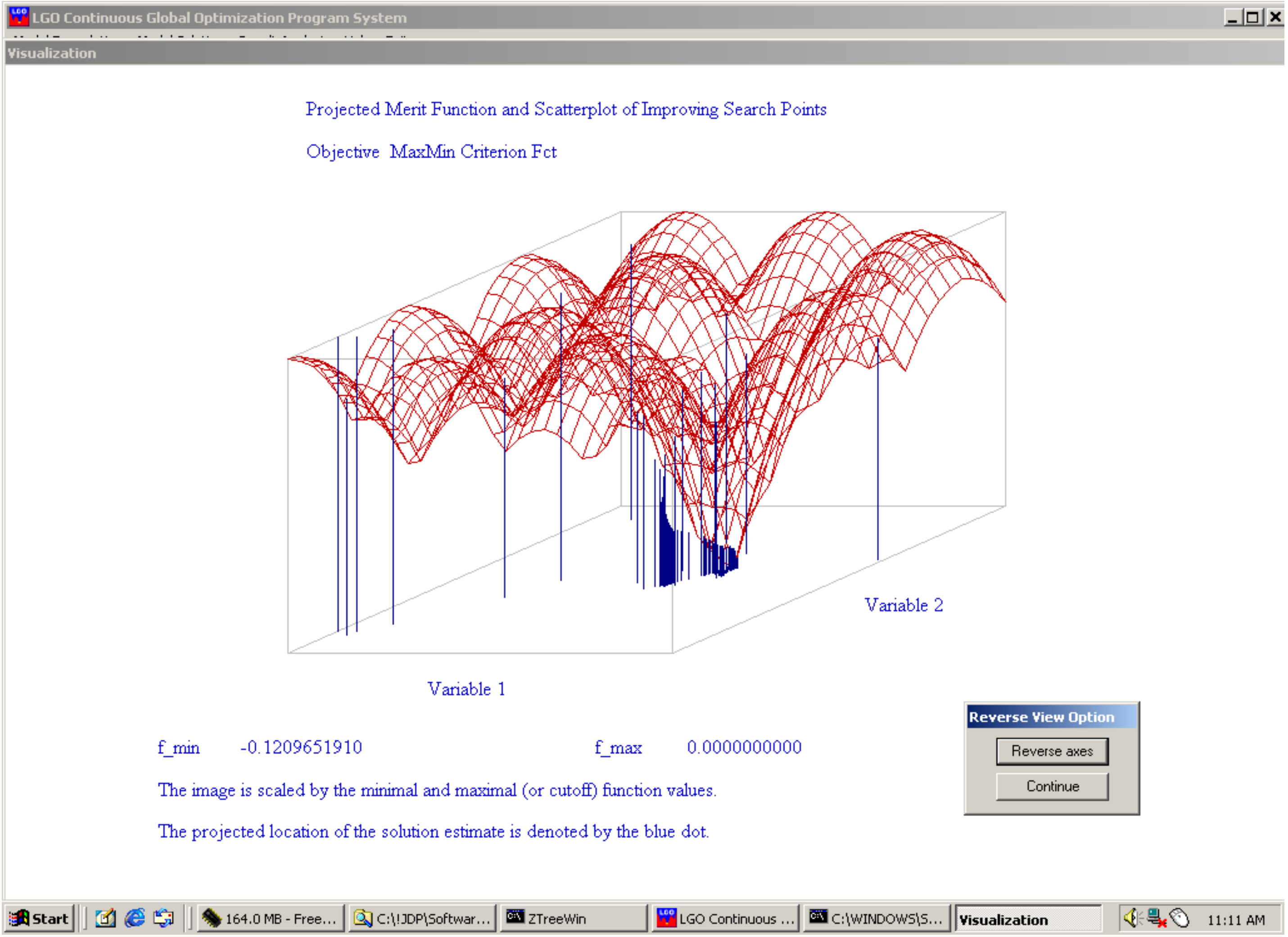
$$\max_{\{x_i\}} \{ \min_{1 \le i < k \le m} \|x_i - x_k\| \} \quad xl \le x_i \le xu \qquad x_i \in R^d \qquad i=1,\ldots,m$$

Additional restrictions, alternative feasible sets, and other quality criteria can also be considered

Permutations avoided by lexicographic point arrangements

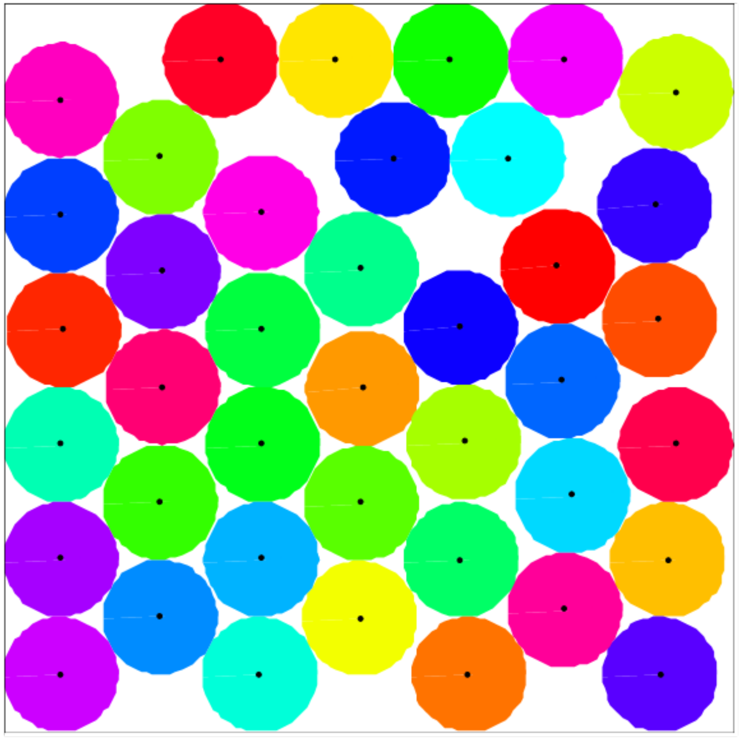
In general, difficult non-convex models arise

MaxiMin Point Arrangement Problem



LGO IDE: model visualization (*m*=13, *d*=2)

Packing Uniform Size Circles in the Unit Square



Example: 40 circles; optimized radius of circles r~0.0787391…
Solution time using MOP: less than 5 minutes (3 GHz PC)
No postulated structural info is exploited: MOP used "blindly"

Non-Uniform Size Circle Packings in a Circle

In such problems, we study the packing of different size circles in an embedding circle. Since this model formulation typically has infinitely many solutions *per se*, we will additionally try to bring the circles as close together as possible.

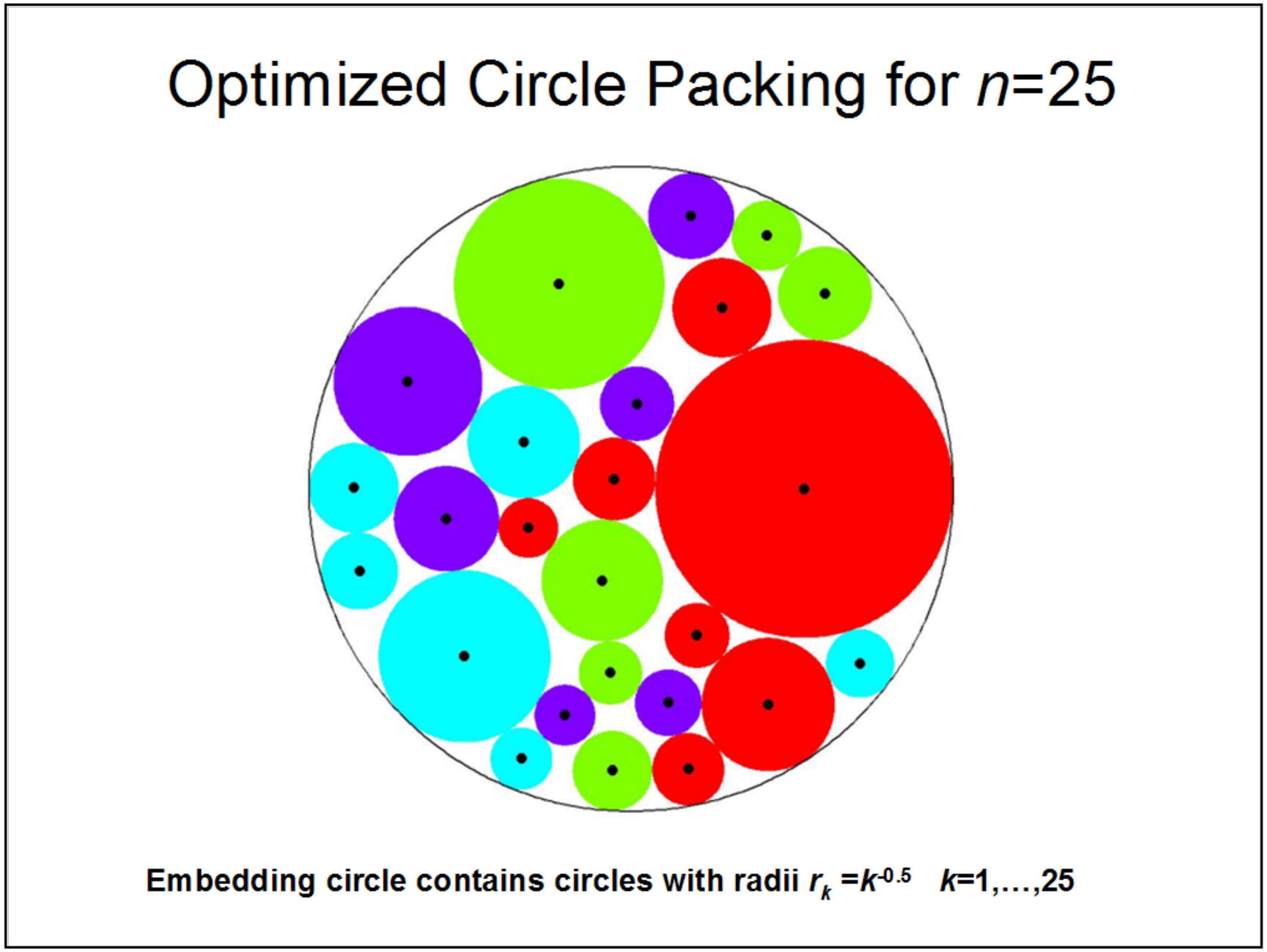
The primary objective (obj1) is to find the circumscribed circle with the smallest radius; the secondary objective (obj2) brings the circles close together by minimizing the average distance among all circle centers.

A scaled linear combination of these two objectives is used. Note that alternative formulations are also possible, and that rotational symmetries of solutions can also be avoided (by added constraints), thereby making the solution of a specific model formulation essentially unique.

Applications: wires packed together in a cable, dashboard design…

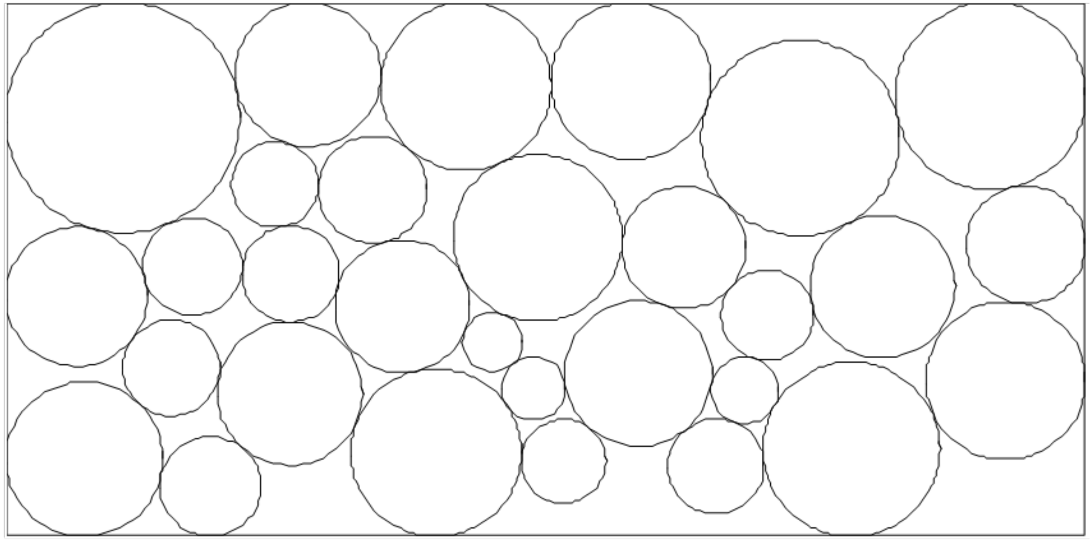
***Mathematica in Education and Research* (2005), *The Mathematica Journal* (2006) Co-author: Frank J. Kampas**

Non-Uniform Size Circle Packings



Embedding circle contains circles with radii $r_k = k^{-0.5}$ $k=1,\ldots,25$

General Circle Packings in Minimal Volume (Length) Container



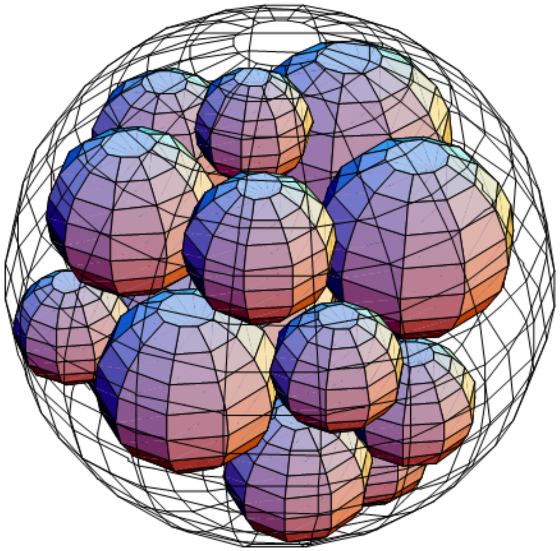
Example: given 30 circles with radii below ; given height of container; find minimal container width

rlist = {1.275, 1.67, 2.05, 1.739, 1.399, 1.18, 0.564, 1.374, 1.237, 0.845, 1.484, 0.868, 0.807, 1.551, 1.274, 0.855, 1.493, 1.281, 1.491, 0.747, 1.085, 1.044, 0.955, 1.404, 1.292, 0.853, 0.76, 0.527, 0.592, 0.887}
Best known radius is 17.291; MOP default option based radius 18.915 in ~ 20 secs; relative quality ~ 91% Further structure based refinements are possible and recommended

Pintér and Kampas (*Mathematica in Edu. and Res.,* 2005), Castillo, Kampas, and Pintér (*EJOR*, 2008), Kampas and Pintér (*WTC* presentations, 2006, 2007; downloadable notebooks)

Sphere Packings in Optimized Sphere

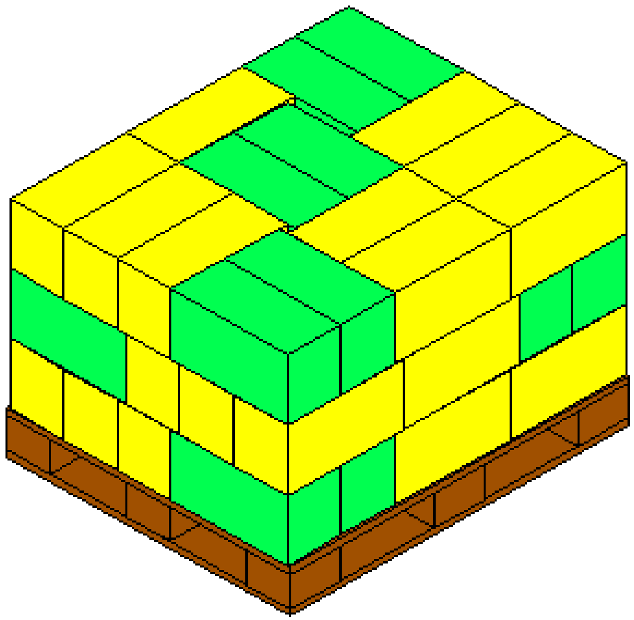
Given a collection of spheres, find the minimal size sphere that includes all of these, in a non-overlapping arrangement

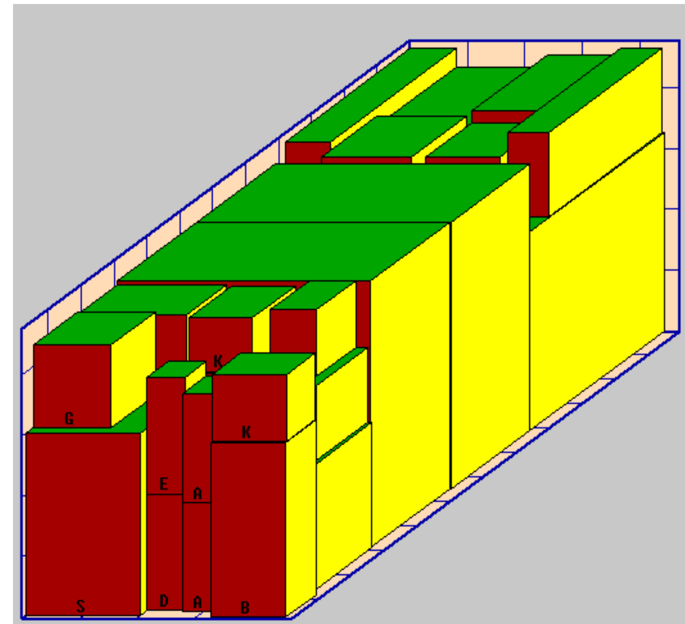


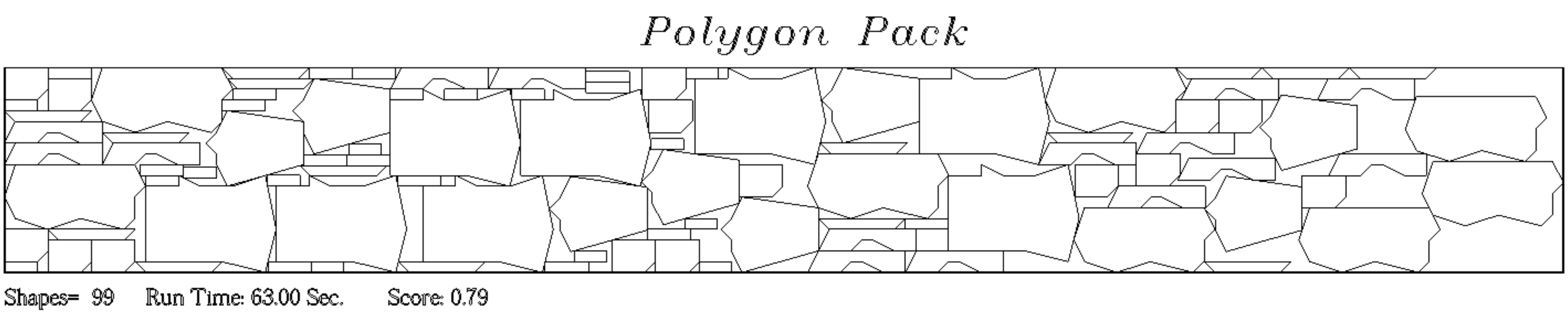
Example: 15 spheres with radii $r_i=i^{-1/3}$ solved numerically by MOP
Radius of embedding sphere: ~1.96308, 1.5 sec runtime on a 2007 PC, vs. ~10 min when using the built-in *Mathematica* function NMinimize

More details: Kampas and JDP, *WTC* talks + notebooks

Industrial Packings and Polygon Cutting Stock Plans







Credits: Ignacio Castillo, University of Alberta, Edmonton, Canada (now at WLU)

Packing Objects with Orientation and Other Characteristics (Constraints Such as Mass Balance): An Example

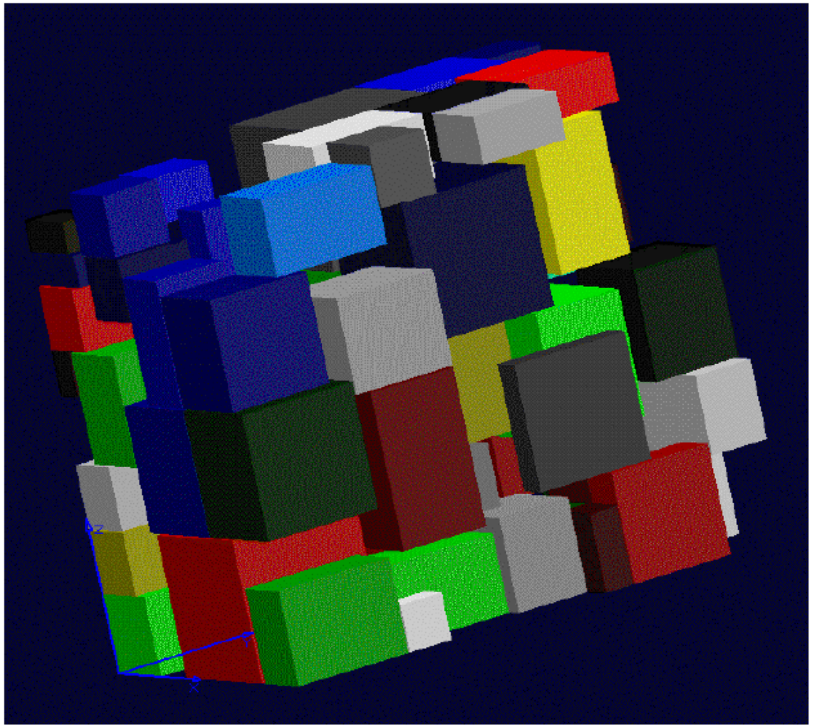


Fig. 8 Case study balanced solution

Credits: G. Fasano, MIP-based heuristics for non-standard 3D-packing problems with additional constraints; technical report and papers by GF

Potential Energy Models

Point arrangements on the surface of unit sphere

$x_i=(x_{i1},x_{i2},x_{i3})$	$\|\|x_i\|\|=1$
$x(m)=\{x_1,\ldots,x_m\}$	m-tuple (point configuration)
$d_{jk}=d(x_j,x_k)$ $1\leq j<k\leq m$	Euclidean distance

Model versions considered:

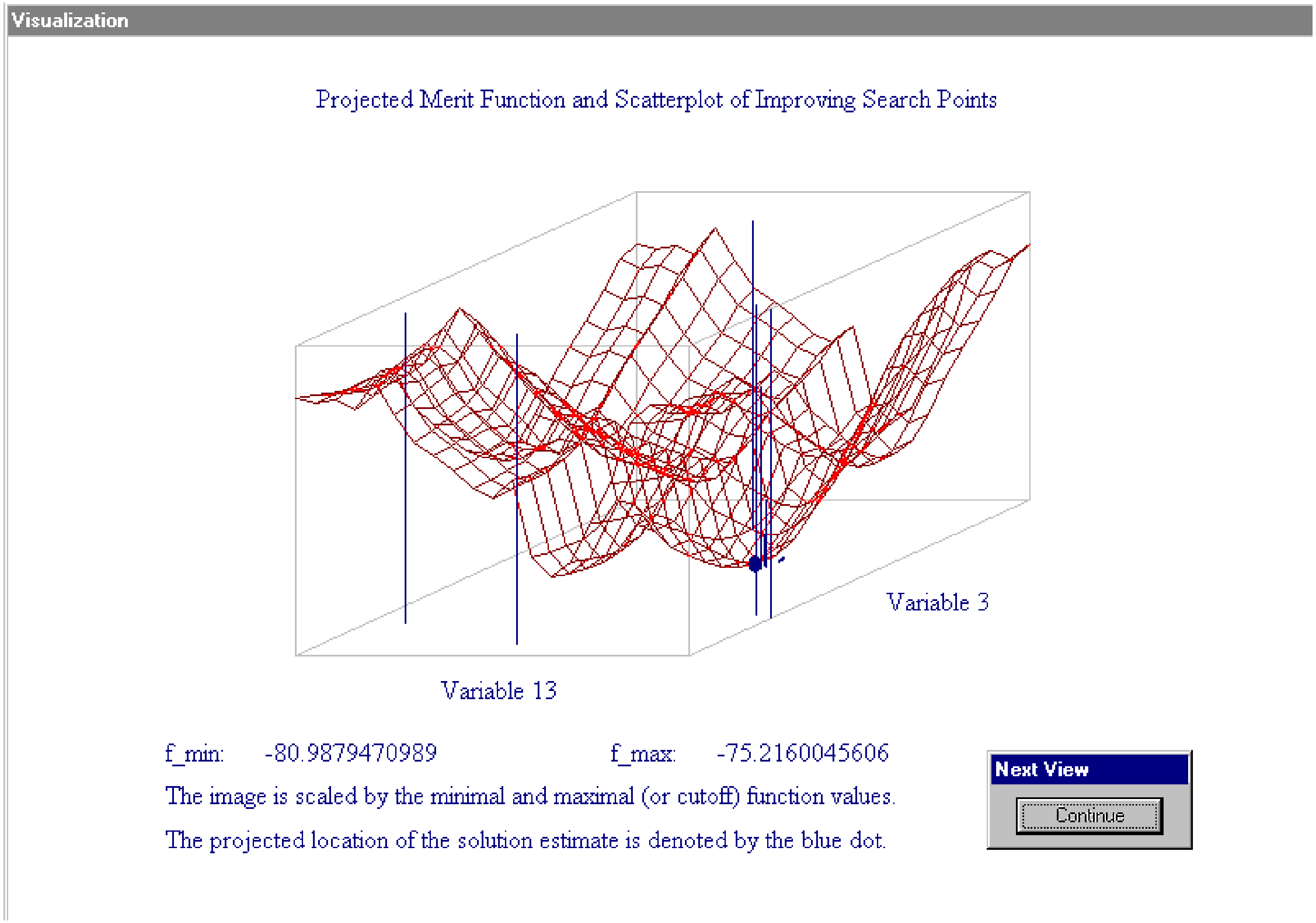
$\max \sum_{1\leq j<k\leq m} \log(d_{jk})$	Fekete (elliptic or log-potential)
$\min \sum_{1\leq j<k\leq m} 1/d_{jk}$ $(d_{jk}>0)$	Coulomb-Fekete
$\max \sum_{1\leq j<k\leq m} d_{jk}^{a}$	Power sum, $0<a<2$
$\max \{\min_{1\leq j<k\leq m} d_{jk}\}$	Tammes (hard sphere)

In all cases, the objective function is multi-extremal;

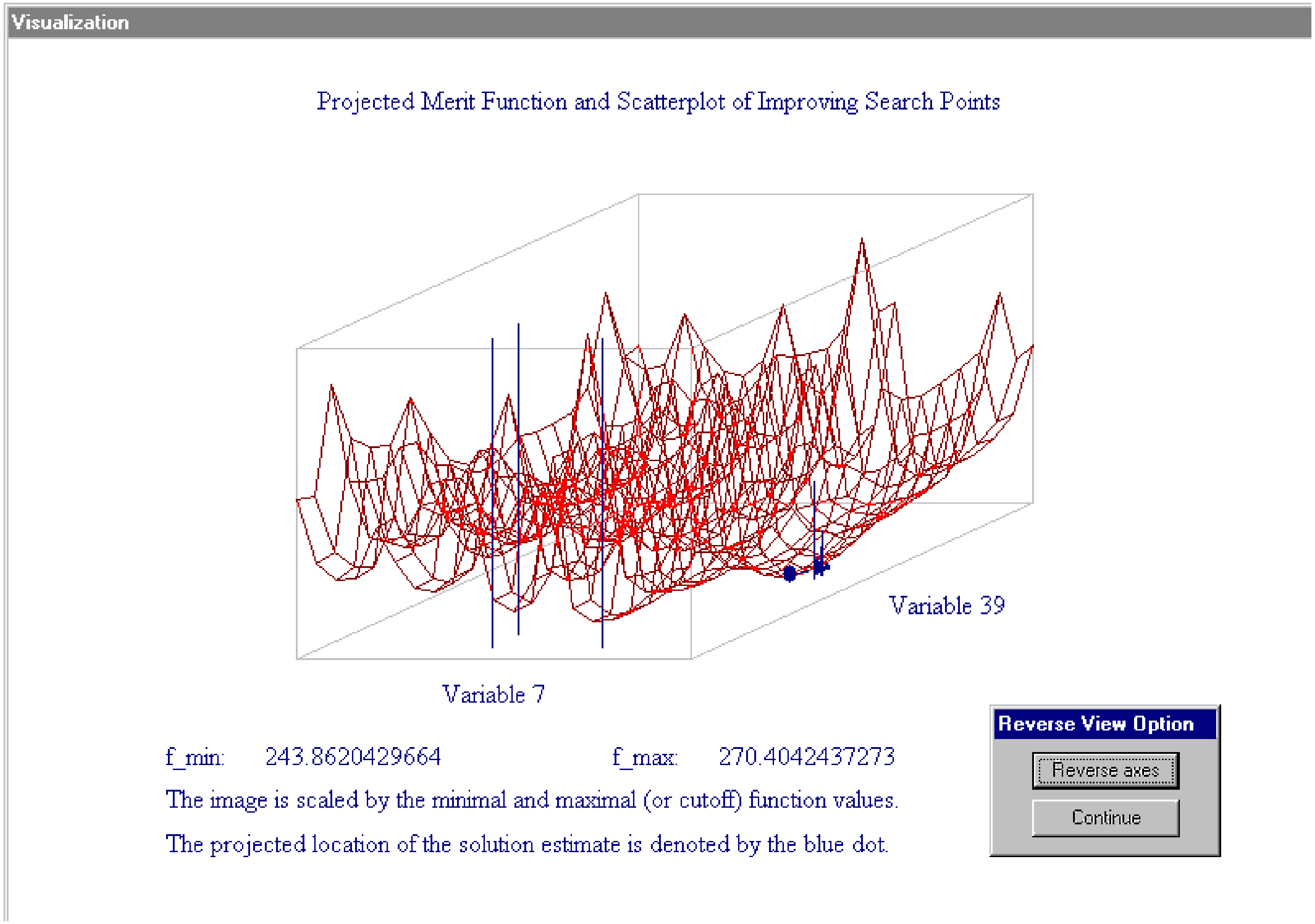
Combined GO + expert knowledge based solution approaches

Applications: math, physics, chemistry, biology,...

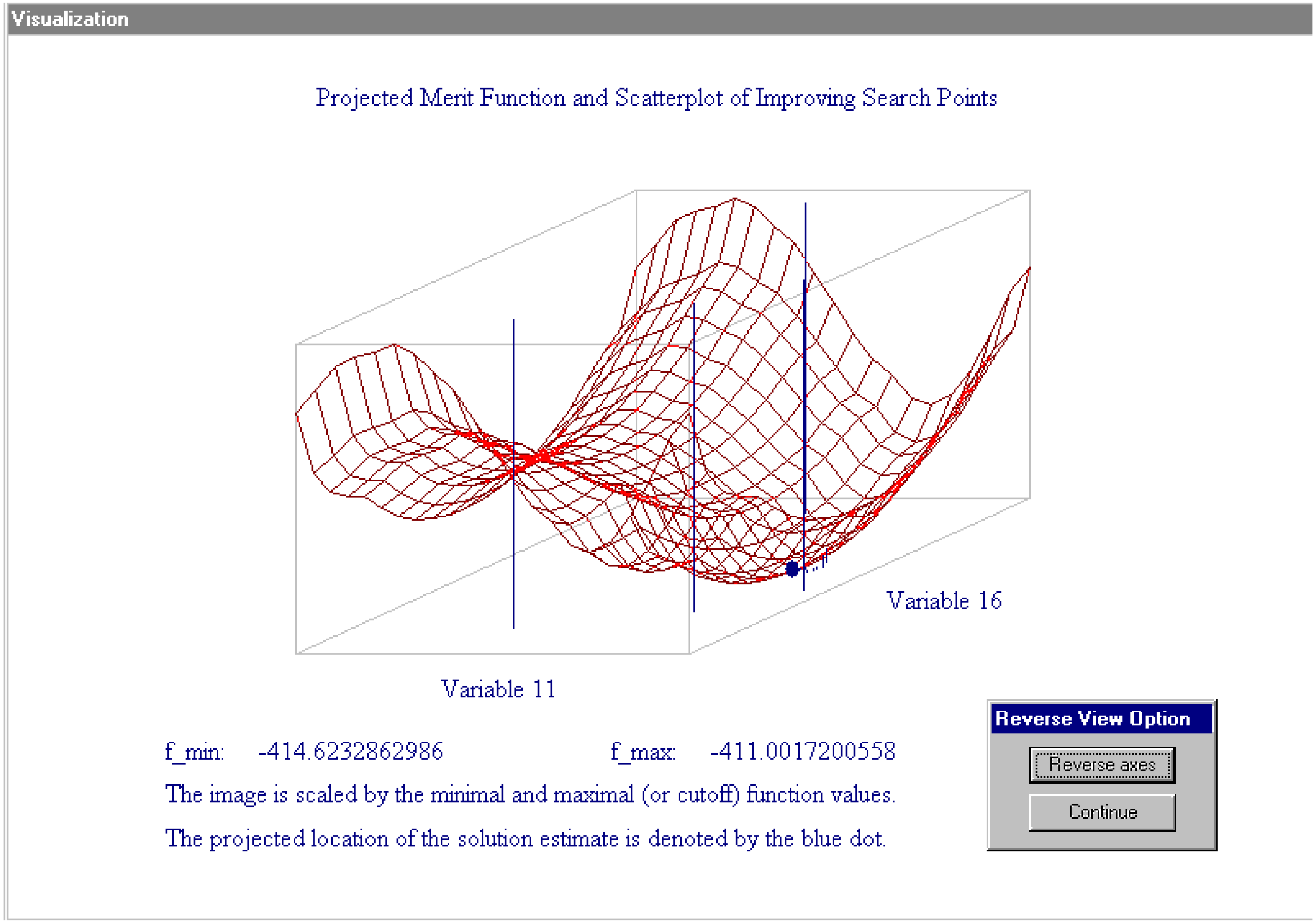
Elliptic Fekete model (*m*=25 points)



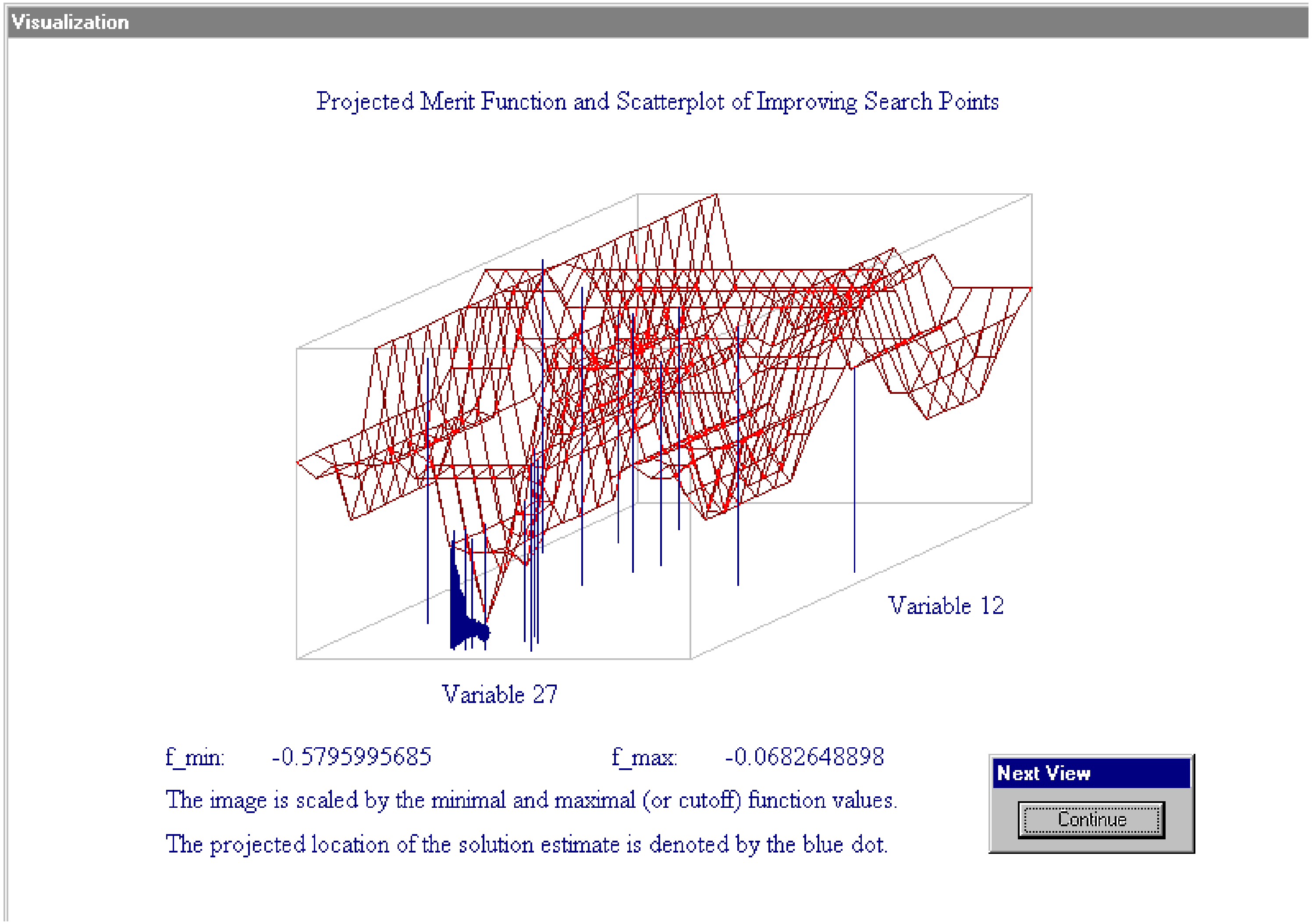
Coulomb-Fekete model (*m*=25 points)



Powersum model (*m*=25 points)



Tammes model (*m*=25 points)



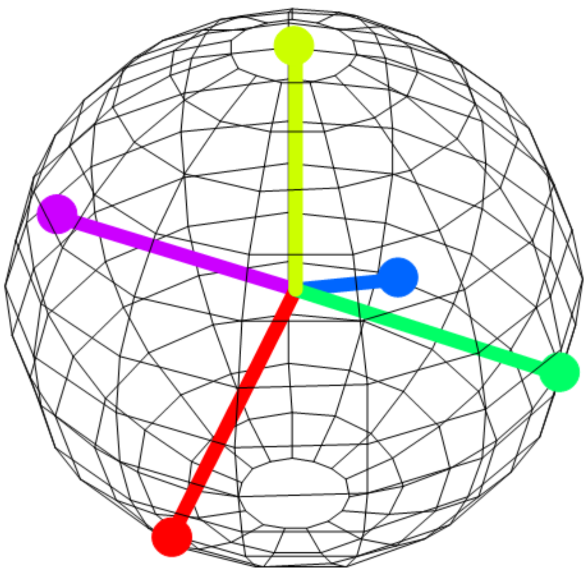
Log-potential (Fekete Potential) Model

In this example charged particles (points that are repelling each other) are confined to stay on the surface of the unit sphere

Our objective is to find their optimized configuration, using the Fekete potential model

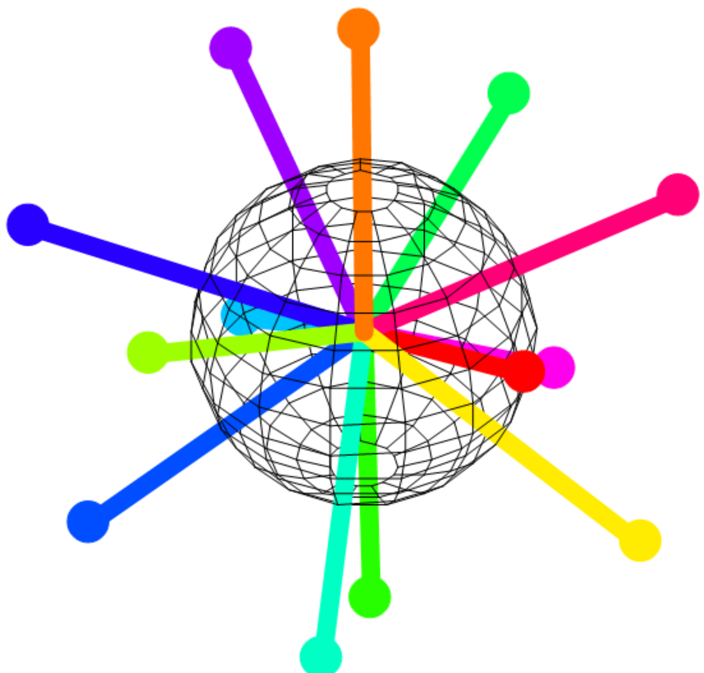
5-particle example: as expected, the arrangement is symmetrical with respect to the configuration elements

Finding such an arrangement is not trivial for arbitrary *m*-point configurations: GO techniques can be used



Mathematica model implementation By Frank J. Kampas

Log-potential Model: 13 Points



Solution found by MathOptimizer expressed in spherical coordinates

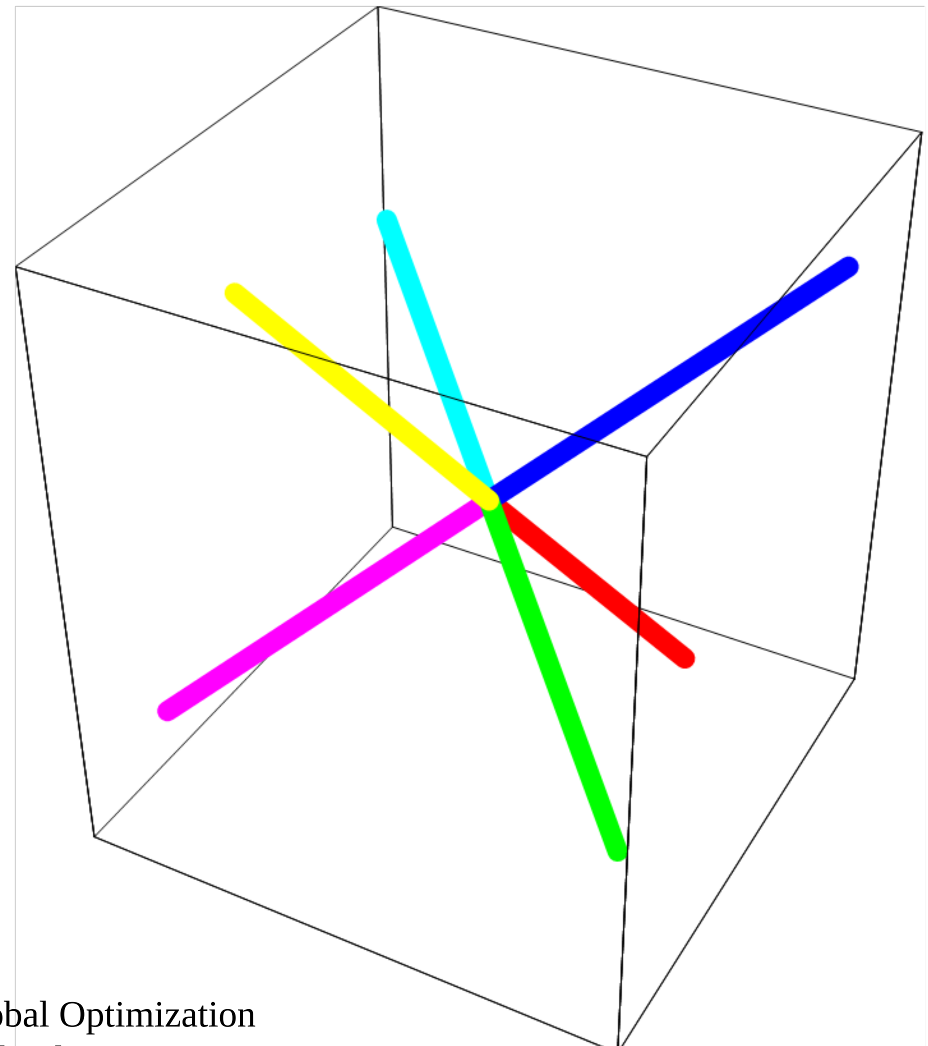
{0.882683, {{θ[1] → 0, ϕ[1] → 0}, {θ[2] → 0.641862 π, ϕ[2] → 0}, {θ[3] → 0.350165 π, ϕ[3] → 1.4503 π},
{θ[4] → 1.01645 π, ϕ[4] → 0.810165 π}, {θ[5] → 0.401262 π, ϕ[5] → 0.49749 π},
{θ[6] → 0.641862 π, ϕ[6] → 1.62032 π}, {θ[7] → 0.666635 π, ϕ[7] → 0.810158 π},
{θ[8] → 0.687546 π, ϕ[8] → 1.23168 π}, {θ[9] → 0.401262 π, ϕ[9] → 1.12283 π},
{θ[10] → 0.333318 π, ϕ[10] → 0.810158 π}, {θ[11] → 0.687545 π, ϕ[11] → 0.388636 π},
{θ[12] → 0.350165 π, ϕ[12] → 0.170016 π}, {θ[13] → 0.349818 π, ϕ[13] → 1.81016 π}}}

Electrons in a Sphere

In this example charged particles that are repelling each other are confined to stay within the unit sphere

Objective: find their optimized configuration

6-electron example: again (as expected), the arrangement found is symmetrical; all optimized points are on the surface of the sphere



Mathematica model implementation
By Frank J. Kampas

Summary of Numerical Studies

Putative global optima collected (to use in comparisons) from the Web site of AT&T Bell Laboratories: these results have been derived by extensive numerical experiments

Our illustrative results (LGO 2000) successfully approximate the corresponding best known results to more than 99.99 precision for the log-potential, Coulomb, and power sum models; LGO solution time was ~ 10-15 sec (P4 1.6 GHz PC)

Hard sphere model solution quality was only ~90% of best: Results significantly improved since that time; more recent published results using LGO and MathOptimizer Pro

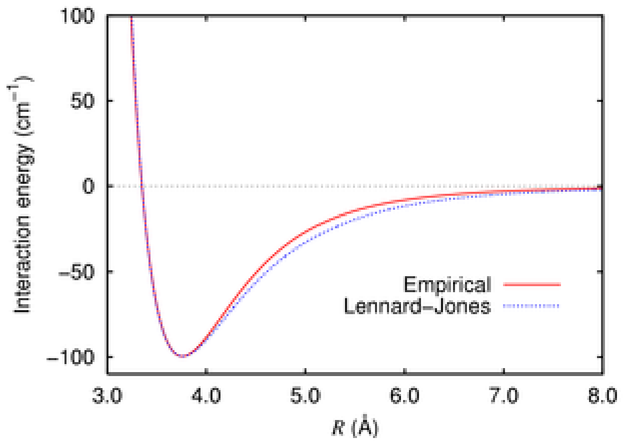
Model variants and illustrative results appeared in *Annals of Operations Research* (2001); *J. of Computational and Applied Mathematics* (2001)
Co-authors of *JCAM* paper: Walter Stortelder and Jacques de Swart
Subsequent work and reports/papers/talks with Frank J. Kampas

Lennard-Jones Potential Energy Model

Credits: Wikipedia

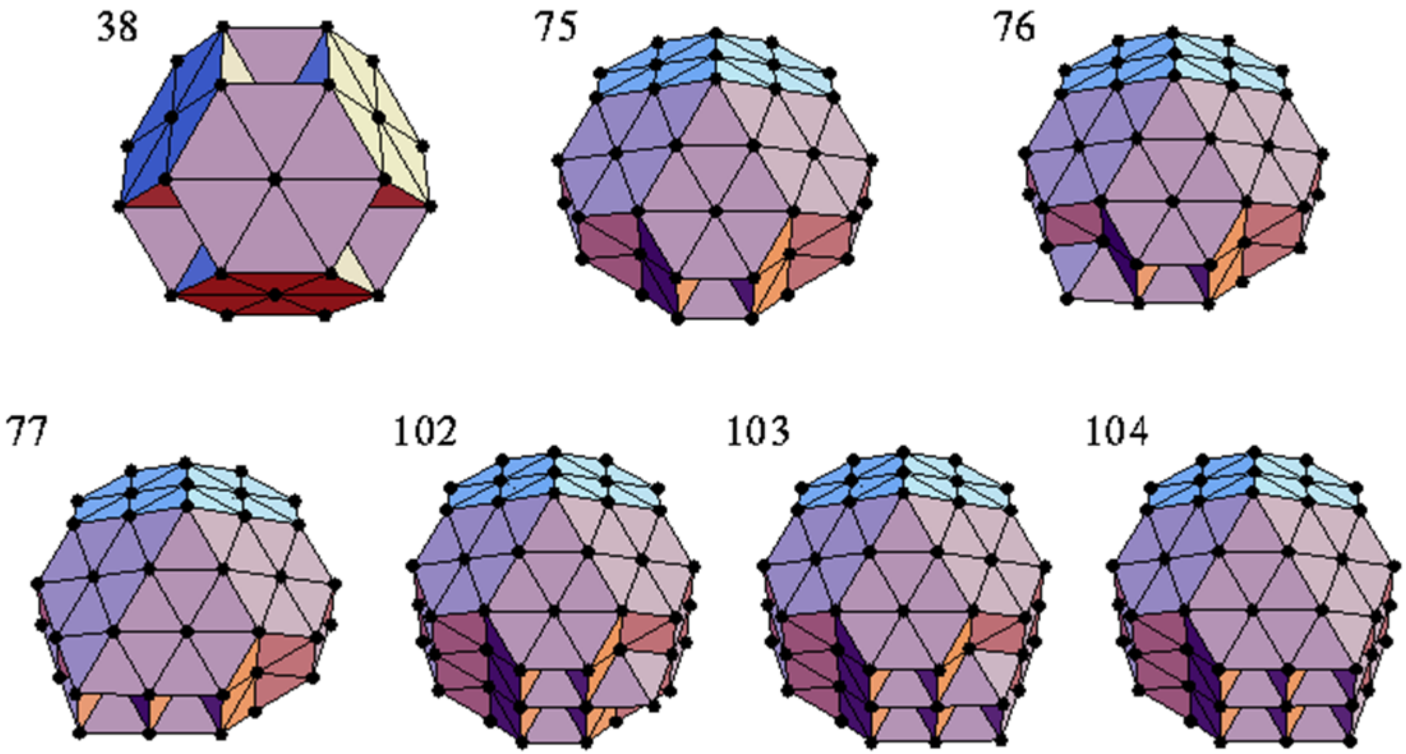
A pair of neutral atoms or molecules is subject to two distinct forces: an attractive force at long ranges (van der Waals force) and a repulsive force at short ranges (Pauli repulsion). The Lennard-Jones potential is a simple mathematical model that approximates these two forces. The L-J potential is of the form

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right]$$



Here ϵ is the depth of the 'potential well'; σ is the distance at which the inter-particle potential is zero; and *r* is the distance between the particles. The aggregated (pairwise) interactions model of a group of particles leads to difficult global optimization problems: these models are used both in GO tests and in (physics, chemistry, biology) research

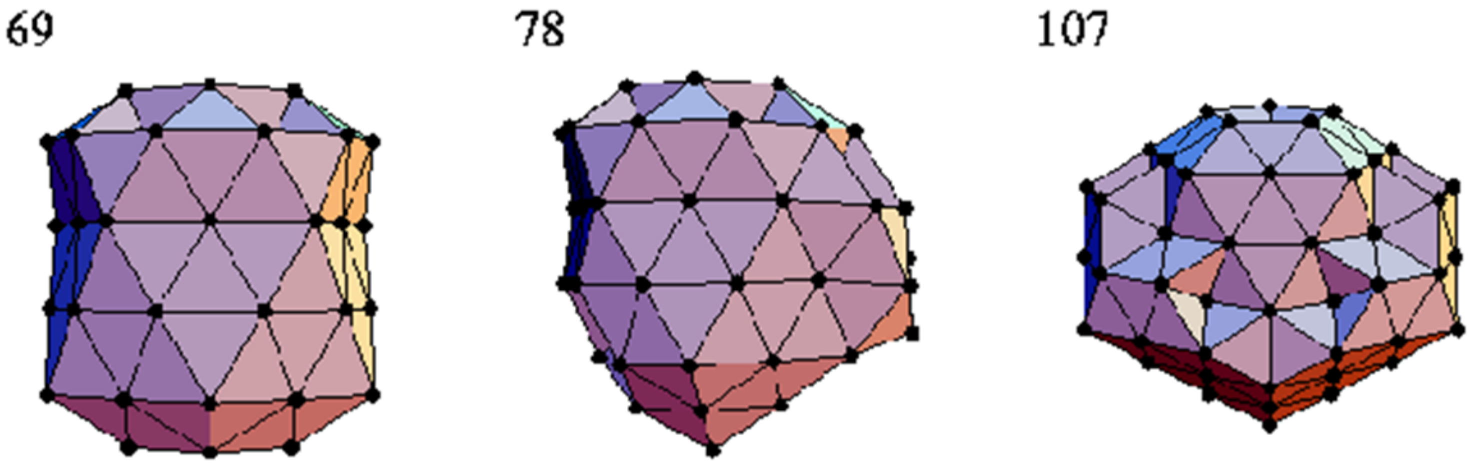
Lennard-Jones Model and Optimized Configurations



Lennard-Jones clusters for some challenging cases

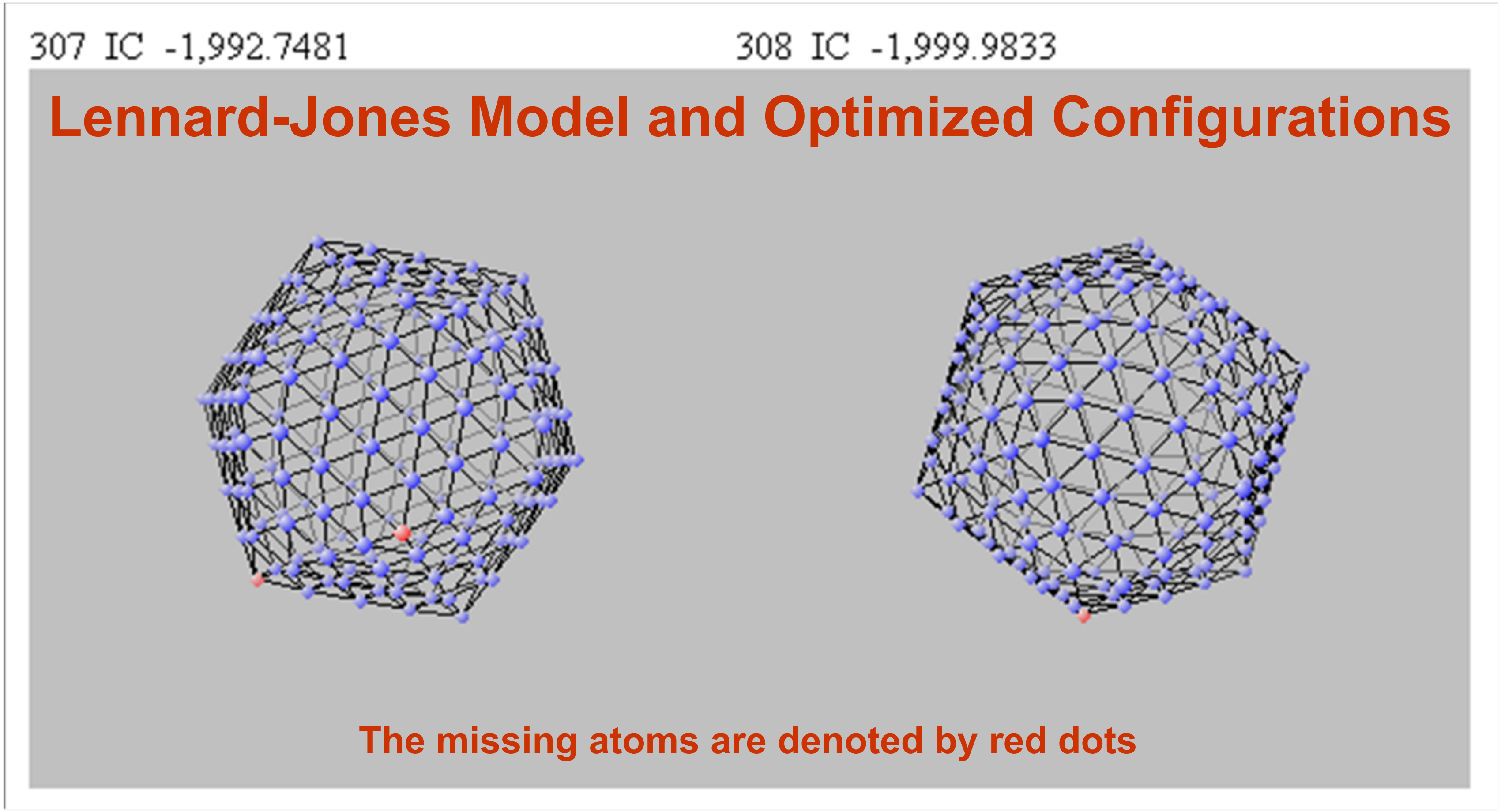
Credits: Jon Doye, University of Cambridge

Lennard-Jones Model and Optimized Configurations



Lennard-Jones clusters for further interesting cases

Credits: Jon Doye, University of Cambridge



Lennard-Jones clusters for higher-dimensional cases

Credits: Carlos Barron (University of Houston)

The optimal geometry of Lennard-Jones clusters: Computer Physics Comm. (1999), 148-309.

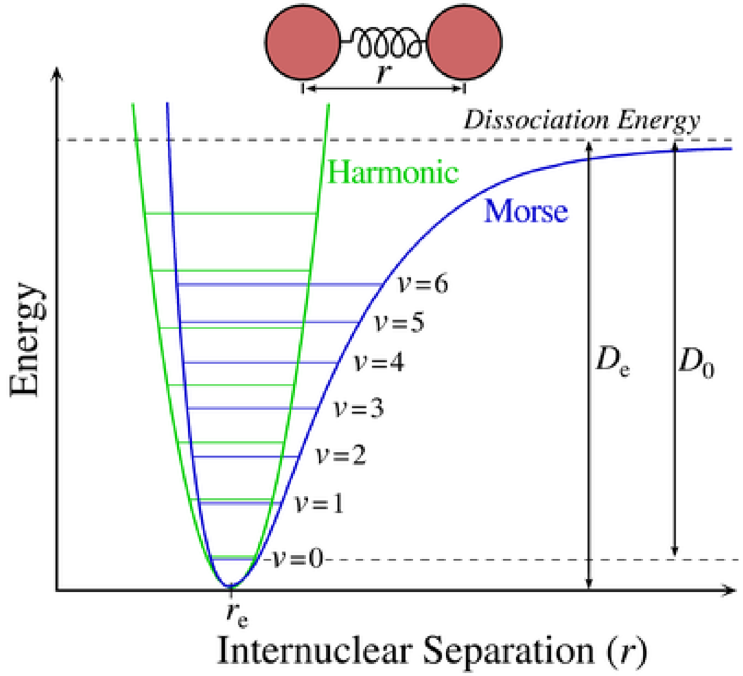
Authors: Romero, D., Barron, C., and Gomez, S.

Morse Potential Energy Model

Credits: Wikipedia

The Morse potential energy function (model) is of the form

$$V(r) = D_e(1 - e^{-a(r-r_e)})^2$$



Here

r **is the distance between the atoms,**

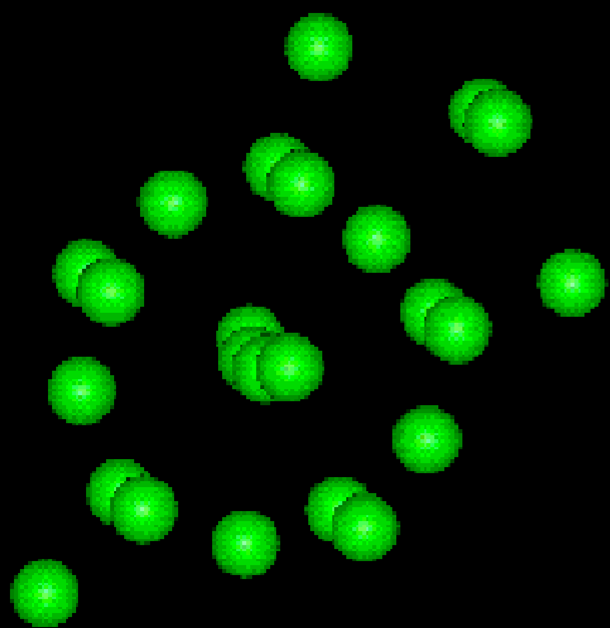
r_e is the equilibrium bond distance,

D_e is the well depth (defined relative to the dissociated atoms), and

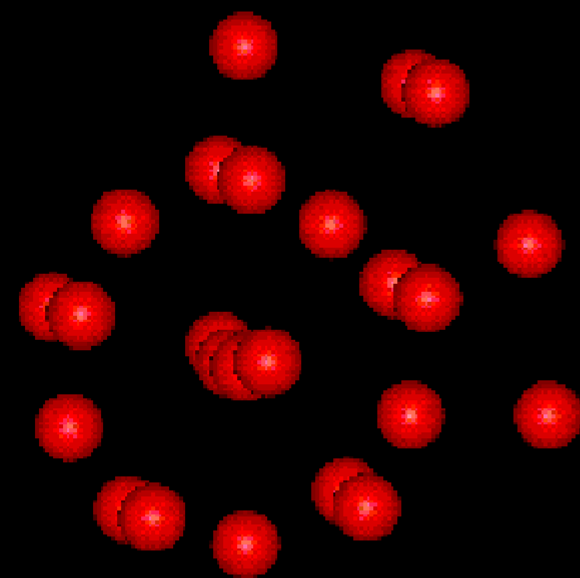
a **controls the 'width' of the potential.**

Again, the minimal energy model of a group of particles leads to difficult global optimization problems that are used both in GO tests and in (physics, chemistry, biology) research

New putative
optimal configuration



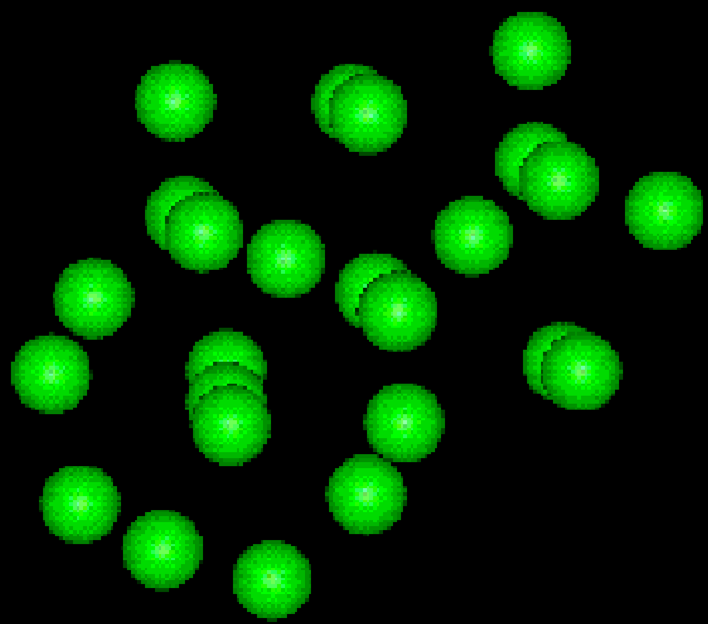
Previous putative
optimal configuration



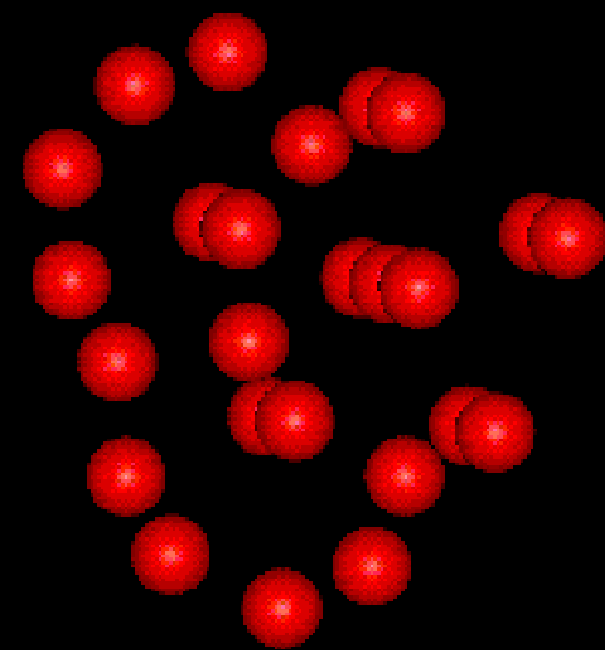
Globally optimized Morse clusters, n=24

Credits: Locatelli and Schoen, 2003

New putative
optimal configuration



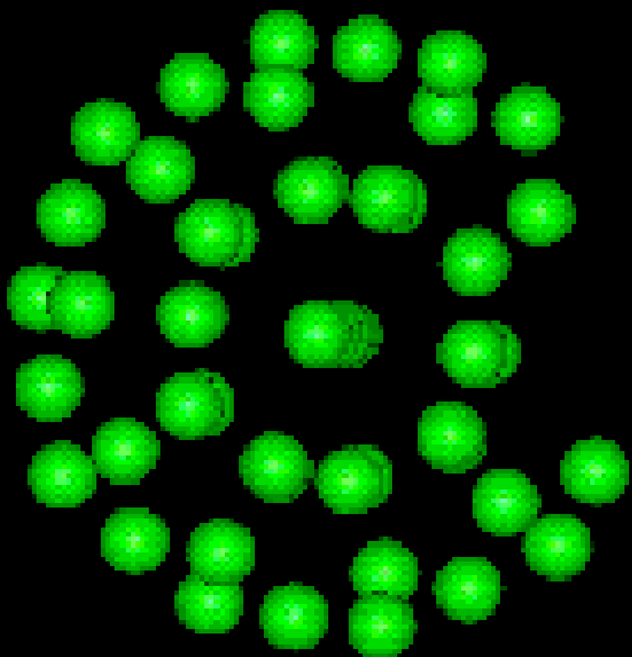
Previous putative
optimal configuration



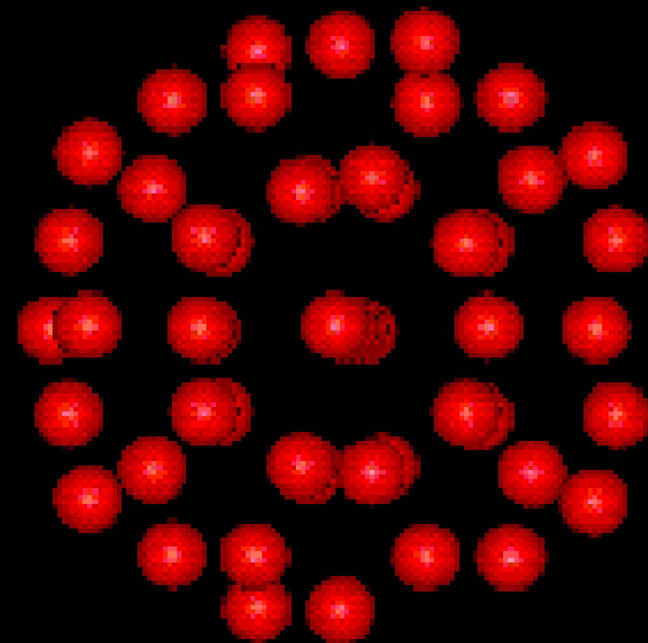
Globally optimized Morse clusters, n=25

Credits: Locatelli and Schoen, 2003

New putative
optimal configuration



Previous putative
optimal configuration



Globally optimized Morse clusters, n=51

Credits: Locatelli and Schoen, 2003

Molecular Alignment and Docking Using *ab initio* Quality Scoring

Docking potential drug candidates into active sites of enzymes in receptor based drug design, or aligning molecules into abstract external fields or to other molecules in ligand based drug design, represents one of the biggest challenges in contemporary drug R&D.

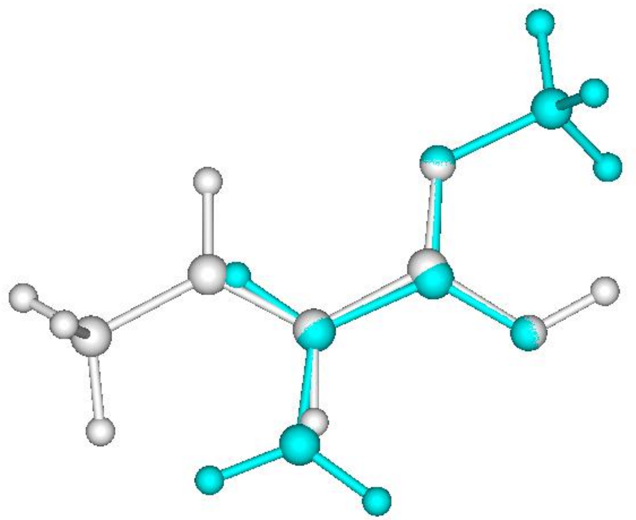
An accurate and efficient molecular alignment technique is presented by the authors named below. It is based on first principle electronic structure calculations. This new scheme maximizes quantum similarity matrixes in the relative orientation of the molecules.

The authors have been using LGO to find the optimal alignment; as a result, they have found noticeably improved alignments.

Credits: László Füsti-Molnár and Kenneth M. Merz

Department of Chemistry, Quantum Theory Project, University of Florida

Molecular Alignment and Docking Using *ab initio* Quality Scoring

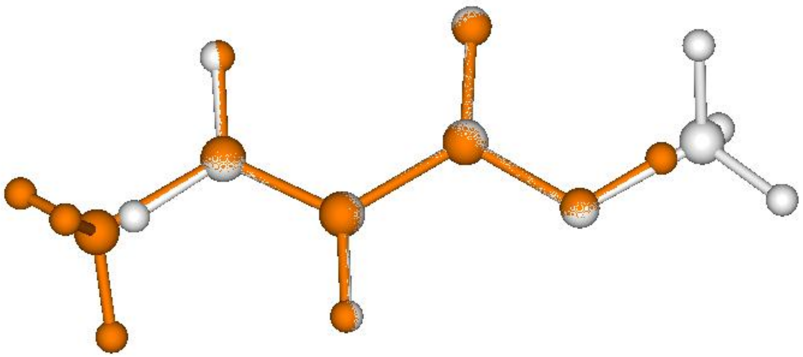


A local optimum in the alignment of Methylacrylate to Crotonic acid

(The color of Crotonic acid is set to white)

Credits: László Füsti-Molnár and Kenneth M. Merz
Department of Chemistry, Quantum Theory Project, University of Florida

Molecular Alignment and Docking Using *ab initio* Quality Scoring



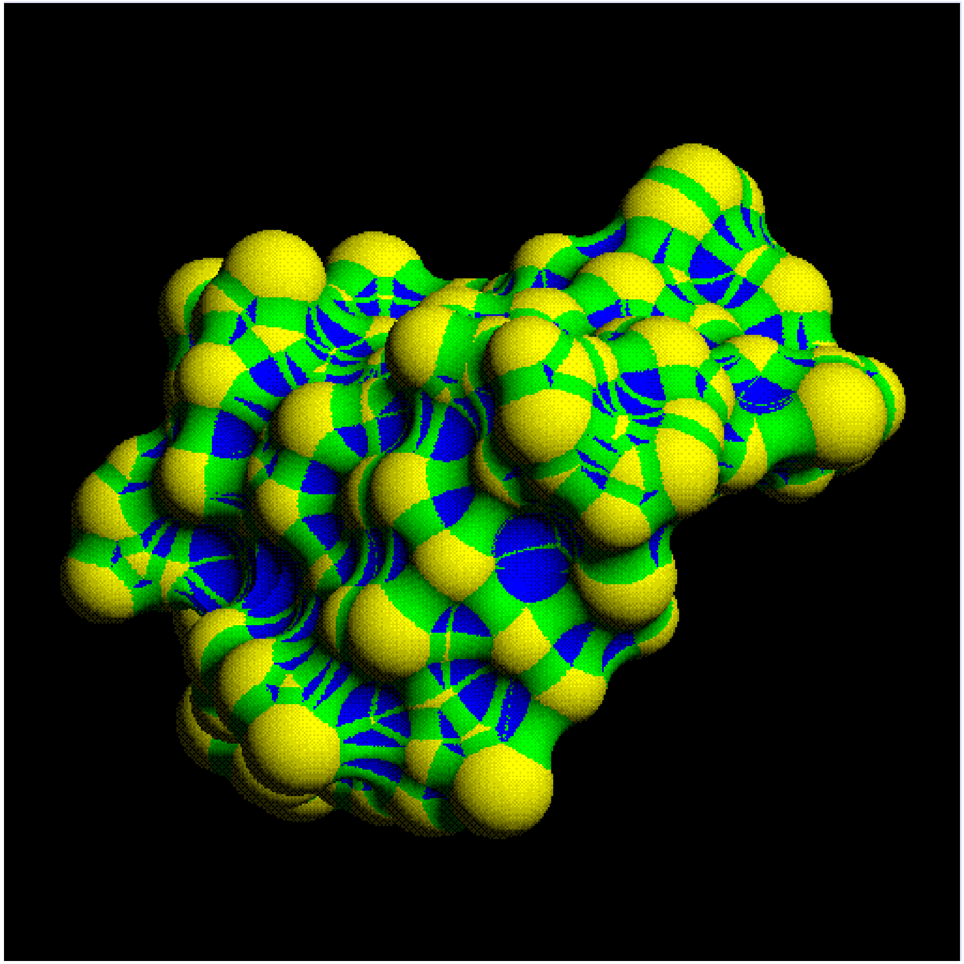
The globally optimized alignment of Methylacrylate to Crotonic acid

(The color of Crotonic acid is set to brown)

Credits: László Füsti-Molnár and Kenneth M. Merz

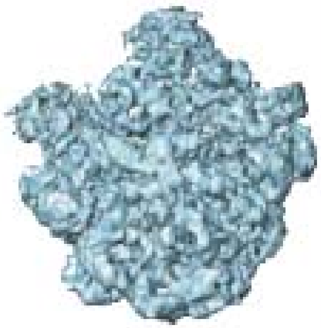
Department of Chemistry, Quantum Theory Project, University of Florida

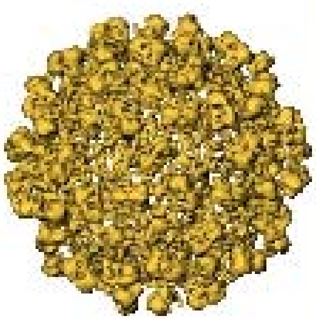
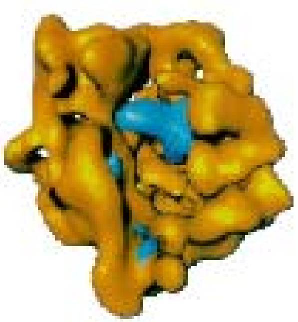
Further Application Perspectives in Chemistry and Biology – The Real Deal…



Example: The molecular surface of crambin

Credits: Molecular Surface Graphics, http://www.netsci.org/Science/Compchem/feature14.html



Atomic Structures of Macromolecules

Credits: Marin van Heel, Imperial College, London, UK

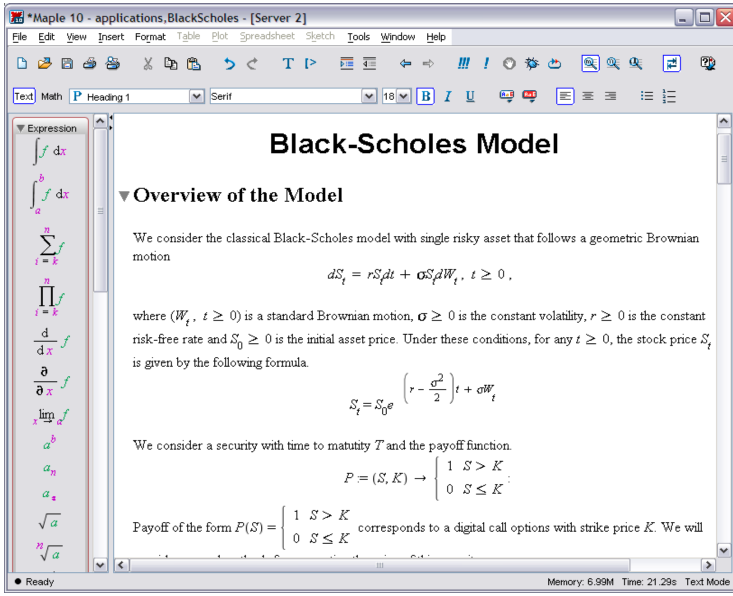
These (and great many other similar, biologically relevant) structures result from a natural tendency towards self-organizing

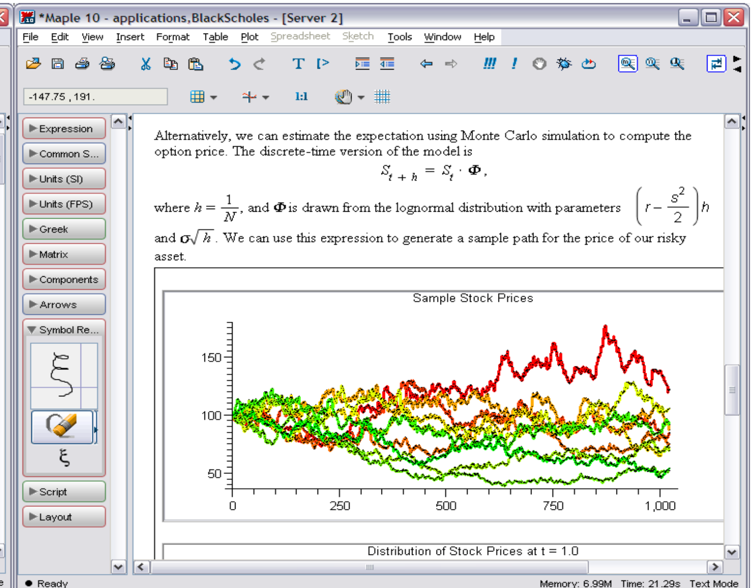
Notice the close conceptual relation among maximin point arrangements, circle packings and the various models of atomic and molecular structures

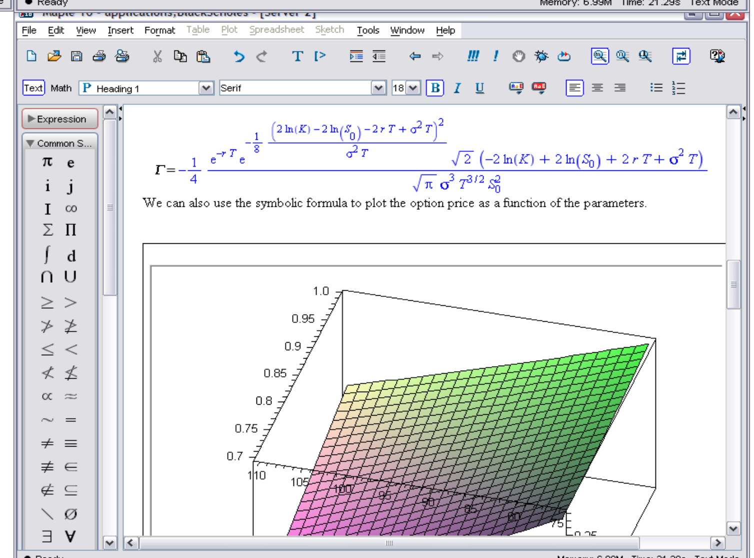
These arrangements are all aimed at finding globally minimal energy configurations of their objects, according to a context-specific criterion function

Therefore GO technology can be brought to a wide range of object configuration problems – as always, in combination with domain-specific expertise

Financial Modeling and Optimization







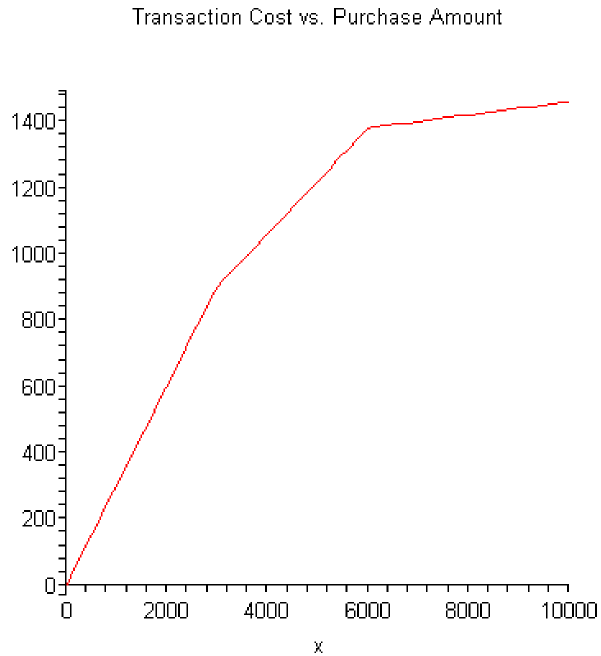
Example:
Model development, solution, and visualization in Maple

Castillo-Lee-Pintér, Integrated Software Tools for the OR/MS Classroom, *AlgOR* (2008)

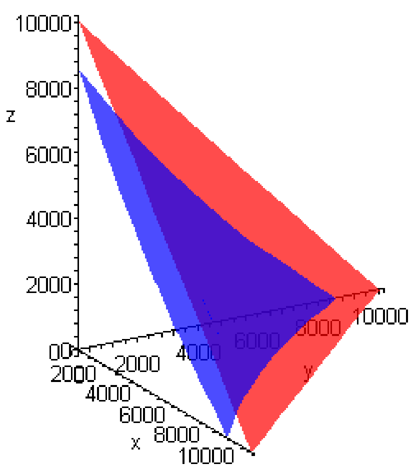
Portfolio Optimization with Transaction Costs

Objective:	**minimize portfolio variance; Q cov. matrix**	x^TQx
Constraints:	**expected return (ER)**	$x^Tr \geq ER$
	asset allocation (of capital C)	$\sum_i x_i + \sum_i t(x_i) \leq C$

Note: other considerations will make model more complex…
GO can be applied to many such (more realistic) models

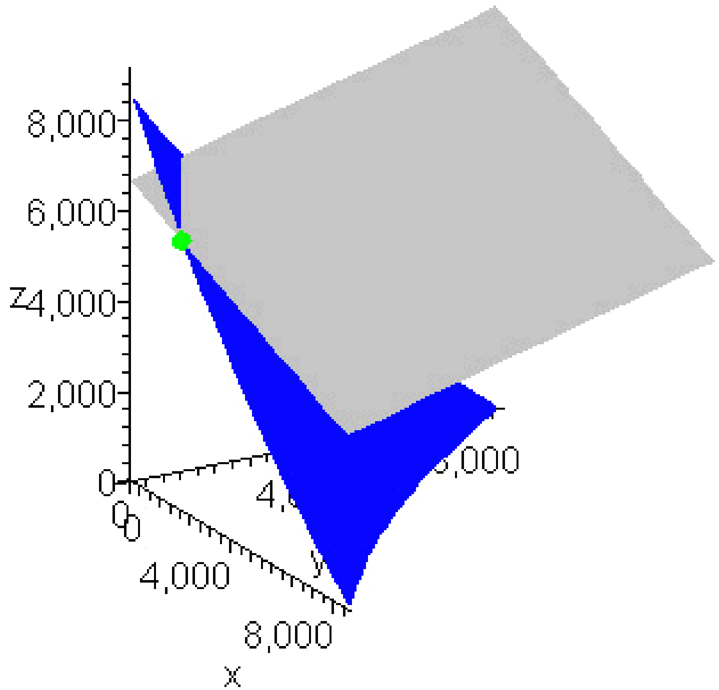


Credits: Jason Schattman, Maplesoft



Portfolio Optimization with Transaction Costs

(continued)



The figure shows the location of the optimal budget allocation point (in green) on the boundary of the feasible region

The surfaces representing the active budget constraint (blue) and the expected return constraint (grey) are also shown – recall KKT theory

Castillo-Lee-Pintér, Integrated Software Tools for the OR/MS Classroom, *AlgOR* (2008)

Supply Chain Management: A Reliability Optimization Example

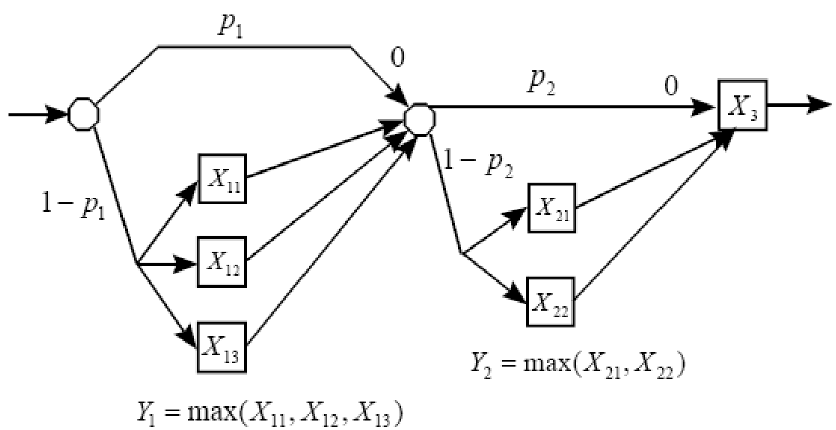


Figure 8: A three-stage assembly-type supply chain. In the first stage, if inventory is not available three components must be ordered from outside suppliers with different leadtimes. In the second stage, when inventory is unavailable, two components must be ordered from suppliers with different leadtime. The last stage always lasts a random length of time and does not require outside components.

Cited from Hum and Parlar (2006); numerical example in the e-book *Global Optimization with Maple* (2006)

Solving a System of ODEs by the "Shooting Method"

The SM consists of adjusting the initial conditions of the solution until the boundary conditions are met. Unless the initial conditions are very close to the correct value, singularities are frequently encountered.

Therefore one can use a finite difference approach and solve the resulting system of equations with *MathOptimizer Professional*. Then, based on the initial condition values found, one can find a more precise solution by the SM.

Note: the model shown is received from an MOP user

Further technical details in

MOP User Guide

The governing equations

$$\frac{d^2S}{dw^2}+\frac{2}{w}\frac{dS}{dw}-\frac{\Phi_1 S\,C}{(S+1)(C+1)}+\frac{\Phi_2 S\,Z}{(S+1)(C+1)(Z+1)} \qquad 6.2.1)$$

$$\frac{d^2Z}{dw^2}+\frac{2}{w}\frac{dZ}{dw}=-\frac{\Phi_5 H\,C}{(H+1)(C+\alpha)}+\frac{\Phi_4 S\,Z}{(S+1)(C+1)(Z+1)} \qquad 6.2.2)$$

$$\frac{d^2C}{dw^2}+\frac{2}{w}\frac{dC}{dw}=\frac{\Phi_6 S\,C}{(S+1)(C+1)}+\frac{\Phi_7 HC}{(H+1)(C+\alpha)} \qquad (6.2.3)$$

$$\frac{d^2H}{dw^2}+\frac{2}{w}\frac{dH}{dw}=\frac{\Phi_3 H\,C}{(H+1)(C+\alpha)} \qquad (6.2.4)$$

Boundary conditions

$$\left.\frac{dS}{dw}\right|_{w\longrightarrow 0}=0,\quad \left.\frac{dS}{dw}\right|_{w=1}=Sh_s(S_b-S) \qquad \left.\frac{dZ}{dw}\right|_{w\longrightarrow 0}=0,\quad \left.\frac{dZ}{dw}\right|_{w=1}=Sh_z(Z_b-Z)$$

$$\left.\frac{dH}{dw}\right|_{w\longrightarrow 0}=0,\quad \left.\frac{dH}{dw}\right|_{w=1}=Sh_H(H_b-H) \qquad \left.\frac{dC}{dw}\right|_{w\longrightarrow 0}=0,\quad \left.\frac{dC}{dw}\right|_{w=1}=Sh_C(C_b-C)$$

The parameters of the system are given in table (6.1) & table (6.2) [42].

Table (6.1)

sh_s	15.45969	S_b	12.5
sh_z	13.16323	Z_b	1
sh_H	12.5708	H_b	1.5
sh_c	11.24284	C_b	95

Table (6.2)

Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	Φ_7
85.103486	72.337963	59.83288	82.6568	103.04552	2.945e+03	5.0336+03

Microsoft Excel - CIRCUITD.XLS

objective | f_x =con_1^2+con_2^2+con_3^2+con_4^2+factor*con_5^2+factor*con_6^2+con_7^2+factor*con_8^2+con_9^2

Model Variables: LBnd, NomValue, UBnd

Var. Index	x_lb	Variable x	x_ub
1	0	1	10
2	0	2	10
3	0	3	10
4	0	4	10
5	0	5	10
6	0	6	10
7	0	7	10
8	0	8	10
9	0	9	10

Circuit Design

NLEQ System

Solved by Excel/LGO

767671534	Objective Function: Minimize the total (scaled, l_2 norm) error of equations

Constraints C1...C9 express physyical system equilibrium conditions

12.062365	C1: prod1*(exp(x5*(g11-g31*x7*rthou-g51*x8*rthou))-one)-g51+g41*x2+cad1 = 0
93.678244	C2: prod1*(exp(x5*(g12-g32*x7*rthou-g52*x8*rthou))-one)-g52+g42*x2+cad2=0
91.053678	C3: prod1*(exp(x5*(g13-g33*x7*rthou-g53*x8*rthou))-one)-g53+g43*x2+cad3=0
174.00957	C4: prod1*(exp(x5*(g14-g34*x7*rthou-g54*x8*rthou))-one)-g54+g44*x2+cad4=0
15.073224	C5: prod2*(exp(x6*(g11-g21-g31*x7*rthou+g41*x9*rthou))-one)-g51+g41*x2+cad5=0
131.4521	C6: prod2*(exp(x6*(g12-g22-g32*x7*rthou+g42*x9*rthou))-one)-g52+g42*x2+cad6=0
-1532.4758	C7: prod2*(exp(x6*(g13-g23-g33*x7*rthou+g43*x9*rthou))-one)-g53+g43*x2+cad7=0
-2763.1953	C8: prod2*(exp(x6*(g14-g24-g34*x7*rthou+g44*x9*rthou))-one)-g54+g44*x2+cad8=0
-5	C9: x1*x3-x2*x4=0

Model Pm	Value
factor=	1.00E+02
rthou=	1.00E-03
one=	1
prod1=	-3
prod2=	-4
cad1=	-3.38889E-05
cad2=	-0.000151406
cad3=	-0.00012015
cad4=	0.000368546
cad5=	15.66836267
cad6=	67.16307487
cad7=	74.74244081
cad8=	126.3450721
g11=	0.485
g12=	0.752
g13=	0.869
g14=	0.982
g21=	0.369
g22=	1.254
g23=	0.703
g24=	1.455
g31=	5.2095
g32=	10.0677
g33=	22.9274
g34=	20.2153
g41=	23.3037
g42=	101.779
g43=	111.461
g44=	191.267
g51=	28.5132
g52=	111.8467
g53=	134.3884
g54=	211.4823

Answer Report LSSQP / Answer Report ES / Feasibility Report IS / Answer Report OQ5 / **Model**

Ready

A Time-Discretized Control Model Solved by Excel/LGO

Microsoft Excel - SPACES~1.XLS

Spaceship Navigator

Demonstration spreadsheet model

Frontline Systems and Pinter Consulting Services

Pilot a spaceship from one planet to another using the least amount of fuel. The trip is divided into 9 time blocks, with one engine blast allowed per time block. The pilot must choose the size and direction of each rocket blast, for each time block. There are two moving suns whose gravitional fields affect the rocket. The best answer will take advantage of this and use the fields to "slingshot" toward the goal...

This is a non-trivial 17-variable multiextremal model with two complicated nonlinear equality constraints. Extra difficulty: a large number of locally optimal solutions can be found on the border of the feasible set.

Sun1

xpos	ypos	ship to sun	grav	angle to sun	xgrav	ygrav
200	400	447.2135955	10	1.107148718	4.472136	8.9442719
180	400	406.504936	12.103148	1.125707647	5.2108643	10.923969
160	400	319.5048347	19.591836	1.098593056	8.9113376	17.447868
140	400	118.8074485	141.69113	0.484360388	125.39282	65.977399
120	400	140.630418	101.12801	-1.946257292	-37.083785	-94.0833
100	400	267.3464798	27.982148	-2.237562568	-17.305505	-21.989091
80	400	366.0161194	14.928961	-2.357613614	-10.571339	-10.541378
60	400	450.237664	9.866119	-2.436672084	-7.514658	-6.392982
40	400	522.2734784	7.3321957	-2.4872459	-5.8176991	-4.4626754
20	400	586.9220194	5.8058952	-2.536249019	-4.7742259	-3.3038138

Sun2

xpos	ypos	ship to sun	grav	angle to sun	xgrav	ygrav	Time	Bl
0	300	300	22.222222	1.570796327	1.3613E-15	22.2222222	0	
10	270	236.9532645	35.620877	1.549626898	0.75401728	35.6128953	1	
20	240	124.6547338	128.71005	1.528052239	5.49991839	128.592483	2	
30	210	134.7658235	110.12108	-1.606856461	-3.9701203	-110.04949	3	
40	180	374.6931255	14.245528	-1.929583627	-5.0021606	-13.33842	4	
50	150	507.9881852	7.750376	-2.008551431	-3.2854411	-7.0195587	5	
60	120	606.5180477	5.4367898	-2.049128029	-2.5025476	-4.8265866	6	
70	90	687.7029927	4.2289073	-2.076151187	-2.04729	-3.7003055	7	
80	60	756.9506925	3.4905576	-2.088182762	-1.7264659	-3.0336955	8	
90	30	816.0012682	3.0036431	-2.100958261	-1.5188614	-2.5913186	9	

Spac

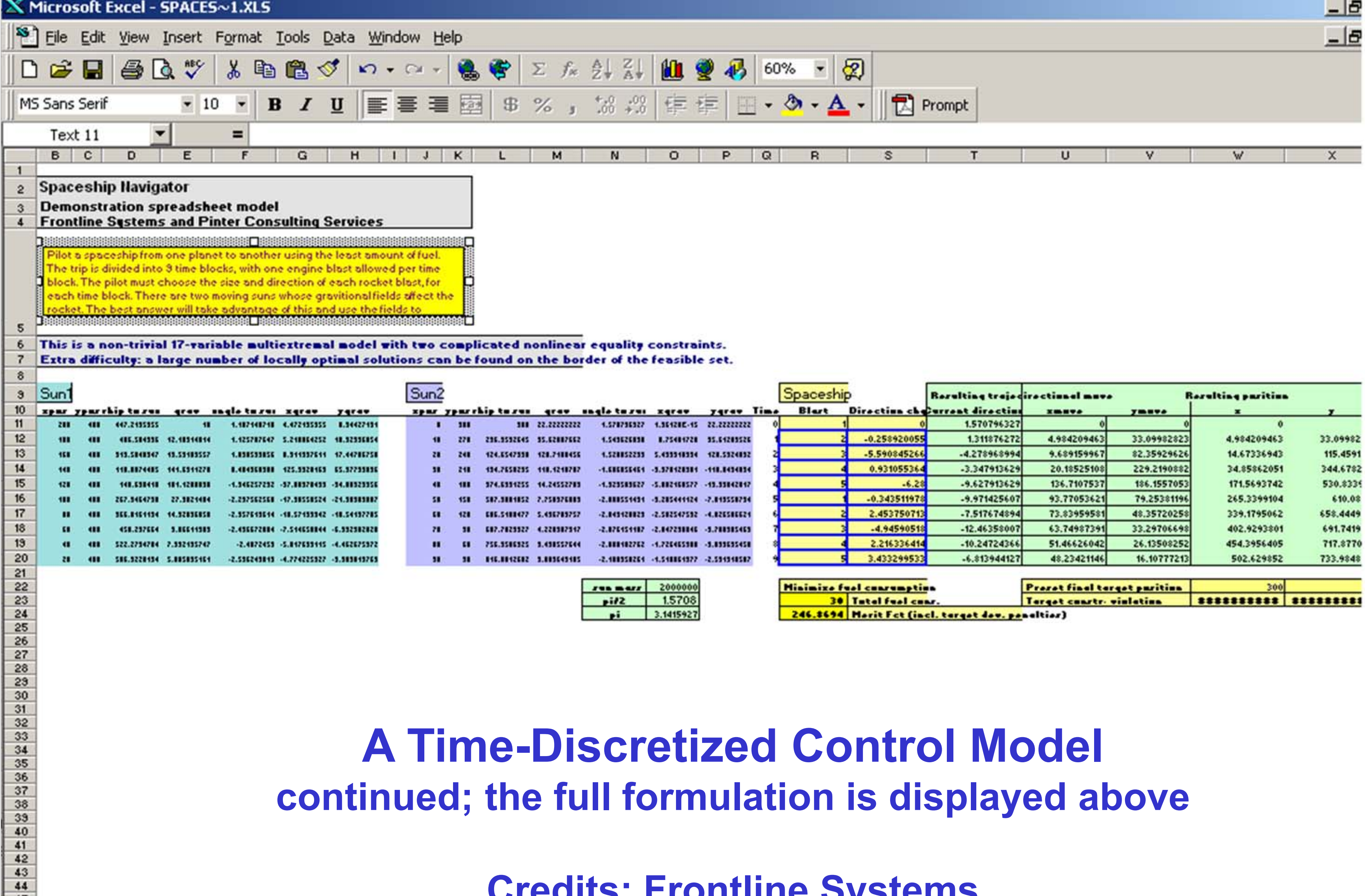
sun mass	2000000
pi/2	1.570796
pi	3.14159265

Minimi

246.8

Answer Report LGO_3 | **SpaceShip Navigator**

Ready



A Time-Discretized Control Model

continued; the full formulation is displayed above

Credits: Frontline Systems

Industrial Design Problems

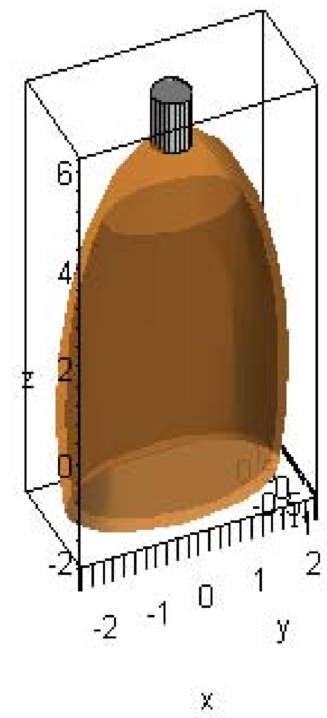
An illustrative application:

Design of an "optimized" parfume bottle, using the Maple GOT

Objective:
minimize package volume

Constraints:
Bottle volume ≥ required
Width of the base ≥ required
Aesthetic proportions

Example by Maplesoft



Circuit Design Problem - Maple Application Center - Maplesoft - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.maplesoft.com/applications/app_center_view.aspx?AID=1894

Getting Started Latest Headlines

Nonlinear optimization, global optimization, s... | Local Minimum -- from Wolfram MathWorld | Chemical Equilibrium Computation - Maple A... | Circuit Design Problem - Maple Appl...

Maple Application Center

- Research PowerTools
- Education PowerTools
- Browse Applications
- Advanced Search
- Top Rated Applications
- New Applications
- Most Downloaded Applications
- Tips & Techniques
- Submit Your Work

Other Resources

- Maple 10 Training
- Maplesoft Books
- Maple Reporter

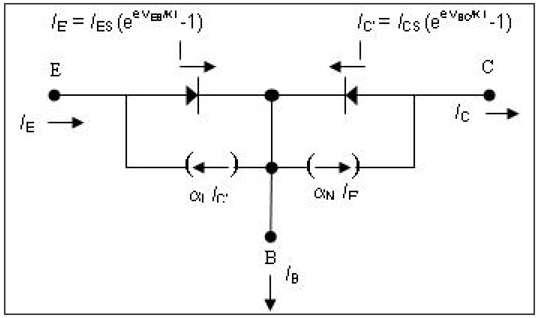
More Information

- Contact Maplesoft

Top Rated Applications | New Applications | Most Downloaded | maple APPLICATION CENTER

Conduct a search: simple : advanced

Circuit Design Problem



Member Rating:	(rate this application)
Author:	Dr. Janos D. Pinter
Application Type:	Maple Worksheet
Publish date:	October, 2005
Related Products:	Maple 10
Language:	English

Options:

- View HTML version
- Download Maple Worksheet *(.mw, 174.6kb)*
- E-mail to a colleague

Abstract:

Based on the classical study of Ebers and Moll (1954), a bipolar transistor is modeled by an electrical circuit (see also e.g., Granvilliers and Benhamou, 2001). The corresponding model leads to a square system if highly nonlinear equations in nine (9) variables that has been studied by numerous researchers, in attempts to solve it, and then prove the correctness of the suggested solution.

Related Application Categories

Engineering : Electrical
Maple Tools : Maple Functionality
Mathematics : Operations Research

Member rating: not rated *(min. 5 ratings required)*
You have not rated this application.
Rate it: - Rate - SUBMIT
View top rated applications

Maple Application Center
- Research PowerTools
- Education PowerTools
- Browse Applications
- Advanced Search
- Top Rated Applications
- New Applications
- Most Downloaded Applications
- Tips & Techniques
- Submit Your Work

Other Resources
- Maple 10 Training
- Maplesoft Books
- Maple Reporter

More Information
- Contact Maplesoft

Top Rated Applications | New Applications | Most Downloaded | maple APPLICATION CENTER
Conduct a search: simple : advanced

Chemical Equilibrium Model

$$H_2O + H_3PO_4 \leftrightarrow H_3O^+ + H_2PO_4^- \Rightarrow K_{a1} = \frac{[H_3O^+][H_2PO_4^-]}{[H_3PO_4]}$$

$$H_2O + H_2PO_4^- \leftrightarrow H_3O^+ + HPO_4^{2-} \Rightarrow K_{a2} = \frac{[H_3O^+][HPO_4^{2-}]}{[H_2PO_4^-]}$$

$$H_2O + HPO_4^{2-} \leftrightarrow H_3O^+ + PO_4^{3-} \Rightarrow K_{a3} = \frac{[H_3O^+][PO_4^{3-}]}{[HPO_4^{2-}]}$$

Member Rating:	(rate this application)
Author:	Dr. Janos D. Pinter
Application Type:	Maple Worksheet MapleNet Application
Publish date:	October, 2005
Related Products:	Maple 10 MapleNet
Language:	English

Options:
- View live with MapleNet
- View HTML version
- Download Maple Worksheet *(.mw, 127.5kb)*
- E-mail to a colleague

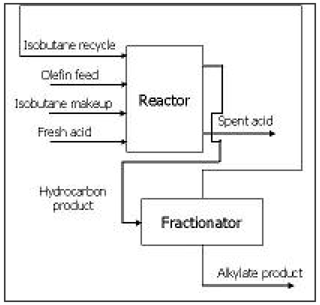
Abstract:
Modeling chemical equilibrium of target compounds is of interest when controlling the pH, alkalinity or corrosivity of drinking water.
As part of this approach, one must determine the contribution rates of various components to mixtures which have given (known or prescribed) chemical characteristics.
In the example presented, we want to determine the concentrations of several components of phosphoric acid such that the resulting pH value is equal to 8, and the total phosphate concentration is 0.1 mols.

This worksheet requires that the Global Optimization Toolbox has been added to Maple.

Related Application Categories
Maple Tools : Maple Functionality
Maple Tools : MapleNet Functionality
Mathematics : Operations Research

Member rating: not rated *(min. 5 ratings required)*
You have not rated this application.
Rate it: - Rate - SUBMIT

Alkylation Process Model



Member Rating:	(rate this application)
Author:	Dr. Janos D. Pinter
Application Type:	Maple Worksheet MapleNet Application
Publish date:	October, 2005
Related Products:	Maple 10 MapleNet
Language:	English

Options:

- View live with MapleNet
- View HTML version
- Download Maple Worksheet *(.mw, 152.2kb)*
- E-mail to a colleague

Abstract:
In this example, we describe a model for the optimization of a typical process operation in the petrochemical industry. Our objective is to determine the optimal set of operating conditions for an alkylation process that combines olefin with isobutane, in the presence of a catalyst, to form alkylate.
Many chemical processes are characterized by nonlinear equilibrium (material and energy balance) constraints. In addition, the processes are typically constrained by restrictions on the operating ranges of the decision variables such as amounts and rates of the components used, temperature, pressure, and so on.

This worksheet requires that the Global Optimization Toolbox has been added to Maple.

Related Application Categories
Education : Operations Research
Maple Tools : Maple Functionality

Member rating: not rated *(min. 5 ratings required)*
You have not rated this application.
Rate it: - Rate - SUBMIT

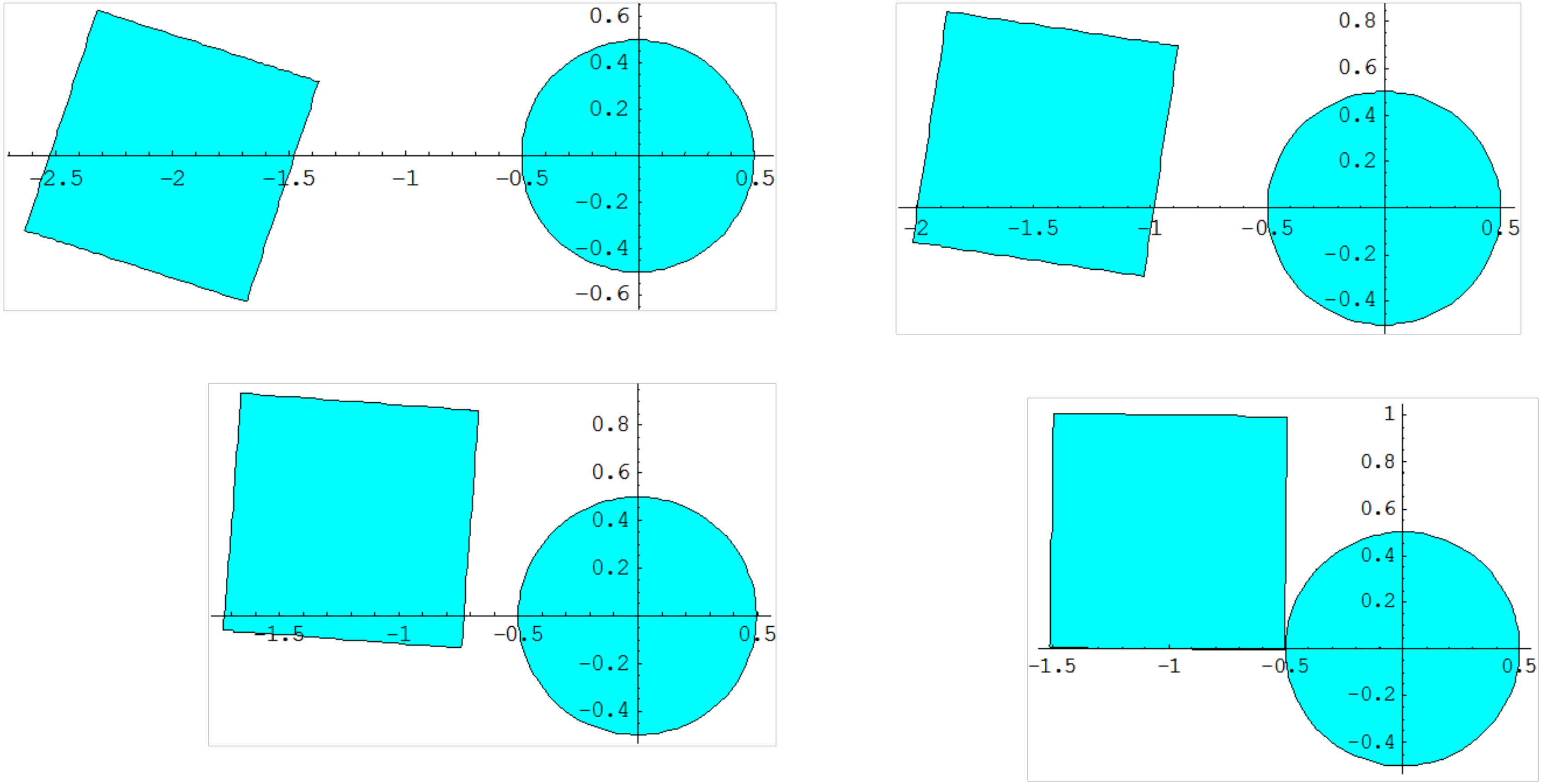
Collision Analysis for Moving Solid Bodies

Given a number of solid bodies, each with corresponding geometry, initial position, and analytical trajectory description: our task is to decide whether they will collide or not

Obviously, this problem-type is of interest in various practical applications: e.g. in robot motion analysis, production (job shop) floor planning, and other areas

One can approach this problem by finding the time moment when the smallest distance between all pairs of the moving bodies is minimal: this is a (generally speaking, far from trivial) global optimization problem

Collision analysis for moving solid bodies: An example



Details, including code implementation: *The Mathematica Journal* (2006)   Co-author: Frank J. Kampas

Kinetic Grasp Feasibility Analysis in Robotics Design

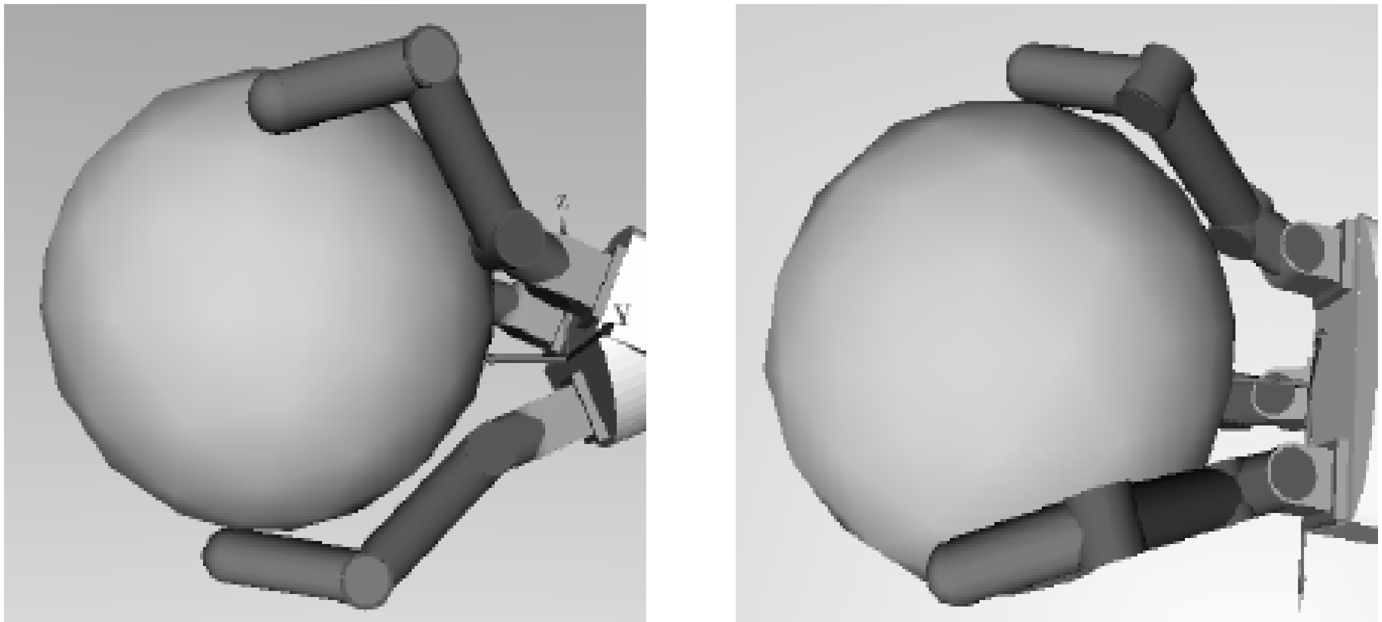


Figure 10: Grasp of the maximum sphere

Credits: Yisheng Guan and Hong Zhang, University of Alberta, Edmonton, Canada

Laser Design

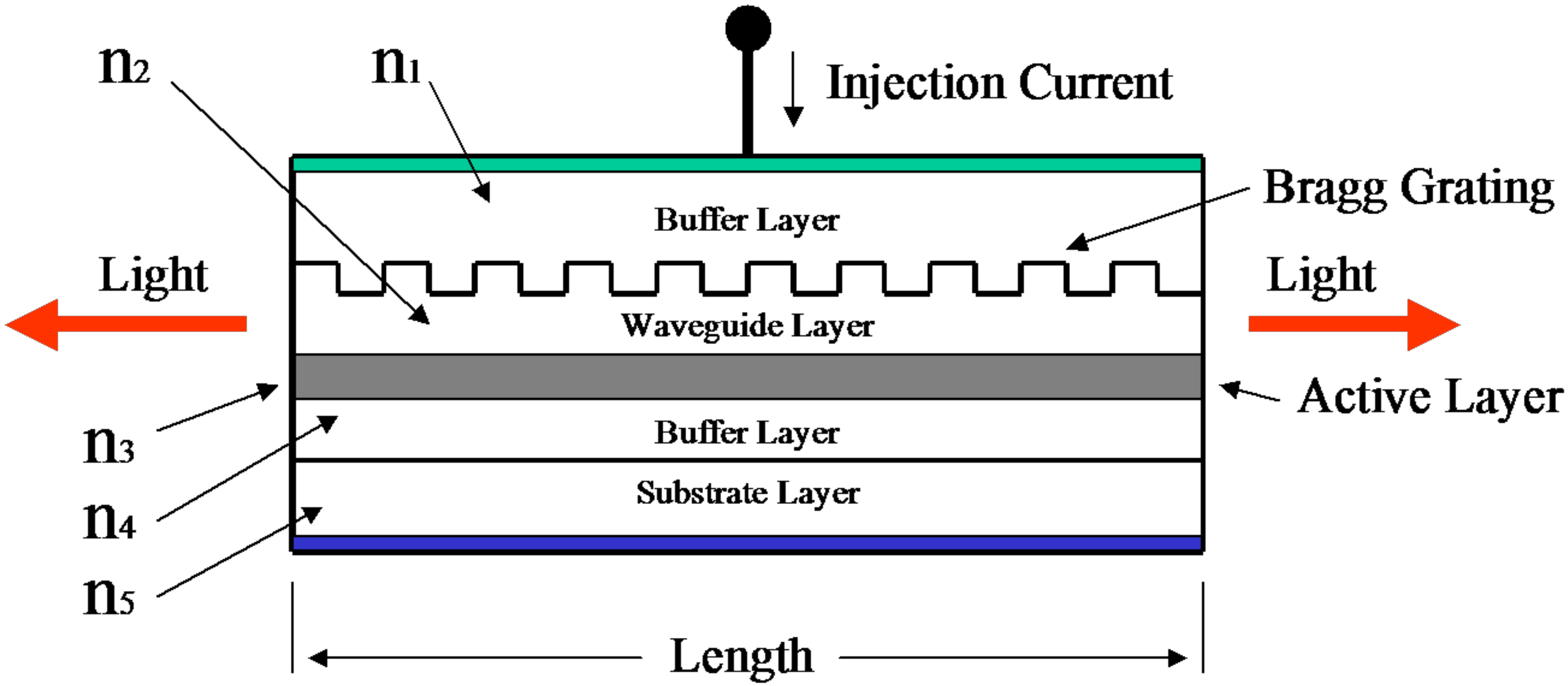
Optimization and Engineering (2003); with G. Isenor & M. Cada

Basic Concepts

The laser is a device that produces a beam of light that is coherent. The beam is produced by a process known as stimulated emission.

The word *laser* is an acronym for the phrase "Light Amplification by Stimulated Emission of Radiation".

The idea of stimulated emission was proposed by Albert Einstein in 1916. It took another four decades to build the first lasers as a scientific research tool; soon they found numerous significant applications.



$n_{1,2,3,4,5}$ = Index of Refraction $\quad n_5 \leq n_1 < n_4 \leq n_2 < n_3$

Index-coupled distributed feedback laser

Various laser design issues can be analyzed using GO

Example:

min $f(x)$	field flatness function (key quality measure)
$g(x) \leq \varepsilon$	boundary condition (error limit)
$xl \leq x \leq xu$	explicit, finite parameter bounds
$x = (KL1, KL2, KL3, \lambda, C_o)$	laser design parameters

Essential difficulty: *f* and *g* are complicated "black box" functions. The LGO IDE software has been used to analyze and solve this model (in several variants).

A very significant improvement (over 90% reduction) of the field flatness function has been attained.

Radiotherapy Planning

Significance of the problem: world-wide interest and R&D activities devoted to cancer therapy by irradiation

Specific area of our research: intensity modulated radiation therapy planning, delivered by multi-leaf collimators, to cure individual patients

Objective: determine the operations (movements) of the leafs in an MLC equipment, to optimally approximate the prescribed dose intensity distribution in 3 dimensions, thereby

- **providing prescribed radiation intensity to a target area or volume (the body parts affected by cancer)**
- **avoiding unwanted radiation as much as possible (especially of organs at risk, as well as other body parts)**

For details, cf. Tervo et al., *Annals of Operations Research,* 2003

Dose Delivery and Effect Modeling

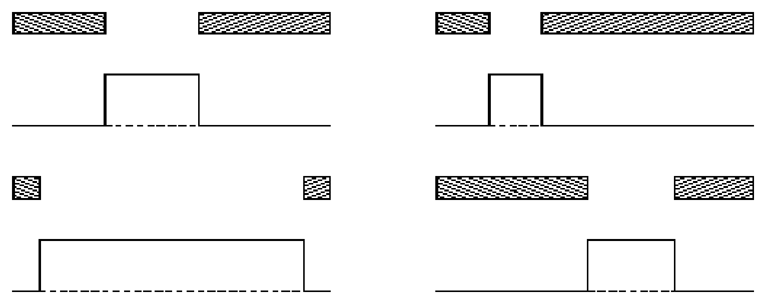
Sophisticated, computationally intensive mathematical models of dose delivery by MLC equipment have been developed in several versions by researchers at the University of Kuopio, Finland. The key novel feature of this approach is to optimize dose distribution directly via adjusting MLC parameters. These optimization models are all characterized by

- **tens or hundreds of variables (leaf positions and their coordinated movements, to describe MLC operations),**
- **a large number of relatively simple constraints (feasible leaf positions),**
- **a few significant complex "black box" constraints; complex objective function (target dose, and limits on unwanted dose in OAR and body tissue).**

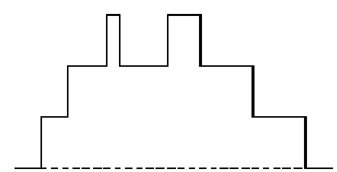
Joint operation of leafs in MLC equipment (simplified scheme)

Leaf positions

Radiation intensity

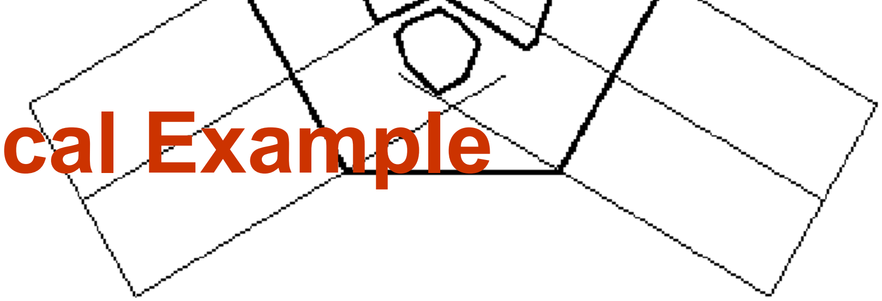


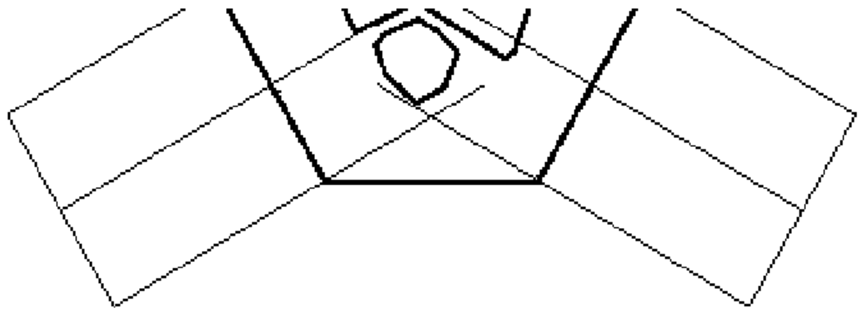
A resulting radiation profile
(based on all leaf positions
that determine total exposure)



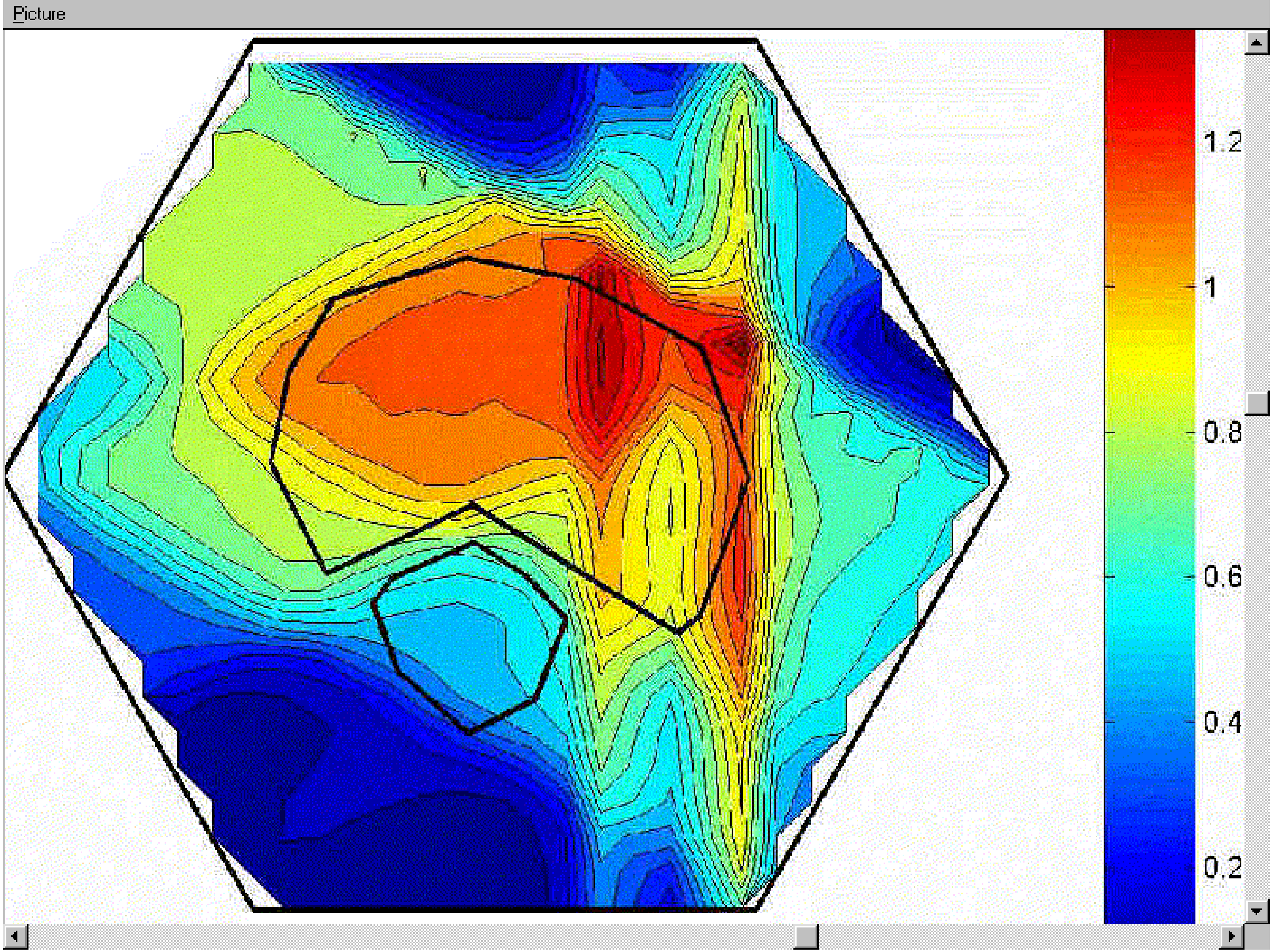
Superposition of overall irradiation effect, as a function of leaf positions and radiation intensity

A Numerical Example

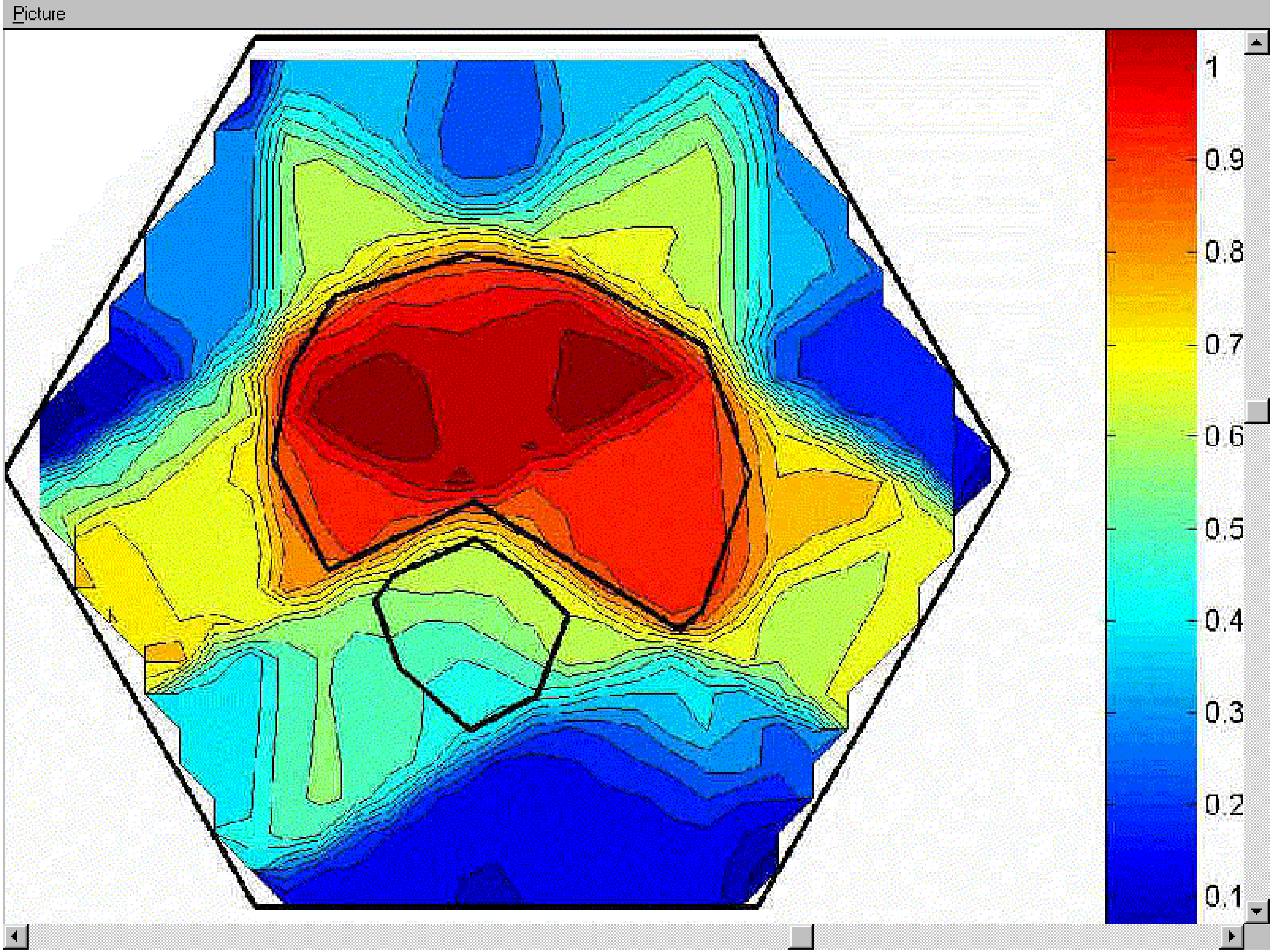




Illustrative model (2D phantom) used in optimized radiation dose distribution test calculations: overall irradiation area, hypothetical target area, and an organ at risk are shown



Dose distribution found by local optimization of nominal solution



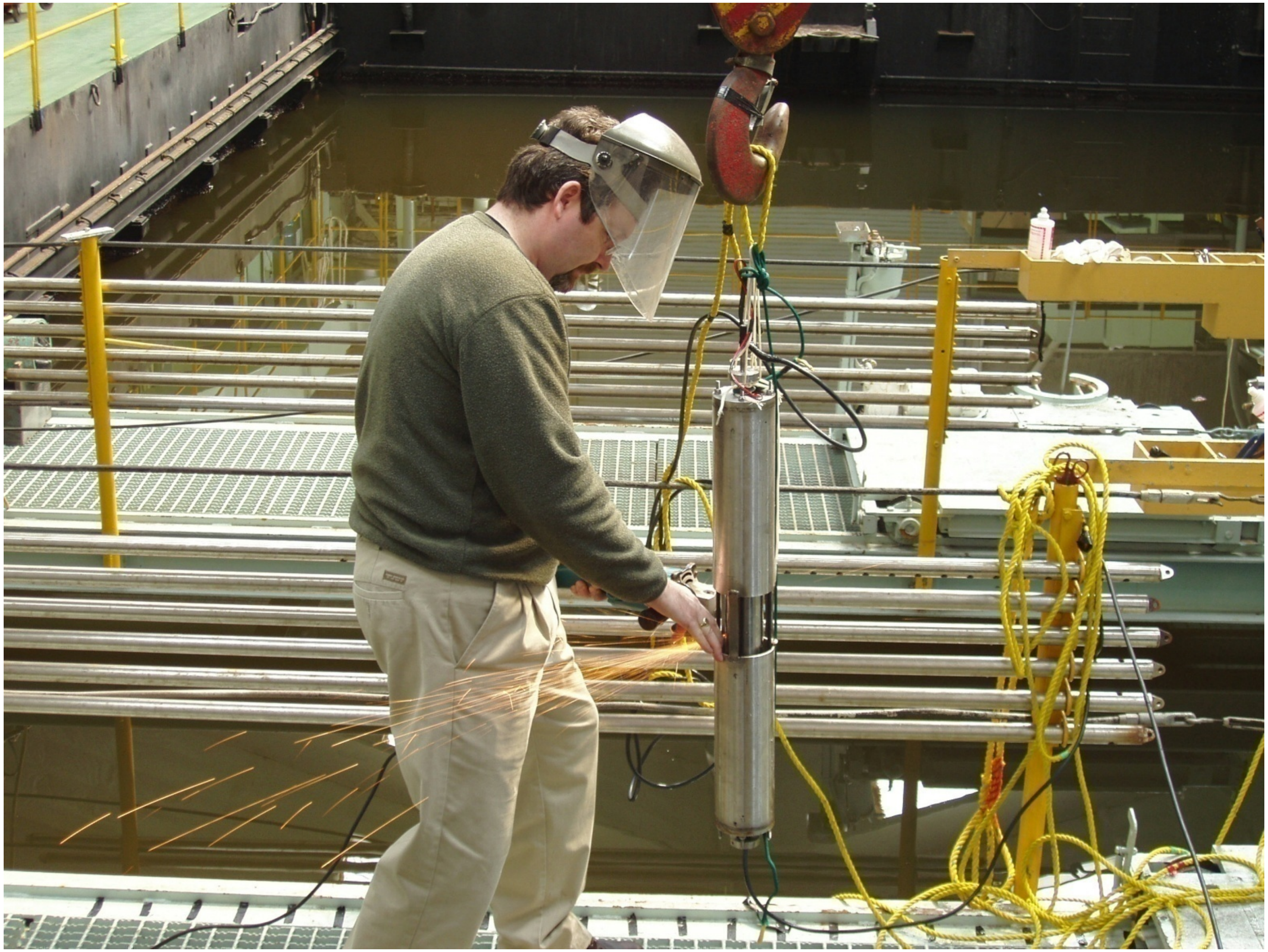
Globally optimized dose distribution

Modeling and Optimization of Transducers

MathOptimizer User Guide, joint presentations with C.J. Purcell

- Traditional engineering design often based on experimental studies: change key parameters and then trace their effect (e.g. by physical experiments and their graphical summaries) – as a rule, expensive and time consuming…
- Parametric studies are ideal tasks for computers: numerical models can (partially) replace experiments
- Parametric models can be directly optimized
- In our study, a combination of detailed system modeling and optimization has been applied; this has resulted in improved (in some cases “surprising” and entirely new) designs

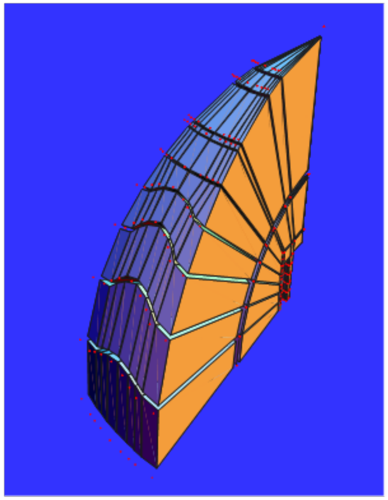
Engineering Design Optimization by Trial and Error

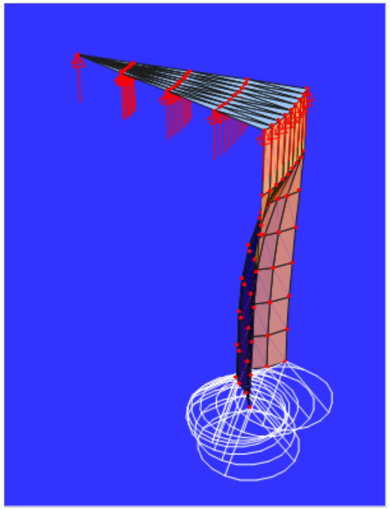


Expensive and time-consuming...

ModelMaker

- *Mathematica* package for developing advanced finite element models (FEM)
- Numeric and symbolic parameterized models can be developed
- Models and results presented in interactive *Mathematica* document (notebook) format
- Built-in, extensible documentation
- Supports other FEM packages (such as Mavart, Mavart3D, MavartMag, Atila,…)
- Developed since 1994 by C. Purcell, DRDC



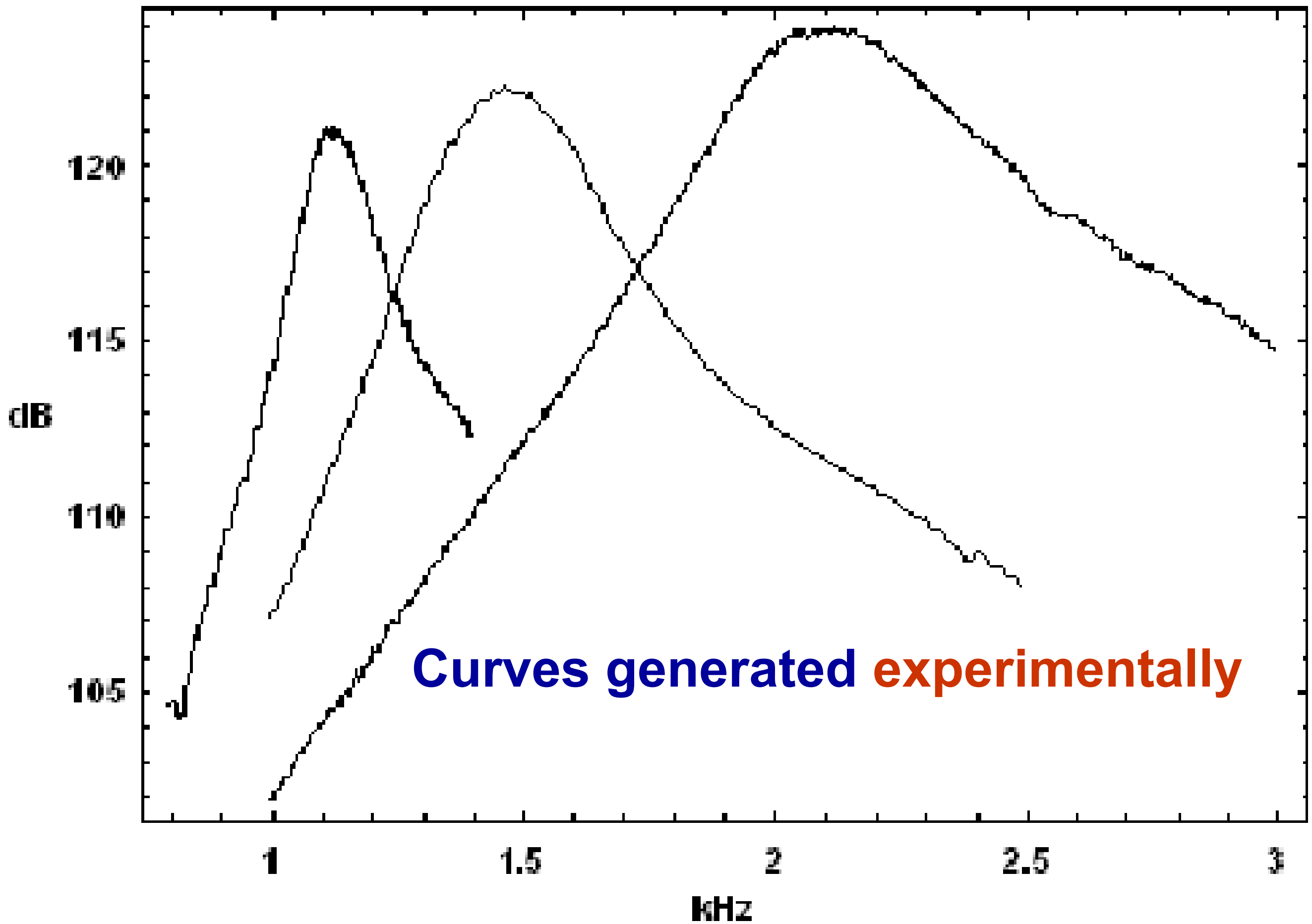


Example: Folded Shell Projector

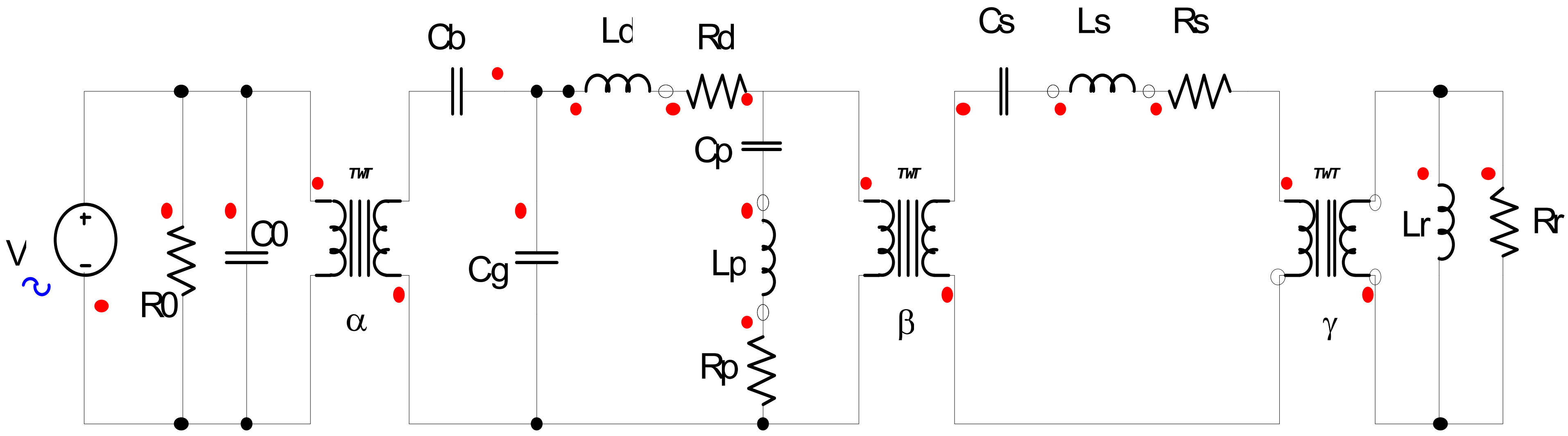
FSP is a sonar projector (or in-air loudspeaker) with overall cylinder shape with corrugations on the sides

Experimental Design

Three FSPs with varying transformer ratio (a key design parameter): optimization needed...



Sonar Transducer Design: Numerical Model



This electric circuit simulates a piezoelectric sonar projector

The optimization problem consists of finding circuit design parameters such that the sonar projector gives a broad efficiency *vs.* frequency. This model has been solved using MathOptimizer. The results have been applied to the actual design of sonar equipment, leading to improved designs.

```
goal[x_]:=N[1-Cos[Pi x]^2];

objective[□_,Cs_,Ls_,Ll_,Cl_,Cd_,Ld_]:=
- Sum[goal[f] * Module
[{Y,w=N[2 Pi f],Zr,Z1,Z2,Z3,V1,V2,V3,I0,I1,V0=1,
powerout,powerin},(  (* Begin Module calculations *)
Zr=1/(1/Rr+1/(w I Lr));     (* impedance to gnd at Rr *)
Z1=1/(w I Cs) + w I Ls + Rs+ Zr/□^2; (* impedance to gnd at input to Cs *)
(* impedance to gnd at input to beta transformer *)
Z2=1/(1/(Rl+w I Ll+1/(w I Cl)) + □^2/Z1);
Z3=1/(w I Cg)+ 1/(w I Cb+ 1/(Rd+1/(w I Cd)+ w I Ld + Z2));
(* impedance at input to Cg *)
Y=(1/R0 + w I C0 +□^2/Z3); (* input admittance *)
I0= Y * V0; (* the input current *)
V1= □ V0(1-1/(I w Cg Z3));
V2= V1 * Z2/(Z2+Rd+1/(w I Cd)+w I Ld);
V3=□ □ V2 *(Zr/□^2)/(Zr/□^2+Rs+w I Ls+1/(w I Cs));
powerout=Re[V3 Conjugate[V3/Rr]]; (* acoustic power Watts *)
powerin=Re[V0 Conjugate[I0]]; (* input electrical power Watts *)
powerout/powerin ) (* return value: relative efficiency *)
],{f, 0.1, 1., 0.01}]; (* End of Module calculations *)

(* Constants *)
{C0,R0}={.5, 10^5};
{Rs,Rl,Rd}={.01, .01, .01};
{Rr,Lr}={.01, .2};
{Cb,Cg}={20., 20.};
{□,□}={.01,.06};
```

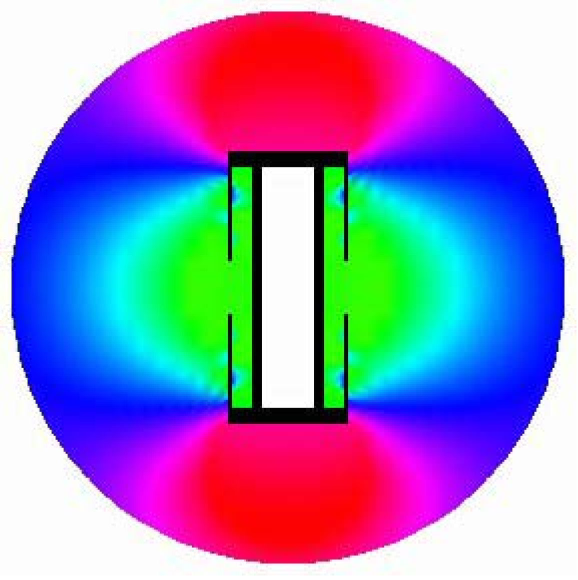
***Mathematica* code of a numerical model (only a portion is shown here), subsequently solved by MathOptimizer**

Example: Optimized FFR Transducer Design

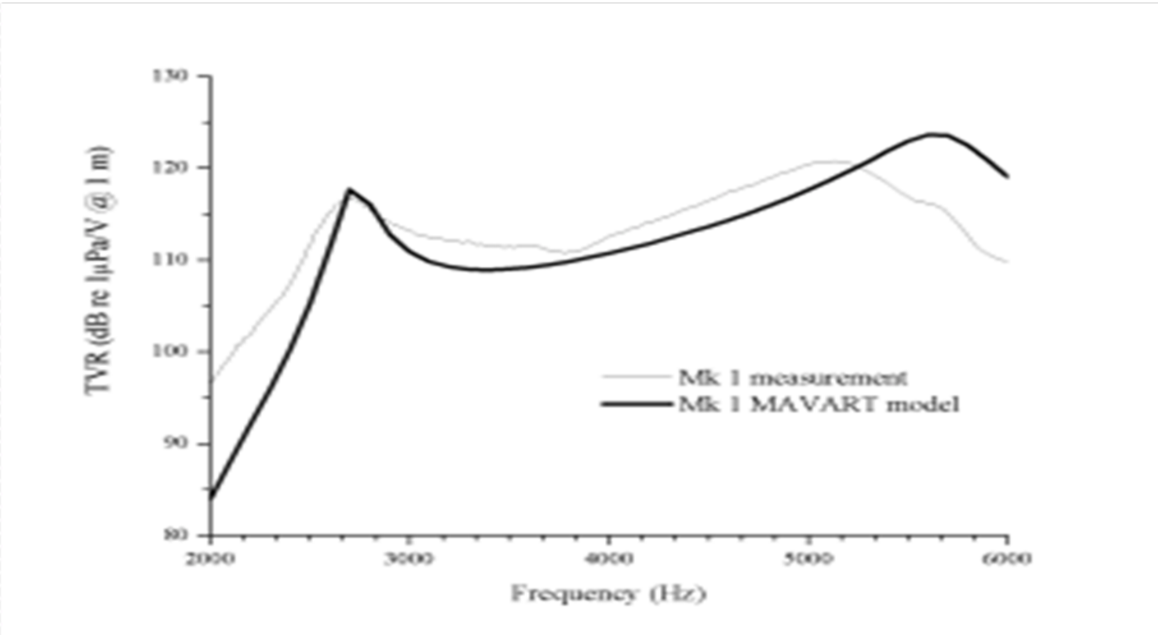
- FFR is a free flooding ring projector and refers to a high power, unlimited depth sonar projector, in the shape of a ring
- Used by Canada and the UK in sonar research over the last 1t years
- TVR is the transmitting voltage response of a sonar projector and gives the response of the device in units of microPascals/Volt (mP/V) measured at 1 meter from the device center, and converted into decibels (by taking $20*log_{10}$ of the resulting mP/V value)
- A higher TVR means more output sound per unit of input voltage, and thus it is a key design objective to provide a uniformly high TVR

Example: Multi-Mode Pipe Projector (MMPP)

- Low frequency, depth insensitive sonar projector
- Prototype MMPP demonstrated reasonable bandwidth from 2.5 to 6 kHz, but TVR (sonar response) was too low

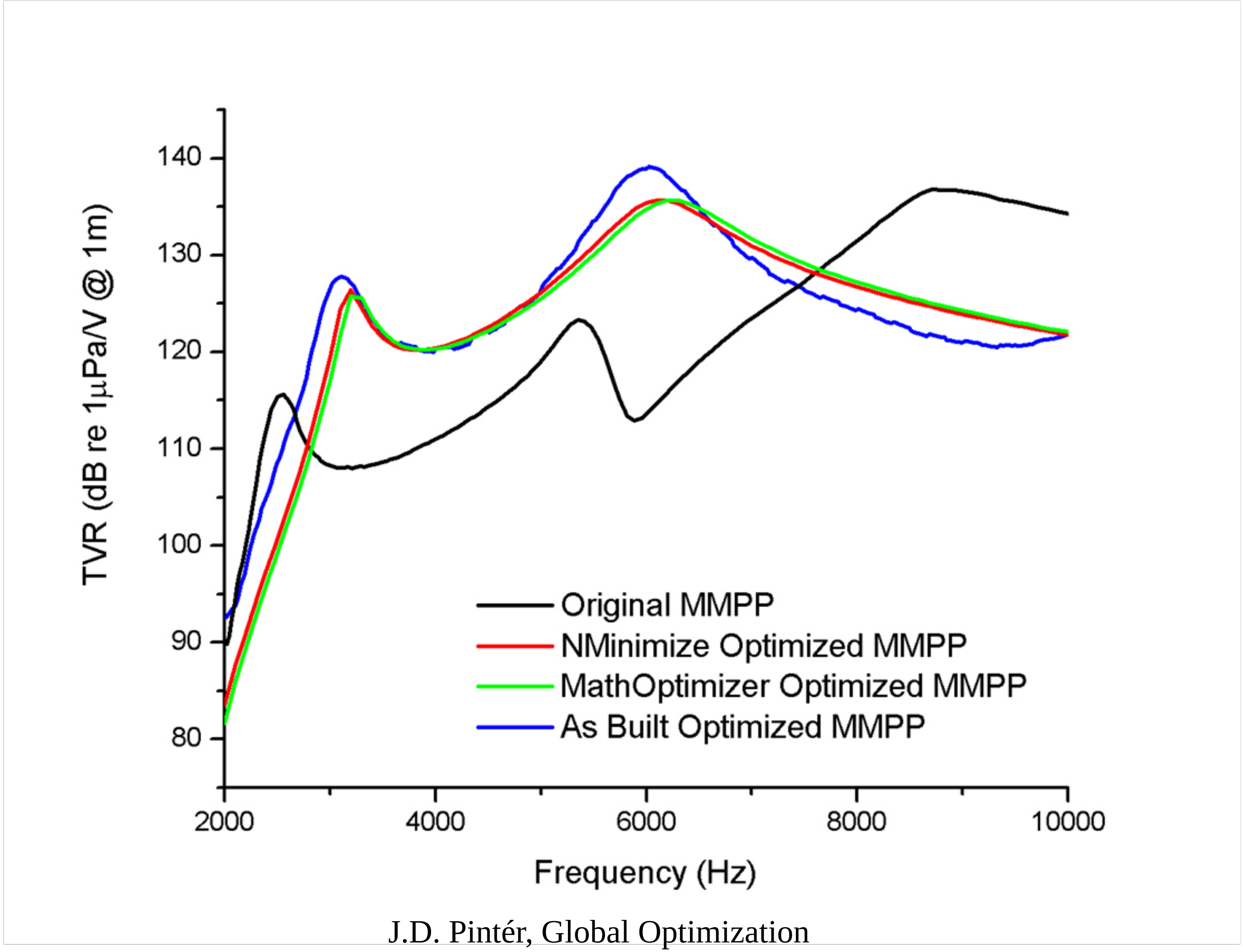


MMPP Modeling



- **Goal of optimization is to improve TVR and to increase bandwidth (3 to 9 kHz)**
- **First optimization done using NMinimize (a *Mathematica* fct)**
- **Second optimization done using MathOptimizer**
- **Optimization was run on wave-guide wall thickness, end-cap thickness and wave-guide wall height**

MMPP Optimization Results



MMPP Optimization: Summary

- Optimization provided solutions enhancing the MMPP design in a very short time period
- Optimized MMPP is a broadband, lightweight, depth-insensitive design which can be employed for numerous applications such as
  - communications
  - active sonar
  - oil well borehole shaker

  as well as some others
  - patents submitted / used for actual designs

Details for a special case described in MathOptimizer User Guide, and in Wolfram Research Developer Conference Proceedings
Co-author: C.J. Purcell

Oil Field Production Analysis and Optimization

2 Description of the Optimization Problem

Our test model considers the blending of gas produced from five fields ($F_1, ..., F_5$) to supply different quality gas at six processing plants ($P_1, ..., P_6$) through a converging-diverging gas gathering/distribution network, as shown in Figure 1 below:

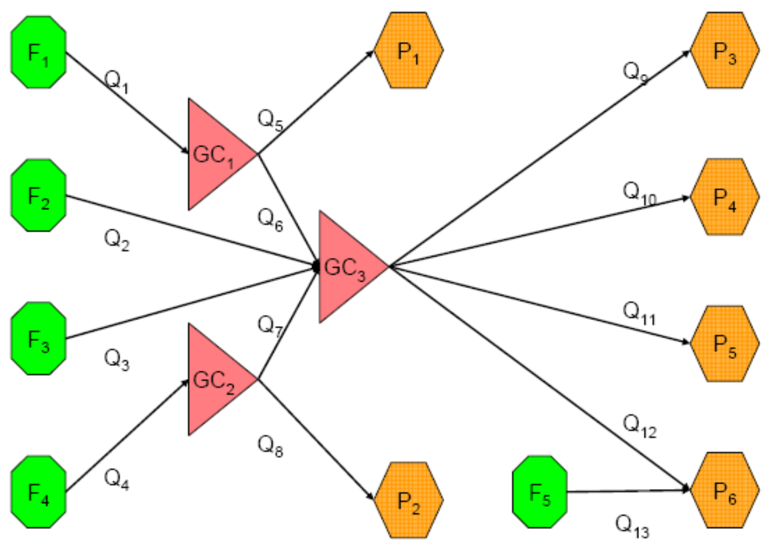
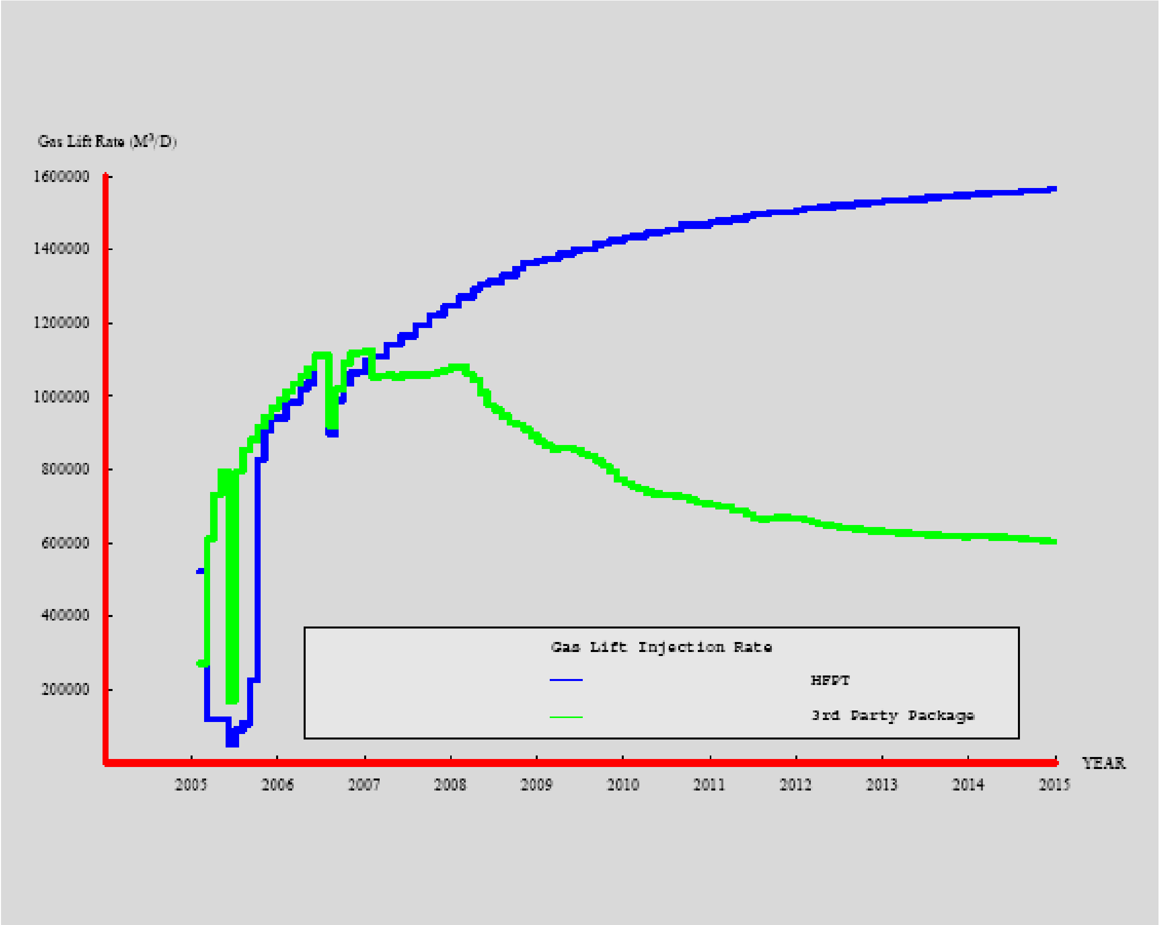


Figure 1: Schematic of the Gas Blending Network

Credits: T. Mason, P. Zwietering, C. Emelle, et al. Shell R&D, Rijswijk, NL

EURO 2006 talk, JIMO 2007 paper by Mason, Emelle, Van Berkel, Bagirov, Kampas, and Pintér

Oil Field Production Analysis and Optimization: The Global Optimization Advantage

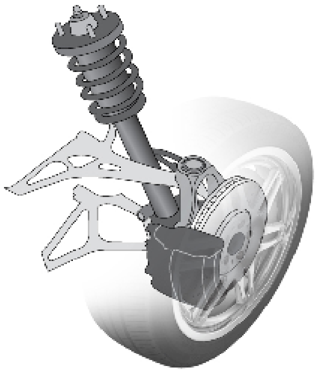


Improved gas lift (production) found by using HFTP/LGO at Shell IEP

Maple 10 - I:\Documents and Settings\Janos Pinter\Desktop\Maple materials and info\Suspension model calibration\SuspensionOptimization.mw - [Server 2]

File Edit View Insert Format Table Plot Spreadsheet Sketch Tools Window Help

# Suspension System Tuning **Calculation Sheet** *© Maplesoft, a division of Waterloo Maple Inc., 2005* *Added noted by János D. Pintér* *Pintér Consulting Services, Halifax, NS, Canada, 2006*	Contract:	6855943/N3	Maplesoft command the brilliance
	Customer:	XYZ Engineering, Inc	



Summary

In this example application, we consider the problem of designing a suspension system that exhibits a specified behavior in response to a bump in the road. The problem variables are the spring constant k and the damper constant b. Given the mass of the car on each wheel, m, and the expected amplitude of a typical bump, we have to find values for k and b to generate a system model response that matches the actual (measured) response as close as possible.

The above problem-type can be cast in an optimization model framework: the model objective function is the squared error between the desired and actual response as a function of k and b measured over a discrete set of time moments. After deriving the actual response by solving the system's differential equation, we use numerical optimization to find the values of k and b that minimize the error function.

Due to the typical multi-modal structure of the associated (nonlinear model calibration) error function, the Global Optimization Toolbox is needed to derive the best possible fit to the given actual data set.

▸ Global Optimization Basics

▸ The Global Optimization Toolbox

▸ An Example: Suspension System Model Selection and Target Response

▸ Deriving the Actual Response

▸ Measuring the Error Between the Target and Actual Responses

▸ Minimizing the Error with the Global Optimization Toolbox

▸ Verification of the Result

▸ Illustrative References

Case study development using Maple

Ready Memory: 0.37M Time: - Text Mode

Suspension System Tuning

Model calibration problem solved by using the Global Optimization Toolbox for Maple

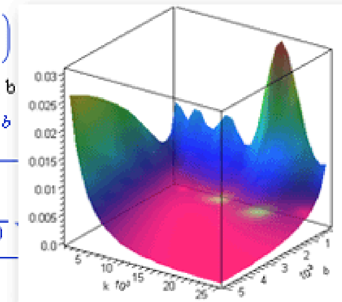
Parameter Optimization: Tuned Parameters

▼ Tuned Parameters

To derive the actual response of the system to a bump, solve the differential equation of the system's behaviour with the initial condition x(0) = 0.1.

The differential equation of an unforced mass-spring-damper system:

$$System\_Equation := m\left(\frac{d^2}{dt^2}x(t)\right) + b\left(\frac{d}{dt}x(t)\right)$$



Now solve the system and define the response as a function of k, b

$$sol := x(t) = \frac{1}{20}\,\frac{\left(b^2 - 1800\,k + b\sqrt{b^2-1800\,k}\right)e^{\left(-\frac{1}{900}b\right.}}{b^2-1800\,k} + \frac{1}{20}\,\frac{\left(b^2 - b\sqrt{b^2-1800\,k} - 1800\,k\right)e^{\left(-\frac{1}{900}b - \frac{1}{900}\right.}}{b^2-1800\,k}$$

The following products provide all the computational power and development tools you need to find the "best" parameter values, given your design constraints, or for rapid modeling parameter matching (or System Identification) from experimental data.

Maple 10 - The most powerful and intuitive tool for solving complex mathematical problems and creating rich, executable technical documents.

Global Optimization Toolbox - Formulate your optimization model easily inside the powerful Maple numeric and symbolic system, and then use world-class Maple numeric solvers to return the best answer, fast!

Database Integration Toolbox - Quickly develop and deploy powerful applications that combine large enterprise datasets with the state-of-the-art analysis and visualization of Maple.

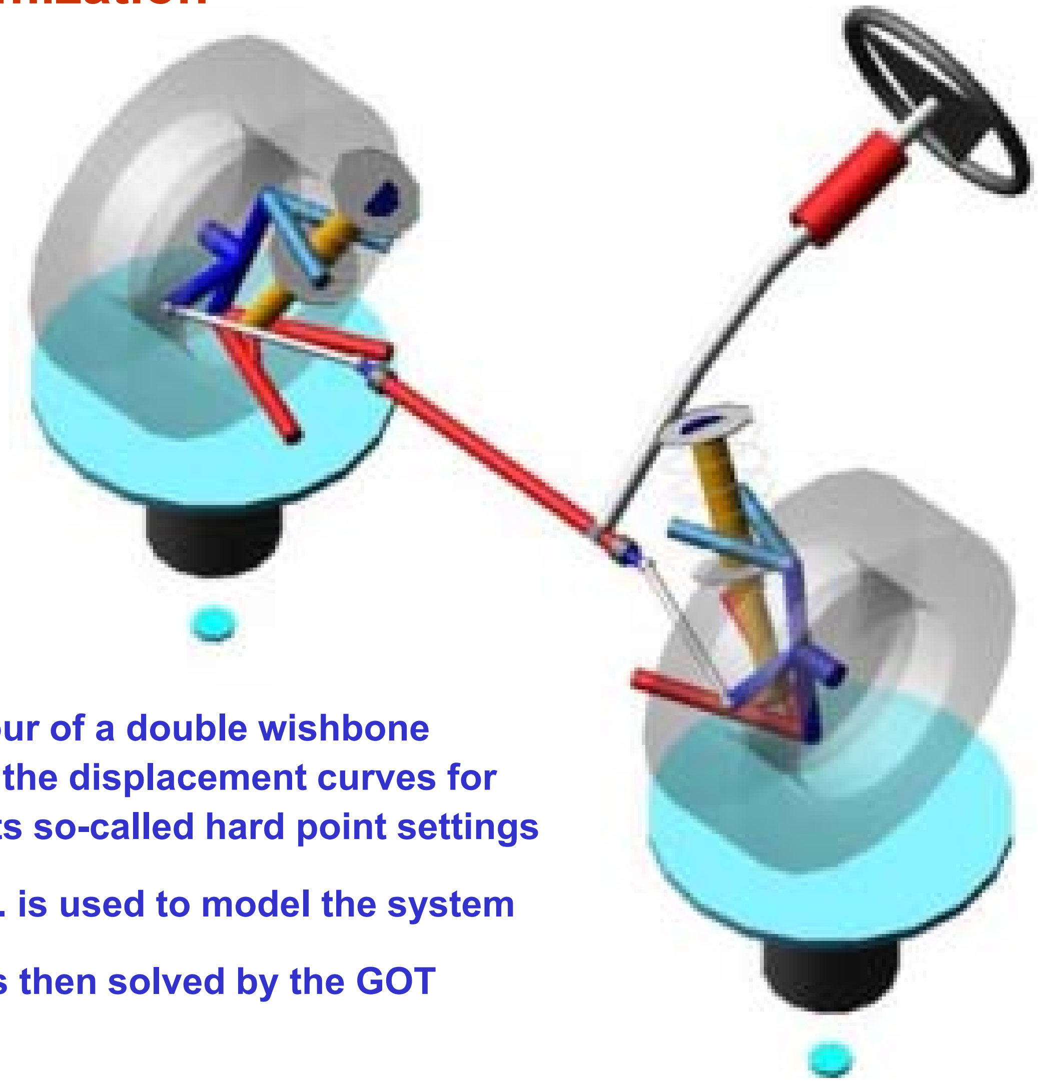
ICP: Intelligent Control Parameterization - A suite of Rapid Control Development tools that enables quick and easy system identification of engineering systems.

Professional Math Toolbox for LabVIEW - Augments LabVIEW with easy access to the sophisticated symbolic and numeric math functionality of Maple.

Global Optimization with Maple (eBook) - Presents Maple as an advanced model development and optimization environment. Special emphasis is placed on solving multiextremal models.

Double Wishbone Suspension and Steering System

Design by Global Optimization



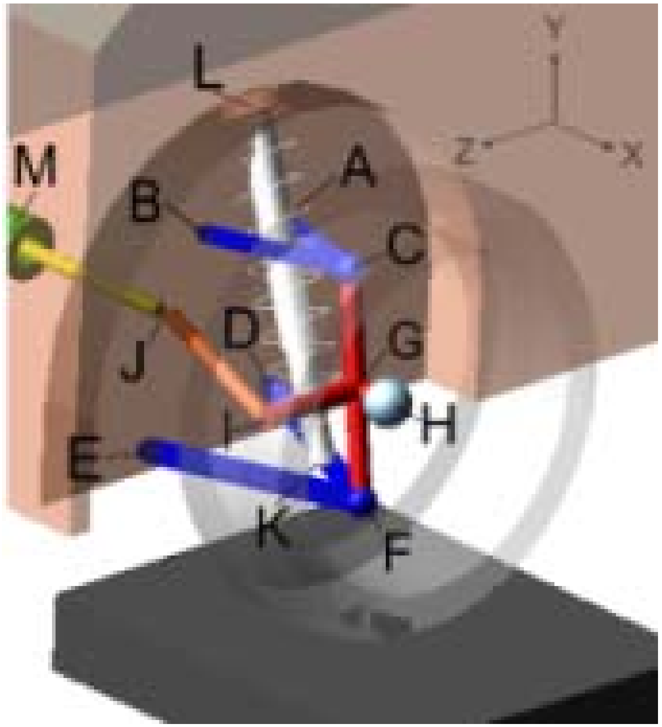
Objective

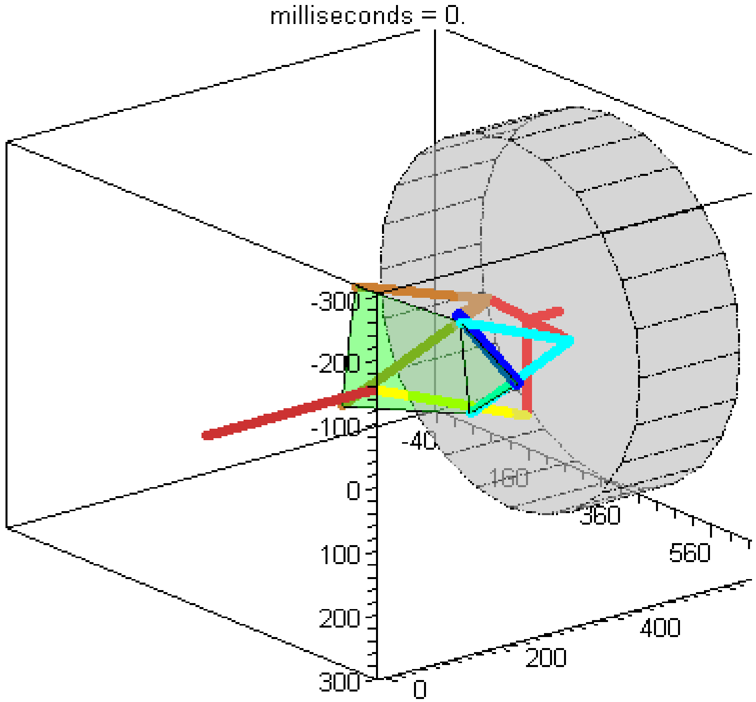
Given a desired (target) behaviour of a double wishbone suspension system, in terms of the displacement curves for bumps on the road, determine its so-called hard point settings

DynaFlex Pro by MotionPro, Inc. is used to model the system

The resulting inverse problem is then solved by the GOT

Designation of the Hard Points





The designer can specify or optimize the Cartesian coordinates of the hard points that define the double wishbone suspension: the label associated with each hard point is indicated in the lhs figure, see A to M

The Cartesian coordinates relative to the chassis-fixed XYZ frame are shown in the rhs figure: the hard point coordinates are expressed in millimeters

Using global optimization, superior new designs have been found; the GOT is now used also be several leading automotive companies as an R&D tool

Credits: Maplesoft and MotionPro, Inc., Waterloo, ON

Global Optimization Software Users: Summary

- Universities
- Research organizations
- Advanced industries, R&D departments
- Consulting organizations
- Scientists, engineers, econometrists and financial modelers
- GO software is used worldwide (software by PCS and partners is used at several hundred organizations)

Global Optimization Applications and Perspectives: Illustrative References

Authors/Editors	Application Areas, with Information on Software (in works denoted by +S)
Grossmann, 1996	Chemical Engineering Design + S
Pardalos, Shalloway & Xue, 1996	Computational Chemistry and Biology
Pintér, 1996	Environmental Modeling/Mgmt, and others + S
Corliss and Kearfott, 1999	Rigorous Optimization in Industry + S
Floudas et al., 1999	Handbook of Test Problems
Papalambros and Wilde (2000)	Engineering Design
Edgar, Himmelblau & Lasdon, 2001	Chemical Engineering Design/Operations+ S
Gao, Ogden & Stavroulakis, 2001	Physics (Mechanics)
Pardalos and Resende, 2002	Topical chapter by Floudas (Chem. Engrg)
Schittkowski, 2002	Model Fitting (Calibration) + S
Tawarmalani and Sahinidis (2002)	Chemical Engineering Design/Operations+ S
Diwekar (2003)	Environmental Modeling/Mgmt + S

Global Optimization Applications and Perspectives: Illustrative References

Authors/Editors	Application Areas, with Details on Software (in works denoted by +S)
Locatelli, Schoen et al. 2000+	Computational Chemistry and Biology + S
Stojanovic, 2003	Financial Modeling + S
Zabinsky, 2003	Engineering Design + S
Neumaier, 2004	See topical review sections + S
Bartholomew-Biggs, 2005	Financial Modeling and Optimization
Liberti & Maculan, 2005	Chapters on Software Implementations + S
Nowak, 2005	MINLP Software Devpt & Tests + S
Pintér, 2006	Global Optimization with Maple + S
Pintér, 2006	GO: Sci & Engrg Case Studies + S
Pintér, 200…	Applied NLO in Modeling Environments + S
Kampas & Pintér, 200…	Modeling & Opt. Using Mathematica + S

Further information is welcome

Note: Keep an eye also on other literature not written by GO researchers — numerous examples discussed by professionals who need GO…

Mathematical Programming Glossary Supplement: Global Optimization

Contributed by János D. Pintér

http://www.pinterconsulting.com
http://myweb.dal.ca/jdpinter/

Portions of this material originally appeared at MathWorld, and they appear here with their consent.

Introduction

The Global Optimization Model

The objective of global optimization (GO) is to find the globally best solution of nonlinear models, in the possible or known presence of multiple local optima. Formally, GO seeks the global solution set of a constrained optimization model of the form:

$$\min f(x): g(x) <= 0, xl <= x <=xu,$$

where x, xl, xu are finite real n-vectors; f is a real-valued (scalar) function; and, g is a real-valued m-vector function. All inequalities are interpreted component-wise. Additional structural assumptions typically include at least the continuity of the model functions in the n-interval [xl,xu]. Denote the feasible set by

$$D := \{x: xl <= x <= xu, g(x) <= 0\},$$

which we assume is nonempty. Then, the above-stated, rather minimal assumptions guarantee that the global optimization model is well-posed since it follows (by the Bolzano-Weierstrass theorem) that the solution set of the global optimization model is nonempty.

Algebra
Applied Mathematics
Calculus and Analysis
Discrete Mathematics
Foundations of Mathematics
Geometry
History and Terminology
Number Theory
Probability and Statistics
Recreational Mathematics
Topology

Alphabetical Index
Interactive Entries
Random Entry
New in *MathWorld*

MathWorld Classroom

About *MathWorld*
Send a Message to the Team

Order book from Amazon

Last updated:
12,646 entries
Sun Mar 4 2007

Applied Mathematics > Optimization
MathWorld Contributors > Pinter

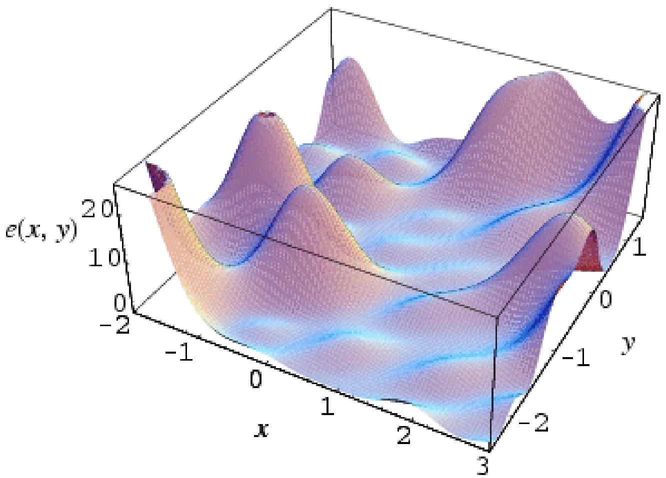
Global Optimization

COMMENT
On this Page

The objective of global optimization is to find the globally best solution of (possibly nonlinear) models, in the (possible or known) presence of multiple local optima. Formally, global optimization seeks global solution(s) of a constrained optimization model. Nonlinear models are ubiquitous in many applications, e.g., in advanced engineering design, biotechnology, data analysis, environmental management, financial planning, process control, risk management, scientific modeling, and others. Their solution often requires a global search approach.

A few application examples include acoustics equipment design, cancer therapy planning, chemical process modeling, data analysis, classification and visualization, economic and financial forecasting, environmental risk assessment and management, industrial product design, laser equipment design, model fitting to data (calibration), optimization in numerical mathematics, optimal operation of 'closed' (confidential) engineering or other systems, packing and other object arrangement problems, portfolio management, potential energy models in computational physics and chemistry, process control, robot design and manipulations, systems of nonlinear equations and inequalities, and waste water treatment systems management.

To formulate the problem of global optimization, assume that the objective function f and the constraints g are continuous functions, the component-wise bounds x_l and x_u related to the decision variable vector x are finite, and the feasible set D is nonempty. These assumptions guarantee that the global optimization model is well-posed since, by the extreme value theorem, the solution set of the global optimization model is nonempty.



Mathematics

Global Optimization in Action
Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications
Series: Nonconvex Optimization and Its Applications, Vol. 6
Pintér, János D.
1996, 512 p., Hardcover
ISBN: 0-7923-3757-3

Ships within 2-5 days

US$299.00

ADD TO CART

Print version
Recommend to others

Additional information
Reviews

All books by this author
Pintér, János D.

Related subjects
Applications
Mathematics

About this book | Table of contents

About this book

In science, engineering and economics, decision problems are frequently modelled by optimizing the value of a (primary) objective function under stated feasibility constraints. In many cases of practical relevance, the optimization problem structure does not warrant the global optimality of local solutions; hence, it is natural to search for the globally best solution(s).
Global Optimization in Action provides a comprehensive discussion of adaptive partition strategies to solve global optimization problems under very general structural requirements. A unified approach to numerous known algorithms makes possible straightforward generalizations and extensions, leading to efficient computer-based implementations. A considerable part of the book is devoted to applications, including some generic problems from numerical analysis, and several case studies in environmental systems analysis and management. The book is essentially self-contained and is based on the author's research, in cooperation (on applications) with a number of colleagues.

This e-book includes hands-on demos of the LGO IDE

HOME | HELP | LOGIN | MY SPRINGER | Please select

Springer

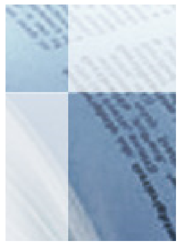
SEARCH | SEARCH BY All | GO | ADVANCED SEARCH

Mathematics

Journals | Series | Textbooks | Contact

Select your subdiscipline | Select a discipline

> Home / Mathematics



Global Optimization
Scientific and Engineering Case Studies
Series: Nonconvex Optimization and Its Applications, Vol. 85
Pintér, János D. (Ed.)
2006, Hardcover
ISBN: 0-387-30408-8

Not yet published. Available: April 2006

approx. US$99.95

Print version
Recommend to others

All books by this editor
Pintér, János D.

Related subjects
Computational Science & Engineering
Engineering
Mathematics

About this book | Table of contents

About this book

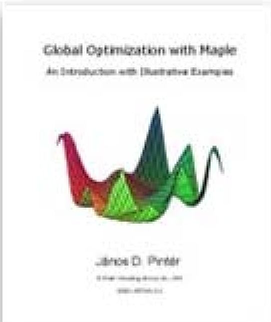
Optimization models based on a nonlinear systems description often possess multiple local optima. The objective of global optimization (GO) is to find the best possible solution of multiextremal problems. This volume illustrates the applicability of GO modeling techniques and solution strategies to real-world problems.

The contributed chapters cover a broad range of applications from agroecosystem management, assembly line design, bioinformatics, biophysics, black box systems optimization, cellular mobile network design, chemical process optimization, chemical product design, composite structure design, computational modeling of atomic and molecular structures, controller design for induction motors, electrical engineering design, feeding strategies in animal husbandry, the inverse position problem in kinematics, laser design, learning in neural nets, mechanical engineering design, numerical solution of equations, radiotherapy planning, robot design, and satellite data analysis. The solution strategies discussed encompass a range of practically viable methods, including both theoretically rigorous and heuristic approaches.

Written for:

Researchers and practitioners in academia, research and consulting organizations, and industry mapleconnect™

Global Optimization with Maple™



More Information:

- Features List
- Download Size
- Technical Requirements
- Pricing
- Technical Support
- Author Information

Product Description:

This electronic book presents Maple as an advanced model development and optimization environment. A special emphasis is placed on solving multiextremal models using the Global Optimization Toolbox™ for Maple™. Following a brief topical introduction, an extensive collection of detailed numerical examples and illustrative case studies is presented.

The following topics are covered:

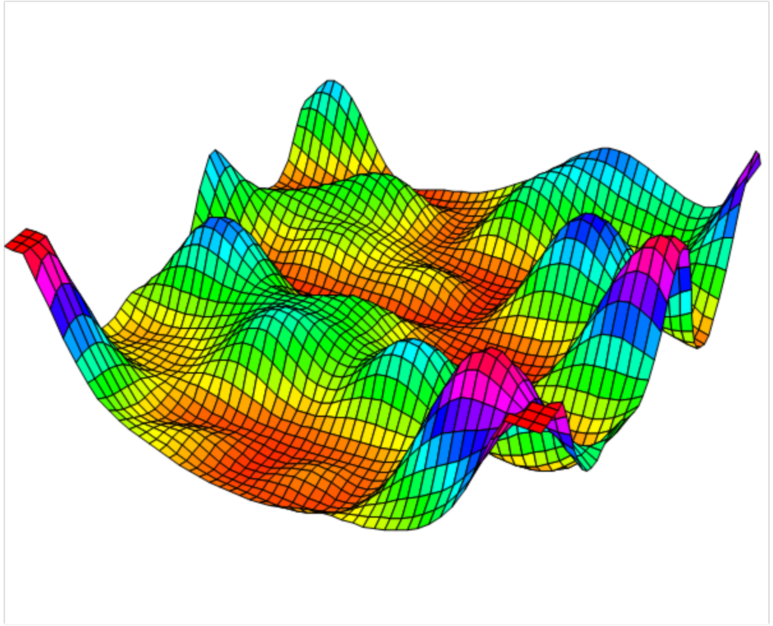
- A brief introduction to Operations Research / Management Science (ORMS)
- Maple as an integrated platform for developing ORMS studies and applications
- A review of the key global optimization concepts
- The Global Optimization Toolbox™ (GOT) for Maple™, including a concise discussion of the core LGO™ solver technology
- Model development tips
- Detailed "hands-on" numerical examples of using the GOT, from a simple illustration of the key tools and options to more advanced challenges
- Illustrative case studies from the sciences and engineering.

Intended Audience:

This electronic book will be of interest to practitioners, researchers, academics, and students in the sciences and engineering.

Optimization with Mathematica

Scientific, Engineering, and Economic Applications



Frank J. Kampas and János D. Pintér

ELSEVIER SCIENCE (forthcoming)

Conclusions 1

- Global optimization is a subject of growing importance: it is relevant in many areas in the sciences, engineering, and economics
- Development and application of sophisticated, complex numerical models frequently requires the use of global scope optimization methodology
- Professionally developed and supported GO solver options are available for a range of platforms; further development in progress

Conclusions 2

Several Key Application Areas

- **Advanced engineering**
- **Chemical and process industries**
- **Defense, security**
- **Econometrics and finance**
- **Math/physics/chemistry/biology**
- **Medical and pharmaceutical R&D**

Conclusions 3

Some Key Challenges and Future Work

- Integrate exact and heuristic methods
- Handle problems with (very) costly functions
- Handle problems w/o any exploitable structure
- Stochastic optimization: simulation and optimization
- Dynamic models: ODE/PDE solvers and optimization

Conclusions 4

Interest in R&D and Business Cooperation

- Customized model, algorithm, software, DSS development and related consulting services
- Workshops and tutorials
- Demonstration software, reports, and articles available
- New test examples and practical challenges are welcome

Thanks for your attention!

Further information:
www.pinterconsulting.com

Comments and questions:
janos.d.pinter@gmail.com