16.888/ESD.77J Multidisciplinary System Design Optimization (MSDO) Spring 2004

Assignment 2

Instructors:	Prof. Olivier de Weck Prof. Karen Willcox
Issued:	Monday Feb. 23, 2004
Due:	Monday March. 8, 2004

Topics: Decomposition, Gradient-Based Optimization, Simulation Code Implementation

Part (a)

(a1) Decomposition

The following non-linear system of equations is given:

$$x_1 x_2 - 2x_3 + 2 = 0 \quad (1)$$
$$x_2 + 3x_5 - 9 = 0 \quad (2)$$
$$x_1 - x_4 x_5 - x_3 + 10 = 0 \quad (3)$$
$$9x_5 - 3x_2 + 7 = 0 \quad (4)$$
$$x_2 x_5 - x_2 x_4 + x_2 - 9 = 0 \quad (5)$$

Solve this system of equations for the set of values x_i, i=1,2,...,5 in two different ways. First, solve the entire system at once choosing a method of your preference. Second, solve the system by decomposition described in class (and in the handout). You can solve this numerically if you wish, but you should attempt a manual solution first.

Show that the solution obtained with each method is identical. Estimate the difference in computational effort (e.g. number of calculations, CPU time) between the all-at-once and the decomposition-based approach.

(a2) Gradient-Based Optimization

Consider the well-known Rosenbrock's "banana" function, which is often used as a test function for numerical optimization algorithms.

$$f = \left(1 - x_1\right)^2 + 100\left(x_2 - x_1^2\right)^2$$

(i) Implement the steepest descent algorithm and determine min(f) using at least two different initial guesses.

(ii) Implement the conjugate gradient algorithm and determine min(f) using the same initial guesses as in (i)

(iii) Verify your computational results against the analytical solution, and compare the performance of the two numerical methods.

Part (b)

We will begin the numerical implementation of your design project.

Note: For Assignment 1, many of you may have described large, complex systems with many disciplines, objective functions and design variables. For this class, you should focus in on a portion of this problem for which you can realistically hope to get numerical results. It does not matter if your initial analysis is not physically comprehensive – you can always increase the complexity of your problem later on. For example, you may want to turn certain design variables into prescribed parameters (especially discrete variables) and start by exploring just a subset of the full design space.

(b1) Simulation Implementation
Numerically implement your system model with all of its disciplinary modules. Demonstrate that the system can be executed in analysis mode, *i.e.* provide a design vector $\boldsymbol{x}$ and show that your simulation returns the correct values of the objective functions $\boldsymbol{J}$ and constraints $\boldsymbol{g}, \boldsymbol{h}$. Describe how you validated your analysis routines.

If there are modules in your system for which the simulation code is not yet available, insert a simple "placeholder" code which relates outputs to inputs. For example, use a simple polynomial expression that at least has the correct sign (output increases/decreases with increasing input). Please include in your write-up a note that describes the development necessary to complete these modules and your expected completion date.

(b2) Feasibility
Describe the initial design vector that you have available. Is this initial guess feasible?

(b3) DOE
Carry out an initial exploration of the design space using a DOE technique of your choice, recall lecture L5. What design variable (factor) and level shows the largest effect? Based on this analysis what initial start point $\mathbf{x}_0$ do you recommend for numerical optimization? You don't have to actually carry out this optimization (yet).

Note: This assignment is aimed at getting you to complete the problem description and implement the simulation code. Do not panic if your first attempt at optimization is a dismal failure – in practice, MDO codes seldom get things right first time. Throughout the semester you will have a chance to improve your formulation and implementation.