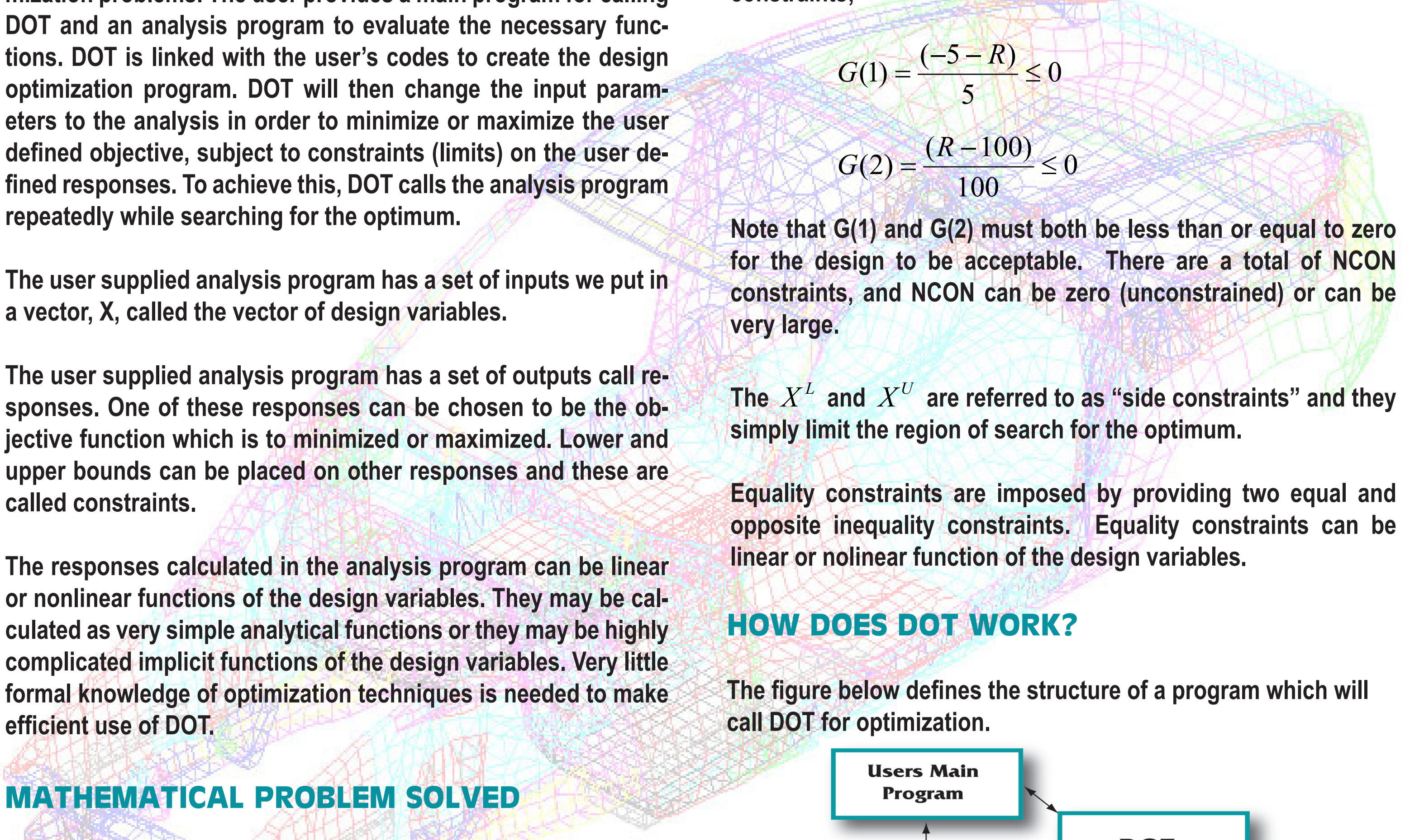
DOT

Design Optimization Tools



DESCRIPTION

DOT is a general purpose numerical optimization software package which can be used to solve a wide variety of nonlinear optimization problems. The user provides a main program for calling DOT and an analysis program to evaluate the necessary functions. DOT is linked with the user's codes to create the design optimization program. DOT will then change the input parameters to the analysis in order to minimize or maximize the user defined objective, subject to constraints (limits) on the user defined responses. To achieve this, DOT calls the analysis program repeatedly while searching for the optimum.

The user supplied analysis program has a set of inputs we put in a vector, X, called the vector of design variables.

The user supplied analysis program has a set of outputs call responses. One of these responses can be chosen to be the objective function which is to minimized or maximized. Lower and upper bounds can be placed on other responses and these are called constraints.

The responses calculated in the analysis program can be linear or nonlinear functions of the design variables. They may be calculated as very simple analytical functions or they may be highly complicated implicit functions of the design variables. Very little formal knowledge of optimization techniques is needed to make efficient use of DOT.

MATHEMATICAL PROBLEM SOLVED

DOT numerically solves the following problem:

Find the values of the N design variables contained in X that will:

Minimize or maximize the objective function OBJ=F(X)

Subject to (such that):

$G_j(X) \le 0$, for $j = 1, \ldots, NCON$

where NCON is number of constraints.

$X_i^L \le X_i \le X_i^U$, for $i = 1, \ldots, NDV$

where NDV is the number of design variables.

Array G contains the constraints that must be satisfied.

For example, if some response, R, must be greater than -5 and less than 100, this would lead to the following two (normalized) constraints;

$$G(1) = \frac{(-5 - R)}{5} \le 0$$

$$G(2) = \frac{(R - 100)}{100} \le 0$$

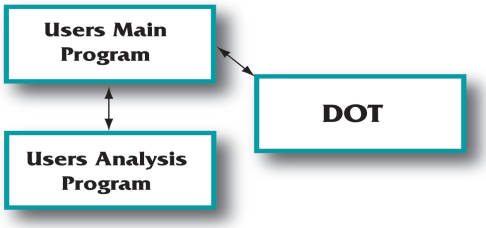
Note that G(1) and G(2) must both be less than or equal to zero for the design to be acceptable. There are a total of NCON constraints, and NCON can be zero (unconstrained) or can be very large.

The X^L and X^U are referred to as "side constraints" and they simply limit the region of search for the optimum.

Equality constraints are imposed by providing two equal and opposite inequality constraints. Equality constraints can be linear or nolinear function of the design variables.

HOW DOES DOT WORK?

The figure below defines the structure of a program which will call DOT for optimization.



The user must provide a main program which defines various arrays and parameters. The user also provides an analysis subroutine or function which evaluates the objective and constraint functions, and calculates gradient information if that option is used. The main program calls DOT to proceed with optimization. DOT will modify the design variables in search of the optimum. When DOT requires the values of the objective function and constraints corresponding to a proposed design, it returns control to the main program and the analysis is called to evaluate them. DOT is then called again, and this process is repeated until DOT returns a parameter to indicate that optimization is complete.

ALGORITHMS

For Constrained Optimization:

- Modified Method of Feasible Directions (MMFD).
- Sequential Linear Programming (SLP) with adjustable move limits.
- Sequential Quadratic Programming (SQP).

For Unconstrained Optimization:

- Brydon-Fletcher-Goldfarb-Shanno (BFGS) Algorithm.
- Fletcher-Reeves (FR) Algorithm.

The algorithm to be used for optimization is controlled with a single input parameter to DOT. These algorithms have been extensively developed and tested and have been demonstrated to be both efficient and reliable for a wide range of engineering applications.

OPTIONS

Default values are provided for all control parameters.

- These control parameters can be easily modified by the user.
- All information is transferred via the DOT subroutine/ function parameter list.
- Optimization can be stopped at any time and restarted later.
- DOT calculates gradient information by finite difference.
- The user can provide analytic gradients if this is desired.
- Design history information can be directed to a separate output file for later graphic processing.

CONFIGURATIONS

RAM Memory:

DOT does not have any fixed memory requirements. You will programmatically set them.

Disk Space:

The DOT libraries and documentation require approximately 3 MB of free disk space.

Program Development Environments:

DOT is delivered as an object library that you can link with your programs. DOT has been tested with Microsoft PowerStation FORTRAN 4, Digital Visual Fortran 5/6, Microsoft Visual C++ 5/6 and all major UNIX workstation compliers.

OTHER VR&D PRODUCTS AVAILABLE

GENESIS - Structural Analysis & Optimization software

GENESIS is a fully integrated finite element analysis and design optimization software package. Analyses include static, normal modes, direct and modal frequency analysis, heat transfer and system buckling. Shape, sizing, topography, topometry and topology optimization are the design options available to the user. Typically the optimization requires less than ten detailed finite element analyses, even for large and complex design tasks.

Design Studio for GENESIS

Design Studio for GENESIS is a design oriented pre- and post-processor graphical interface for the GENESIS program

VisualDOC

VisualDOC is a software system that simplifies adding optimization to almost any design task. It uses powerful, intuitive graphical interface, both gradient based and non-gradient based optimization, response surface (RS) approximate optimization, and design of experiments (DOE) methods. VisualDOC interfaces easily to your own code or third-party analysis programs.

SMS Eigensolver

The SMS eigensolver may be added to existing NASTRAN installations to offer significant performance advantages over the default method when a large number of eigenmodes is required for a system with many degrees of freedom. Benchmark tests and user experiences have seen solutions times anywhere between 2 - 5 times faster when using SMS. SMS may also be embedded into your product/software.

BIGDOT

BIGDOT is intended to solve very large, nonlinear, constrained problems where gradient information is available, and function and gradient evaluation is efficient. BIGDOT is capable of solving continuous, discrete/integer or mixed variable problems. Problems in excess of 100,000 variables have been solved by BIGDOT.

Vanderplaats Research & Development, Inc.

Headquarters:
1767 S. 8th Street
Colorado Springs, CO 80905
Ph. 719-473-4611
Fax. 719-473-4638
Email: Sales@vrand.com
www.vrand.com

Michigan Office:
41700 Gardenbrook
Suite 115
Novi, MI 48375
Ph. 248-596-1611
Fax. 248-596-1911

California Office:
126 Bonifacio Place
Suite F
Monterey, CA 93940
Ph. 831-373-4611
Fax. 831-373-4638

11/19/2008