One Dimensional Problems

(Dated: December 23, 2015)

CONTENTS

`code: https://github.com/anandpratap/microstructures_1d (private repository)`
`** utilizes symbolic differentiation to generate sigma and beta, all it needs is the expression for energy function and a function for boundary conditions. Can solve for arbitrarily complex energy function.`

I. INTRODUCTION

The governing equation in one dimension is given by,

$$\sigma_x - \beta_{xx} = 0, \tag{1}$$

where,

$$\sigma = \frac{\partial W}{\partial \epsilon}, \tag{2}$$

$$\beta = \frac{\partial W}{\partial \epsilon_x}, \tag{3}$$

where, W is the strain energy density and ϵ is the strain. The governing equation is solved for: (i) a convex energy function which results in a linear relationship; and (ii) a non–convex energy function which leads to non–linear equation resulting in micro-structures. Inverse problem is solved for the linear case to build the energy density function from the displacement field. For the non linear case, the solution contains multiple bifurcation points in the parameter space and thus needs better strategy for inversion.

II. CONVEX ENERGY FUNCTION `EXAMPLES/LINEAR_01`

The equation for the energy function is given by

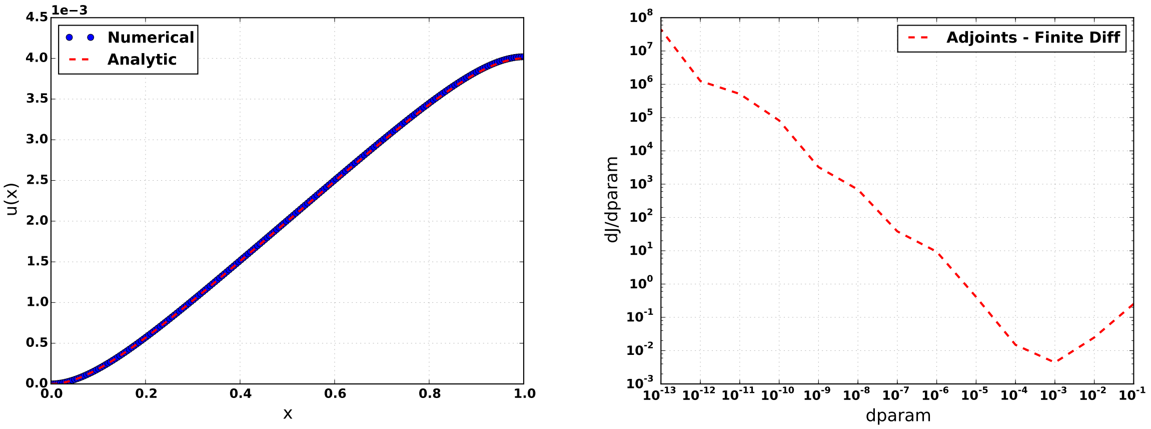
$$W(\epsilon, \epsilon_x) = \frac{1}{2}\mu\epsilon^2 + \frac{1}{2}\mu l^2 \epsilon_x^2. \tag{4}$$

Following boundary conditions are imposed on the displacement field:

$$u_x = 0 \text{ at x = 0 and L}, \tag{5}$$

$$u = 0 \text{ at x = 0}, \tag{6}$$

$$-\mu l^2 u_{xxx} = t \text{ at x = L }. \tag{7}$$



(a) Solution

(b) Difference between the finite difference derivative and the adjoint derivative

FIG. 1: Convex energy function

For this work, I use $\mu = 1.0$, $l = 0.1$, $L = 1$, and $t = 0.005$. The governing equation is solved using a second order finite difference scheme (refer appendix for details). A uniform grid with 200 points is used. Fig. 1a shows the finite difference and the analytic solution. Sensitivities are required for an inversion process, which are calculated using discrete adjoints (refer appendix for details). The difference between the sensitivities from the adjoint and the finite difference is shown in figure 1b. For this case, the parameter β is the multiplier of the length scale l (i.e. $l_{effective} = \beta l$) and the following objective function is used:

$$\mathfrak{J} = 10^4 \left(\sum_j u_j^2 \right) + \beta^2. \tag{8}$$

III. NONLINEAR EXAMPLES/NONLINEAR_02

For this case the following energy function is used:

$$W(\epsilon, \epsilon_x) = \frac{\mu}{\alpha^4}(\epsilon^4 - 2\alpha^2\epsilon^2) + \frac{1}{2}\mu l^2 \epsilon_x^2. \tag{9}$$

At the boundaries, the displacement field satisfies the following relations:

$$u_x = 0 \text{ at x} = 0 \text{ and L} \tag{10}$$

$$u = 0 \text{ at x} = 0 \tag{11}$$

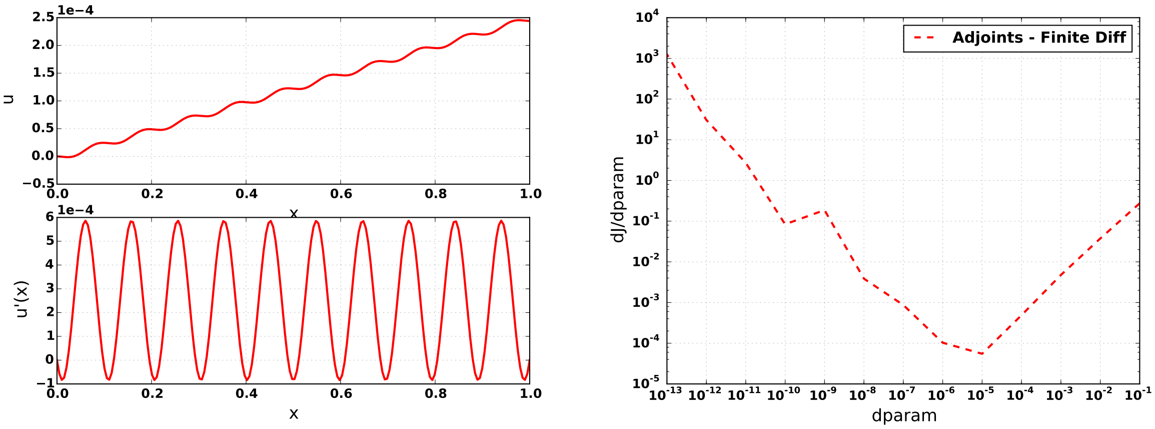
$$u = g \text{ at x} = \text{L} \tag{12}$$

I use $\mu = 1.0$, $g = 2^{-12}$, $l = 1/8$, $\alpha = 1/4$. Problem setup is same as in the previous case. Fig. 2 shows the solution and difference between the adjoint and the finite difference sensitivity.

IV. CONVEX INVERSION EXAMPLES/LINEAR_INVERSE_03

This section demonstrates a simple inversion test case applied to the system with a convex energy function. The benchmark data is generated using $l = 0.1$. Starting from $l = 0.01$ – the prior model – the following objective function is minimized:

$$\mathfrak{J} = \frac{1}{2\sigma_{obs}^2} \sum_j (u_j - u_{obs,j})^2 + \frac{1}{2\sigma_{prior}^2}(l - l_{prior})^2 \tag{13}$$

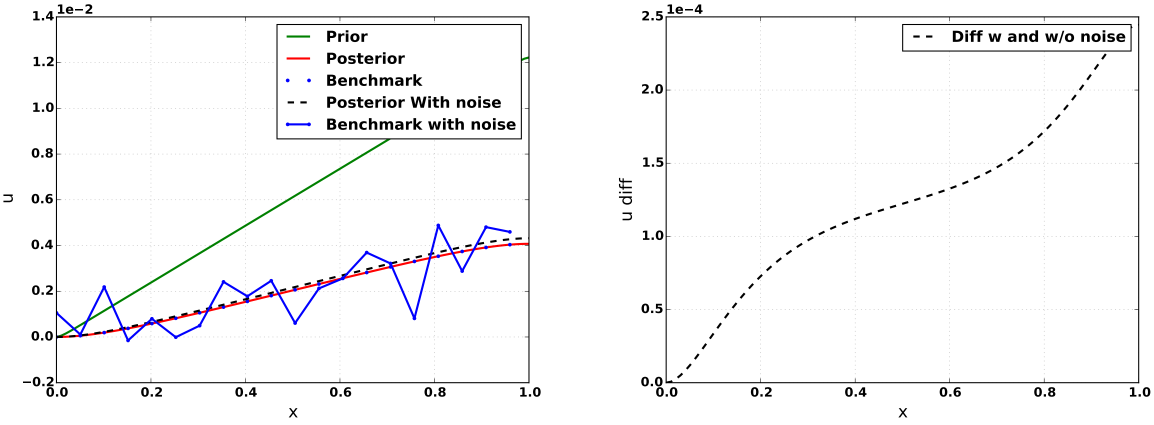


(a) Solution

(b) Difference between the finite difference derivative and the adjoint derivative

FIG. 2: Non Convex energy function

where σ_{obs} and σ_{prior} are the respective observational and the prior standard deviations representing the confidence. $\sigma_{obs} = 10^{-5}$ and $\sigma_{prior} = 1$ is used, the low observational standard deviation represents a high confidence in the data (in fact it should be zero as our data is exact!). A steepest descent algorithm with fixed step size is used to get the posterior model (note that the term *posterior* is used in a loose sense, ideally posterior refers to the probability distribution, the solution which minimizes this objective function is called maximum a posteriori (MAP) solution). Another variation of the inversion with imperfect data is tested. Gaussian error, with standard deviation of 0.001, is introduced in the benchmark. For this variation, σ_{obs} is set to 0.001. Figure 3 shows the prior, posterior and benchmark data, and difference between the solution. For the case with perfect data, the inversion gives $l_{posterior} = 0.1$ (exactly same as the one used for data generation) and with noisy data $l_{posterior} = 0.09$.



(a) Inverse solution

(b) Difference between the displacement with noisy data and perfect data

FIG. 3: Convex energy function inversion

V. NON CONVEX INVERSION `EXAMPLES/NONLINEAR_TESTS_04`

Strategy from the previous section did not work for this case. Fig. 4 shows the variation of the maximum value of the displacement as a function of parameter β (hence l). Multiple bifurcation points exist for this case, therefore *normal* optimization algorithms do not suffice.

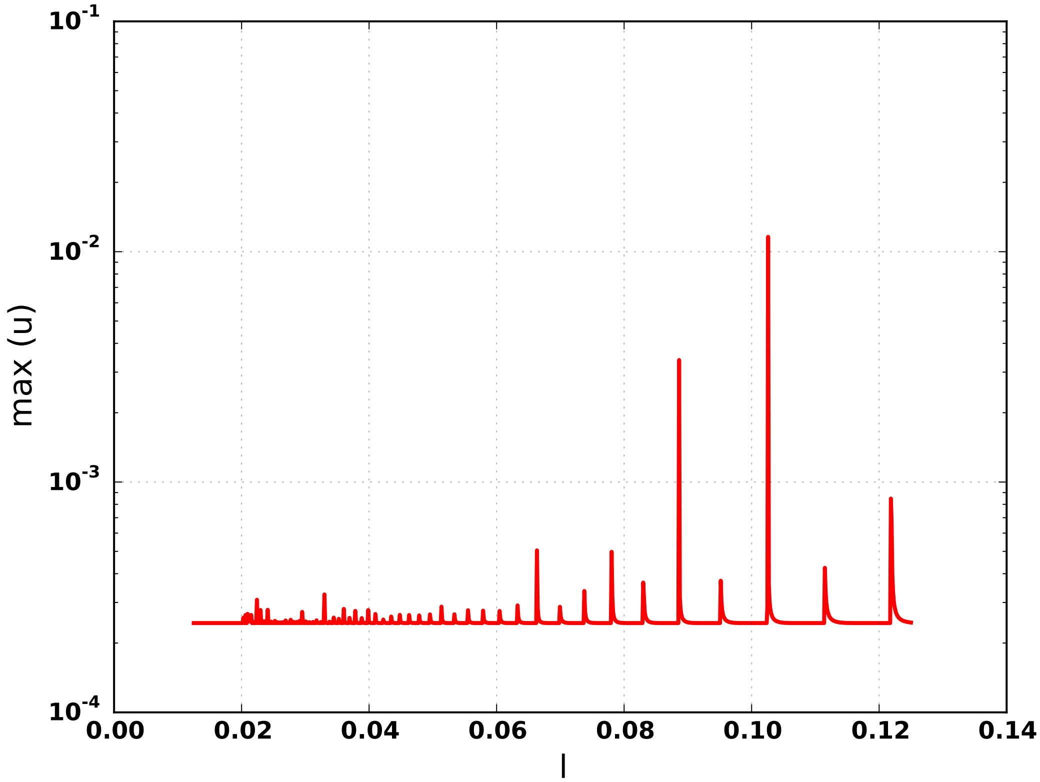


FIG. 4: Maximum displacement (max of u(x)) with variation in l (or β)

VI. NON CONVEX INVERSION USING MCMC `EXAMPLES/NONLINEAR_MCMC_05`

The failure of gradient based methods motivates a need for non gradient based optimization which looks for global minima. For this case, I am using Markov chain Monte Carlo. I generate the data assuming the length scale(l) to be 1 (and call it the Benchmark). I start from a prior $l = N(5, 0.1^2)$, where N represents normal distribution with given mean and standard deviation. Further, since I am using exact data, I set $\sigma_{obs} = 10^{-10}$. For now, I am only interested in obtaining the MAP solution, so I am running for 1000 samples which gives decent result for the MAP length scale, distribution might not be accurate. Results from the inversion are presented in figs 5 - 7. The posterior solution is almost identical to the benchmark, also note (from the posterior distribution) that the MAP solution is almost one, which is the true solution.

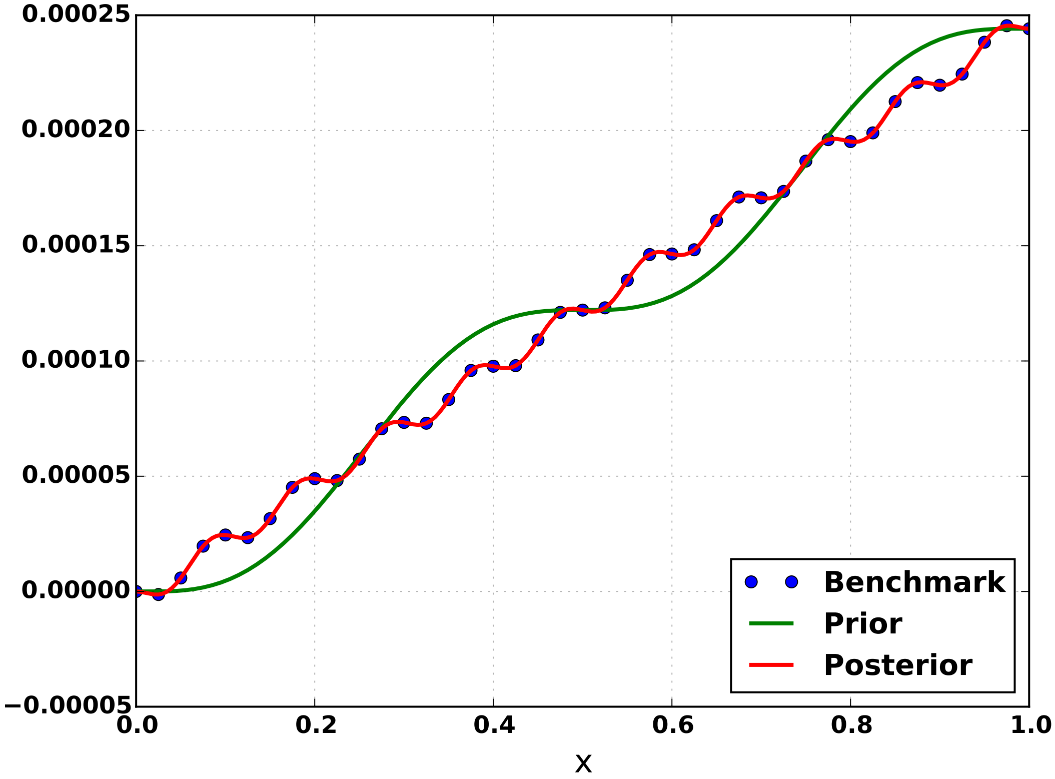


FIG. 5: Solution u(x).

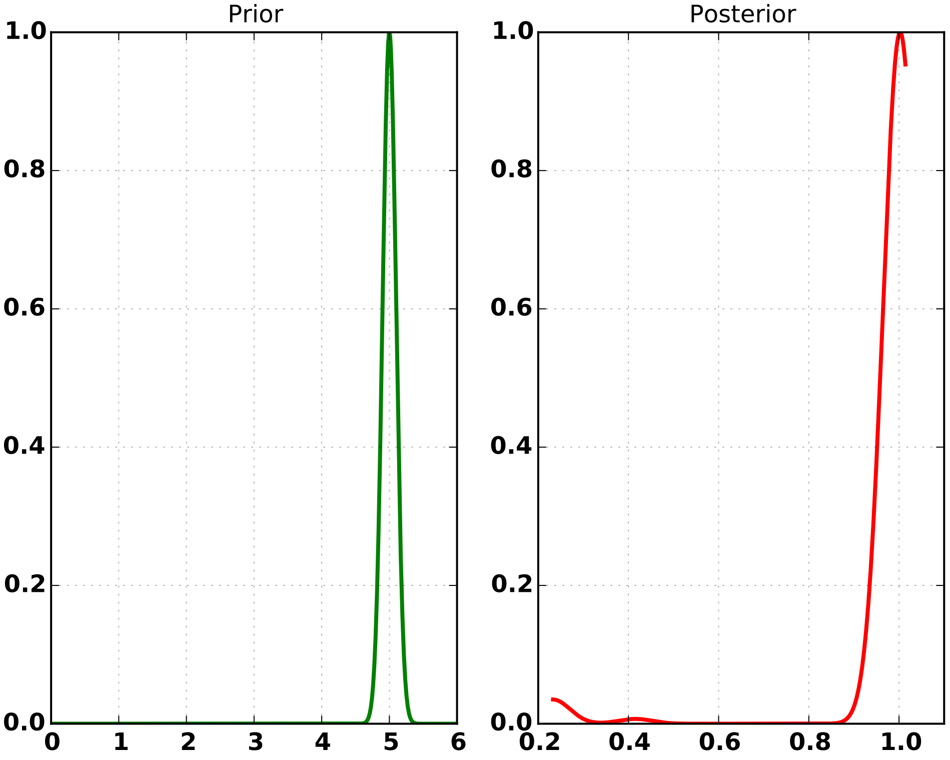


FIG. 6: Normalized prior and posterior probability distributions.

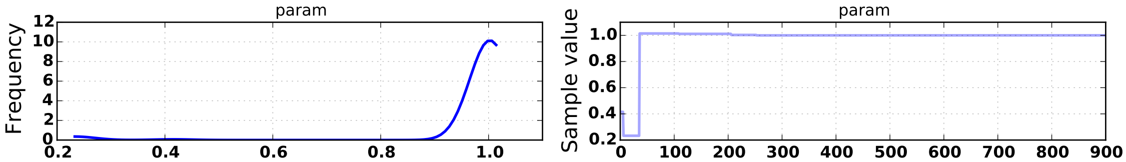


FIG. 7: MCMC evolution.

Appendix A: Implementation

Second order central difference approximation is used to discretize the derivatives at the interior points.

$$u_{xxxx}(i) = \frac{u(i-2) - 4u(i-1) + 6u(i) - 4u(i+1) + u(i+2)}{\Delta x^4} \tag{A1}$$

$$u_{xx}(i) = \frac{u(i-1) - 2u(i) + u(i+1)}{\Delta x^2} \tag{A2}$$

$$u_x(i) = \frac{u(i-1) - u(i+1)}{2\Delta x} \tag{A3}$$

$$\tag{A4}$$

Boundary condition at the second and the second last grid point is imposed by setting $u(i \pm 2)$ to u(i). For the first and last grid point, the residual is specified depending upon the boundary condition. For example, for the linear case,

$$R(1) = u(1), \tag{A5}$$

$$R(n) = -\mu l^2 u_{xxx}(i) + t, \tag{A6}$$

where, $u_{xxx}(i)$ is computed using one sided second order finite difference. For the non linear case,

$$R(1) = u(1), \tag{A7}$$

$$R(n) = u(n) - g. \tag{A8}$$

Then, the problem of $\mathbf{R}(\mathbf{u}) = 0$ is solved using backward Euler scheme.

$$\Delta \mathbf{u} = \left(\frac{\mathbf{I}}{dt} - \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right)^{-1} \mathbf{R}(\mathbf{u}^n) \tag{A9}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta \mathbf{u} \tag{A10}$$

$$\tag{A11}$$

The jacobian is calculated using complex step differentiation.

Discrete adjoints are used to calculate the sensitivity of an objective function, $\mathfrak{J}$, with respect to a parameter β (it can be a vector). The total derivative with respect to β is given by,

$$\frac{d\mathfrak{J}}{d\beta} = \frac{\partial \mathfrak{J}}{\partial \beta} + \psi^T \frac{\partial \mathbf{R}}{\partial \beta}, \tag{A12}$$

where the adjoint variable, ψ, is calculated using,

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right]^T \psi = - \left[\frac{\partial \mathfrak{J}}{\partial \mathbf{u}} \right]^T . \tag{A13}$$

All the partial derivatives are calculated using complex step differentiation.