CEJOR (2008) 16:165–178
DOI 10.1007/s10100-007-0045-8

ORIGINAL PAPER

Model development and optimization in interactive computing environments

János D. Pintér

Published online: 26 March 2008

Abstract Integrated computing systems—such as Maple, Mathematica, and MATLAB—enable the development of "live" electronic documents that include explanatory text, calculations, custom programming (code development), visualization, and other features. As a result, teachers and students, researchers and practitioners can develop applications in a completely interactive format. Such e-documents can be put to good use in developing textbooks, lecture notes, assignments and presentations, as well as in the context of research and development (R&D) projects. The interactive approach accelerates and enhances the process of learning and research. To illustrate this approach, we discuss a nonlinear (global and local) optimization software product and a topical electronic book that support interactive model development and optimization in Maple. We highlight the key features of the e-book and the software, present illustrative examples, and point towards a range of scientific and engineering applications.

Mathematics Subject Classification 65K30 · 90C05 · 90C31

Keywords Advanced systems modeling and optimization · Interactive electronic documents for research and education · Integrated computing systems · Maple · Global Optimization Toolbox · Illustrative examples and applications

J. D. Pintér (✉)
PCS Inc., 129 Glenforest Drive, Halifax, NS, Canada B3M 1J2
e-mail: jdpinter@hfx.eastlink.ca
URL: http://www.pinterconsulting.com

Present Address:
J. D. Pintér
Department of Industrial Engineering, Bilkent University,
06800 Bilkent, Ankara, Turkey
e-mail: janos.d.pinter@gmail.com; jdpinter@bilkent.edu.tr

1 Introduction

Operations research provides a scientifically established methodology to assist analysts and decision makers in finding efficient (optimized) decisions under ever-present resource constraints. As it is well-known, a "generic" O.R. project consists of the following key steps:

- Conceptual description of the decision problem
- Development of a corresponding quantitative model
- Development and/or adaptation of a suitable algorithmic solution procedure
- Numerical solution of the model and its verification
- Implementation of the decision selected.

Many of the real-world problems tackled by O.R. are complicated, and the "best possible" model development and solution procedure may not be clear at the beginning of the project. As a result, interdisciplinary teams of decision makers, domain experts, modelers and algorithm developers often have to work together in an iterative fashion. Specifically, the O.R. team repeatedly has to modify the model formulation and solution procedure until the model captures the essence of the problem, it is adequately supported by data, computationally tractable, and its solution is deployable in the real-world setting.

The outlined stages of decision support system development point towards the advantages of using (also) high-level software tools. We specifically refer here to integrated scientific–technical computing (ISTC) systems. The most prominent currently available professional ISTC platforms are Maple (Maplesoft 2007a), Mathematica (Wolfram Research 2007), and MATLAB (The MathWorks 2007). The alternative terminology "problem solving environments" is used by Landau (2005), in relation to Maple and Mathematica.

Integrated scientific–technical computing systems can be put to good use in R&D, as well as in education. A key feature of ISTC systems is their capability to develop interactive electronic work documents. Such e-documents can be modified and extended according to the needs and interests of their user(s). Their interactivity enables active learning and hands-on experimentation, rapid prototyping and quick tests, incremental and modular development, to create, capture and preserve knowledge. We will illustrate the advocated approach by developing and solving nonlinear optimization models using Maple and the Global Optimization Toolbox (GOT) (Maplesoft 2007b).

The structure of the paper is as follows. Section 2 introduces Maple and the GOT. Section 3 discusses model development and optimization using (electronic) Maple worksheet documents. Section 4 introduces several examples: these serve to illustrate a simple-to-follow model formulation and solution approach, feasibility verification, model visualization, incremental model development, parameterized models, and models with embedded computational procedures. Section 5 discusses the development of complete case studies, briefly reviewing one such study. The conclusions in Sect. 6 are followed by references.

For technical and pedagogical details that are not discussed here, we refer to Pintér et al. (2006), Castillo et al. (2008), as well as to the Maple software documentation (Maplesoft 2007a), and to the e-book (Pintér 2006b). From a growing collection of electronically available books that can be used in conjunction with ISTC environments, we refer here only to several representative works by Landau (2005), Lopez (2005), and Trott (2004, 2005).

2 Maple and the GOT

2.1 Maple

Maple supports the development of interactive documents that combine technical description, computing, program code development, and visualization, with many additional features. Maple incorporates an extensive mathematical function library: its more than 3,500 built-in functions cover a multitude of scientific and technical research areas. Maple also includes a detailed on-line documentation system with hyperlinks, debugging tools, data communication (import from and export to various software packages) Web connectivity, automated (ANSI C, Fortran 77, Java, MATLAB and Visual Basic) code generation, and document production (including Maple to HTML, MathML, TeX, and RTF converters). Maple is portable across all major hardware platforms and operating systems, including most current Windows, Macintosh, Linux, and Unix versions.

For further details, we refer to Maplesoft's web site (http://www.maplesoft.com) that offers extensive topical information. This information includes quick start guides, training materials, complete user manuals, as well as interactive product and feature demonstrations. For illustration, see Fig. 1 that shows an integrated demonstra-

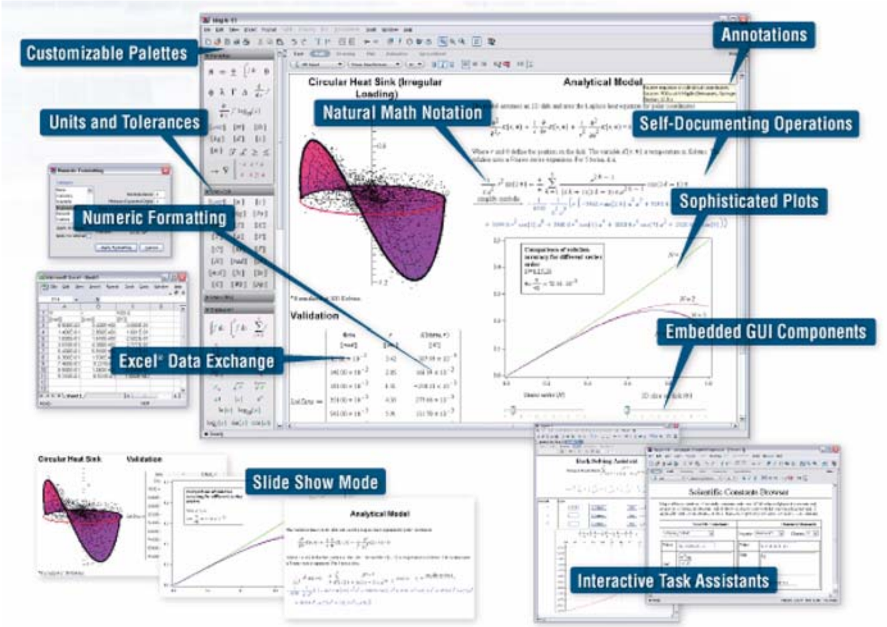


Fig. 1 Maple feature demonstrations through an integrated interface

tion interface available at Maplesoft's web site: clicking on the fields of the (active website) figure leads to the corresponding feature demonstrations. Additionally, downloadable white papers, articles, demo applications, course materials and complete case studies are available: these serve to illustrate the use of Maple in educational, engineering design, scientific research, operations research, financial analysis, and other contexts.

2.2 Global Optimization Toolbox for Maple

Within the general quantitative framework of systems modeling and optimization, we see a particularly strong case for using ISTC software to assist the analysis and management of nonlinear systems. Such systems are often defined by individually formulated model objective and constraint functions, without a ready-to-use "canonical" description (as opposed, e.g., to linear programming). Solving an arbitrary system of nonlinear equations that could express system equilibrium conditions is a case in point. To cite further examples, the optimization model functions may require, e.g., the execution of computational procedures defined by special functions, the numerical evaluation of integrals, the numerical solution of systems of differential equations, calls to external function modules, deterministic or stochastic simulation, and so on.

Nonlinear descriptive models are essential in various scientific and engineering studies and applications. Here we refer only to an illustrative selection of topical discussions by Aris (1999), Bracken and McCormick (1968), Diwekar (2003), Gershenfeld (1999), Grossmann (1996), Hansen and Jørgensen (1991), Lopez (2005), Mastinu et al. (2006), Murray (1983), Papalambros and Wilde (2000), Pardalos and Resende (2002), Pintér (1996, 2006a), Schittkowski (2002), and Steeb (2005).

Nonlinear optimization models—typically based on an underlying descriptive model—frequently have a multitude of (local and global) optima. The objective of global optimization is to find the "absolutely best" solution of such multimodal problems. The continuous global optimization model form considered here is

$$\min f(x) \quad x \in D := \{x : x_l \leq x \leq x_u \quad g(x) \leq 0\}. \tag{1}$$

Here $x \in \boldsymbol{R^n}$ is the real n-vector of decision variables, x_l and x_u are finite, component-wise n-vector bounds of x, $f : \boldsymbol{R^n} \to \boldsymbol{R}$ is a continuous (scalar) objective function, and $g : \boldsymbol{R^n} \to \boldsymbol{R^m}$ is the continuous m-vector function of constraints (the inequality $g(x) \leq 0$ is interpreted component-wise).

Without going into technical details that are unnecessary within the scope of our discussion, let us point out that the model statement (1) covers a broad class of formally more general or detailed models. Specifically, instead of $\leq$ relations, arbitrary combinations of inequality and equality relations could be used in the functions g; one could state explicit bounds also on the constraint function values; and one could even use a combination of continuous and discrete (finitely bounded) decision variables. Observe furthermore that if the feasible set D is non-empty, then model (1) has a non-empty global solution set X^*. In spite of this key existence result, if we use traditional local scope search methods to solve multimodal instances of (1), then—as a rule, depending on the starting point of the search—we will find only corresponding locally optimal solutions. In order to find the "true" solution (i.e., to properly approximate it after a finite number of algorithmic search steps) in such cases, a genuine global scope search strategy is required.

The GOT for Maple has been developed to meet these needs (Maplesoft 2007b). The core of the GOT is the LGO global–local nonlinear optimization solver suite. LGO abbreviates Lipschitz(-continuous) Global Optimizer, originally named after one of its global solver components. The theoretical foundations of LGO are discussed in Pintér (1996), together with basic implementation details, test examples, and a collection of detailed case studies. The LGO software has been further developed to a significant extent. It is currently available to use in conjunction with C and Fortran compiler platforms, several optimization modeling languages (AIMMS, AMPL, GAMS, MPL), and ISTC systems (Maple, Mathematica, and MATLAB via Tomlab). For details, consult for example Pintér (2001, 2002, 2005, 2007), as well as the platform-specific user manuals not cited here.

Since its 2004 release, the GOT has been used worldwide by industrial and government organizations, R&D laboratories and consulting companies, as well as by academic researchers and educational users. The GOT has also been peer-reviewed by academic and industrial experts, consult Castillo (2005), Wass (2006), Henrion (2006), Deschaine (2006) and Forst (2008).

The GOT product page includes links to its key technical features and illustrative applications, to a recorded web seminar, technical datasheet, white paper, several peer reviews, and to topical books, articles, and presentations. In the following sections we will use Maple and the GOT, to illustrate the application of interactive documents in the context of O.R.

3 Global Optimization with Maple: an electronic book

The e-book (Pintér 2006b) has been written entirely using the Maple environment. The book presents Maple as a model development and optimization environment. The emphasis is placed on nonlinear (often multi-extremal) models: to solve these, the GOT is used throughout the book. However, readers can also experiment with Maple's built-in (local nonlinear) optimization functionality, or perhaps with their own solvers.

The e-book is available in the form of a single fully functional, printable Maple worksheet, and/or as a set of nearly 60 hyperlinked worksheets (at the time of this writing). The content of the e-book can be modified or expanded, as deemed suitable. The book consists of five chapters, in addition to front material, conclusions, and references. Chapter 1 presents a brief introduction and background regarding O.R., Maple, global optimization, and LGO. Chapter 2 introduces the GOT for Maple: the discussion covers installation and initialization steps, basic option settings, model formulation and function evaluation options, model development and solution tips. Chapter 3 presents several dozen numerical examples to illustrate GOT usage basics. The discussion covers the solution of standard nonlinear optimization test examples and some more challenging problems. Chapter 4 discusses more advanced examples: these include optimization models with various embedded computable functions (such as a parametric integral or a Bessel function in the model objective), calls to MATLAB to define model functions, and the use of external (precompiled C) functions in a model. Chapter 5 presents illustrative case studies (these will be mentioned later on); the last section discusses further application perspectives.

This electronic book is written primarily for model developers—practitioners, researchers, academics, and students—working in the sciences and engineering, with or without an in-depth optimization background. The e-book can be used as a self-study resource, or as part of a practically oriented modeling and optimization curriculum. To follow the exposition, an undergraduate level of mathematical knowledge and basic Maple programming experience are sufficient. In order to execute all "live" examples presented in the Maple worksheet(s), the reader should have access to a current version of Maple (version 9.5 or higher) and to a compatible version of the GOT.

In the next section, we will highlight some key features of the interactive model development and solution approach. The examples presented are similar to some of those found in the e-book. Due to space limitations (to be followed by this article), we will discuss here only small-size—yet non-trivial—models that can be presented in a sufficiently compact fashion.

4 Illustrative examples

4.1 Model development, solution and analysis

A key feature of ISTC systems is their interpreted (as opposed to compiled) language environment. An apparent manifestation of this feature is that one can enter commands one by one, and the system will respond "immediately" (barring time-intensive calculations), unless we suppress its response. The interpreted environment supports interactive exploration, model prototyping, and modular development.

We will not discuss Maple programming details here, but the examples presented should be easy to follow. All Maple input will be typeset using `Courier Bold` fonts. Maple's output will be shown using standard (Times New Roman) fonts, immediately following the corresponding input line(s). Individual Maple input lines typically end with the symbol "`;`" that leads to a corresponding Maple output. To suppress the resulting output, the symbol "`:`"can be used after an input line(s). Except minimal editing done for the purposes of this article, all Maple input and output is directly copied and pasted from a corresponding Maple worksheet document.

The following Maple command allows us to launch directly the (subsequently issued) commands of the GOT.

```
> with(GlobalOptimization):
```

The GOT automatically sets default parameter values for its operations, partly based on the model to solve. According to extensive numerical experience, the suggested default settings are suitable in many cases, but the user can always directly assign (i.e., override) them. In all examples presented here, the default settings are used.

Our first example shows a simple-to-follow "standard" model development style. First we define the key model components, essentially following the structure of the mathematical model formulation (1). The model (2) displayed below has two variables (x and y), and three nonlinear constraints given in a list between the braces {}, in addition to bound constraints.

```
> objective := (2 * sin(1 + 2 * x) * sin(3 * x + y))^2 :        (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 :
```

Notice that the objective function and two of the constraints are non-convex: this structure leads to a global optimization problem. The model is solved using the GOT: the GlobalSolve command computes a numerical global solution, in a fraction of a second (on today's average personal computer).

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

solution := [1.61054103392516999 $\cdot 10^{-32}$,
[x = –0.00925260522200139526, y = 0.0277578156660041112]]

The solution output consists of the objective function value, followed by the assigned list of optimal solution vector components. Since—by its definition—the model objective function in (2) is always non-negative, a feasible solution with an approximately zero objective function value is a numerical global solution. The next statement serves to verify numerical feasibility, by evaluating the constraints at the solution argument.

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

0.01002287059 ≤ 0.3, 0 ≤ 0.0008561070340, 0.001369771254 ≤ 0.1

Let us point out that we do need global optimization technology to solve this model. The next two statements illustrate this point: first we invoke the built-in local nonlinear solver, and then attempt to solve the model.

```
> with(Optimization):
> localsolution:=
Minimize(objective, constraints, bounds);
```

localsolution := [3.45511225499958430,
[x = .474341655354528546, y = .158113889337685638]]

Let us also note that if one is interested in other (quite possibly existing, "noticeably different") global solutions, then, e.g., a "sufficiently small" spherical neighborhood of the current global solution can be excluded by adding a single constraint to the model formulation (2). After adding such a constraint, the model can be immediately resolved. The modified model displayed in (3) implements this approach. Here we prefer to show a very simple way to do this, instead of perhaps more sophisticated Maple programming. For clarity, we repeat also the unchanged model components. Notice also the rounding of the previous solution, for facilitating the display of the modified model.

```
> objective := (2 * sin(1 + 2 * x) * sin(3 * x + y))^2 :                (3)
constraints2 :=
{x^2 - 3 * x * y >= 0,
sin(y) * y - sin(x) <= 0.3,
(x - y)^2 <= 0.1,
(x + 0.0092526)^2 + (y - 0.0277578)^2 >= 0.0001} :
bounds := x = -1..3, y = -2..3 :
> solution2 := GlobalSolve
(objective, constraints2, bounds);
solution2 := [8.67322388733747448 · 10^-32,
[x = 0.0424073335558855652, y = -0.127222000667656876]]
```

The second global numerical solution found is also feasible, as it can be easily verified. Evidently, this approach can be iterated to find additional global solutions (until no further solutions can be found). Before dismissing such a pragmatic approach, recall that the general problem of finding all global solutions to an arbitrary instance of (1) is theoretically NP-complete.

One can also visualize the model functions: the command shown below displays the objective function, see Fig. 2.

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

The constraint functions can be visualized similarly. In higher-dimensional models, interactively selected subspace projections can be displayed. As Maple users will know, one can also rotate the surface plots generated: this option—and the many other visualization options not discussed here—can help users to gain further insight regarding their model.

4.2 Parameterized optimization problems

Model (3) illustrates the possibility of incremental development via inline model changes. To show another model development example, here we introduce a

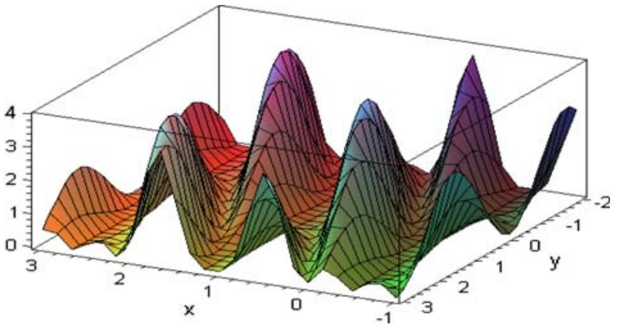


Fig. 2 Objective function in model (2)

parameterized model-class. The model objective includes a trigonometric term which can be made to "dominate" a simple underlying quadratic function, by increasing its multiplier parameter. When solving these models, one can observe that the precision of the numerical solution becomes somewhat less as the noise increases, but—in the examples presented—not too much. (In all cases, the theoretical unique global solution is the vector $x = y = 0$, with the corresponding objective function value 0.)

$$\begin{aligned}
&> \texttt{GlobalSolve}(\texttt{x\^{}2} + \texttt{y\^{}2} + 1 * (\texttt{sin}(\texttt{x\^{}2} + \texttt{x} + \texttt{y\^{}2} - \texttt{y}))\texttt{\^{}2},\\
&\texttt{x} = -8..1, \texttt{y} = -3..10);\\
&[2.65363041289996510 \cdot 10^{-22},\\
&[\mathrm{x} = 2.66241931525791942 \cdot 10^{-12}, \mathrm{y} = 1.26168271945189798 \cdot 10^{-11}]]\\
&> \texttt{GlobalSolve}(\texttt{x\^{}2} + \texttt{y\^{}2} + 10 * (\texttt{sin}(\texttt{x\^{}2} + \texttt{x} + \texttt{y\^{}2} - \texttt{y}))\texttt{\^{}2},\\
&\texttt{x} = -8..1, \texttt{y} = -3..10);\\
&[1.30003498875953083 \cdot 10^{-20},\\
&[x = 7.17520512867093428 \cdot 10^{-11}, y = 4.82487515260131190 \cdot 10^{-11}]]\\
&> \texttt{GlobalSolve}(\texttt{x\^{}2} + \texttt{y\^{}2} + 100 * (\texttt{sin}(\texttt{x\^{}2} + \texttt{x} + \texttt{y\^{}2} - \texttt{y}))\hat{}2,\\
&\texttt{x} = -8..1, \texttt{y} = -3..10);\\
&[2.98780842650330138 \cdot 10^{-20},\\
&[\mathrm{x} = 1.18729856020084858 \cdot 10^{-10}, \mathrm{y} = 1.21804064110639752 \cdot 10^{-10}]]
\end{aligned} \tag{4}$$

Of course, the parameterized models themselves are becoming more and more difficult as the multiplier of the trigonometric term is increased. Figure 3 displays the objective function of the third model version shown in (4). This figure illustrates the fact that global optimization models—even low-dimensional instances of the general problem statement (1)—can be truly difficult. In higher dimensions, as well as in the presence of added general constraints, nonlinear models tend to become even harder. Hence, the need for global scope search immediately becomes clear (e.g., in a classroom environment, or for non-mathematician members of an interdisciplinary R&D team). Parameterized model-classes similar to (4)—and optionally equipped with a new randomized solution vector for each instance—can be put to good use also in testing global optimization software as suggested and done in Pintér (2002).

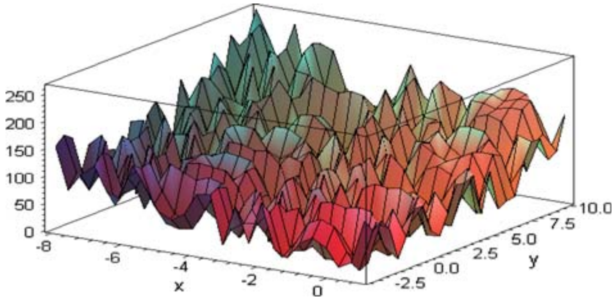


Fig. 3 Objective function in the third parameterized model shown in (4)

4.3 Models with embedded computational procedures

One of the apparent advantages of ISTC systems is the range of built-in functions—and their programmable extensions—that can be directly used in optimization model formulations. In order to illustrate this feature, in the next example we will use the built-in gamma function of Maple. Let us recall that the gamma function $\Gamma(z)$ is defined for complex z arguments with $Re(z) > 0$ by the integral expression

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

The gamma function is extended to the entire complex plane, except for non-positive integers, by analytic continuation. (Consult, e.g., Maple's Help system for more information.) Consider first the model shown in (5), and its numerical solution. Notice that for simplicity and consistency we use the same model component symbols as above: this could be changed if necessary, of course.

```
> objective := sin(4 * x^2 + 10 * y − GAMMA(x + 1))^2+        (5)
    sin(x − 2 * y − GAMMA(y))^2 :
constraints:={x^2 − y^5 <= −10,
    cos(x − y) * sin(−x + y^3 − 3) = 0} :
bounds := x = 1..8, y = 1..5 :
solution := GlobalSolve
    (objective, constraints, bounds);
solution := [0.00145307767952264008,
[x = 4.45034916356521926, y = 2.87955283435428422]]
```

Next, we will modify (only) the constraints of this model, and then resolve it.

```
> constraints2 := {x^2 − y^5 <= −7,                            (6)
        x * sin(−x + y^3 − 3) = 0} :
solution2 := GlobalSolve
        (objective, constraints2, bounds);
solution2 := [2.90055483780898152 · 10^−16,
[x = 5.00000000010406876, y = 2.00000000001128520]]
```

Inspecting now the model objective and constraints in (6), and recalling the fact that $\Gamma(z) = (z - 1)!$ for positive integer arguments z, one can directly verify that the solution found to the modified model (6) is a close numerical approximation of the theoretical global solution $x = 5$, $y = 2$. In the formulation (6) both constraints are active, and this fact helps to drive the GOT towards the exact solution. Notice also that the values returned in (5) using default GOT settings provide a verifiably

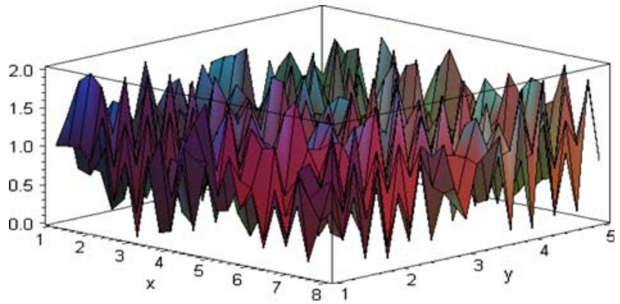


Fig. 4 Objective function in models (5) and (6)

feasible, high-quality, global search based solution (since the objective function in (5) is always non-negative). Barring numerical problems—that could occur even in the realm of local optimization—such a result can be expected by default from a numerical global solver. Figure 4 shows the surface plot of the objective function in the last two examples, indicating again the potential difficulty of instances within the very general model-class addressed by the GOT.

5 Developing case studies and applications

As it has been already noted, Maple enables the modular development of complete R&D projects. Due to space limitations, the numerical examples introduced above could only hint at these possibilities. To present a more realistic (but still fairly simple) illustrative application, we briefly review an automotive suspension system tuning problem. For further details—including the industrial background and motivation, as well as other aspects of modeling and optimizing vehicle dynamics—consult Goossens et al. (2007).

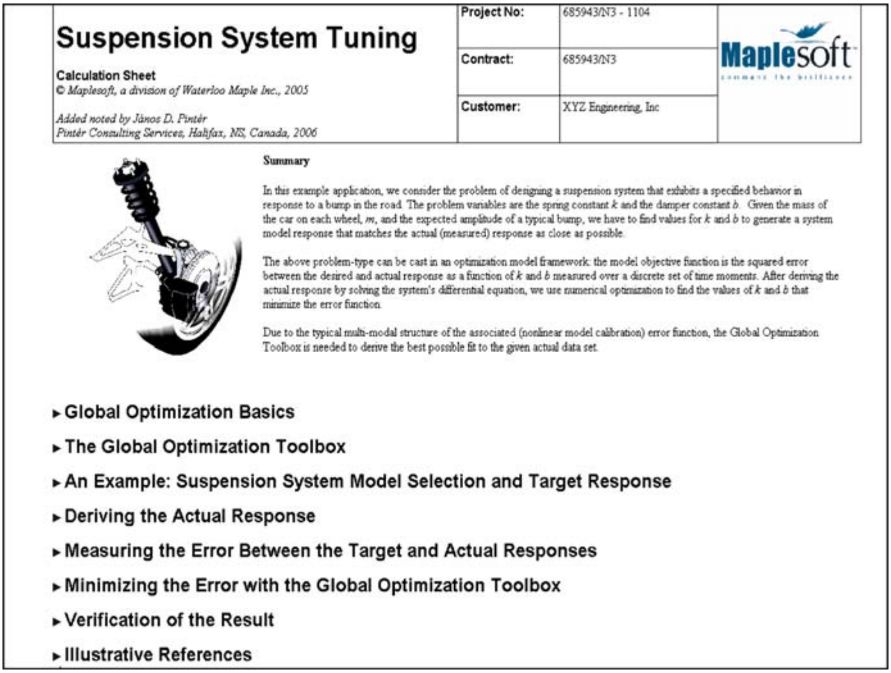
One of the many areas where the GOT is used extensively by industry is model fitting or calibration. Frequently, engineers will have a good theoretical model of a system component or module, but they do not have the necessary parameter values to make the model output (such as a trajectory) behave the same as the observed or desired system. In situations like this, engineers can capture system output data, and then use optimization techniques to find the best model input parameter values to match the target output as closely as possible. Since engineering and many other professional studies typically deal with highly non-linear dynamic systems, the use of global optimization in such applications is key, to obtain the best possible model fit. This is a more general, flexible and successful approach than using traditional statistical regression techniques that often restrict the class of stochastic models to be considered, and/or involve only local scope search methods.

For illustration, let us consider a vehicle suspension system parameterized by the spring stiffness parameter k and the damper parameter b. We shall use the basic second-order suspension (unforced mass-spring-damper) system model

$$m\left(\frac{d^2}{dt^2}x(t)\right)+b\left(\frac{d}{dt}x(t)\right)+k\cdot x(t)=0. \tag{7}$$

In (7) m is the mass of the car on each wheel. Using this model, for any given parameter combination (k, b) we can determine the dynamic response of the system using Maple's built-in ordinary differential equation solvers. However, one would have to use initial parameter "guesses" that will most likely not give the desired response, unless we are very experienced—or lucky... Given a prescribed suspension system behavior at a "typical" (specific size and shape) road bump, the designer's task is to find values for k and b to generate a system response that matches the target response at the bump. Obviously, this problem can be cast in an optimization model framework: the objective function is defined as the squared error between the desired and actual response measured over a discrete set of time moments, as a function of k and b. After deriving the parameterized response by solving the differential equation (7), one can use the GOT to find the values of k and b that globally minimize the error function. This general strategy has been followed in several topical case studies which validated the approach outlined above.

Figure 5 displays one of the related Maple worksheet documents in its concise format: its expandable sections (denoted by black right-pointing triangles) lead to



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

Project No:	685943/N3 - 1104
Contract:	685943/N3
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

Fig. 5 An illustrative case study developed in Maple and solved by the GOT

the complete documentation, computing and visualization details regarding the corresponding stages of our demonstration project.

A growing number of other examples and case studies that illustrate the application potentials of using Maple and the GOT are available from Maplesoft's website, and from Pintér (2006b). Current (publicly available) examples are related, e.g., to alkylation process modeling, calibration of multiple process Arrhenius models, aspherical lens surface identification, calibration of a molecular similarity index model, chemical equilibrium analysis, industrial design, electrical circuit design, general (non-uniform) circle packing problems, nonlinear equations, perfume bottle design, portfolio management, and supply chain performance (reliability) optimization. Further examples are available upon request from Maplesoft and from the author.

6 Conclusions

In this article, we have discussed the advantages of using "live" electronic documents in R&D and education. To illustrate this point, we have presented examples of interactive model development and optimization using Maple and the GOT.

We see especially strong application potentials for ISTC systems and related tools in the analysis of highly nonlinear systems. Several broad (and obviously overlapping) model-classes with a potential need for both ISTC systems and advanced optimization tools include models with a provably non-convex structure, "black box" models (with an unknown structure), as well as dynamic and stochastic system models. Models belonging to these categories are ubiquitous in educational, research and commercial contexts. These include, e.g., automotive, chemical, electrical, electronic, environmental and process engineering, econometric and finance studies, and a range of mathematical, physical, chemical, biological, medical and pharmaceutical applications.

Acknowledgments Thanks are due to my colleagues at Maplesoft—especially to Paulina Chin, Paul Goossens, Tom Lee, David Linder, William Spaetzel, and Christina Spirou—for useful discussions and cooperation related to the Global Optimization Toolbox, topical web seminars, the e-book, and other publications. I also wish to acknowledge the work and constructive suggestions of all reviewers of the GOT and/or the e-book: published reviews were written (so far) by Ignacio Castillo, Larry Deschaine, Wilhelm Forst, Didier Henrion, Mahmut Parlar, Tamás Terlaky, and John Wass. The constructive comments received from two anonymous referees of this article are also appreciated.

References

Aris R (1999) Mathematical modeling: a chemical engineer's perspective. Academic Press, San Diego

Bracken J, McCormick GP (1968) Selected applications of nonlinear programming. Wiley, New York

Castillo I (2005) Maple 10 and the global optimization toolbox (software review). ORMS Today 32(6): 56–60. Available online at http://www.lionhrtpub.com/orms/orms-12-05/swr.html

Castillo I, Lee T, Pintér JD (2008) Integrated software tools for the ORMS Classroom. Algorithm Oper Res (in press)

Deschaine LM (2006) Global optimization toolbox for Maple and global optimization with Maple (book and software review). Int J Model Identif Control 1(4):338–339

Diwekar U (2003) Introduction to applied optimization. Kluwer, Boston

Forst W (2008) Global optimization with Maple (book review). Algorithm Oper Res (in press)

Gershenfeld N (1999) The nature of mathematical modeling. Cambridge University Press, Cambridge

Goossens P, McPhee J, Schmitke C, Pintér JD, Stahl H (2007) Driving innovation: how mathematical modeling and optimization increase efficiency and productivity in vehicle design. Technical memorandum published by Maplesoft, Waterloo

Grossmann IE (ed) (1996) Global optimization in engineering design. Kluwer, Dordrecht

Hansen PE, Jørgensen SE (eds) (1991) Introduction to environmental management. Elsevier, Amsterdam

Henrion D (2006) A review of the Global Optimization Toolbox for Maple. IEEE Control Syst Mag 26 (October issue):106–110

Landau RH (2005) A first course in scientific computing. Princeton University Press, Princeton

Lopez RJ (2005) Advanced engineering mathematics with Maple, electronic book edition. Waterloo Maple Inc., Waterloo

Maplesoft (2007a) Maple (Version 11). Maplesoft, Waterloo. http://www.maplesoft.com

Maplesoft (2007b) Global optimization toolbox for Maple (Version 11). Maplesoft, Waterloo. http://www.maplesoft.com/products/toolboxes/globaloptimization/

Mastinu G, Gobbi M, Miano C (2006) Optimal design of complex mechanical systems with applications to vehicle engineering. Springer, Berlin

Murray JD (1983) Mathematical biology. Springer, Berlin

Papalambros PM, Wilde DJ (2000) Principles of optimal design—modeling and computation, 2nd edn. Cambridge University Press, Cambridge

Pardalos PM, Resende MGC (eds) (2002) Handbook of applied optimization. Oxford University Press, Oxford

Pintér JD (1996) Global optimization in action. Kluwer, Dordrecht

Pintér JD (2001) Computational global optimization in nonlinear systems: an interactive tutorial (includes LGO demo software). Lionheart Publishing, Marietta. See http://www.lionhrtpub.com/books/globaloptimization.html

Pintér JD (2002) Global optimization: software, test problems, and applications. In: Pardalos PM, Romeijn HE (eds) Handbook of global optimization, vol 2, chap 15. Kluwer, Dordrecht, pp 515–569

Pintér JD (2005) Nonlinear optimization in modeling environments: software implementations for compilers, spreadsheets, modeling languages, and integrated computing systems. In: Jeyakumar V, Rubinov AM (eds) Continuous optimization: current trends and applications, chap 5. Springer Science + Business Media, New York, pp 147–173

Pintér JD (ed) (2006a) Global optimization: scientific and engineering case studies. Springer Science + Business Media, New York

Pintér JD (2006b) Global optimization with Maple—an introduction with illustrative examples. An electronic book distributed by Maplesoft and Pintér Consulting Services. See http://www.maplesoft.com/products/thirdparty/GOT/index.aspx

Pintér JD (2007) LGO—a model development and solver system for continuous global optimization. User Guide (current edition). Pintér Consulting Services, Inc., Halifax. For related information, see http://www.pinterconsulting.com

Pintér JD, Linder D, Chin P (2006) Global optimization toolbox for Maple: an introduction with illustrative applications. Optim Methods Softw 21(4):565–582

Schittkowski K (2002) Numerical data fitting in dynamical systems. Kluwer, Dordrecht

Steeb W-H (2005) The nonlinear workbook, 3rd edn. World Scientific Publishing, Singapore

The MathWorks (2007) MATLAB (Version: R2007b). The MathWorks, Inc., Natick. http:/www.mathworks.com

Trott M (2004, 2005) The Mathematica GuideBooks, vol 1–4. Springer Science + Business Media, New York

Wass JA (2006) Global optimization with Maple—an add-on toolkit for the experienced scientist (software review). Scientific Computing, p.16 and fwd

Wolfram Research (2007) Mathematica (Version: 6.0). Wolfram Research, Inc., Champaign. http://www.wolfram.com