

Funneling Algorithms for Multi-Scale Optimization on Rugged Terrains

by

Angelo Lucia*, Peter A. DiMaggio and Praveen Depa

Department of Chemical Engineering

University of Rhode Island

Kingston, RI 02881-0805

August 2003

Revised November 2003

* Corresponding Author: Tel: 401.874.2814; Fax: 401.874.4689; Email: lucia@egr.uri.edu

Abstract. Global optimization problems for which it is either intractable or undesirable to find all stationary points and that contain so-called ‘rough’ or rugged objective function landscapes are studied. These problems often show considerable differences between small-scale (or local) geometry and large-scale (or more global) geometry. A new class of functions called generalized exponential funnel functions is proposed for modeling large-scale geometry. This class of functions is capable of modeling folds, cups, cones, funnels and other geometric objects that recur in many physical applications. The basic mathematics of exponential funnels is described along with their numerical and algorithmic implications. It is shown that funnels provide a non-convex model of large-scale geometry which, when approximated correctly, possesses a unique and easily calculated minimum that can be estimated from a small amount of local objective function and derivative information. Ways of extracting ‘average’ gradient and Hessian matrix information are presented and novel interpolating formulae for building funnel approximations of large-scale geometry are given. A funneling algorithm, intended to guide or ‘funnel’ iterates to regions where the most promising global optimizers are expected to lie, is described.

A multi-scale global optimization algorithm based on the combined use of terrain methods and funneling algorithms is proposed. The terrain methodology is used to gather small-scale information while a funneling algorithm is used to guide the overall optimization calculations and to make ‘large’ moves within the feasible region. Communication between scales is also addressed. Typical empirical force field functionality, many stationary points and changes in convexity are used to clearly illustrate by example but not theoretically that the combined terrain/funneling methodology is capable of finding a global minimum without calculating all stationary points and can lead to significant reductions in overall computational work.

Keywords: global optimization, large-scale geometry, exponential funnel functions, funneling algorithms

1. Introduction. Optimization is still an active area of research and an important computational tool in many branches of science and engineering. For example, optimization can be used to determine molecular conformation and transition pathways in computation biology. Often times, however, there are dramatic differences between the local and global geometry of a given objective function that can be important in the optimization process. These geometric differences can be inherent or they can be due to the presence of multiple length scales (e.g. short and long-range, intra- and inter-particle forces). While quadratic Taylor series expansions represent reasonable models of local geometry, they are generally inadequate for describing global geometry, which is usually strongly non-quadratic. Figure 1 shows the least-squares function for estimating binary interaction parameters for the Non-Random Two-Liquid (NRTL) equation for activity coefficients by fitting the model to liquid-liquid equilibrium data. Note that the overall shape of the objective function surface is clearly non-quadratic with large sections that appear loosely folded, cupped or flat. Nevertheless all sections ‘funnel’ trajectories of the gradient vector field to the dark region where the stationary points lie. Figure 2, on the other hand, is a reproduction of Fig. 4 in Onuchic et al.¹ that is a schematic representation of the protein folding free energy landscape. Here the global geometry of the free energy surface is truly shaped like a funnel because it’s a schematic. However, one current accepted view of protein folding is described by a phenomenon called funneling. That is, while the Gibbs or Helmholtz free energy landscape associated with folding is ‘rough’ at the small scale, the large-scale geometry of the free energy surface is funnel-shaped. Moreover, it is believed that this large-scale funnel-shaped geometry guides protein conformation dynamics toward the bottom of the funnel where the native protein structure or global minimum in the free energy

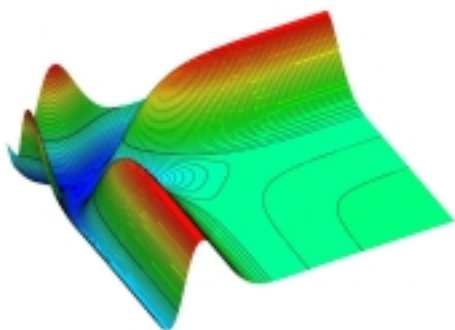


Figure 1: Parameter Estimation Surface for NRTL Equation

sits. However, because of the roughness (or frustration) of the landscape a protein can fold incorrectly or get trapped in any number of local minima that are encountered during the dynamics of folding. Understanding protein folding is an important problem because the function of a protein is closely connected to its native conformation. Misfolding, on the other hand, can destroy protein function and has been linked to a variety of diseases. Tovchigrechko and Vakser² have shown that the

potential energy surface used to describe protein-protein interactions (or docking) is also funnel-shaped. Then there's Fig. 3. This is the least-squares surface for the mass and phase equilibrium equations for finding all azeotropes of a binary mixture. Note the region on the right looks like an inverted cone while the region on the left is sharply folded. Like the parameter estimation example, the solutions of interest lie in the valley at the bottom of the figure.

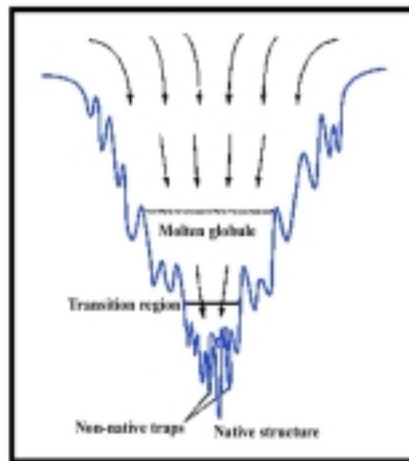


Figure 2: Schematic of Protein Folding Energy Landscape*
From Douchik, Luthey-Schultze & Wolynes, Theory of Protein Folding: The Energy Landscape Perspective, Annual Reviews in Physical Chemistry, v 48, 505-509 (1997).

These and other examples we have studied clearly show that diverse applications have many common topological similarities in large-scale geometry (i.e., folds, cups, cones, funnels). This includes thermodynamic potential surfaces in multiphase equilibrium, distillation problems, molecular conformation examples and others. **However, the discovery that this recurring large-scale or global geometry can be adequately modeled by a single, rather simple non-quadratic function, which we call a generalized exponential funneling function, came as a complete surprise to us.**

The free energy surface in protein folding is also an example of a multi-scale optimization problem. These problems generally have many, many stationary points on a local scale and a unique global optimizer on a larger

scale. However, protein folding is not the only example of the so-called many minima problem in computational biology. Protein-protein interactions can be described by potential energy surfaces that have similar characteristics. There are also examples of the many minima problem in engineering. Recent work by Heidemann et al.³ in phase equilibrium in wax precipitation clearly shows the presence of many solid solution phases. We have calculated that the results

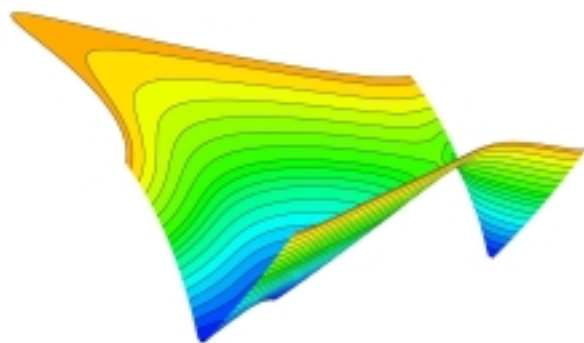


Figure 3: Azeotropic Surface

given in Heidemann et al. for a 14-component system with 12 equilibrium phases correspond to 1002 equilibrium points and hundreds of meta-stable solutions. However, there is only one global minimum. Thus wax precipitation equilibrium has the same generic geometric properties as protein folding – distinctly different local and global geometry in the presence of many stationary points. We have also encountered a number of other physical problems in which distinct differences in local and global geometry exist in the presence of many stationary points. These include optimal parameter estimation problems in reactor modeling, shape optimization in the design of prosthetic devices, and stagnation point calculations in dynamic wetting problems.

Protein Structure and Interactions. Proteins are extremely complex molecules generally made up of primary amino acid sequences, secondary α -helices and β -strands and more detailed three-dimensional components that constitute tertiary structure. The amino acids are linked together by amide bonds, form polypeptide chains and these polypeptide chains often form secondary helical or stranded structures. In addition, β -strands can be linked by hydrogen bonding to form β -sheets with varying parallel or anti-parallel structure while α -helices and β -strands have been observed to form TIM barrels. Other organization elements such as loops, turns, and domains can also be present in protein structures. To say the least, proteins organize on many levels (or length scales), each level can have its own geometry, and they organize into very complex geometric objects quickly and efficiently (Levinthal⁴).

While there are many important aspects of protein structure and interactions, it is widely recognized that the three-dimensional molecular conformation that a protein adopts has a significant impact on its function and its interaction with other proteins. The ability to predict the native structure of proteins has remained one of the important grand challenges in science for many years. However, its importance cannot be overstated. A clear understanding of protein structure and the pathways by which a protein folds and unfolds would have enormous social and economic impact in health and medicine. Knowledge of protein-protein and protein-DNA interactions would also have similar impacts. One way to build an understanding of molecular conformation and pathways is through stochastic and/or deterministic global optimization and many approaches have been used. General-purpose deterministic methods like the tunneling algorithms of Levy and Montalvo⁵ and Bahren and Protopopescu⁶, the α BB algorithm of Maranas and Floudas⁷, the diffusion method of Piela and co-workers⁸ and other surface deformation methods, and the interval analysis methods of Hansen⁹ and Schnepfer and Stadtherr¹⁰ can be used. Also general-purpose stochastic methods for global optimization like simulated annealing, the stochastic differential equations

approach of Aluffi-Pentini and co-workers¹¹ and the work of Bilbro¹² can be used. Most general-purpose methods tend not to exploit any problem specific information and are therefore widely applicable. There are also deterministic and stochastic algorithms designed for particular classes of problems. Methods here include the chain-of-states method of Seveck et al.¹³, the build-up procedures for peptides and proteins developed by Scheraga¹⁴, threading algorithms for protein folding¹⁵, the nudged elastic band (NEB) methods of Henkelman et al.¹⁶ for transition state and reaction pathway calculations, and Monte Carlo and molecular dynamics approaches for conformational analysis by potential energy minimization like the work of Bolhuis et al.¹⁷. For a good survey of the application of global optimization in molecular modeling, see Westerberg and Floudas¹⁸.

Recent theoretical and experimental evidence that supports the ideas that protein dynamics are guided by funneling has sparked a great deal of interest. It is now accepted as a guiding principle for understanding the complexity of protein folding and protein-protein interactions. However, the role of funneling in algorithmic development for predicting protein structure and interactions has not been exploited. **We believe that the large-scale funnel shape of the energy landscape can be used to construct reliable and efficient global optimization algorithms for predicting protein structures and pathways** in ways that permit known structural elements like α -helices, β -strands, β -sheets, loops, turns, and others structural objects to also be incorporated. One way to do this is to use a multi-scale optimization methodology that uses a model of large-scale geometry based on the non-quadratic exponential funnel functions alluded to earlier.

Problem Statement. The problem we are interested in is that of finding the global optimum of a twice continuously differentiable function, say $f(z)$, that has many, many stationary points on some fixed feasible region defined by bounds on the variables. That is, f is a non-convex C^2 function of the n unknown variables, say z , such that f takes R^n into R .

Some New Ideas. The relatively few non-quadratic models that have been proposed over the years for use in optimization have been specifically intended to model local geometry (see, for example, Jacobson and Oksman¹⁹, Davidon²⁰, Sorenson²¹, and Banerjee et al.²²). **In this work, we propose to use a new class of non-quadratic functions to model large-scale geometry.** These non-quadratic models will be used in an outer level for multi-

scale and/or global optimization to ‘funnel’ iterates to regions where the most promising global optimizers are expected to lie. The new ideas contained in this paper include

- 1) new exponential funnel functions for modeling global geometry,
- 2) a rigorous study of the mathematical properties of funnels,
- 3) the development of interpolating formulae for funnel approximations,
- 4) novel funneling algorithms for use within a global optimization framework.

We have searched the optimization and applied mathematics literature and have found no evidence that generalized exponential funnels have been proposed, studied or used before in any theoretical or numerical context.

Our overall multi-scale and/or global optimization framework consists of at least two, and perhaps more, alternating levels of optimization. The fine or small-scale level makes continued use of a quadratic model for local geometry and uses the terrain methodology recently proposed by Lucia et al.^{23,24} to gather local information (i.e., stationary points, singular and bifurcation points, pathways in valleys and on ridges, changes in curvature, etc.). The large-scale or global geometry is modeled by generalized exponential funnel approximations and related funneling algorithms are used to ‘funnel’ iterates to regions where the most promising global optimizers are expected to lie.

2. Generalized Exponential Funnels. To a large degree, the qualitative and quantitative nature of the global geometry shown in Figs. 1 through 3 can be captured by the function

$$F(z) = F_0 - \Gamma e^{-q(z)} \tag{1}$$

where $q(z) = \frac{1}{2}z^T A z + b^T z + c$, and where $\Gamma > 0$, F_0 and c are scalar parameters, b is an n -dimensional vector and A is an $n \times n$ symmetric matrix. **The class of functions defined by Eq. 1 is a completely new class of functions that we call generalized exponential funnels.** They are called this because they are defined on \mathbb{R}^n , are generated using exponential functionality, are shaped like a funnel for certain parameter values and, unlike real funnels, are not open at the bottom. The functional form of Eq. 1 closes and rounds the opening at the bottom, giving the funnel a well-defined minimum.

Note that there are $n(n+1)/2$ elements of the symmetric matrix A , n elements of the vector b , and one each for F_0 , Γ and c giving a total of $\frac{1}{2}(n^2 + 3n + 6)$ parameters in Eq. 1. These parameters F_0 , Γ , A , b and c ‘control’

everything about the shape of the resulting geometric object – the funnel depth (i.e., $F_0 - \Gamma$), the shape of the funnel and funnel mouth (largely through the eigenvalues of A), the location of the bottom of the funnel (from A and b), etc. Thus by carefully determining the $\frac{1}{2}(n^2 + 3n + 6)$ parameters in Eq. 1, which can be estimated from iterative function and derivative information, the shape of the funnel object can be controlled.

Exponential Funnel Geometry. Before discussing the mathematical properties of generalized exponential funnels, we show that this class of functions is capable of modeling a wide variety of geometric shapes by simply adjusting the parameters in Eq. 1. To see this, consider the geometric objects depicted in Fig. 4 for the feasible region given by $z_1, z_2 \in [-1 \times 10^4, 1 \times 10^4]$. Note that they are all quite different. However, they have all been generated using Eq. 1. Figure 4a in the top left corner shows the classical funnel cloud typically associated with an approaching

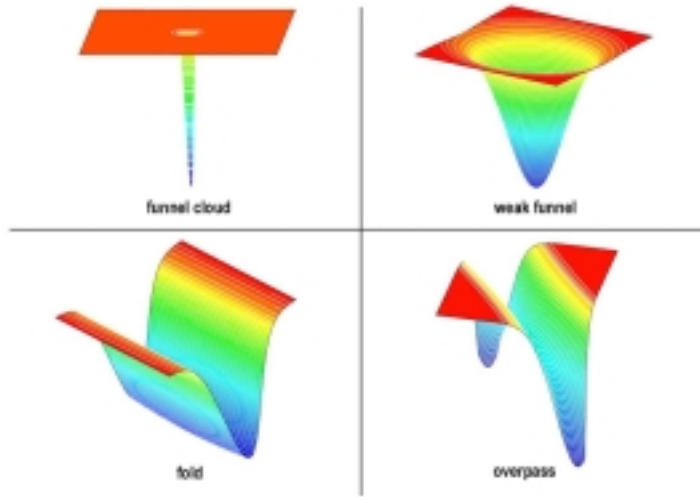


Figure 4: Various Funnel Geometries

tornado. This figure was generated using parameters $F_0 = 4$, $\Gamma = 1$, $A = /1.25 \times 10^{-5}, 0, 0, 8 \times 10^{-6} /$, $b = /-1.25 \times 10^{-5}, -1.6 \times 10^{-5} /$ and $c = 2.225 \times 10^{-5}$. The elements of the matrix A are in the order A_{11} , A_{12} , A_{21} and A_{22} and the components of b are b_1 and b_2 respectively. Figure 4b, on the other hand, shows a more gradual funnel shape similar to the one on the right hand side of Fig. 3.

This figure, which is just to the right of Fig.

4a, was generated using the set of parameters $F_0 = 4$, $\Gamma = 1$, $A = /1.25 \times 10^{-7}, 0, 0, 8 \times 10^{-8} /$, $b = /-1.25 \times 10^{-7}, -1.6 \times 10^{-7} /$ and $c = 2.225 \times 10^{-7}$. Figure 4c on the bottom left is an illustration of a fold similar to the ones shown in Figs. 1 and 3 where $F_0 = 4$, $\Gamma = 1$, $A = /1.25 \times 10^{-7}, 0, 0, 8 \times 10^{-10} /$, $b = /-1.25 \times 10^{-7}, -1.6 \times 10^{-9} /$ and $c = 6.41 \times 10^{-8}$. Finally there is Fig. 4d, which also folds but looks more like an overpass on a highway. Here the parameter values are $F_0 = 4$, $\Gamma = 1$, $A = /1.25 \times 10^{-7}, 2.22 \times 10^{-7}, 2.22 \times 10^{-7}, 8 \times 10^{-8} /$, $b = /-1.25 \times 10^{-7}, -1.6 \times 10^{-7} /$ and $c = 2.225 \times 10^{-7}$. Note that Figs. 4a-c have a unique global minimum. This is due to the fact that A is positive definite in each of these cases. Figure 4d, on the other hand, does not share this property because A is indefinite.

The point here is this. Each of the surfaces shown in Figs. 1 through 3 have large portions described by one or more of the funnel shapes shown in Fig. 4. Other applications show strong similarities in large-scale geometry. **Thus it should be possible to use generalized exponential funnels to capture the large-scale geometry of many surfaces including problems in protein folding, protein-protein and protein-DNA interactions and to guide the optimization calculations over large portions of the feasible region to a global optimum.**

Mathematical Properties of Exponential Funnels. The geometric shapes in Fig. 4 suggest that generalized exponential funnels defined by Eq. 1 have some rather interesting mathematical properties. This includes the facts that

- 1) $F(z)$ has a unique global minimum when A is positive definite,
- 2) $F(z)$ is non-convex,
- 3) there exists a family of maxima of $F(z)$ at infinity,
- 4) there is also a ring of inflection points.

Proof of 1). We show that $F(z)$ has a unique global minimum provided A is positive definite. To see this, note that the gradient of the funnel function defined by Eq. 1 is

$$G(z) = \Gamma e^{-q(z)} [Az + b] \quad (2)$$

and the Hessian or second derivative matrix of $F(z)$ is given by

$$H(z) = \Gamma e^{-q(z)} [A - (Az + b)(Az + b)^T] \quad (3)$$

From Eq. 2 it follows that $F(z)$ has an isolated stationary point at $z = -A^{-1}b$ provided A is nonsingular. Moreover, if $z = -A^{-1}b$, then $b = -Az$ and the Hessian matrix reduces to $H(z) = \Gamma \exp[\frac{1}{2}z^T Az - c]A$. Thus if A is positive definite, $H(z)$ is positive definite because $\Gamma > 0$ and the term $\exp[\frac{1}{2}z^T Az - c] > 0$. This means the isolated stationary point, $z = -A^{-1}b$, is a minimum of $F(z)$. Because \exp is a positive function, the only other stationary points of $F(z)$ occur when $q(z) = (\frac{1}{2}z^T Az + b^T z + c)$ goes to \pm infinity and this can only happen when z approaches infinity. However if

A is positive definite, then for large z , the exponential term in Eq. 1 must tend to zero and this implies that $F(z)$ must approach a maximum value of F_0 . As a result, $z = -A^{-1}b$ is a unique global minimum of $F(z)$.

Proof of 2). The preceding arguments show that $F(z)$ has a unique global minimum and a full rank Hessian matrix when A is positive definite. They also show that $F(z)$ also has at least one maximum. Thus as z gets large, $H(z)$ approaches $-\Gamma e^{-q(z)}[Azz^T A]$, which is a rank one, negative semi-definite matrix. Since $F(z)$ is a smooth function it is convex in some region around its minimum and concave in some region around any maximum. Therefore, it follows that $F(z)$ is a non-convex function. Note that Fig. 4 clearly shows the non-convex nature of various funnel shapes.

Proof of 3). In reality $F(z)$ has a singular set or family of maxima at infinity. This is because as any component of z , say z_j , approaches $\pm \infty$, the term, $e^{-q(z)}$, goes to zero.

Proof of 4). Let $F(z)$ be defined by Eq. 1 and let $G[z(\alpha)]$ for $\alpha \in [0, \infty]$ be the geodesic path from the unique global minimum to any maximum of $F(z)$. Since $F(z)$ is smooth, $G[z(\alpha)]$ is also smooth and must go through a smooth change in curvature. Thus there exists a point on this geodesic for which the second derivative is zero and this point corresponds to an inflection point of $F(z)$. Moreover since there is a family of maxima at infinity, there must be a family of geodesics each with their own inflection point of $F(z)$. Hence there is a ‘ring’ of inflection points of $F(z)$.

Note that Eqs. 1, 2 and 3 clearly show that there is a connection between local information (i.e., $F(z)$, $G(z)$ and $H(z)$) and more global information (i.e., specifically A and b). The importance of this connection will be made clearer in the next section.

Numerical Aspects of Exponential Funnels. Exponential funnels also have interesting numerical characteristics. One important numerical property of the funnels defined by Eq. 1 is that their unique global minimum can be calculated in a single iteration. However, to do this and to also use this class of functions to quickly ‘funnel’ iterates to parts of the feasible region where the most promising global optimizers lie, we must be able to build good global funnel approximations from function and derivative information gathered along the way.

Local Approximations. In local optimization, iterative quadratic models are generally built from a second-order Taylor series expansion of the objective function using function values, gradients and some type of second derivative information (i.e., quasi-Newton, analytical or finite difference derivatives). That is, an arbitrary objective function, say $f(x)$, is approximated about the point x_k using the expansion

$$f(x) = f(x_k) + g_k^T \Delta x + \frac{1}{2} \Delta x^T h_k \Delta x \quad (4)$$

where $g_k = g(x_k)$ is the gradient and h_k is the Hessian matrix (or some approximation of it) at x_k . From this, the change in the variables, $\Delta x = (x - x_k)$, is calculated by solving the linear system

$$h_k \Delta x = -g_k \quad (5)$$

For Newton's method, the quadratic model matches the function value, gradient and Hessian matrix at one point, x_k . For quasi-Newton methods, the quadratic model matches the function value at x_k and the gradient at two points, x_k and x_{k+1} . Also while function evaluations at points other than x_k are typically used in line searching or trust region strategies to produce a monotonic sequence of function values and force convergence, they are not used directly in building the local quadratic model. All of this is rather standard and long-established technology (see, p. 155 in Luenberger²⁵).

Approximating Global Geometry. One way of building iterative global funnel approximations is to match function, gradient and second derivative information of an arbitrary twice continuously differentiable objective function, say f , g and h , with F , G and H of the funnel at a variety of points. That is, we replace the values $F(z)$, $G(z)$ and $H(z)$ of the funnel on the left hand sides of Eqs. 1, 2 and 3 with values of $f(z)$, $g(z)$ and $h(z)$ for various values of z . Ideally one would like these points to be spread throughout the feasible region.

To see how this is done, note that by using Eq. 1, when $f(z)$ replaces $F(z)$ we have

$$\gamma(z) = F_0 - f(z) = \Gamma e^{-q(z)} \quad (6)$$

where $\gamma > 0$ is a positive scaling factor that depends on a single numerical measurement, $f(z)$, and the scalar parameter, F_0 . Moreover, from Eq. 2 it follows that

$$[Az + b] = g(z)/\gamma \quad (7)$$

and substitution of this expression in Eq. 3 gives

$$A = [\gamma h(z) + g g^T]/\gamma^2 \quad (8)$$

Equations 6, 7 and 8 reveal that it is possible to estimate A and b from values of $f(z)$, $g(z)$ and $h(z)$ at two or more iterates. We describe one way to do this for the simple case of two distinct iterates.

Let z_k be any value of the unknown variables with corresponding objective function, gradient and Hessian matrix values f_k , g_k and h_k respectively. Moreover, let z_{k+1} be some other arbitrary but not necessarily nearby or successive iterate with corresponding function, gradient and Hessian matrix values f_{k+1} , g_{k+1} and h_{k+1} . Now write Eq. 6 for z_k and z_{k+1} and subtract the latter from the former. This eliminates F_0 and gives

$$\gamma_{k+1} - \gamma_k = f_k - f_{k+1} \quad (9)$$

Repeating the same algebra using Eq. 8 yields

$$\gamma_k^2 [\gamma_{k+1} h_{k+1} + g_{k+1} g_{k+1}^T] - \gamma_{k+1}^2 [\gamma_k h_k + g_k g_k^T] = 0 \quad (10)$$

Equations 9 and 10 form a set of $[1 + n(n+1)/2]$ nonlinear equations in the two unknowns γ_k and γ_{k+1} when the symmetry of the associated matrices is taken into account. There is one nonlinear equation for each element of the matrices h_k , $g_k g_k^T$, h_{k+1} , and $g_{k+1} g_{k+1}^T$ in Eq. 10 but some of these are redundant due to symmetry. Actually only $n(n+1)/2$ independent equations from Eq. 10. This plus Eq. 9 gives a total of $[1 + n(n+1)/2]$ nonlinear equations. For $n = 1$, there are two equations and two unknowns while for $n > 1$ there are more equations than unknown

variables. Irrespective of this, two equations for which γ_k and $\gamma_{k+1} > 0$ can be determined using the Routh criterion²⁶. Using calculated values of γ_k and γ_{k+1} the matrix A can be calculated from Eq. 8 using gradient and Hessian information either at z_k or z_{k+1} . The parameter b can then be computed by rearranging Eq. 7 to give

$$b = g(z)/\gamma - Az \quad (11)$$

while F_0 can be calculated from $F_0 = f(z) + \gamma$. Like A , the values of b and F_0 can be determined using function and gradient values at either z_k or z_{k+1} .

With values of A and b , it is straightforward to calculate an estimate of the unique global minimum of the funnel approximation, say y , by simply solving

$$Ay = -b \quad (12)$$

Note that Eq. 12 requires the same information and the same amount of work (i.e., the solution of a linear system of equations) as required for locating the minimum of the conventional quadratic model.

There are several important aspects of the proposed iterative calculations for building a funnel approximation to global geometry. First, the funneling algorithm described in this section requires objective function, gradient and second derivative information at two or more points. However, there is an obvious trade off between the amount of information retained and robustness, computational work and storage. A minimum of function, gradient and Hessian matrix information at two distinct points is required to build a funnel approximation of an arbitrary objective function and results in two nonlinear equations in two unknown values of γ (see Eqs. 9 and 10). Retaining more points clearly provides a better large-scale funnel approximation but it increases the storage and makes the determination of funnel parameters more difficult because it increases the number of nonlinear equations and unknown values of γ . In general, retaining K distinct points results in K nonlinear equations and K unknown γ 's. When terrain methods are used in an inner loop, the required function, gradient and Hessian matrix information is readily available. Moreover, we have tested this algorithm on functions of the form defined by Eq. 1 and have observed that it converges to the unique global minimum in a single iteration, excluding the iterations required to solve Eqs. 9 and 10. Second, note that Eq. 8 shows that A is computed from a symmetric rank one

correction to the actual Hessian matrix. This suggests that it might also be possible to calculate iterative approximations to A using a symmetric rank one (SR1) quasi-Newton update (see, p. 193 in Luenberger²⁵) of the form

$$A_{k+1} = [\gamma_k A_k + g_k g_k^T] / \gamma_k^2 \quad (13)$$

Because $\gamma_k > 0$, this guarantees hereditary positive definiteness provided the initial approximation was positive definite. Finally, communication between scales is extremely important. This is because some derivative information is required to approximate large-scale geometry but analytical derivatives on rough objective function landscapes like the protein folding problem can often be misleading. Thus, deciding what information and how much of the information gathered at the small scale should be used to build a funnel approximation on any given outer iteration is crucial.

Funneling Calculations. Everything we have described in the previous section on approximating global geometry is part of what we call a funneling algorithm. In summary, funneling requires no user-supplied parameters and can be described by the following steps.

- 1) Let z_k and z_{k+1} and corresponding objective function, gradient and Hessian matrix values, $f(z_k)$, $g(z_k)$ and $h(z_k)$ and $f(z_{k+1})$, $g(z_{k+1})$ and $h(z_{k+1})$ be given.
- 2) Calculate the funnel parameters A and b as follows
 - a) Solve Eqs. 9 and 10 for the scaling factors, γ_k and γ_{k+1} .
 - b) Define $\gamma = \max(\gamma_k, \gamma_{k+1})$ and choose corresponding values z , $f(z)$, $g(z)$ and $h(z)$.
 - c) Calculate A using Eq. 8.
 - d) Compute b from Eq. 11.
- 3) Calculate the funnel minimum, y , by solving the linear system of equations defined by Eq. 12.
- 4) Compute the remaining funnel parameters F_0 , c and Γ from the equations

$$F_0 = f(z) + \gamma; \quad c = q(z) = \frac{1}{2}y^T A y + b^T y; \quad \Gamma = [F_0 - f(z)]e^{q(z)}$$

$$\text{where } q(z) = \frac{1}{2}z^T A z + b^T z + c.$$

3. Multi-Scale Optimization. We now show how to combine funneling with the terrain methodology of Lucia and co-workers^{23,24} to give a novel geometric optimization tool that interacts at local and global scales. The proposed multi-scale optimization methodology uses two-way communication where information gathered at the small scale is used to drive global optimization at the large scale and information at the large-scale is used to guide local optimization at the small scale. We make no theoretical guarantees that the resulting multi-scale optimization algorithm will find a global optimum.

Small Scale Information Using Terrain Methods. For ‘rough’ optimization landscapes many traditional local optimizers often suffer from convergence problems and large computational overhead. The presence of many stationary points in compact regions creates convergence problems and even if convergence can be controlled the performance of many good algorithms is painfully slow because they visit many local optima along the way. What is really needed at the small scale is a robust algorithm that will reliably take a number of different snapshots of the local geometry (i.e., local minima, saddle points, singular points, their respective objective function values, parts of valleys and ridges, eigen-information, etc.). This local information can then be accumulated and used intelligently to deduce the large-scale geometry of the objective function landscape. It is important to understand that some number of local optimization cycles may be required before a useful picture of global geometry emerges. However it is conjectured that, in the end, this will result in a considerable reduction in overall computational work when compared to the work required to visit the many, many local stationary points on a rough surface.

Small-scale information can be gathered using recently developed terrain methods (Lucia et al.^{23,24}). **The terrain methodology, which is based on the central idea that stationary points of any smooth objective function are connected along valleys and ridges, provides a wealth of information – minima, saddles, singular points, changes in curvature, pathways, eigen-information, etc.** Using a reliable local optimization method such as a trust region or Levenberg-Marquardt method, a first stationary point – any stationary point – is located. Successful movement from one stationary point to the next then proceeds by following appropriate valleys or ridges in the landscape either up or downhill using predictor-corrector calculations and equation solving respectively. Uphill movement is initiated in eigen-directions associated with the smallest positive eigenvalues of the Hessian matrix of the objective function and followed by predictor-corrector calculations to get ‘near’ a saddle point. Predictor steps are calculated by integrating the uphill Newton vector field, which tends to follow valleys reasonably

well but does drift some. Corrector steps, on the other hand, are used only intermittently to return iterates to the current valley. Once near a saddle point, quadratic acceleration (or a second-order Newton's method) is used to actually converge to the saddle (or next stationary point). Downhill movement is initiated in eigen-directions associated with the largest negative eigenvalues of the Hessian matrix of the objective function and followed by the use of a trust region method to force convergence to the next stationary point (either a saddle point or minimum). Irrespective of the stationary point found first (i.e., some saddle point, a local minimum, another type of singular point, or a global minimum), movement from one stationary point to the next follows this same logic until it is determined that the algorithm has finished. Numerical results for data regression, reactor models, flash and distillation calculations, potential energy minimization, phase split calculations and Gibbs free energy minimization, and other applications have clearly shown that terrain methods represent a reliable and efficient global optimization strategy for problems with a relatively small number of stationary points.

Figures 5 and 6 illustrate the ability of the terrain methodology to calculate stationary points and transition pathways for problems in molecular modeling. Figure 5 shows the connection between the collinear and equilateral conformations for a three-particle system modeled by the Lennard-Jones 6-12 potential. **Note that there is a clearly defined conformational path** between the global minimum (equilateral conformation) and the saddle point (collinear configuration) and there is no need to solve separate problems to find each conformation. The saddle point, which is a local constrained minimum, the global minimum, two singular points that also lie on the path and the pathway itself are easily computed by our terrain method. Figure 6, on the other hand, is a slightly more complicated reacting system. Note again that all minima and saddle points as well as the path connecting them are

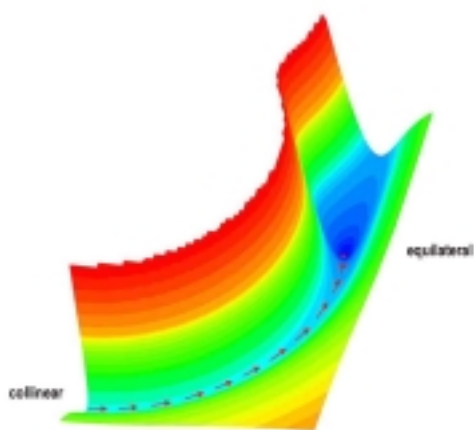


Figure 5: Conformational Pathway for a Simple Lennard-Jones Fluid

easily computed using the terrain methodology. **While the geometric illustrations in Figs. 5 and 6 are necessarily small in dimension, we have successfully solved problems with as many as 336 unknown variables using our terrain methodology and only a modest amount of computer resources.**

Because terrain methods are designed to move from one stationary point to another by following pathways in valleys and ridges, we believe they are ideally suited for optimization at the small scale. The use of trust regions and the integration of the Newton vector field provide a means of controlling the calculations and confining them to a portion of the feasible region. These

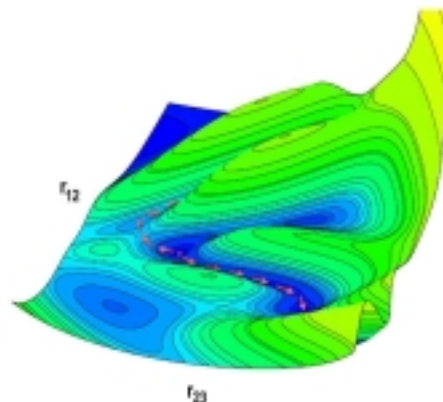


Figure 6: Reaction Pathway for a Simple Reacting System

small-scale optimization calculations using the terrain methodology provides a wealth of local information – a set of stationary points, corresponding objective function values, eigen-information, the parts of any valleys and/or ridges followed in finding these stationary points, and more. This local information can, in turn, be used to deduce large-scale geometry using simple averages, partial Hessian matrix information, difference formulae or the mean value theorem. For example, ‘average’ gradient and second-derivative information can be computed as a by-product of the path following aspects of each set of terrain calculations. Any ‘average’ gradient and Hessian matrix, say $\langle g \rangle$ and $\langle h \rangle$ respectively, is just a path integral along the relevant portion of the terrain path and given by

$$\langle g \rangle = (1/\alpha) \int g[z(\alpha)]d\alpha \quad \langle h \rangle = (1/\alpha) \int h[z(\alpha)]d\alpha \quad (14)$$

where α is some relevant length of the terrain path connecting any set of connected stationary points and where the lower and upper limits of either integration are 0 and α respectively.

It is these ‘average’ values, $\langle g \rangle$ and $\langle h \rangle$, together with a set of stationary points that are communicated to the funneling algorithm for large-scale optimization. Note that values of $\langle g \rangle$ and $\langle h \rangle$ can be accumulated for any number of stationary points. Thus it is straightforward to provide the objective function, ‘average’ gradient and ‘average’ Hessian information for some number of stationary points (or distinct iterates) as described in the section on Approximating Global Geometry.

In problems like protein folding, there is at least a qualitative way to decide how much small-scale optimization is required to provide adequate information about the global geometry. In particular, the small-scale calculations must perform enough local optimization to capture the inflection points (or changes in convexity) that straddle at least two low frequency minima on the energy landscape. See Fig. 2, the description of smoothing on p. 1574 in Tovchigrechko and Vakser² or the examples in section on proof of concept for reasons why this is important in these applications.

Large-Scale Optimization Using Funneling Algorithms. What is really needed at the large scale is an optimization methodology that will drive the calculations to a region where the global optimum lies so that the local optimization algorithm can accurately locate that global optimum. Funneling provides such a large-scale strategy and thus our proposed large-scale optimization calculations are based on the theory presented in the section on Approximating Global Geometry. In particular, given some set of stationary points and corresponding values of $\langle g \rangle$ and $\langle h \rangle$, values for the funnel parameters A and b can be determined by the procedure described earlier. From these funnel parameters, the global minimum of the funnel, say y , can be estimated by solving $Ay = -b$. This value y is then communicated to the small-scale and used as the starting point for the next set of local optimization calculations by the terrain methodology. However, **it also important to understand that the critical aspect of these large-scale calculations is not solving the linear system of equations for the funnel minimum but the correct use of the function, average gradient and average Hessian matrix information that has accumulated from small-scale local optimization calculations.** By this we mean retaining points that reflect more large-scale information (i.e. points that are not necessarily nearby, points that perhaps show changes in average curvature, points associated with low frequency local minima, etc.).

Termination. For each set of small-scale optimization calculations that has been conducted, there is a set of stationary points, say M . Thus if k small-scale optimization calculations have been conducted, there will be sets of stationary points M_1, M_2, \dots, M_k . Termination occurs when the predicted minimum of the funnel approximation on the j th funnel iteration, say y_j , is a member of one of the sets M_1, M_2, \dots, M_k .

4. Proof of Concept. In this section two examples are used to illustrate that funneling algorithms can be used to find global optima without finding all stationary points.

Example 1. The purpose of the first example is to give sufficient details associated with building funnel approximations. Therefore, consider the objective function defined by the nonlinear equation

$$f(z) = \sum_{i=1}^3 \{2000 - \frac{1}{2}\cos^2[\omega i(z-3000)/200] - e^{-q(z)}\} \quad (15)$$

where $\omega = 0.005$, $q(z) = [(z-3000)/1200]^2 - (z-3000)/4000$, and the feasible region is given by $0 \leq z \leq 6000$. Note that the function defined by Eq. 15 consists of typical force field-like functions from molecular modeling - a \cos^2 function for describing torsion angles and an exponential or Morse-like potential function for bond length forces with stretch-stretch cross terms. Moreover, a plot of Eq. 15 shows that it looks very much like the schematic in Fig. 2. Note that there are high and low frequency fluctuations on the objective function landscape. The high frequency fluctuations give the surface its roughness while the low frequency fluctuations result in higher barriers and stronger local minima.

Results of Using Only the Terrain Methodology to Solve Example 1. From $z_0 = 500$, it takes 1352 objective function and gradient evaluations and 0.11 s on a pc equipped with a Pentium III processor to find the 57 stationary points and 58 inflection points within the feasible region using the terrain method of Lucia et al.^{23,24}. Similar numerical results are observed for any arbitrary starting point in the given feasible region. The unique global minimum for this function is at $z^* = 3002.12$ where the objective function takes on a value of $f(z^*) = 5995.5$. The other 56 stationary points in the feasible region consist of 28 local minima and 28 local maxima.

Results Using Combined Terrain/Funnel Algorithms for Example 1. To test the idea that it is possible to find the global minimum without finding all stationary and inflection points using funneling, we also performed multi-scale optimization calculations using a terrain methodology in an inner loop and a funneling algorithm in an outer loop from the same starting point.

Results for the Initial Set of Small Scale Terrain Calculations for Example 1. The numerical results for the first set of terrain calculations are shown in Table 1, in the order in which they were computed. Notice how flat the objective function is despite the fact that there is an overall funnel-shape to the surface. Local optimization calculations were performed until two low frequency minima were located, resulting in the computation of 23 stationary and inflection points in 273 function and gradient evaluations and 0.03 s of computer time. The two low frequency minima are entries 1 and 22 in Table 1 while the inflection points that bracket these minima are shown in bold. During the course of this first set of local optimization calculations, ‘average’ gradient and ‘average’ Hessian matrix information was accumulated using the (terrain) path integrals in Eqs. 14. For example, $\langle g \rangle$ at the low frequency minimum $z = 486.939$ was calculated from

$$\langle g \rangle = (1/\alpha) \int g[z(\alpha)]d\alpha = [f(547.342) - f(426.091)]/[547.342 - 426.091] = -8.247 \times 10^{-5}$$

where the objective function values for the inflection points 426.091 and 547.342 are given in Table 1. ‘Average’ Hessian matrix information was calculated using ‘like’ inflection points straddling each of the low frequency minima. In particular, $\langle h \rangle$ at the low frequency minimum $z = 486.939$ was calculated from

$$\langle h \rangle = (1/\alpha) \int h[z(\alpha)]d\alpha = [g(1054.49) - g(426.091)]/[1054.49 - 426.091] = -5.2912 \times 10^{-7}$$

Note that to evaluate $\langle h \rangle$ local not ‘average’ gradient information is needed at the appropriate pair of inflection points. Also by ‘like’ inflection points we mean inflection points that show the same qualitative local behavior. For example, the inflection points 426.091 and 1054.49 both have a local gradient that is negative along the terrain path and each points to the low frequency minimum to its right. That is, the inflection point at 426.091 points to the low frequency minimum at $z = 486.939$ and the inflection point at 1054.49 points to the low frequency minimum at $z = 1116.35$. ‘Average’ gradient and Hessian matrix information for only the two low frequency minima is shown in Table 1.

Results for Initial Large Scale Funnel Calculations for Example 1. To build a model of large-scale geometry we let $z_k = 486.939$ and $z_{k+1} = 1116.35$. Using the corresponding values of $\langle g \rangle$ and $\langle h \rangle$ from Table 1, Eqs. 9 and 10 give values of $\gamma_k = 1.20713 \times 10^{-2}$ and $\gamma_{k+1} = 1.52071 \times 10^{-1}$. Since $\gamma_{k+1} > \gamma_k$, we choose to use γ_{k+1} and the associated information at z_{k+1} in Eqs. 8 and 11. This gives $A = 2.84209 \times 10^{-6}$ and $b = -8.21584 \times 10^{-3}$ respectively. Using these values for A and b in Eq. 12 provides an estimate of $y = 2890.77$ for the global minimum of the funnel. Note that A is positive (convex) even though $\langle h \rangle$ at both low frequency minima in Table 1 is negative (concave). Also note that y is a reasonably good estimate of the true global minimum of the rough objective function defined in Eq. 15, which is 3002.12. These funnel calculations take essentially no time for this small example.

Results for a Second Set of Small Scale Terrain Computations for Example 1. Using this calculated global minimum of the funnel as a starting point, a second set of local optimization calculations is performed using the terrain methodology. The results of these terrain calculations are shown in Table 2, again in the order in which they were computed. The twenty-three stationary and inflection points shown in Table 2 required 274 function and gradient calls and 0.03 s of computer time. The two low frequency minima are entry 4, which is the global minimum, and entry 21 while the relevant inflection points that bracket these minima are shown in bold. Note that this second set of local optimization calculations contains the global minimum of the objective function defined by Eq. 15.

Results for a Second Set of Large Scale Funneling Calculations for Example 1. Using the information in Tables 1 and 2, the large-scale geometry of the objective function is refined and a second iteration of the funneling algorithm is performed. Here we choose $z_k = 486.939$ from Table 1 and $z_{k+1} = 3002.12$ from Table 2 because they provide a mix of average curvature information. Using the corresponding values of $\langle g \rangle$ and $\langle h \rangle$, Eqs. 9 and 10 give values of $\gamma_k = 1.25725 \times 10^{-2}$ and $\gamma_{k+1} = 2.99257$. Again, using $\gamma_{k+1} > \gamma_k$ and the associated information at z_{k+1} in Eqs. 8 and 11 gives $A = 9.42156 \times 10^{-7}$ and $b = -3.07452 \times 10^{-3}$ respectively. Using these values of A and b in Eq. 12 provides a new estimate of $y = 3263.28$ for the global minimum of the funnel. Since this new estimate of the global minimum of the funnel is contained in Table 2, the multi-scale optimization algorithm terminates with correct global minimum, $z^* = 3002.12$. Thus the combined terrain/funneling algorithm finds only 46 of the 115 stationary

and inflection points, uses 547 instead of 1352 function and gradient evaluations and reduces the overall computer time by one half.

Figure 7 gives a geometric illustration of the terrain/funneling calculations for this example. The figure on the left shows the first set of terrain path calculations on the rough objective function landscape (red portion) and the first funnel approximation of the surface (green curve). Note that the location of the minimum of the funnel approximation on this first iteration is quite good but the depth of the funnel is a bit too deep. In the figure on the right, the second funnel approximation provides both a good estimate of the minimum and depth of the funnel for the rough objective function landscape.

While this illustrative example is admittedly simple and the reduction in work is not dramatic, it clearly goes to show that when information is gathered and used correctly, the proposed multi-scale optimization algorithm is capable of finding a global optimum without calculating all stationary points. However, if we increase the degree of frustration or roughness on this surface, the overall reduction in work in favor of the combined terrain/funneling algorithm becomes more pronounced. In particular, by increasing ω in the \cos^2 function, say from 0.005 to 0.02, the number of stationary and inflection points increases to 459 and the location of the global minimum shifts to $z = 3157.10$. If all stationary and inflection points are located, then 19,478 function and gradient evaluations and 1.59 s

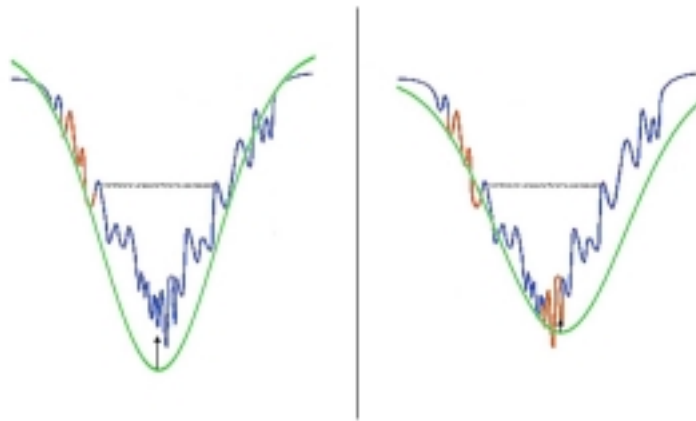


Figure 7: Geometric Illustration of Terrain / Funneling Computations

on a Pentium III computer are needed while the multi-scale terrain/funneling methodology needs again just 547 function and gradient evaluations and 0.06 s. Here the improvements that result from using the combined terrain/funneling algorithm are more dramatic because of the increase in the number of stationary points or roughness.

The reader might also be tempted to argue that the good numerical results are consequences of the use of the exponential term in Eq. 15 and its striking similarity to our proposed exponential funnel model of global geometry. However, this is not true. We have also observed similar reliable and efficient behavior of the combined

use of terrain methods and funneling on other small dimensional problems in molecular conformation. These examples use many of the common empirical models for molecular interactions - harmonic bond forces, exponential bond forces, three-body interactions, and the Lennard-Jones 6-12 potential to model long-range forces.

Example 2. This second example is used to illustrate funneling concepts on a two-dimensional ‘rough’ energy surface with considerable frustration. For this we chose the Muller-Brown²⁷ potential with the modified coefficients used in Lucia et al.²⁴ and added frustration to it. The Muller-Brown function is given by

$$E_{MB} = \sum_{i=1}^4 D_i \exp \{ A_i (r_{23} - x_{1i})^2 + B_i (r_{23} - x_{1i})(r_{12} - x_{2i}) + C_i (r_{12} - x_{2i})^2 \} \quad (16)$$

where the unknown variables, r_{12} and r_{23} , are inter-particle distances, $x_{1i} = /3, 2, 1.5, 1/$, $x_{2i} = /1, 1.5, 2.5, 2/$ and $A_i = /-1, -1, -6.5, -0.7/$, $B_i = /0, 0, 11, 0.6/$, $C_i = /-10, -10, -6.5, 0.7/$, $D_i = /-200, -100, -170, 15/$ for $i = 1, \dots, 4$. The feasible is defined by $r_{12}, r_{23} \in [0, 3.5]$. This ‘smooth’ potential energy function has three minima and two saddle points as shown in Table 3.

To create small-scale roughness but retain the large-scale multiple funnel geometry of the Muller-Brown surface, we added torsion terms given by

$$R_{MB} = \sum_{i=1}^4 \Omega \{ \cos([i\omega_1 r_{12} - \omega_2 r_{23}]^3) + \cos([i\omega_1 r_{23} - \omega_2 r_{12}]^3) \} \quad (17)$$

where $\Omega = 20.$, $\omega_1 = 0.6$ and $\omega_2 = 0.15$. The resulting ‘rough’ potential energy function is

$$E = E_{MB} + R_{MB} \quad (18)$$

Figure 8 shows the ‘rough’ Muller-Brown surface while Fig. 9 gives an enlargement of the deepest energy well. Note that the roughness function, R_{MB} , creates havoc on the surface and that just one well of the ‘rough’ potential energy function has an enormous number of stationary points! We have estimated that there are on the order of 10^5

stationary and singular points on this surface. The addition of roughness shifts the global minimum of the smooth Muller-Brown function a bit and also introduces several strong local minima in the same well. These strong local minima can be viewed as trapping local minima similar to the non-native traps shown in Fig. 2. Some of the minima, saddle points and singular points on the ‘rough’ Muller-Brown function are shown in Table 4.

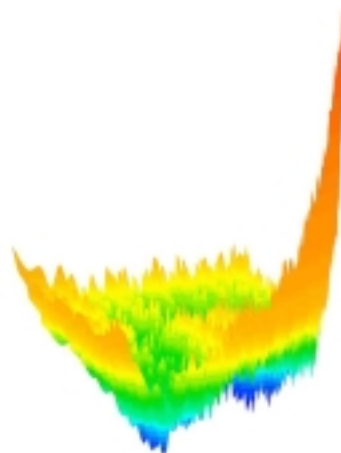


Figure 8: ‘Rough’ Muller-Brown Potential Energy Surface

One interesting numerical experiment that is very much like a protein folding computation is to start with an extended conformation near the rim of a given ‘rough’ energy well and attempt to determine the global minimum of that well. Therefore consider the ‘rough’ energy well containing the global minimum shown in Table 3.

Results for the Initial Set of Small Scale Terrain Calculations for Example 2. For the starting point of $(r_{23}, r_{12}) = (0.8, 2.)$ an initial set of terrain calculations results in the computation of 21 stationary and singular points requiring 495 function and gradient evaluations and 0.05 s on a Pentium III computer. Some of these 21 stationary and singular points are shown at the top of Table 4. We choose $z_k = (1.01156, 2.01765)$ and $z_{k+1} = (1.17277, 2.10699)$. Average gradient and average Hessian matrix information is then computed using path integrals along the appropriate terrain path.

Results for the Initial Large Scale Funneling Calculations for Example 2. From this average information, the funnel parameters A and b are determined using the proposed interpolating formulae. Equations 9 and 10 give values of $\gamma_k = 101.930$ and $\gamma_{k+1} = 121.711$ which, in turn, give funnel parameters of $A = /13.8382, -11.2797, -11.2797, 12.1535/$ and $b = /7.63177, -13.8052/$. Using these values of A and b , the minimum of the funnel approximation to this energy well on the ‘rough’ Muller-Brown surface is $y = -A^{-1}b = /1.53753, 2.56288/$.

Results for the Second Set of Small Scale Terrain Calculations for Example 2. Using this value of the estimated funnel minimum, y , as a starting point, a second set of terrain calculations is conducted. Here 19 stationary and singular points were computed requiring 390 function and gradient evaluations and 0.11 s. Some of these stationary points are shown at the bottom of Table 4.

Results for the Second Set of Large Scale Funneling Calculations for Example 2. This time we choose $z_k = (1.01156, 2.01765)$ and $z_{k+1} = (1.74143, 2.72743)$ and use ‘average’ gradient and Hessian matrix information along the corresponding terrain path. This gives $\gamma_k = 27.1485$ and $\gamma_{k+1} = 41.4833$ from Eqs. 9 and 10 and the funnel parameters $A = /65.2245, -27.7272, -27.7272, 58.1797/$ and $b = /-34.0564, -107.737/$. The corresponding funnel minimum is $y = /1.64201, 2.63434/$. Since the estimated funnel minimum is in the convex hull defined by the set of stationary and singular points in the bottom half of Table 4 the calculations are terminated with the correct global minimum of this energy well.

The total amount of work needed to find this global minimum using the combined terrain/funneling algorithm is 885 function and gradient evaluations and 0.16 s. When this is compared with the work that would be required to find the estimated 10^5 stationary and singular points in this energy well, it is clear that the proposed multi-scale global optimization algorithm results in significant computational savings.

5. Conclusions. A new class of functions called generalized exponential funnels was proposed for modeling large-scale geometry. The mathematical, algorithmic and numerical consequences of exponential funnels were discussed. In particular, it was shown that exponential funnels are capable of modeling the large-scale folds, cups, funnels and cones that recur on many objective function surfaces for physical problems. New interpolating formulae were developed to compute funnel parameters from a small amount of local gradient and Hessian matrix information. These interpolating formulae also show that there is a low rank connection between this local derivative information and the shape and location of the minimum of the funnel. With properly determined parameters, exponential funnels provide non-convex approximations to large-scale geometry that have unique minima. An outline of a funneling algorithm was presented.

A multi-scale global optimization methodology that makes combined use of terrain methods and the proposed funneling algorithm was presented. In particular, terrain methods were suggested for reliably accumulating small-scale information (i.e., sets of stationary points, changes in curvature, pathway information, etc.). A funneling algorithm, on the other hand, was

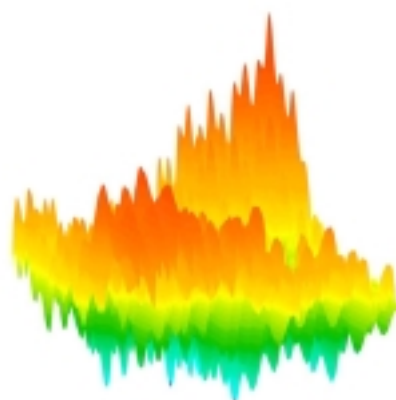


Figure 9: Enlarged 'Rough' Muller-Brown Surface

recommended to guide the overall optimization calculations and make 'large' moves by 'funneling' iterates to parts of the feasible region where the most promising global optimizers are expected to lie. A method for calculating 'average' large-scale derivative information from small-scale information based on integrals along terrain paths was suggested and communication between scales was also discussed.

Two examples with 'rough' objective function surfaces and many stationary points were presented as proof of concept. Numerical results for these examples show that

- 1) exponential funnels can provide adequate models of large-scale geometry,
- 2) the proposed multi-scale global optimization methodology is capable of finding the global optimum without finding all stationary points,
- 3) often times the reduction in overall computational overhead can be significant, and
- 4) the combined terrain/funneling methodology has considerable promise as a global optimization tool for problems in molecular conformation.

This is particularly important for optimization problems like protein-folding and protein-protein interactions that are characterized by astronomically large numbers of local optima and rough energy landscapes where we anticipate orders of magnitude savings in computational overhead.

Acknowledgments. The authors would like to thank the National Science Foundation for support of this work under Grant No. CTS-0113091.

Literature Cited

- (1) Onuchic, J.N.; Luthey-Schulten, Z.; Wolynes, P.G. Theory of Protein Folding: The Energy Landscape Perspective. *Annu. Rev. Phys. Chem.* **1997**, 48, 545.
- (2) Tovchigrechko, A.; Vakser, I. A. How Common is the Funnel-like Energy Landscape in Protein-Protein Interactions? *Protein Sci.* **2001**, 10, 1572.
- (3) Heidemann, R.A.; Madsen, J.; Stenby, E.H.; Anderson, S.I. Wax Precipitation Modeled with Many Mixed Solid Phases. AIChE Spring National Meeting, New Orleans, April 2003; Paper 137c.
- (4) Levinthal, C. in *Mossbauer Spectroscopy in Biological Systems*; Debrunner, P.; Tsibris, J.C.M.; Munck, E., Eds.; 1969.
- (5) Levy, A.V.; Montalvo, A. The Tunneling Method for the Global Minimization of Functions. *SIAM J. Sci. Stat. Comp.* **1985**, 6, 15.
- (6) Bahren, J.; Protopopescu, V. Generalized TRUST Algorithms for Global Optimization. In *State of the Art in Global Optimization*; Floudas, C.A.; Pardalos, P.; Eds.; Kluwer: Dordrecht, Netherlands, 1996.
- (7) Maranas, C.D.; Floudas, C.A. Finding All Solutions to Nonlinearly Constrained Systems of Equations. *J. Global Optim.* **1995**, 7, 143.
- (8) Piela, L.; Kostrowicki, J.; Scheraga, H.A. The Multiple Minima Problem in Conformational Analysis of Molecules. Deformation of the Potential Energy Surface by the Diffusion Equation Method. *J. Phys. Chem.* **1989**, 93, 3339.
- (9) Hanson, E.R. Global Optimization Using Interval Methods – The Multidimensional Case. *Numer. Math.* **1980**, 34, 247.
- (10) Schnepper, C.A.; Stadtherr, M.A.; Robust Process Simulation Using Interval Methods. *Comput. Chem. Engng.* **1995**, 20, 187.
- (11) Aluffi-Pentini, F.; Parisi, V.; Zirilli, F. Global Optimization and Stochastic Differential Equations. *J. Opt. Theory and Appl.* **1985**, 47,1.
- (12) Bilbro, G.L. Fast Stochastic Global Optimization. *IEEE Trans. Sys. Man. Cyber.* **1994**, 4, 684.
- (13) Sevick, E.M. ; Bell, A.T.; Theodorou, D.N. A Chain of States Method for Investigating Infrequent Event Processes Occurring in Multistate, Multidimensional Systems. *J. Chem. Phys.* **1993**, 98, 3196.

- (14) Scheraga, H.A. Prediction of Protein Conformation. In *Current Topics in Biochemistry*; Anfinsen, C.A.B.; Schechter, A.N., Eds.; Acad. Press: New York 1974.
- (15) Jones, D.T.; Taylor, W.R.; Thornton, J.M. A New Approach to Protein Fold Recognition. *Nature* **1992**, 358, 86.
- (16) Henkelman, G.; Johansson, G.; Jonsson, H. Methods for Finding Saddle Points and Minimum Energy Paths. In *Progress in Theoretical Chemistry and Physics*, Vol. 5; Schwartz, S.D., Ed.; Kluwer: Dordrecht, Netherlands, 2000.
- (17) Bolhuis, P.G.; Chandler, D.; Dellago, C.; Geissler, P.L. Transition Path Sampling: Throwing Ropes Over Rough Mountain Passes, in the Dark. *Annu. Rev. Phys. Chem.*; **2002**, 53, 291.
- (18) Westerberg, K.M.; Floudas, C.A. Locating All Transition States and Studying the Reaction Pathways of Potential Energy Surfaces. *J. Chem. Phys.* **1999**, 110, 9259.
- (19) Jacobson, D.H.; Oksman, W. An Algorithm that Minimizes Homogeneous Functions of N Variables in $N + 2$ Iterations and Rapidly Minimizes General Functions. *J. Math Anal. Appl.* **1972**, 38 535-552.
- (20) Davidon, W.C. Conic Approximations and Collinear Scalings for Optimizers. *SIAM J. Numer. Anal.* **1980**, 17, 268.
- (21) Sorensen, D.C. The Q-superlinear Convergence of a Collinear Scaling Algorithm for Unconstrained Optimization. *SIAM J. Numer. Anal.* **1980**, 17, 84.
- (22) Banerjee, A.; Adams, N.; Simons, J.; Shepard, R. Search for Stationary Points on Surfaces. *J. Phys. Chem.* **1985**, 89, 52.
- (23) Lucia, A.; Yang, F. Global Terrain Methods. *Comput. & Chem. Engng.* **2002**, 26, 529.
- (24) Lucia, A.; DiMaggio, P.; Depa, P. A Geometric Terrain Methodology for Global Optimization. *J. Global Optim.* in press.
- (25) Luenberger, D.G. *Introduction to Linear and Nonlinear Programming*; Addison-Wesley: Reading, MA, 1973.
- (26) Routh, E.J. *Dynamics of a System of Rigid Bodies, Part II*; MacMillan: London, 1905.
- (27) Muller, K.; Brown, L.D. Location of Saddle Points and Minimum Energy Paths by a Constrained Simplex Optimization Procedure. *Theoret. Chim. Acta.* **1979**, 53, 75.

Table 1: First Set of Small-Scale Terrain Optimization Results for Illustrative Example 1

stationary pt.	z	f	<g>	<h>
min	486.939	5998.48	-8.247×10^{-5}	-5.2912×10^{-7}
inflection	426.091	5999.00		
inflection	547.342	5998.99		
max	357.237	5999.57		
max	615.558	5999.55		
inflection	313.802	5999.40		
inflection	659.035	5999.38		
min	264.543	5999.23		
min	710.026	5999.19		
inflection	221.217	5999.35		
inflection	753.282	5999.31		
max	172.407	5999.49		
max	799.563	5999.44		
inflection	129.080	5999.38		
inflection	851.965	5999.28		
min	80.894	5999.23		
min	894.203	5999.15		
inflection	28.428	5999.43		
inflection	942.594	5999.32		
max	984.393	5999.47		
inflection	1054.49	5998.88		
min	1116.35	5998.34	-4.1312×10^{-4}	-6.9009×10^{-7}
inflection	1175.52	5998.83		

Table 2: Second Set of Small-Scale Terrain Optimization Results for Illustrative Example 1

stationary pt.	z	f	<g>	<h>
max	2865.38	5996.71		
inflection	2938.89	5996.08		
inflection	2831.26	5996.62		
min	3002.12	5995.50	-7.3632×10^{-4}	2.6383×10^{-6}
min	2786.57	5996.49		
inflection	3061.09	5995.99		
inflection	2733.66	5996.69		
max	3128.40	5996.52		
max	2673.45	5996.92		
inflection	3171.85	5996.35		
inflection	2639.78	5996.86		
min	3221.33	5996.17		
min	2606.42	5996.81		
inflection	3264.79	5996.30		
inflection	2544.18	5997.10		
max	3318.04	5996.47		
max	2492.73	5997.36		
inflection	3361.08	5996.37		
inflection	2432.43	5996.94		
min	3400.39	5996.27		
min	2377.96	5996.54	-2.1413×10^{-3}	1.96359×10^{-6}
inflection	3456.67	5996.51		
inflection	2311.01	5997.20		

Table 3: Minima and Saddle Points for the 'Smooth' Muller-Brown Function

stationary point	r_{23}	r_{12}	E_{MB}
local min	2.62350	1.02804	-108.167
saddle	2.21249	1.29299	-72.2489
local min	1.94999	1.46669	-80.7678
saddle	1.17800	1.62431	-40.6648
global min	1.44178	2.44173	-146.700

Table 4: Some Stationary and Singular Points on the ‘Rough’ Muller-Brown Surface

stationary point	r_{23}	r_{12}	$E = E_{MB} + R_{MB}$
singular	0.82856	1.99138	0.15084
singular	0.95012	1.99859	-47.0750
saddle	1.00924	2.00166	-79.9580
local min	1.01156	2.01765	-100.409
local min	1.01804	2.09670	-118.373
singular	1.06339	2.09985	-97.3774
saddle	1.10511	2.10249	-77.6757
singular	1.14076	2.10474	-92.6712
local min	1.17277	2.10699	-106.563
saddle	1.47643	2.54713	-145.639
singular	1.49746	2.54831	-164.899
local min	1.51881	2.54981	-184.305
saddle	1.48020	2.61975	-122.033
global min	1.59652	2.57736	-231.622
saddle	1.55603	2.62469	-143.723
local min	1.59990	2.62757	-202.799
singular	1.58529	2.71786	-153.941
singular	1.65781	2.72235	-144.495
saddle	1.70923	2.72551	-88.8338
local min	1.74143	2.72743	-132.898