UNCLASSIFIED

TISE

~~XDC 57-10-112~~

APEX-378

OPTIMIZING NUCLEAR DESIGNS BY MEANS OF
NON-LINEAR PROGRAMMING

D. S. Selengut
REACTOR ANALYSIS UNIT

October 14, 1957

GENERAL ELECTRIC-AIRCRAFT NUCLEAR PROPULSION DEPARTMENT-CINCINNATI 15, OHIO

UNCLASSIFIED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor the Air Force, nor any person acting on behalf of the Commission or the Air Force:

- A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above "person acting on behalf of the Commission or Air Force" includes any employee or contractor of the Commission or Air Force to the extent that such employee or contractor prepares, handles, or distributes, or provides access to, any information pursuant to his employment or contract with the Commission or Air Force.

Dont

Photostat

DISTRIBUTIONInternal

F. A. Aschenbrenner
J. R. Beeler
D. F. Burton
R. M. Cohen
W. C. Cooley
F. G. Dawson
B. H. Duane
P. G. Fischer
W. B. Henderson
C. R. Horner
J. G. Keppler
J. M. Krase
F. L. Langford
M. J. Langford
S. R. Lenihan
M. C. Leverett
J. E. MacDonald
R. R. McCready
F. W. Mezger
J. W. Morfitt
D. S. Selengut (20)
J. R. Terrall

External

Chief Nuclear Branch - Weapons Systems, WADC (2)
Chief Nuclear Branch - Power Plant Lab., WADC (2)
Chief Aircraft Reactor Branch - AEC
Technical Branch - AEC Lockland Office
AEC Lockland Office Representation, Idaho Falls
Technical Information Service Extension, Oak Ridge
COO - Program Reactors Division

Idaho Library
Document Control - Evendale (4)

ABSTRACT OF REPORT NO. XDC 57-10-112

TITLE: Optimizing Nuclear Designs by Means of Non-Linear Programming

AUTHOR AND ORIGINATING UNIT: D. S. Selengut, Reactor Analysis

DATE: October 14, 1957

The mathematical problem of non-linear programming consists of maximizing a function $f_0(x_1 \dots x_n)$ of n independent variables, subject to constraints in the form of inequalities, $f_m(x_1 \dots x_n) \geq 0$. If the independent variables are considered the design parameters and the function to be maximized describes the system performance, this set of equations is equivalent to the general problem of optimizing a complex engineering system. The case where the performance can be approximated by a quadratic function of the design variables and the constraints by upper and lower bounds is sufficient to exhibit the general properties of the system performance near the design point.

Because the constraints are in the form of inequalities, usual calculus of variations techniques cannot be applied. However, an iterative procedure suitable for machine computation has been developed which is comparable in efficiency to the Gauss-Seidel solution of n simultaneous algebraic equations. A computer program has been coded which will allow the optimization of a system containing up to 25 independent variables, subject to limit constraints. An example is discussed in which the radial power distribution in a hypothetical reactor is flattened by varying reflector thickness and core composition, subject to limits on maximum thickness and fuel loadings of various regions. Details of the FORTRAN source program for the 704 are given in the Appendix.

1. STATEMENT OF THE PROBLEM*

The problem I would like to consider is that of optimizing the performance of a complex engineering system, that is, finding the set of design variables which will make the system the best possible in some sense. The technique that has been developed for accomplishing this is closely related to the problem of dynamic programming. I would therefore like to first briefly discuss the relation of linear and non-linear dynamic programming to optimizing engineering designs, second, indicate the mathematical procedure for explicitly carrying out this optimization, and finally, discuss a specific problem that this technique has been applied to, namely flattening the power density in a particular reactor design.

The problems that one is led to in trying to optimize an engineering design are typically maximization problems; however, they have special features which prevent one from applying the usual calculus of variations techniques to them. A large class of such problems can be described according to the formalism given in equation (1):

$$\begin{aligned} &\text{Maximize } f_0(x_1 \dots x_n) \text{ subject to the} \\ &\text{constraints } f_i(x_1 \dots x_n) \geq 0, i = 1 \dots m \end{aligned} \quad (1)$$

In this form the problem is equivalent to the general problem of dynamic programming, which in the linear case (where all the functions f_m are linear in the independent variables) has had extensive applications in economics, for example. The difficulty lies in the nature of the constraints: rather than being in the form of equalities, they are statements that a certain function of the independent variables must be greater than zero. This is the reason that one cannot apply usual techniques to the problem.

The relation of dynamic programming to the problem of optimizing an engineering design is quite straightforward, if we identify the x_n with the design variables of the system, and consider the dependent functions f_m as describing certain properties of the system that are of interest. The problem then reduces to the statement that there is some property of the system (which we would naturally call the performance) that we want to maximize, subject to constraints in the form of inequalities on other properties of the system.

*This paper was presented at the June 1957 meeting in Pittsburgh of the American Nuclear Society.

The meaning of these rather abstract statements can be made a little clearer if they are interpreted in terms of a reactor design problem. In this case the performance f_0 might be the reactivity of the system, which one would want to maximize. The remaining functions might describe limitations on the amount of fissionable material that might be loaded in various regions, or geometrical constraints such as limitations on cooling channel diameter. Alternatively, one might choose to flatten the power to the greatest extent possible while maintaining the reactivity of the system above a fixed level and keeping reflector thickness, for example, within certain limits. I think it is clear that this formalism covers a very wide variety of engineering optimization problems.

Unfortunately this general problem of non-linear programming is quite difficult to solve and relatively little work has been done on it*. Instead of dealing with the problem in its full generality, I would like to consider a special case which, however, retains the general features that characterize this type of problem and which does have useful applications. This restricted problem is stated in equation (2):

$$\text{Maximize } y = a + \sum_{i=1}^n a_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \quad (2)$$

subject to the constraints $A_i \leq x_i \leq B_i, i = 1 \dots n$

The function y describing the performance, which is to be maximized, is given as a quadratic function of the design variables, which in turn are subjected only to the special linear constraint that they lie within a given range.

While this is a considerable simplification of the general case, it still retains the characteristic features of the optimization problem. Since the performance is written as a quadratic function of the design variables, it permits the existence of a maximum within the allowed region, in contrast to linear programming. Furthermore, since one is primarily interested in the behavior of the system only near the optimum point, and since the performance is usually a smoothly varying function of the design variables near the optimum, it seems reasonable to make a Taylor series expansion, neglecting higher than

*J. H. McCampbell, "On the Application of Multivariant Analysis to Nuclear Design", WAPD-T-538; R. D. Sheffield, "Gradient Non-Linear Method of Optimization", paper 7 in Reports to the 3rd Semi-Annual ANP Shielding Information Meeting, May 1957, APEX 325 (classified).

quadratic terms. Finally, the constraints that are indicated, while not sufficiently general to cover some cases of interest, nevertheless do include a great many typical limitations on reactor design: for example, limitations on fuel loading imposed by mechanical considerations, or limitations on the dimensions or geometry of reactor components.

2. METHOD OF SOLUTION

The method of solution is an iterative one and arises quite naturally from considering the procedure for solving the problem in the absence of constraints. In this case, the maximum value of the quadratic form is given simply by setting the derivatives of y with respect to all the x 's equal to zero as indicated in equation (3):

$$\frac{\partial y}{\partial x_i} = a_i + \sum_{j=1}^n a_{ij}x_j = 0 \quad (3)$$

The problem then reduces to the solution of n simultaneous equations for the values of x .

One way to carry this out is to solve each of the n equations for the corresponding independent variable as is done in equation (4):

$$\bar{x}_i = x_i - \frac{1}{a_{ii}} \left(a_i + \sum_{j=1}^n a_{ij}x_j \right), \quad i = 1 \dots n \quad (4)$$

If one then starts with an arbitrary set of x 's, substitutes them in the right-hand side and evaluates the corresponding new values (designated here by $\bar{x}_i$) successively for each value of i , then eventually, under the right conditions, the procedure will converge to the solution of the linear equations. This procedure, of course, is the well-known Gauss-Seidel method. Geometrically it corresponds to improving an approximate solution by a series of displacements parallel to each coordinate axis in turn, in the general direction of the maximum of the quadratic function.

Now, the restricted maximum is either going to lie within the region defined by the constraints, or else on the boundary. It is therefore plausible that we can take the boundary into account by stopping at each iteration and deciding whether or not the improved value of x actually satisfies the constraints. If it does not, then the correct value at that stage will lie on one of the boundaries.

Deciding whether the improved value of x lies at the point of zero derivative or on one or the other of the two boundaries is a straightforward matter of case analysis. The five possible cases are illustrated in Figure 1.

To summarize, the iteration procedure consists of using the Gauss-Seidel process to improve each of the x 's in succession, and choosing at each stage the allowed value of the independent variable which gives the maximum value of y . This procedure is iterated until a succeeding set of values x are close enough together so that convergence has taken place; one convenient criterion is given in equation (5):

$$\frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - x_i'}{B_i - A_i} \right)^2 \leq \epsilon^2 \quad (5)$$

The procedure should be a little clearer in terms of its geometric interpretation, which is shown in Figure 2 for a two-variable case. Here the allowed region is the rectangle blocked off in the diagram, and the ellipses are level surfaces or contour lines $y = \text{constant}$. The unrestricted maximum is given by the point in the center of the ellipses and lies outside the constraints. As a trial value to start with, the point in the lower left-hand corner might be chosen. After one iteration, improving x_1 and x_2 , the second point is reached. After two iterations the improved values of x_1 and x_2 lie on the upper boundary, and it requires only one more to reach the optimum within the limitations imposed by the constraints.

Several points should be noted. First, while at least one solution always exists, there may quite possibly be many solutions or even a continuum of them. The one which the procedure will converge to depends on where the initial value is chosen. In a practical problem one usually has some idea of what region the design variables should lie in and starting at this point would lead to the nearest maximum. However, there is no guarantee that the absolute maximum will be obtained, and the procedure may converge to a saddle point.

Second, the choice of the Gauss-Seidel procedure was made only because it was the simplest one to use; any other iterative procedure could be used since the method depends basically on the maximization of the dependent variable with respect to one independent variable at a time. The extension of the procedure, in principle at least, could be made to the case of a general function to be optimized within an arbitrary region, since as long as the maximum can be found

Figure 1

CHOOSING THE RESTRICTED MAXIMUM

If $\bar{x} < \frac{A+B}{2}$, then $x = B$

If $\bar{x} \geq \frac{A+B}{2}$, then $x = A$

If $\bar{x} < A$, then $x = A$

If $A \leq \bar{x} < B$, then $x = \bar{x}$

If $\bar{x} \geq B$, then $x = B$

UNCLASSIFIED

UNCLASSIFIED

UNCLASSIFIED

101 303

Figure 2
TWO-VARIABLE CASE

UNCLASSIFIED

for the dependent variable as a function of each of the independent variables in succession, the same general procedure will apply.

Finally, the time required for computation has to be evaluated in terms of the fact that for the special case in which the constraints define a very large region, and the a_{ij} matrix corresponds to a convex surface, the problem is equivalent to solving a set of n simultaneous equations. This gives one a lower limit on the amount of labor involved. For the method just described, the extra work involved beyond the straightforward iteration procedure consists in the examination, at each stage in the process, of whether the maximum lies inside or outside the constraints. One can conclude that the amount of computational time required for this method is comparable to that required for the iterative procedure; perhaps a factor of two or three larger, but in any case of the same order of magnitude.

3. APPLICATION TO A REACTOR DESIGN PROBLEM

As a specific example of the method, the problem of minimizing the peak to average power in a reactor will be considered. The motivation behind this lies in the assumption that the reactor is temperature limited, so that one would like to flatten the radial power distribution, which is normally a cosine-like curve, in order to obtain the greatest heat output consistent with the material limitations of the core.

The reactor chosen is a hypothetical one and is intended only for illustrative purposes. The core, shown in Figure 3, consists of a square cylinder with three radial regions in which various fuel loadings are allowed, and is surrounded by a radial reflector. The nuclear constants used correspond roughly to a water moderated core and a beryllium reflector. The design variables to be varied to achieve a lower peak to average power are the fuel loading within each of the three regions, and the reflector thickness. The constraints to be imposed are that the fuel loading within each region lies between 100 and 200 pounds (these are total core loadings that would result from a uniform loading with this density in each region) and the reflector is to be allowed to vary between zero and six inches in thickness.

In order to apply the method it is necessary first to represent the peak to average power as a quadratic function of each of the four design variables. This was carried out by the use of the synthetic experiment design technique*. A

*J. M. Krase and C. Cyl-Champlin, "Synthetic Experiment Design Techniques in Reactor Analysis", APEX 303.

Figure 3

series of twenty-five synthetic experiments, consisting of power distribution calculations using a two-group model were carried out. (This is, incidentally, less than a third of the number of calculations that would be needed to cover this range of variables if a factorial design had been used, that is, a set of calculations in which the effect of each variable was studied on the response at three different levels.) Figure 4 shows the power distributions from four typical calculations.

A computer program carrying out the calculations outlined above has been written for the 704, and was used to compute the values of the design variables which would give a minimum peak to average power. For equivalent loadings in the three regions of 100, 100 and 200 pounds of fuel, and a reflector thickness of 4.48 inches, a minimum peak to average power of 1.16 was obtained.

While the present example is sufficiently simple to allow a good guess to be made of just where the optimum is, it should be remembered that in many problems the physical effect of changing any one of the design variables is by no means as obvious as the effect of loading on the power distribution. Moreover, it is not entirely obvious that within the restrictions of this problem the optimum should occur with this particular combination of design variables; one might expect an intermediate loading would be better in the middle core region, for example. In more realistic cases where 10 or 15 independent variables are present in the system, and where the system must be optimized on all of these variables simultaneously if high performance is to be obtained, it would be extremely difficult to find the optimum without some systematic procedure.

Finally, it should be pointed out that the application just given is quite special and that the procedure should apply to a wide variety of other optimization problems.

ACKNOWLEDGEMENTS

I would like to thank J. G. Keppler for carrying out the experiment design calculations, M. J. Langford for help in preparing the figures, F. Gagliano of the IBM Service Bureau for checkout of the FORTRAN program, and J. E. Dolby of the General Engineering Laboratory for many stimulating discussions.

Figure 4

TYPICAL POWER DISTRIBUTIONS

4. APPENDIX

This appendix gives details of the computer program used in Section 3. Coding was done in FORTRAN*, an automatic coding system developed by IBM for the 704.

4.1 Statement of the Problem

The program is intended to find the maximum of the quadratic function

$$y = a + \sum_{i=1}^n a_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

subject to the constraints

$$A_i \leq x_i \leq B_i, \quad i = 1 \dots n.$$

The application is to optimization problems in which y is the system performance, while the x_i are design variables. The case where y is to be minimized can be obtained by changing the sign of all the coefficients a , a_i , a_{ij} .

The present code can handle up to 25 independent variables. An iterative procedure is used which is similar to the Gauss-Seidel solution of a set of simultaneous linear equations. It should be noted that the problem may have several solutions; the one obtained by the machine depends on the initial guess, and may be only a saddle point or relative maximum.

4.2 Notation (FORTRAN notation is given in the second column)

4.2.1 Input

-	CASE	case number
n	N	number of independent variables (≤ 25)
ϵ	EPSLN	constant determining convergence
x_i	X(I)	initial estimate of values of independent variables (need not be given as input)
a	A0	coefficients of the quadratic function
a_i	A1(I)	
a_{ij}	A2(I,J)	
A_i, B_i	A(I), B(I)	lower and upper limits of the independent variables

*"The FORTRAN Automatic Coding System for the IBM 704 EDPM", and "FORTRAN Introductory Programmer's Manual", Programming Research Department, International Business Machines Corporation.

4.22 Variables Used Within the Program

old x_i	OLDX(I)	independent variable from previous iteration
$\bar{x}_i$	XB(I)	trial value of x_i
-	S	partial sum

4.23 Output

y	Y	dependent variable
x_i	X(I)	values of independent variables which maximize y
-	SUM	sum to be compared to ϵ to determine convergence

4.3 Computations

- (1) Read and print input data.
- (2) Complete a_{ij} matrix ($a_{ij} = a_{ji}$).
- (3) Prepare for the next iteration.
 - (i) Re-label x_i as old x_i .
 - (ii) Compute y and print, together with x_i .
- (4) Improve each x_i successively, $i = 1 \dots n$:
 - (i) For the current value of i, calculate the trial value of x_i :

$$\bar{x}_i = x_i - \frac{1}{a_{ii} + 10^{-10}} \left(a_i + \sum_{j=1}^n a_{ij} x_j \right)$$

- (ii) Calculate the improved value of x_i :

If $a_{ii} < 0$ and $\bar{x}_i < A_i$, then $x_i = A_i$

$A_i \leq \bar{x}_i < B_i$, then $x_i = \bar{x}_i$

$\bar{x}_i \geq B_i$, then $x_i = B_i$

If $a_{ii} \geq 0$ and $\bar{x}_i < \frac{A_i + B_i}{2}$, then $x_i = B_i$

$\bar{x}_i \geq \frac{A_i + B_i}{2}$, then $x_i = A_i$

- (5) Test for convergence:

$$\text{If } \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \text{old } x_i}{x_i} \right)^2 \leq \epsilon^2, \text{ print its value and stop; otherwise}$$

go back to step (3).

4.4 Operation

The problem has been coded for an IBM 704 with an 8192 word (or larger) core memory, card reader, and on-line printer.

During operation, sense switch 1 should be up. Insert the input data for all cases at the end of the program deck and load into the 704. After each case the machine will stop with a (7777)₈ displayed in the address portion of the storage register. To start the next case, press the start button.

The program required about 10 minutes for assembly. Calculation time for cases run so far has been under a minute, and is apparently determined by the speed of the on-line printer. If the program were re-assembled, it might be desirable to include the option of tape input and output in addition to on-line operation. More than 25 independent variables could be handled by changing the dimension statement before re-assembly.

4.5 FORTTRAN Source Program

```

C      OPTIMIZER 1, MAY 9, 1957      D. SELENGUT, ANPD
      DIMENSION X (25), OLD X (25), A1(25), A2(25,25), A(25), B(25)
C      READ AND PRINT INPUT DATA
915  READ1,CASE,N,EPSLN,(X(I),I=1,N),AO,(A1(I),I=1,N),
      1((A2(I,J),J=1,I),I=1,N),(A(I),I=1,N),(B(I),I=1,N)
0001 FORMAT(F8.4,I4,(F10.5))
      PRINT 2, CASE, N, EPSLN
      PRINT 3, (I,X(I), I=1,N)
      PRINT 4, AO, (I,A1(I), I=1,N)
      PRINT5,((I,J,A2(I,J),J=1,I),I=1,N)
      PRINT 6, (I,A(I),B(I), I=1,N)
0002 FORMAT(25H OPTIMIZER 1 CASE NUMBER F8.4//4H N =I3,
      110H EPSILON =F8.4
0003 FORMAT(3H X(I3,4H) = F10.4)
0004 FORMAT(6H AO = F8.4,(4H A1(I3,4H) = F10.4))
      5 FORMAT(4H A2(I3,1H,I3,4H) = F8.5)
0006 FORMAT(20H      1      A(I)      B(I)/(1H I3,2F8.5))
C      COMPLETE A2(I,J) MATRIX
      100 DO 110 I=1,N
      105 DO 110 J=1,I
      110 A2(J,I) =A2(I,J)
C      PREPARE FOR NEXT ITERATION
0120 DO121I=1,N
      121 OLD X(I)=X(I)
      130 Y=0
0131 DO133I=1,N
0132 DO133J=1,N
      133 Y=Y+A2(I,J) *X(I) *X(J)
      Y=AO+Y/2.0
0135 DO137I=1,N
      137 Y=Y+A1(I) *X(I)
      PRINT 3, (I,X(I), I=1,N)
      PRINT 8,Y
0008 FORMAT(4H Y= F10.6)
C      ITERATE ON XS
0200 DO300I=1,N
C      CALCULATE XB, TRIAL VALUE OF X
      S=A1(I)
      203 DO 205 J=1,N
      205 S=S+A2(I,J) *X(J)
      XB=X(I)-S/(A2(I,I)+1.0E-10)
      IF(SENSESWITCH1)910,221
910  PRINT900,I,XB,A2(I,I)
900  FORMAT(I3,2F20.9)
C      CALCULATE X(I)
      221 IF (A2(I,I)) 223,222,222
      222 IF (XB-(A(I)+B(I)) /2.0) 233,231,231
      223 IF (XB-B(I)) 224,233,233
      224 IF (XB-A(I)) 231,232,232
      231 X(I)=A(I)
      GO TO 300
      232 X(I)=XB
      GO TO 300
      233 X(I)=B(I)

```

```
300 CONTINUE
C   ALL X(I) REPLACED BY NEW VALUES.  CALCULATE CONVERGENCE.
   IF(SENSESWITCH1)911,912
911 PRINT900,I,X(I)
912 S=0
305 DO 310 I=1,N
310 S=S+(OLDX(I)-X(I))**2
   FN=N
315 IF(S/FN-EPSLN**2)400,400,120
400 PRINT 9,S
9   FORMAT(26H SUM (1-OLD X(I)/X(I))**2=E16.9)
   PAUSE 77777
   GO TO 915
   FREQUENCY 100(10), 105(10), 120(10), 131(10), 132(10), 135(10),
1200(15), 203(10), 221(5,1,5), 222(5,1,5), 223(5,1,5), 224(5,1,5),
2305(10), 315(1,1,20)
```

4.6 Test Case

OPTIMIZER 1 CASE NUMBER 6.0000

N = 4 EPSILON = 0.0001

X(1) = -0.

X(2) = -0.

X(3) = -0.

X(4) = -0.

A0 = -1.2593 A1(1) = -0.0490

A1(2) = 0.0327

A1(3) = 0.0215

A1(4) = 0.2043

A2(1, 1) = 0.02116

A2(2, 1) = 0.00291

A2(2, 2) = -0.00236

A2(3, 1) = 0.00956

A2(3, 2) = 0.00006

A2(3, 3) = -0.00036

A2(4, 1) = 0.02094

A2(4, 2) = -0.01006

A2(4, 3) = -0.00906

A2(4, 4) = -0.19936

I A(I) R(I)

1-1.41420 1.41420

2-1.41420 1.41420

3-1.41420 1.41420

4-1.50000 1.50000

X(1) = -0.

X(2) = -0.

X(3) = -0.

X(4) = -0.

Y = -1.259310

X(1) = -1.4142

X(2) = 1.4142

X(3) = 1.4142

X(4) = 0.7404

Y = -1.164843

SUM (1-OLD X(I)/X(I))**2 = 0.555111513E-16