

Construction of Gaussian Surrogate Process Using Numerical and Modeling Error Uncertainty

H. Aghakhani, A. K. Patra and E. Spiller

SIAM CSE 2015

March 11, 2015

Table of contents

Adjoint Definition and Derivation

Error Estimation

Burger's Equation

Inclined Plane

Adjoint Definition

Let:

- ▶ Let U and V be two vector spaces, and L be a linear operator that maps any $u \in U$ into $v \in V$.
- ▶ $\langle \cdot, \cdot \rangle$ be a bilinear map that maps any two vectors like u, v two a real number, $U \times V \rightarrow \mathbb{R}$.

Then the adjoint operator, L^* , of L is defined:

$$\langle Lu, v \rangle = \langle u, L^*v \rangle.$$

Discrete Adjoint

Assume:

- ▶ $R(U, \alpha)$ as a system of governing equations,
- ▶ U is the solution vector,
- ▶ α is the vector of design parameters.

The object is to minimize $J(U, \alpha)$ subject to $R(U, \alpha) = 0$

Discrete Adjoint Cont.

With writing the variation of the functional and governing equation w.r.t design parameters, we will have:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial U} \frac{dU}{d\alpha} + \frac{\partial J}{\partial \alpha}$$

and:

Discrete Adjoint Cont.

With writing the variation of the functional and governing equation w.r.t design parameters, we will have:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial U} \frac{dU}{d\alpha} + \frac{\partial J}{\partial \alpha}$$

and:

$$\frac{\partial R}{\partial U} \frac{dU}{d\alpha} + \frac{\partial R}{\partial \alpha} = 0.$$

Discrete Adjoint Cont.

With writing the variation of the functional and governing equation w.r.t design parameters, we will have:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial U} \frac{dU}{d\alpha} + \frac{\partial J}{\partial \alpha}$$

and:

$$\frac{\partial R}{\partial U} \frac{dU}{d\alpha} + \frac{\partial R}{\partial \alpha} = 0.$$

Now, if we replace $\frac{dU}{d\alpha}$ from the second equation into the first equation, we have:

$$\frac{dJ}{d\alpha} = -\frac{\partial J}{\partial U} \left(\frac{\partial R}{\partial U} \right)^{-1} \frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha}$$

Discrete Adjoint Cont.

With writing the variation of the functional and governing equation w.r.t design parameters, we will have:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial U} \frac{dU}{d\alpha} + \frac{\partial J}{\partial \alpha}$$

and:

$$\frac{\partial R}{\partial U} \frac{dU}{d\alpha} + \frac{\partial R}{\partial \alpha} = 0.$$

Now, if we replace $\frac{dU}{d\alpha}$ from the second equation into the first equation, we have:

$$\frac{dJ}{d\alpha} = -\frac{\partial J}{\partial U} \left(\frac{\partial R}{\partial U} \right)^{-1} \frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha}$$

Sensitivity Computation

Assume that n is the size of vector U , and m is the size of vector α :

$$dJ_{scalar} = - \underbrace{\left[\frac{\partial J}{\partial U} \right]_{1 \times n} \left[\frac{\partial R^{-1}}{\partial U} \right]_{n \times n} \left[\frac{\partial R}{\partial \alpha} \right]_{n \times m}}_{\text{scalar}} d\alpha_{m \times 1} + \underbrace{\left[\frac{\partial J}{\partial \alpha} \right]_{1 \times m}}_{\text{scalar}} d\alpha_{m \times 1}$$

Sensitivity Computation

Assume that n is the size of vector U , and m is the size of vector α :

$$dJ_{\text{scalar}} = - \underbrace{\left[\frac{\partial J}{\partial U} \right]_{1 \times n} \left[\frac{\partial R^{-1}}{\partial U} \right]_{n \times n} \left[\frac{\partial R}{\partial \alpha} \right]_{n \times m}}_{\text{scalar}} d\alpha_{m \times 1} + \underbrace{\left[\frac{\partial J}{\partial \alpha} \right]_{1 \times m}}_{\text{scalar}} d\alpha_{m \times 1}$$

Above sensitivity can be computed in two ways:

1. Forward mode: first computes $(\frac{\partial R}{\partial U})^{-1} \frac{\partial R}{\partial \alpha}$

Sensitivity Computation

Assume that n is the size of vector U , and m is the size of vector α :

$$dJ_{\text{scalar}} = - \underbrace{\left[\frac{\partial J}{\partial U} \right]_{1 \times n} \left[\frac{\partial R^{-1}}{\partial U} \right]_{n \times n} \left[\frac{\partial R}{\partial \alpha} \right]_{n \times m}}_{\text{scalar}} d\alpha_{m \times 1} + \underbrace{\left[\frac{\partial J}{\partial \alpha} \right]_{1 \times m}}_{\text{scalar}} d\alpha_{m \times 1}$$

Above sensitivity can be computed in two ways:

1. Forward mode: first computes $(\frac{\partial R}{\partial U})^{-1} \frac{\partial R}{\partial \alpha}$
2. Adjoint mode: first computes $\frac{\partial J}{\partial U} (\frac{\partial R}{\partial U})^{-1} \rightarrow (\frac{\partial R}{\partial U})^T v = \frac{\partial J}{\partial U}^T$, v is adjoint solution

Adjoint definition

Sensitivity Computation

Assume that n is the size of vector U , and m is the size of vector α :

$$dJ_{\text{scalar}} = - \underbrace{\left[\frac{\partial J}{\partial U} \right]_{1 \times n} \left[\frac{\partial R^{-1}}{\partial U} \right]_{n \times n} \left[\frac{\partial R}{\partial \alpha} \right]_{n \times m}}_{\text{scalar}} d\alpha_{m \times 1} + \underbrace{\left[\frac{\partial J}{\partial \alpha} \right]_{1 \times m}}_{\text{scalar}} d\alpha_{m \times 1}$$

Above sensitivity can be computed in two ways:

1. Forward mode: first computes $(\frac{\partial R}{\partial U})^{-1} \frac{\partial R}{\partial \alpha}$
2. Adjoint mode: first computes $\frac{\partial J}{\partial U} (\frac{\partial R}{\partial U})^{-1} \rightarrow (\frac{\partial R}{\partial U})^T v = \frac{\partial J}{\partial U}^T$, v is adjoint solution

Adjoint definition

Advantage of Adjoint:

If the number of design parameters are more than the objective functionals, then the computational cost of the adjoint is much lower than the forward method.

Sensitivity Computation

Assume that n is the size of vector U , and m is the size of vector α :

$$dJ_{\text{scalar}} = - \underbrace{\left[\frac{\partial J}{\partial U} \right]_{1 \times n} \left[\frac{\partial R^{-1}}{\partial U} \right]_{n \times n} \left[\frac{\partial R}{\partial \alpha} \right]_{n \times m}}_{\text{scalar}} d\alpha_{m \times 1} + \underbrace{\left[\frac{\partial J}{\partial \alpha} \right]_{1 \times m}}_{\text{scalar}} d\alpha_{m \times 1}$$

Above sensitivity can be computed in two ways:

1. Forward mode: first computes $(\frac{\partial R}{\partial U})^{-1} \frac{\partial R}{\partial \alpha}$
2. Adjoint mode: first computes $\frac{\partial J}{\partial U} (\frac{\partial R}{\partial U})^{-1} \rightarrow (\frac{\partial R}{\partial U})^T v = \frac{\partial J}{\partial U}^T$, v is adjoint solution

Adjoint definition

Advantage of Adjoint:

If the number of design parameters are more than the objective functionals, then the computational cost of the adjoint is much lower than the forward method.

Computing Adjoint

To solve adjoint equation we need Transpose of Jacobian Matrix:

$$\left(\frac{\partial R}{\partial U} \right)^T v = \frac{\partial J}{\partial U}$$

TITAN2D uses Godunov finite volume with Euler explicit time scheme, so the discretized form of equations are:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \{ F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n \} - \frac{\Delta t}{\Delta y} \{ G_{i+\frac{1}{2}}^n - G_{i-\frac{1}{2}}^n \}$$

$$\left(\frac{\partial R}{\partial U} \right)_{m \times m}^T = K_{ij}$$

where m is the number of time steps, and each K_{ij} is a

Computing Adjoint Cont.

For Euler explicit:

$$K_{i,i} = I \quad \text{and} \quad K_{i,i+1} = \left(\frac{\partial R_p^{i+1}}{\partial U_q^i} \right)^T,$$

- ▶ p and q are degrees of freedom
- ▶ rest of the components are zero
- ▶ depend on the stencil is used. K matrices are also block bounded

$$\left(\frac{\partial R}{\partial U} \right)_{m \times m}^T = \begin{pmatrix} I & K_{1,2} & & & \\ & I & K_{2,3} & 0 & \\ & & \ddots & \ddots & \\ 0 & & & I & K_{m-1,m} \\ & & & & I \end{pmatrix}$$

Computing Adjoint Cont.

Important conclusion:

To compute adjoint for an explicit system, there is no need to solve a system of equation, and adjoint solution can be found marching backward in time.

$$\begin{aligned}v_1 + K_{1,2}v_2 &= \left(\frac{\partial J}{\partial U}\right)_1^T \\&\vdots \\v_{m-1} + K_{m-1,m}v_m &= \left(\frac{\partial J}{\partial U}\right)_{m-1}^T \\v_m &= \left(\frac{\partial J}{\partial U}\right)_m^T\end{aligned}$$

Error Estimation

The goal is to minimize the numerical error due to mesh size on the objective functional $J(Q)$, given the solution on the coarse mesh $J(Q_H)$.

With Taylor expansion we can write¹:

$$J(Q_h) \approx J(Q_h^H) - \underbrace{(\psi_h^H)^T R(Q_h^H)}_{\text{Adjoint correction term}} - \underbrace{(\psi_h - \psi_h^H)^T R(Q_h^H)}_{\text{Remaining term}},$$

where $J(Q_h)$ is the functional value on a finer mesh, and all \square_h^H is the projection of \square from the coarse mesh to the fine mesh.

¹Marian Nemec, MJ Aftosmis, and Mathias Wintzer. "Adjoint-based adaptive mesh refinement for complex geometries". In: *AIAA Paper* (2008), pp. 1–23. URL: http://people.nas.nasa.gov/~nemec/MYWeb/aiaa%5C_2008%5C_0725%5C_small.pdf.

Error Estimation Cont.

So to compute the error $\varepsilon = |J(Q) - J(Q_H)|$:

- ▶ we have to first find ψ_h .
- ▶ since computing ψ_h is not reasonable to just compute the error we approximate it with higher order construction.

Consequently:

$$J(Q_h) \approx J(Q_L) - (\psi_L)^T R(Q_L) - (\psi_L - \psi_C)^T R(Q_L),$$

where \square_L , \square_C represent linear and constant reconstruction respectively.

Case 1: Burger's Equation

$$R(x, t) = \frac{\partial u}{\partial t} + \frac{\alpha}{2} \frac{\partial u^2}{\partial x} = 0, \quad x \in (-1, 1), \quad t \in (0, 1),$$

$$u(x, 0) = \beta \cos\left(\frac{\pi}{2}x\right),$$

$$u(1, t) = 0,$$

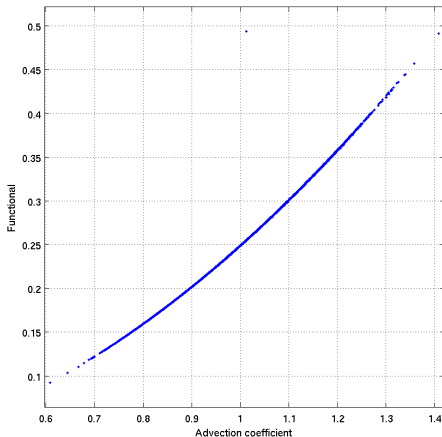
$$J = 0.5 \int_T \int_x u^2 \, dx \, dt,$$

where α and β are uncertain parameters, and are selected from $\mathcal{N}(1, 0.1)$.

Case 1: Burger's Equation

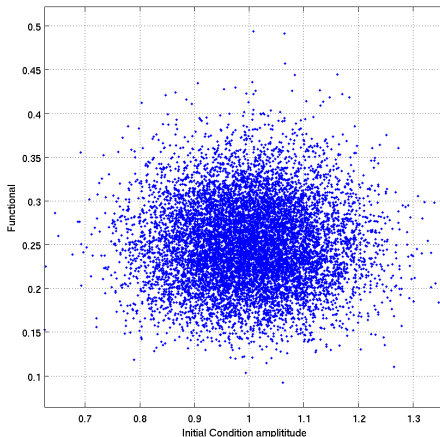
Results are for a Monte Carlo simulation with 10,000 samples:

Functional of interest as a function of advection coefficient



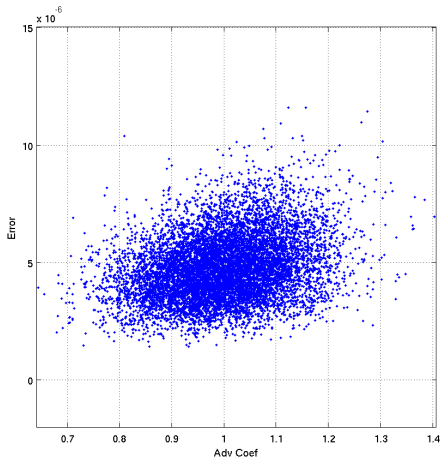
Case 1: Burger's Equation

Functional of interest as a function of initial condition uncertain coefficient



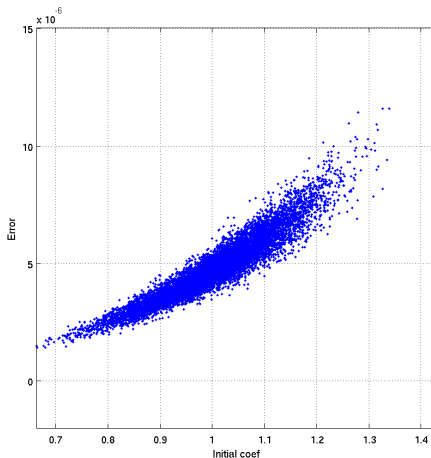
Case 1: Burger's Equation

Error at final stage as a function of advection coefficient



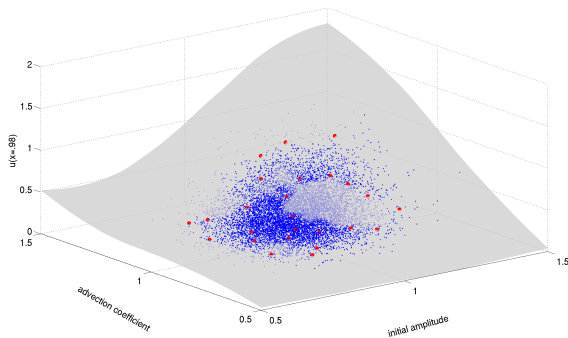
Case 1: Burger's Equation

Error at final stage as a function of initial condition uncertain coefficient



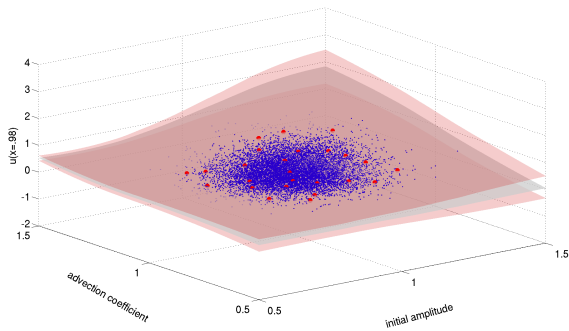
Case 1: Burger's Equation

Emulator result



Case 1: Burger's Equation

Emulator result + error



Governing Equations

$$U_t + F(U)_x + G(U)_y = S(U)$$

Where:

$$U = (h, hv_x, hv_y)^T$$

$$F = (hv_x, hv_x^2 + 0.5k_{ap}g_z h^2, hv_x hv_y)^T$$

$$G = (hv_y, hv_x v_y, hv_y^2 + 0.5k_{ap}g_z h^2)^T$$

Governing Equations Cont.

$$S = (0, S_x, S_y)^T$$

$$S_x = g_x h - \frac{V_x}{\sqrt{V_x^2 + V_y^2}} \left(g_z h + \frac{h V_x^2}{r_x} \right) \tan(\phi_{bed})$$

$$- h k_{ap} \operatorname{sgn} \left(\frac{\partial V_x}{\partial y} \right) \frac{\partial (g_z h)}{\partial y} \sin(\phi_{int})$$

$$S_y = g_y h - \frac{V_y}{\sqrt{V_x^2 + V_y^2}} \left(g_z h + \frac{h V_y^2}{r_y} \right) \tan(\phi_{bed})$$

$$- h k_{ap} \operatorname{sgn} \left(\frac{\partial V_y}{\partial x} \right) \frac{\partial (g_z h)}{\partial x} \sin(\phi_{int})$$

Case 2: Result

For this case, we considered an inclined plane with the uncertain parameters of:

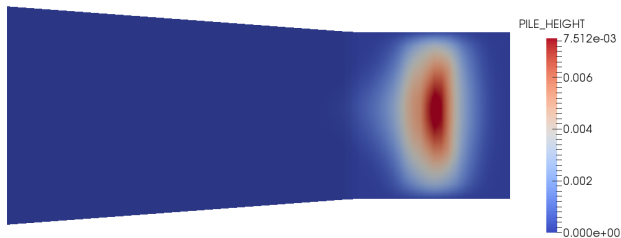
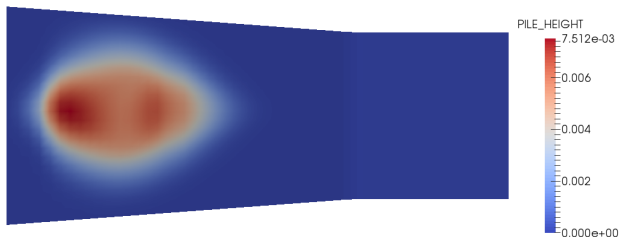
- ▶ Initial Volume $\pm 30\%$ of $2.64 \times 10^{-4} m^3$
- ▶ Bed Friction angle $(23^\circ, 40^\circ)$

we simulated this process with 256 Latin Hyper cube samples. The functional that we are interested in this case is:

$$J = 0.5 \int_T \int_\Omega h^2 dX dt$$

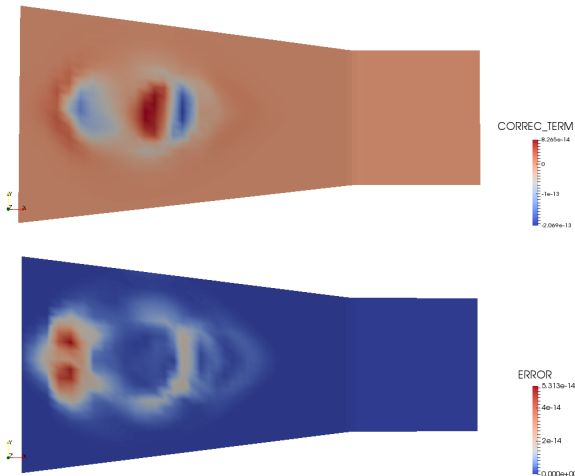
Case 2: Result

Pile height result for $t = 0.3$ sec and $t = 1$ sec



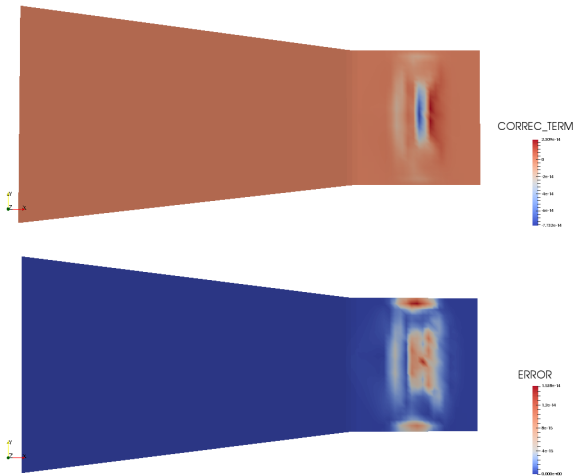
Case 2: Result

Correction term and Error for incline at $t = 0.3$ sec



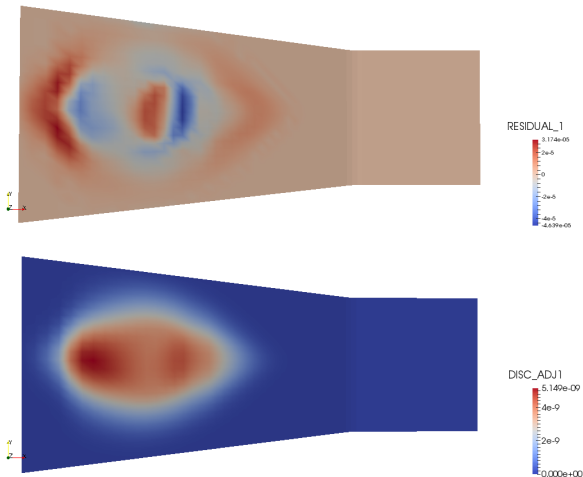
Case 2: Result

Correction term and Error for incline at $t = 1$ sec



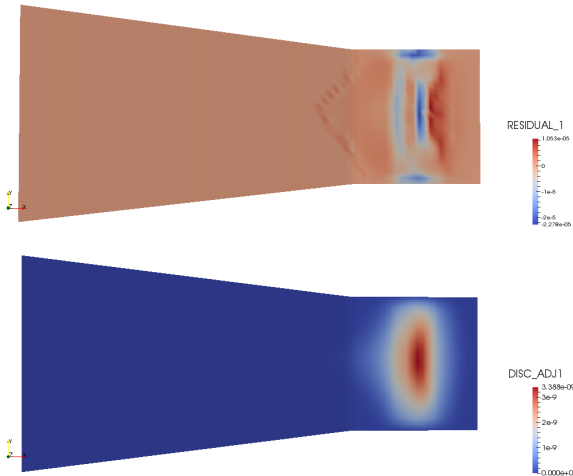
Case 2: Result

Residual and Adjoint result at $t = 0.3$ sec



Case 2: Result

Residual and Adjoint result at $t = 1$ sec



Connection to Adjoint definition

$$\begin{aligned} u &= \frac{dU}{d\alpha}, & A &= \frac{\partial R}{\partial U} \\ g^T &= \frac{\partial J}{\partial U}, & f &= -\frac{\partial R}{\partial \alpha} \end{aligned} \quad (1)$$

Forward method:

$$\begin{aligned} \frac{dJ}{d\alpha} &= g^T u + \frac{\partial J}{\partial \alpha} \\ \text{Subject to } Au &= f \end{aligned} \quad (2)$$

Adjoint Method:

$$\begin{aligned} \frac{dJ}{d\alpha} &= v^T f + \frac{\partial J}{\partial \alpha} \\ \text{Subject to } A^T v &= g \end{aligned} \quad (3)$$

Back to [Connection adjoint definition](#).