

Assignment: A Gentle Introduction to the Manifold-Mapping Technique for PDE-Constrained Optimization Problems

D. Lahaye

DIAM - Delft Institute of Applied Mathematics
Department of Electrical Engineering, Mathematics and Computer Science
TU Delft, Mekelweg 4, 2628 CD Delft, The Netherlands
phone: +31.15.27.87.257 fax: +31.15.27.87.209
<http://ta.twi.tudelft.nl/nw/users/domenico/>
d.j.p.lahaye@tudelft.nl

1 Introduction and Problem Statement

In this assignment the manifold-mapping optimization algorithm is to be studied, implemented and applied to two optimization problems. For both problems the performance of this technique (in terms of number of iterations and required CPU time) is to be compared with that of classical optimization algorithms. Although for the implementation any programming language is allowed (MATLAB, JAVA, C++, C, Fortran or other), an implementation in Python will be especially appreciated. A written report detailing the findings is to be compiled and sent to me.

First, read the paper [2] having an operational understanding of the manifold-mapping algorithm in mind, i.e., mainly focussing on Section 5. This paper is available on my web site. In the following we describe the two optimization problems taken from [1], also described in Section 6.1 and Section 6.2 of [2]. We adopt the notation as in [1, 2].

The first problem has a single variable $\mathbf{x} \in X \subset \mathbb{R}$ (i.e., $n = 1$). It denotes a distance (the radius of a permanent magnet, e.g.) and can range between $\mathbf{x}_m = 1\text{e-}3$ and $\mathbf{x}_M = 45\text{e-}3$ (in meters), i.e.,

$$X = [\mathbf{x}_m, \mathbf{x}_M] \quad [1\text{D design variable set}].$$

The fine model response $\mathbf{f}(\mathbf{x})$ is a nine component vector function, i.e., $\forall \mathbf{x} \in X$, $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^9$ ($m=9$). Each component represent the magnetic force on a moving part of a device. The nine components correspond to different positions on the moving part. Assume that in the fine model the response is computed after having solved a partial differential equation for the magnetic field and that this process is computationally costly. The goal in the optimization problem is to find the size such that the computed force comes close a prescribed values or objective denoted by $\mathbf{y} \in Y \subset \mathbb{R}^9$. The fine model cost function is denoted by $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2$. The single variable fine model optimization problem can then be stated as:

find $\mathbf{x}^* \in X = [\mathbf{x}_m, \mathbf{x}_M]$ such that

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{z} \in X} \|\mathbf{f}(\mathbf{z}) - \mathbf{y}\|_2 = \operatorname{argmin}_{\mathbf{z} \in X} F(\mathbf{z}) \quad [1\text{D fine model problem}]. \quad (1)$$

The second optimization problem is a two-dimensional extension of the first one. Instead of having a single distance, we now have two distances (radius and height of a permanent magnet) that form the components $\mathbf{x}(1)$ and $\mathbf{x}(2)$ of the vector of design variables $\mathbf{x} \in X \subset \mathbb{R}^2$ (i.e., $n = 2$). On the components we impose the following lower and upper bounds: $\mathbf{x}_m(1) = 1\text{e-}3$, $\mathbf{x}_M(1) = 45\text{e-}3$, $\mathbf{x}_m(2) = 5\text{e-}3$ and $\mathbf{x}_M(2) = 45\text{e-}3$, i.e.,

$$X = [\mathbf{x}_m(1), \mathbf{x}_M(1)] \times [\mathbf{x}_m(2), \mathbf{x}_M(2)] \quad [2\text{D design variable set}]$$

The fine model response and objective have the same meaning as in the 1D problem. The two variables fine model optimization problem can be stated as find $\mathbf{x}^* \in X = [\mathbf{x}_m(1), \mathbf{x}_M(1)] \times [\mathbf{x}_m(2), \mathbf{x}_M(2)]$ such that

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{z} \in X} \|\mathbf{f}(\mathbf{z}) - \mathbf{y}\|_2 = \operatorname{argmin}_{\mathbf{z} \in X} F(\mathbf{z}) \quad [2D \text{ fine model problem}]. \quad (2)$$

To avoid the use of (or having to develop) a partial differential equation solver and to facilitate the experimentation with different algorithmic variants, the evaluation of the fine model response function is made available in the form of a lookup table. That is, for both optimization problems (1) and (2) we provide a table of function values $\mathbf{f}(\mathbf{x})$ for a sufficiently fine sampling of the space X . We provide a sampling of X on a uniform mesh consisting of 41 and 41 times 41 nodes, respectively. We also provide functionality implemented in MATLAB to read this table and to evaluate the fine model response $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^9$ and the fine model cost function $F(\mathbf{x})$ for a given $\mathbf{x} \in X$.

A distribution of files structured according to Fig. 1 accompanies this assignment and is available for download from my website. In the following we describe first how the input data is organized and what functionality is provided to manipulate this data.

For the problem (1), the files

1. `assignment/1dproblem/data/fm_resp_mat.dat`: is an ASCII (i.e., human readable) file that provides a matrix of size 9 by 41 in which each column corresponds to a vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^9$, for \mathbf{x} being one of the 41 grid points on X ;
2. `assignment/1dproblem/code/lookup_fm_resp.m`: given an $\mathbf{x} \in X$ on input, returns the fine model response $\mathbf{f}(\mathbf{x})$. This function reads data from `fm_resp_mat.dat`;
3. `assignment/1dproblem/code/compute_fm_cost.m`: given an $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ on input, returns the fine model cost function $F(\mathbf{x})$.

The fact that the data files in the directory `assignment/1dproblem/data` are written in ASCII format, should facilitate the conversion of the provided building blocks to a non-MATLAB environment (if desired) and enhance portability across different versions of MATLAB. To preserve this facility for the 2D problem, we opted for a slightly more complex structure.

For the problem (2), the directory/files

1. `assignment/2dproblem/data/fm_data/`: provides a collection of 41^2 ASCII files each containing a vector of size 9 each corresponding to a vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^9$, for \mathbf{x} being one of the 41 by 41 grid points on X ;
2. `assignment/2dproblem/tools/build_fm_resp_mat.m`: provides a loop to read all of the 41 by 41 files in `assignment/1dproblem/data/fm_data/` and to construct three-dimensional arrays of dimension 9 by 41 by 41 of fine model responses. This matrix is written as the MATLAB **binary** data file `assignment/1dproblem/data/fm_resp_data.mat`.
3. `assignment/2dproblem/code/lookup_fm_resp.m`: given an $\mathbf{x} \in X$ on input, returns the fine model response $\mathbf{f}(\mathbf{x})$. This function reads data from `../data/fm_resp_mat.mat`;
4. `assignment/2dproblem/code/compute_fm_cost.m`: given an $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ on input, returns the fine model cost function $F(\mathbf{x})$.

2 Single Level Optimization

Choose for problem (1) a particular objective $\mathbf{y} \in Y$. Set \mathbf{y} constant and equal to 12, 24 or 48 Newtons, respectively. In MATLAB this corresponds to setting, e.g.,

$$\mathbf{y} = 24 * \text{ones}(9, 1).$$

Use the function `compute_fm_cost.m` to plot the fine model cost function over the design space X . Use a logarithmic scale in the vertical direction (y -axis) to better locate the minimum. Solve the fine model problem using any (stochastic, deterministic, gradient-based or other) optimization algorithm. In the absence of a favorite optimization algorithm, we suggest to use the Nelder-Mead simplex algorithm implemented in MATLAB in the `fminsearch` function. Use the plot of the cost function to check the answer you obtain. Report on the number of iterations and fine model evaluations required to solve the problem.

Repeat this procedure for the problem (2). Use a surface or contour plot to visualize the location of the minima. Use a logarithmic scale as before to sharpen the location of the minimum. Observe that this problem is ill-posed for \mathbf{y} constant and equal to 12, 24 or 48 Newtons in the sense that a set of equivalent minima $\mathbf{x}^* \in X$ exist. Show that the optimization algorithm with different parameters (different starting guesses e.g.) finds different solutions. Report on the number of iterations and fine model evaluations required to solve the problem.

3 Multilevel Optimization using the Manifold-Mapping Technique

For both optimization problems a coarse model is available. The response function in this model represents the magnetic force on the moving part of the device computed using a simplified partial differential equation model (neglecting non-linear effects). The coarse model optimization problem with one and two design variables reads:

find $\mathbf{x}^* \in X = [\mathbf{x}_m, \mathbf{x}_M]$ such that

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{z} \in X} \|\mathbf{c}(\mathbf{z}) - \mathbf{y}\|_2 = \operatorname{argmin}_{\mathbf{z} \in X} C(\mathbf{z}) \quad [1\text{D coarse model problem}] \quad (3)$$

and, find $\mathbf{x}^* \in X = [\mathbf{x}_m(1), \mathbf{x}_M(1)] \times [\mathbf{x}_m(2), \mathbf{x}_M(2)]$,

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{z} \in X} \|\mathbf{c}(\mathbf{z}) - \mathbf{y}\|_2 = \operatorname{argmin}_{\mathbf{z} \in X} C(\mathbf{z}) \quad [2\text{D coarse model problem}], \quad (4)$$

respectively. As before we make the model available in the form of a lookup table.

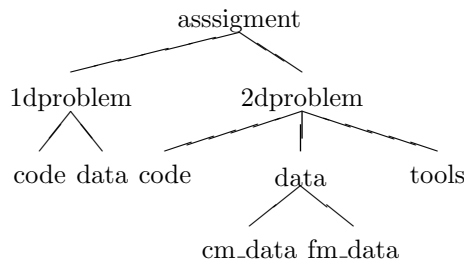


Figure 1: Distribution of files.

For the problem (3), the following functions are available

1. `assignment/1dproblem/data/cm_resp_mat.dat`: an ASCII file that provides a matrix of size 9 by 41 in which each column corresponds to a vector $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^9$, for \mathbf{x} being one of the 41 grid points on X ;
2. `assignment/1dproblem/code/lookup_cm_resp.m`: given an $\mathbf{x} \in X$ on input, returns the coarse model response $\mathbf{c}(\mathbf{x})$;
3. `assignment/1dproblem/code/compute_cm_cost.m`: given an $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ on input, returns the fine model cost function $C(\mathbf{x})$.

For the problem (4), the directory/files

1. `assignment/2dproblem/data/cm_data/`: provides a collection of 41 times 41 ASCII files each containing a vector of size 9 each corresponding to a vector $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^9$, for \mathbf{x} being one of the 41 by 41 grid points on X ;
2. `assignment/2dproblem/tools/build_cm_resp_mat.m`: provides a loop to read all of the 41 by 41 files in `assignment/1dproblem/data/cm_data/` and to construct three-dimensional arrays of dimension 9 by 41 by 41 of fine model responses. This matrix is written as the MATLAB **binary** data file `assignment/1dproblem/data/cm_resp_data.mat`.
3. `assignment/2dproblem/code/lookup_fm_resp.m`: given an $\mathbf{x} \in X$ on input, returns the fine model response $\mathbf{c}(\mathbf{x})$. This function reads data from `../data/cm_resp_mat.mat`;
4. `assignment/2dproblem/code/compute_cm_cost.m`: given an $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ on input, returns the fine model cost function $C(\mathbf{x})$.

Techniques to solve the fine model problem immediately carry over to solve the coarse model problem.

Implement the manifold-mapping algorithm described in Section 5 of [2] and solve the problem (1). In this implementation the functions previously described can be used as building blocks. Use the optimization algorithm previously applied to the fine model problem to solve the coarse model problem at each iteration of the manifold-mapping algorithm. Report on the number of iterations and fine model evaluations manifold-mapping requires to solve the problem. Compare these numbers with those for the optimization algorithm directly applied to the fine model problem. Use the assumptions that the computation of the coarse model response is twenty times cheaper than that of the fine model response to draw conclusions about the computational speed-up that the manifold-mapping algorithm delivers. Do the same for the problem (2).

4 Report

Write a report describing the manifold-mapping algorithm and the numerical results of its application to solving the problems (1) and (2).

References

- [1] D. Echeverría, D. Lahaye, L. Encica, E. A. Lomonova, P. W. Hemker, and A. J. A. Vandeput. Manifold mapping applied to linear actuator design. *IEEE. Trans. on Magn.*, 42(4):1183–1186, 2006.
- [2] D. Echeverría, D. Lahaye, and P. W. Hemker. *Model Order Reduction*, chapter Space Mapping and Defect Correction, pages 107–126. 2008.