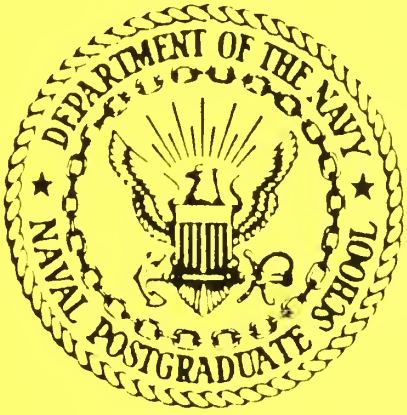
NPS-53-86-014

NAVAL POSTGRADUATE SCHOOL

Monterey, California



AN ALGORITHM FOR NOISY FUNCTION MINIMIZATION
FOR USE IN DETERMINING OPTIMAL TRAJECTORIES

by

I. B. Russak

J. B. Bassingthwaighte

I. S. Chan

A. A. Goldstein

January 1987
Technical Report For Period AY '86

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

R. C. AUSTIN
Rear Admiral, U.S. Navy
Superintendent

D. A. SCHRADY
Provost

This report was prepared by:

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS-53-86-014	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): An Algorithm For Noisy Function Minimization For Use In Determining Optimal Trajectories	5. TYPE OF REPORT & PERIOD COVERED: Technical Report, Academic Year 1986
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): I. B. Russak, J. B. Bassingthwaighte, I. S. Chan, A. A. Goldstein	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93943	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93943	12. REPORT DATE: January 1987
	13. NUMBER OF PAGES: 7
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

optimal trajectory, noisy function, kinetic energy, minimization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This work concerns a technique to be used in the solution of optimal trajectory problems associated with kinetic energy weapons. In this problem, it is desired to solve for a control function (which might be thrust magnitude and direction of a gimbaled engine) in time in order to minimize time to intercept an enemy missile.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AN ALGORITHM FOR NOISY FUNCTION MINIMIZATION FOR USE IN DETERMINING OPTIMAL TRAJECTORIES

I. B. Russak
Department of Mathematics, Naval Postgraduate School
Monterey, California

J.B. Bassingthwaighte, I.S. Chan, A.A. Goldstein
Center for Bioengineering University of Washington
Seattle, Washington

BACKGROUND

This work concerns a technique to be used in the solution of optimal trajectory problems associated with kinetic energy weapons. In this problem, it is desired to solve for a control function (which might be thrust magnitude and direction of a gimbaled engine) in time in order to minimize time to intercept an enemy missile.

Such problems are really infinite dimensional in nature (i.e., determining the control at each time point along the trajectory). However, in using a digital computer to solve such problems, certain operations occur which make the problem discrete and so viewable in a finite dimensional setting. For example, to numerically integrate the differential equations of motion, only values of thrust at a finite number of time points (typically, the beginning of each integration interval) affect the trajectory.

The problem then is to determine these values so as to minimize the time to intercept. For any particular trajectory, this quantity is computed through a complicated flight equation simulation model. Also inherent in this computation is noise so that the computed time to intercept is really a noisy quantity. The current algorithm considers the noise in solving for the optimal control.

INTRODUCTION

We are concerned here with the problem of minimizing functions in which the only data available are function values. These values could be obtained by: a) observation and, therefore, be subject to measurement errors, or by b) simulation and so, be subject to computational errors of the simulation model. A difficulty present in any type of numerical optimization and often, in particular, when these errors exist, is "stalling". This is the inability of the algorithm to further reduce the function from its value at a non-minimizing point. The object of the current work is to postpone "stalling" as long as possible.

"Stalling" arises when the magnitudes of the true gradient and of the error in determining the gradient become comparable. In the current problem, gradients are estimated from noisy function values which can be expected to include an absolute noise component. As the minimizing point is approached, the magnitude of the true gradient decreases but the absolute noise present in estimating it does not so that "stalling" results.

The procedure employed to delay "stalling" uses data smoothing in computations involving descent direction determination. The criterion of least squares is used to fit the data over a mesh of points. A secondary optimization problem is solved to optimize the location of the mesh points so that maximal descent direction determination accuracy will be achieved.

This latter optimization is very demanding in the number of data points and, hence, also the number of function evaluations required. For this reason, the determination of the mesh is done by three methods henceforth referred to as A, B and C. These are of increasing sophistication and expense and are used progressively as needed to provide continued decrease of the function f. Method B with associated numerical results is discussed in this paper. Method A appears in [1] and method C is the subject of a future paper.

DESCRIPTION OF THE ALGORITHM

The basic scheme is described in [2]. The data is assumed to include an absolute noise that is bounded by the input parameter epsilon. It is possible to also include relative noise, however, for small function values in the "stalling" phase (these are the type of cases that were run) absolute errors dominate relative erros. Thus, absolute errors were concentrated on in this paper.

The procedure of [2] consists of computing relaxed Newton and gradient steps. These directions are determined by using least squares to fit a quadratic smoothing polynomial over a local mesh of points surrounding the current estimate of the minimizing point. This fit yields first and second order coefficients for the function that define the Newton and gradient directions.

Method A is the first method used to determine the mesh. It is based on the fact that in the least squares process the accuracy in estimating the coefficients of the smoothing polynomial is directly related to the error in function value differences across the mesh. The method proceeds as follows: the spacing along each coordinate axis is selected with the purpose of maintaining at least a minimum significance in the above referred to difference of function values. A linear model is assumed for the function and the spacing predicted on the

basis of it. In order to keep function evaluations to a minimum at the current estimate of minimizing point, the success in achieving the above purpose is not verified. At the next estimate of minimizing point, the spacing is updated, based on the previous estimate. This scheme has proven to be adequate from far-off starting points where, due to a large gradient magnitude, the determination of the mesh is not so critical.

If, while performing mesh determination by method A, "stalling" occurs, the algorithm then switches to method B. This method has the same stated purpose as method A, but does not use a linear model. Also, the predicted value of spacing is verified by function evaluation and solution is accomplished through an iterative technique. Furthermore, in order to minimize truncation errors, the smallest feasible spacing is used.

In the event that "stalling" occurs while method B is being used for mesh determination, then the algorithm switches to method C. This method requires extensive computation. Hence, it is suited for use only in the final stages as the solution is approached. The goal now is to determine the mesh which minimizes the fitting error in the first and second order coefficients. This is done by adjusting the location of each mesh point so as to minimize the above error.

There are two points to note here. First, the optimal adjustment of each coordinate of each mesh point would normally lead to a large dimensional problem. Second, since the true coefficients are not known, then reduction of the fitting error cannot be done directly. Both of these difficulties are discussed and procedures developed to overcome them in a future paper.

MESH DETERMINATION BY METHOD B

Optimal spacing is different for first and second order coefficients. Because of this, a separate mesh and fit is computed for each of these. Following is a description of the construction of the mesh used to estimate the first order coefficients.

In addition to the absolute noise (discussed above) in function values, roundoff noise is present in representing these values in the computer. This is approximately equal to $r|f(x)|$ where r is the roundoff in representing unity in the computer. The criterion used for mesh spacing in this paper is the following folk dictum: across the mesh, first order function differences (used in estimating first order coefficients) should retain at least approximately one-half the number of significant digits present in f(X) itself. In order to satisfy this, the spacing for the jth coordinate axis (j=1,...,n) of the mesh is computed by iteration such that, with the notation listed next, then the formula below is true. The symbols f, X_0, h_j, I_j are respectively the function to be

minimized, the currest estimate of minimizing point, the spacing along the jth axis and the jth column of the identity matrix

$$|f(X_0 + h_j I_j) - f(X_0)| \approx \sqrt{2(\varepsilon + r|f(X_0)|)}$$

The above equation is solved by a method of bisection, thus determining the mesh used for least squares. Only the first order coefficients of this fit are used in the algorithm since these are the only ones for which the mesh was adjusted.

A scheme similar to the one above is used to determine the mesh for the second order coefficients. The distinction is in retaining the significance of second order differences. Analagous to the above, and using the notation there, the spacing is adjusted to satisfy

$$|f(X_0 + h_j I_j) + f(X_0 - h_j I_j) - 2f(X_0)| \approx \sqrt{4(\varepsilon + r|f(X_0)|)}$$

A new least squares determination is made with this mesh and the resulting hessian is used to compute a Newton direction. Next, line searches (described below) along both the Newton and gradient directions are made and the smallest function value used to determine the next estimate of the minimizing point.

THE LINE SEARCH

The first and second order coefficients resulting from the above fits are used to define the negative gradient direction D_g and the Newton direction D_n. These vectors define the directions of separate one dimensional searches (described next) to obtain a value of f lower than $f(X_0)$. That vector associated with the lowest attained value of f defines the direction and step for the next value of X_0 about which to form meshes and restart the process.

The one dimensional search for each vector consists of starting with an initial step size for Newton and gradient directions respectively sn = 1, sg = 2/ $H(X_0)$, (these stepsizes give guaranteed decreases in f along Newton and gradient direction) where H is the fitted Hessian of the function f. These values are halved repeatedly if f increases from $f(X_0)$. Tne halving process continues until an input value lower bound exceeds the current step size in which case no decrease is assumed along that vector. If f is decreased from $f(X_0)$ then

repeated steps of that size are taken until f starts to increase. As soon as a local dip occurs in the value of f, then an overdetermined quadratic is fit through the dip, the critical point determined and compared to the other sampled values of f along that vector and the lowest value recorded.

NUMERICAL RESULTS

Method B was tested in stand-alone mode, using test problems with standard starting values near the solution. Method A is counted on to provide the movement from far-off starts to points near the solution where method B will take over. For comparison purposes, the IMSL routine ZXMIN (a quasi-Newton method) and tne Nelder-Mead Simplex method were used.

Since function values for the runs were computed (rather than measured) then the roundoff error of r f(X) outlined above should be replaced by the actual roundoff error in computing f(X). This was done as follows: The runs were made on a computer with roughly seven decimal places of accuracy (single precision). In order to approximate the roundoff in computing f at location X, a vector of $|X|e^{-7}$ was added to X (as a bound on the single precision error in representing X) and this sum which shall be called X_d, was represented in double precision. Next, a double precision computation of f at X_d was performed. Calling this as $f_d(X_d)$, then the difference $|f_d(X_d)-f(X)|$ (where both function values were computed without input noise) was used as the roundoff error in computing f(X).

Four standard problems with standard starting values were run for each of the algorithms. These are the Rosenbrock,Beale and Freudenstein-Roth problems in two dimensions and the Helical Valley problem in three dimensions. For each problem the input noise bound epsilon was run at the levels of 0, .001 and .01. The input noise consisted of multiplying these values by the output of a uniform random number generator with mean 0 and variance 1. Runs were terminated when stalling occurred. The results are listed below with $\max_i |(\Delta X_j)|$, $|\Delta f|$ being respectively the maximum absolute component miss in achieving the minimizing point and the associated absolute miss in function value

PROBLEM & STARTING POINT	INPUT NOISE	METHOD B		ZXMIN		SIMPLEX	
		mx	|Δf|	mx	|Δf|	mx	|Δf|
ROSENBROCK	0.000	0.0	0.0	0.4E-3	0.3E-7	0.0	0.0
-1.2, 1.	0.001	0.104	0.4E-2	0.19	1.4	0.7E-2	0.6E-3
	0.010	0.330	0.37E-1	1.65	0.76E-1	0.368	0.35E-1
F-ROTH	0.000	0.0	0.0	0.1E-3	0.1E-3	0.2E-3	0.0
0.5, -2.	0.001	0.0	0.5E-3	0.1E-1	0.8E-3	0.7E-4	0.1E-3
	0.010	0.16E-2	0.21E-2	0.1E-1	0.8E-3	0.1E-1	0.33E-2
HELICAL	0.000	0.2E-9	0.2E-17	0.2E-6	0.6E-14	0.6E-8	0.5E-16
VALLEY	0.001	0.3E-1	0.8E-3	0.479	0.229	0.15	0.23E-1
-1.0, 0.0, 0.0	0.010	0.11	0.55E-2	2.9	8.48	0.189	0.26E-1
BEALE	0.000	0.000	1.E-13	overflow		0.00	0.000
10., 10.	0.001	0.39E-1	0.7E-3	overflow		7.19	0.297
	0.010	0.119	0.11E-1	overflow		7.26	0.302

CONCLUSIONS

Generally, the best performance was exhibited either by method B or the simplex method. In particular, for the lowest values of input noise, these methods either shared or alternated in attaining the best results. However, for the highest input noise level, method B showed definite dominance over the simplex method.

BIBLIOGRAPHY

1) Chan, Joseph, "Ggopt, a Method for Optimizing Whose Values are Suoject to Errors", National Simulation Resource, Department of Bioengineering, University of Washington report.

2) Glad, T. and Goldstein, A. A., "OPTIMIZATION OF FUNCTIONS WHOSE VALUES ARE SUBJECT TO SMALL ERRORS", BIT Vol. 17, 1977, 160-169.

DISTRIBUTION LIST

DEFENSE TECHNICAL INFORMATION CENTER (2)
CAMERON STATION
ALEXANDRIA, VA 22217

CHAIRMAN, CODE 53Fs (1)
DEPARTMENT OF MATHEMATICS
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

DEPARTMENT OF MATHEMATICS (1)
CODE 53
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

LIBRARY, CODE 0142 (2)
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

RESEARCH ADMINISTRATION OFFICE (1)
CODE 012
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

MR. DAN PARKS (3)
NAVAL SURFACE WEAPONS CENTER
DAHLGREN, VA 22448

PROFESSOR I. B. RUSSAK (3)
CODE 53Ru
DEPARTMENT OF MATHEMATICS
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943