D_3D: A SOFTWARE FOR GRAPHICAL REPRESENTATION OF OBJECTIVE FUNCTIONS

P.A. SIMIONESCU
Dept. of Mechanical Engineering
and Solidification Design Center
Auburn University, 202 Ross Hall,
Auburn AL, 36849

D. BEALE
Dept. of Mechanical Engineering
Auburn University, 202 Ross Hall,
Auburn AL, 36849

ABSTRACT
The main features of a software named ***D_3D*** specially devised for visualizing objective functions as contour plots and 3D surfaces are presented. These include detailing the minimum areas by logarithmically spacing the level curves, the possibility of truncating the upper parts of the surface (particularly useful in representing penalized objective functions) and representing the gradient of the function as a vector field projected on the bottom plane in combination with 3D surface plots.

INTRODUCTION

Objective functions arising in optimisation problems are single valued functions of one or more variables. Functions of one, two, even three variables allow for graphical representation in the 2 or 3 dimensional Cartesian space aaa1] (Encarnacao et al 1990). Such representations permit attaining a good feeling of the monotonicity, convexity, and the existence of multiple minima of the function, and are therefore usual encounter in most numerical optimisation and operations research publications.

The purpose of this paper is to present a computer software named ***D_3D*** specially devised for representing objective functions of two variables, that has several features not yet at hand in commercially available mathematical and visualisation software. These features will be introduced based on a simple optimisation problem, of finding the equilibrium position of a spring-restrain double pendulum.

EXAMPLE PROBLEM

The dynamic analysis of constrained multibody systems includes, among time response and kinetostatic analysis, the problem of determining the static equilibrium configuration. Several methods for solving the equilibrium problem are known (Simionescu and Fawcett 1997) of which searching for the minimum of total potential energy is the most simple and easy to apply. On the other hand, determining the equilibrium position of spring and weight systems are good illustrative examples of optimisation problems that can arise in practice (Vanderplaats 1984).

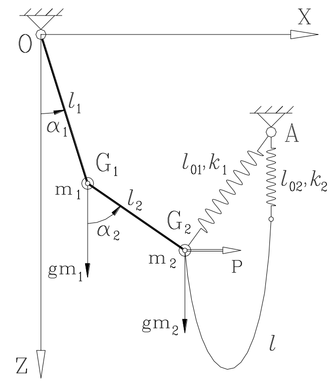

Fig. 1 A double pendulum restrained with elastic springs (g=9.81m/s^2 is the acceleration due to gravity)

The mass spring system considered for determining the equilibrium position is that of the double pendulum in Fig. 1 amplified with a linear spring (k_1, l_{01}) having one end attached to the outer lumped mass and the other end fixed to the frame at point A(x_A,y_A). Mounted in between the same points G_2 and A, there are an ideal inextensible massless thread of length l, connected in series with a second spring (k_2, l_{02}).

For the system under consideration, the total potential energy is the sum of the gravity potentials of masses m_1 and m_2, that due to the constant force P (which can also be considered as deriving from a potential) and that of the elastic energy of the two springs. Choosing the reference position in deriving the potentials of m_1, m_2 and P is arbitrary. In this example the extreme position of point G_2 along the positive axis Ox for the potential of P, and G_1 and G_2 along Oz for the gravity potentials, have been used as reference. Therefore the total potential energy of the system is:

$$U(\alpha_1,\alpha_2) = \left(l_1 - z_{G1}\right)\cdot m_1 g + \left(l_1 + l_2 - z_{G2}\right)\cdot m_1 g + \\ + \left(l_1 + l_2 - x_{G2}\right)\cdot P + \frac{1}{2}\cdot k_1 \cdot (AG_2 - l_{01})^2 + Q(\alpha_1,\alpha_2) \tag{1}$$

where

$$\mathrm{Q}(\alpha_1,\alpha_2) = \begin{cases} 0 & \text{if } \mathrm{AG}_2 \le l + l_{02} \\ \frac{1}{2} k_2 (AG_2 - l - l_{02})^2 & \text{if } \mathrm{AG}_2 > l + l_{02} \end{cases} \tag{2}$$

and

$$\mathrm{A}G_2 = \sqrt{(x_{G2} - x_A)^2 + (z_{G2} - z_A)^2} \tag{3}$$

$$z_{G1} = l_1 \cdot \cos\alpha_1 \qquad z_{G1} = l_1 \cdot \cos\alpha_1 \\ x_{G2} = l_1 \cdot \sin\alpha_1 + l_2 \cdot \sin\alpha_2 \qquad z_{G2} = l_1 \cdot \cos\alpha_1 + l_2 \cdot \cos\alpha_2 \tag{4}$$

It is to observe that the term Q(α1,α2) acts as a barrier penalty function (Press et al 1992), with the penalty factor $k_{2,}$ corresponding to the case when the point G_2 is constrained to remain inside a circle of centre A and radius l+l_{02}.

NUMERICAL RESULTS

The parameters of the system in Fig. 1 were considered to be: l_1=0.45m, l_2=0.35m, m_1=0.75kg, m_2=0.5kg, x_A=0.25m, z_A=-0.25m, k_1=9.5N/m, l_{01}=0.2m, $k_2=2\cdot10^4$N/m, l_{02}=0.1m and P=1.8N. Initially the thread length *l* was chosen sufficiently long so that the spring k_2 never becomes effective i.e. the penalty term Q equals zero irrespective of the value of α_1 and α_2 [identical results are obtained considering k_2=0 in equation (2)]. For this particular case, the potential function $U(\alpha_1, \alpha_2)$ has the shape in Fig. 2. Two points of extrema can be identified, of which the maximum point corresponds to an unstable equilibrium position.

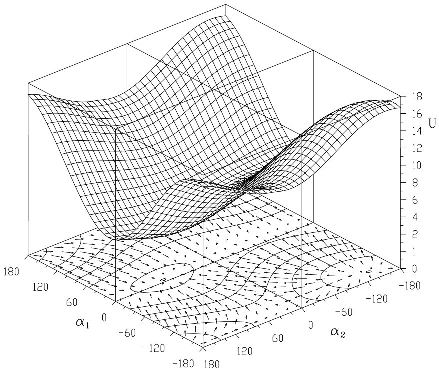

Fig. 2 Graphical representation of the potential function $U(\alpha_1,\alpha_2)$ for k_2=0. Also shown as arrows projected on the bottom plane is the gradient of the function.

The static equilibrium position was determined by minimising the function (1) using Fletcher-Reeves algorithm (Press et al 1992), resulting in $\alpha1=6.0762°$, $\alpha2=88.0589°$ for which U=3.72088 Nm.

For *l*=0.65m the shape of the potential function $U(\alpha_1, \alpha_2)$ changes to that in Fig. 3, showing that the point of maximum have moved to a different location, however without modifying the location and value of the global minimum.

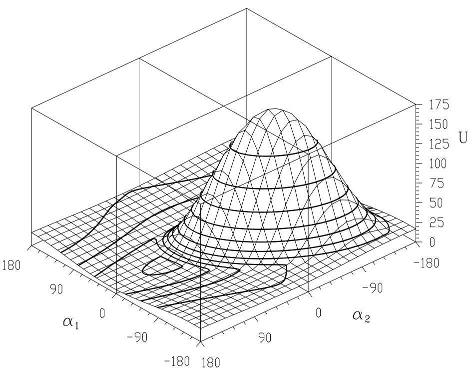

Fig. 3 A whole range representation of the potential function for *l*=0.65m and $k_2=2\cdot10^4$N/m.

As one can notice, a whole z-axis range representation of the function U in this case is less suggestive. The solution chosen for detailing the minimum area of the function as shown in Fig. 3, was to map logarithmically spaced level curves on the function surface. This way the level curves are concentrated in the lower region of the function surface and the global minimum highlighted.

The relation used for calculating the height of the horizontal cutting plane that generates the j-th level curve was:

$$z_j = z_{\min} + \exp\left[(j-1)\cdot\ln\frac{z_{\max}-z_{\min}+1}{n-1}\right]-1 \qquad (5)$$

where n is the total number of level curves, and z_{min} and z_{max} are the lower and upper range of the z-axis, which can coincide or not with the minimum and

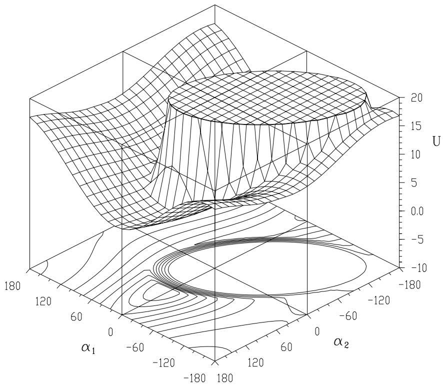

Fig. 4 Truncated representation of the potential function for l=0.65m and k_2=2·10^{4}N/m.

maximum function values from among the grid points used for sampling the function surface.

The idea of generating unequally spaced level curve diagrams is not a new concept, but as proposed, there is the certitude that the minimum region is well detailed, without the need of interactively editing the level of some equally spaced level curves, as it is the case of many graphs in Reklaitis et al (1983).

A second method that can be used for detailing the minimum region of the function-surface, is to trim at a certain height the upper region of the graph. The ***D_3D*** program has the feature of permitting upon request an accurate truncating of the upper (or lower) regions of the function surface (Fig. 4). Commercially available software like MS EXCEL™ and MATHEMATICA™ allow truncating the function surface, but the method used (of replacing the z values outside the [z_{min}..z_{max}] domain with exactly z_{min} or z_{max}) is not very accurate (Fig. 5), mainly if the number of points used in sampling the function is small. A different solution in service of MATLAB™ through the *Axis* function is to blind the upper and lower parts of the graph, which is obviously even less satisfactory.

In order to generate an equivalent of the infinite-barrier penalty function (Reklaitis et al 1983), for the same double-pendulum, l=1.0m and k_2=1020N/m have been considered. The shape of the potential function U(α_1,α_2) looks as shown in Fig. 6, case in which the lower region of the function surface appears completely flatten, hence any details of the minimum region are hidden. Thus a truncated representation of the function surface proves even more useful that in the previous case.

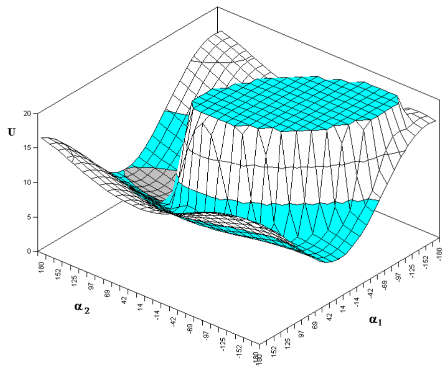

Fig. 5 The same representation as in Fig. 4 produced with MS-EXCEL™. A similar effect has the *Plotrange* option in MATHEMATICA™ software.

By reducing the upper limit of the z-axis domain from 1.5·10^{17} to 20, the diagram in Fig. 7 has been obtained. To increase the accuracy of representing the intersection between the function surface and vertical cylinder induced by the penalty term, the number of points used for sampling the function surface has been increased. However, in order not to darken completely the surface, only the lines of constant a1 have been generated. This feature is also useful when representing noisy data or highly multimodal functions such as Ackley's test objective function (Gen and Cheng 1997):

$$F(x_1 \ldots x_n) = 20\left[1-\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)\right]-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+e \qquad (6)$$

shown in Fig. 8 for n=2 as lines of constant x_1 combined with contour lines projected on the bottom plane. shown in Fig. 8 for n=2 as lines of constant x_1 combined with contour lines projected on the bottom plane.

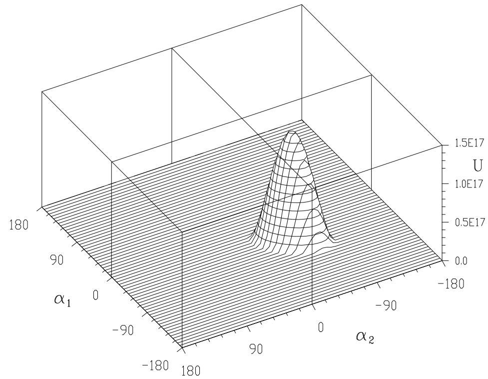

Fig. 6 A whole range representation of $U(\alpha_1,\alpha_2)$ for l=1.0m and $k_2=10^{20}$N/m.

CONCLUSIONS

The case of the total potential function of a spring restrained double pendulum has been considered to illustrate the features of the ***D_3D*** function representation software, particularly design to facilitate representing graphically objective functions of two variables. These features are summarised briefly in the following:

-plotting single-valued functions of two real variables z=f(u,v) as polylines of constant u, polylines of constant v, crossed-hatch representations or raised level-curves (constant z profiles) with or without the hidden lines removed;

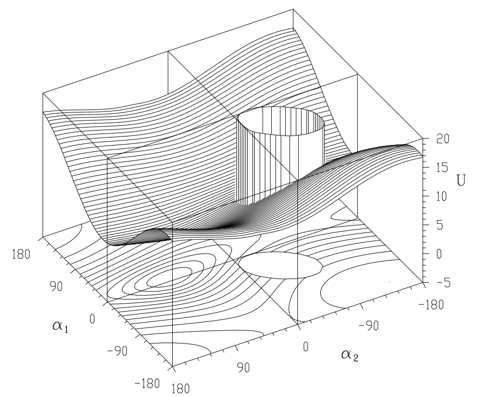

Fig. 7 Truncated representation of the potential function surface $U(\alpha_1,\alpha_2)$ for l=1.0m and $k_2=10^{20}$N/m.

-accurate solving of the intersection problem between horizontal plane(s) and the function surface in truncated plots over z-axes, both in crosshatch and in constant x, y and z profile representations;

-generating contour maps of the same types of functions;

-plotting the gradient of the function as a vector field projected on the bottom plane using the same grid of heights for numerically calculation of its components [resent implementations of this feature in mathematical and visualization software allow the representation of the gradient only in top view projections, and require the user to provide

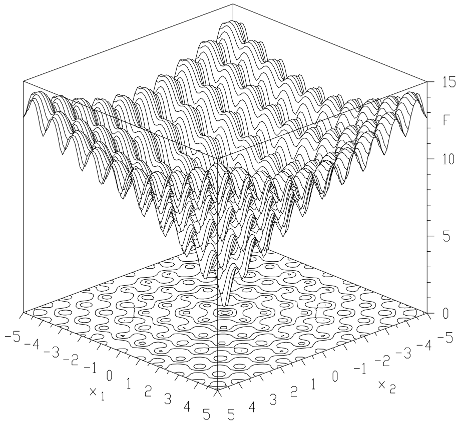

Fig. 8 Ackley's function represented as lines of constant x_2 and 15 contour lines projected on the bottom plane, equally spaced over the [0..15] interval.

separately the components of the gradient over the x and y axes – see Quiver function in MATLAB 6 or Champ function in SCILAB 2.5 (http://www-rocq.inria.fr/scilab/)]
-generation of logarithmically spaced level curve diagrams (projected, or mapped on the function surface) for detailing the global minimum area in objective functions;
-the height of the level curves generated automatically (equally spaced or logarithmically spaced), can be further modified interactively and saved to a file from where they can be read later on;
-the orientation of the z-axis can be reversed on request, an alternative solution for viewing function surfaces from below;
-all the graphic images can be exported as PCX or DXF files, and further used in spreadsheets and reports.
More representations produced with D_3D software can be found on the Internet at: http://www.auburn.edu/~simiope/fxy.htm

REFERENCES

Encarnacao, J.L., Lindner, R. and Schlechtendahl, E.G., 1990, *Computer Aided Design. Fundamentals and System Architecture*, Springer-Verlag, pp. 335-342.

Gen, M., and Cheng, R., 1997, *Genetic Algorithms and Engineering Design*, Willey, New York, pp. 44.

Kota, S. and Chiou, S.J., 1993, "Use of Orthogonal Arrays in Mechanism Synthesis," *Mechanism and Machine Theory*, Vol. 28, pp. 777-794.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1992, *Numerical Recipes*," Cambridge University Press.

Reklaitis, G.V., Ravindran, A., and Ragsdell, K.M., 1983, *Engineering Optimization, Methods and Applications*, Wiley, New York.

Simionescu, P.A. and Fawcett, J.N., 1997, "Static Equilibrium Determination of Multi-DOF Systems. A Review," Proceedings of the Seventh International Symposium on Linkages and CAD Methods, SYROM'97, Bucharest, Romania, Vol. 1, pp. 281-288.

Vanderplaats G.N., 1984, *Numerical Optimisation Techniques for Engineering Design*, McGraw-Hill.