# Bayesian inference with partial differential equations using Stan

**Yi Zhang[1], William R. Gillespie[1], Minjie Zhu[2]**

[1] Metrum Research Group  [2] School of Civil and Construction Engineering, Oregon State University

- ***How to apply Stan to problems involving PDEs?***
- ***How to compute QoI sensitivities?***

## Method 1: user-defined Stan function

- Simple PDE problems(1D or 2D problems on a regular domain).
- Example: Stan function for 1D wave equation with wave speed as parameter `theta`, using central difference scheme for space and time.

```
real[,] wave_1d(real theta, int n, real[] u0, real h, int nt, real k) {
  real sol[nt, n];
  real u[n];                /* u at time i */
  real u_prev[n];           /* u at time i-1 */
  real u_next[n];           /* u at time i+1 */
  real c;
  c = theta * k / h; /* CFL number is based on wave speed theta */
  u = u0; u_prev[1] = 0; u_prev[n] = 0; /* initial conditions */
  for (j in 2:n-1) u_prev[j] = u[j] + 0.5 * c^2 * (u[j+1] - 2*u[j] + u[j-1]);
  for (i in 1:nt) {for (j in 2:n-1) u_next[j] = 2 * u[j] - u_prev[j] + c^2 * (u[j+1] - 2 * u[j] + u[j-1]);
    u_next[1] = 0; u_next[n] = 0; /* update boundary conditions */
    u_prev = u; sol[i, ] = u; u = u_next; }
  return sol; }
```

## Method 2: user-defined external C++ function

- Utilizing Stan interface to external C++ functions.

$$\frac{du}{dt} = \mathcal{L}(u,x,t) \rightarrow \frac{d\mathbf{u}}{dt} = A\mathbf{u} \rightarrow \texttt{stan::math::integrate\_ode\_xxx} \qquad (1)$$

## Method 3: interface to external PDE libraries

```
functions {
  real[,] solve_with_sensitivity(real[] theta);
  real[,] solve(real[] theta);
  real[,] pde_model(real[] theta, int need_sens, real[] x_r, int[] x_i){
    if(need_sens)
      return solve_with_sensitivity(theta);
    else
      return solve(theta);}}
/* ... */
QoI = forward_pde(pde_model, k, x_r, x_i);
```

- `solve_with_sensitivity`: user-supplied function that maps PDE parameter $\theta$ to *quantity of interest* (QoI) $\widetilde{\mathcal{J}}(\theta) = \mathcal{J}(u;\theta)$ and sensitivity $d\widetilde{\mathcal{J}}/d\theta$.
- `forward_pde` : interface function(type conversion, resource management, etc).

## Example 1: Laplace equation

- LibMesh [KPSC06]: QUAD9 elements with AMR.

$$\nabla^2[(\theta_1 + 2\theta_2)u] = 0, \quad \forall x \in \Omega_0 \subset \mathbb{R}^2. \quad \rightarrow \quad \mathcal{R}(u,v;\theta) = 0, \quad \forall v \in V \qquad (2)$$

$$\widetilde{\mathcal{J}}(\theta) = \mathcal{J}(u(\theta)) = \int_\Omega u(x;\theta), \qquad (3)$$

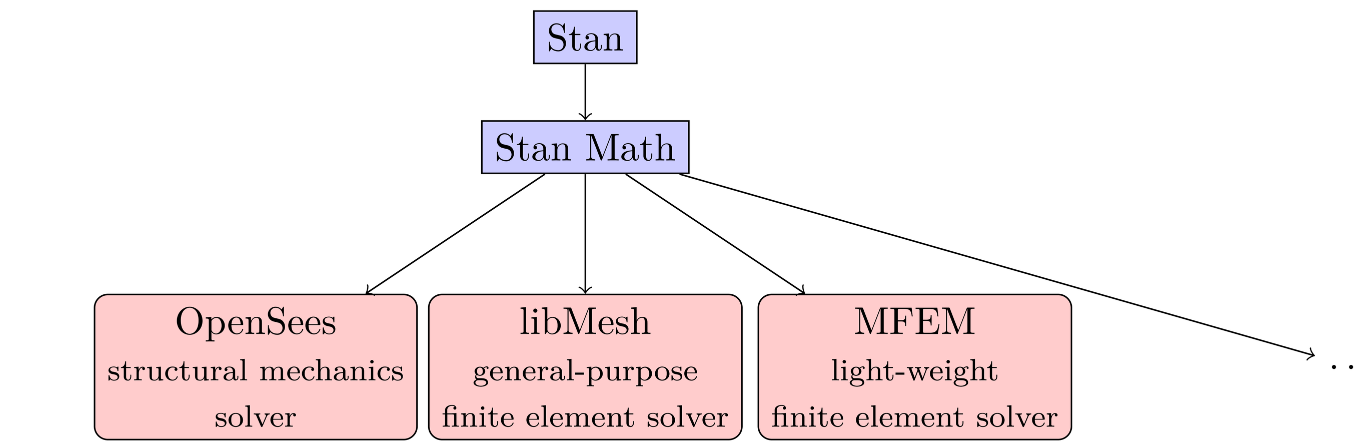
$$\hat{\mathcal{J}} = \widetilde{\mathcal{J}}(\theta) + \mathcal{N}(0,\sigma^2). \qquad (4)$$

- Sensitivity solution: discrete adjoint method

$$\frac{d\mathcal{R}}{d\theta} = \mathcal{R}_\theta + \mathcal{R}_u u_\theta = 0, \qquad (5)$$

$$\mathcal{R}_u(u,u^*;\theta) = \mathcal{J}_u(u;\theta), \qquad (6)$$

$$\frac{d\widetilde{\mathcal{J}}}{d\theta} = \mathcal{J}_\theta + \mathcal{J}_u u_\theta = \mathcal{J}_\theta + \mathcal{R}_u(u,u^*;\theta)u_\theta = \mathcal{J}_\theta + \mathcal{R}_\theta(u,u^*;\theta). \qquad (7)$$



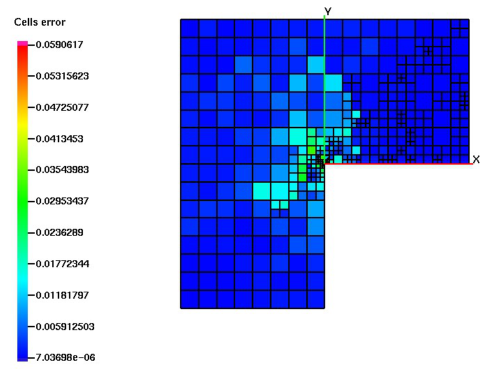




Figure 1: $u$ error

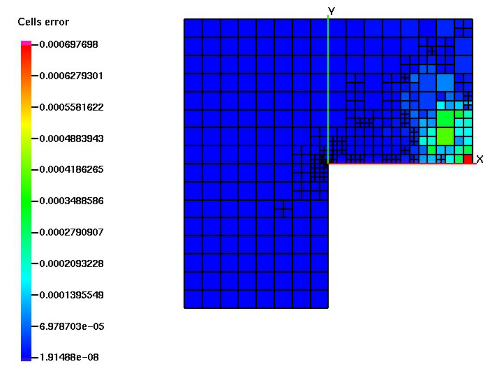
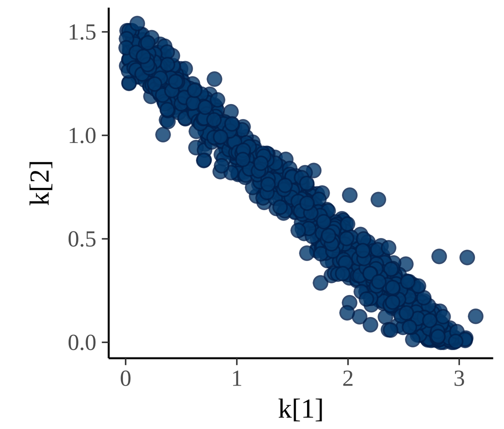
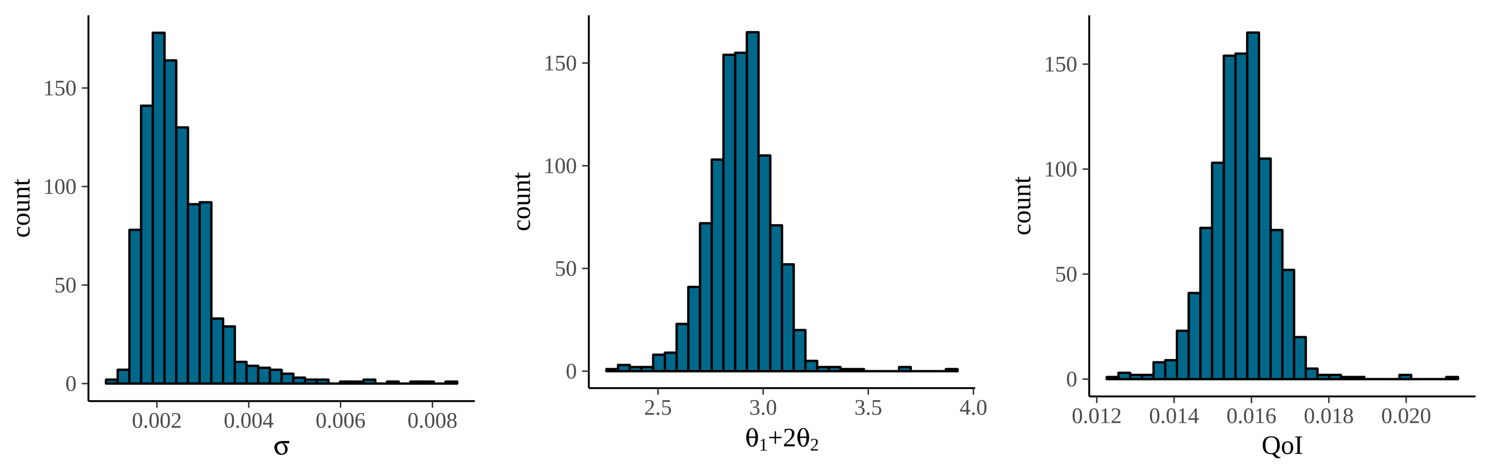



Figure 2: $u^*$ error




## Example 2: Darcy's flow

- MFEM [mfe]: Raviart-Thomas for $\vec{u}$ and $P_0$ for $p$.
- Sensitivity solution: finite difference method

$$\begin{bmatrix} k & \nabla \\ \nabla\cdot & 0 \end{bmatrix}\begin{bmatrix} \vec{u} \\ p \end{bmatrix} = \begin{bmatrix} \vec{f} \\ g \end{bmatrix} \quad \longrightarrow \quad \begin{bmatrix} A(k) & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \qquad (8)$$

$$\widetilde{\mathcal{J}}(\theta) = \mathcal{J}(u(\theta)) = \|u_h - u_0\|_2, \qquad \hat{\mathcal{J}} = \widetilde{\mathcal{J}}(\theta) + \mathcal{N}(0, 0.01^2). \qquad (9)$$

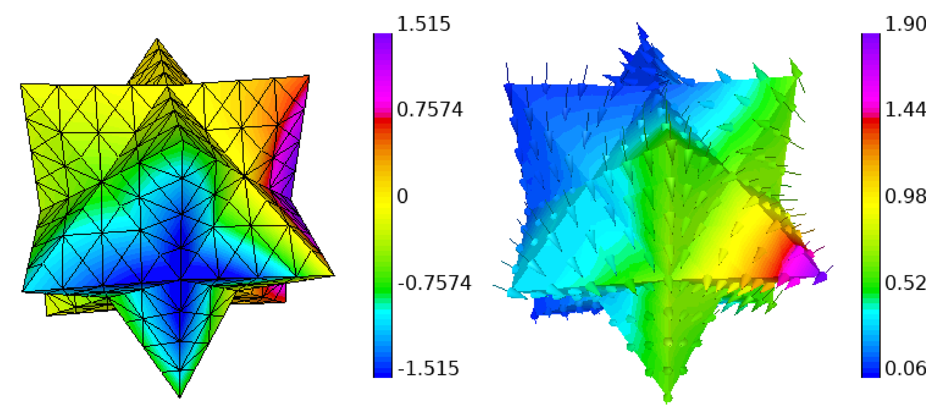
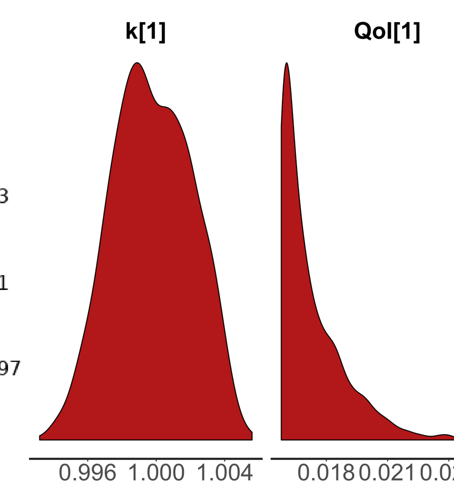


Figure 3: $p$  Figure 4: $\vec{u}$




## Example 3: nonlinear beam

- Bayesian calibration of mechanical material property.
- OpenSees [MSF10]: Displacement-Based Beam-Column Element + corrotational transformation
- Sensitivity solution: direct differentiation method(DDM) [CVM03]

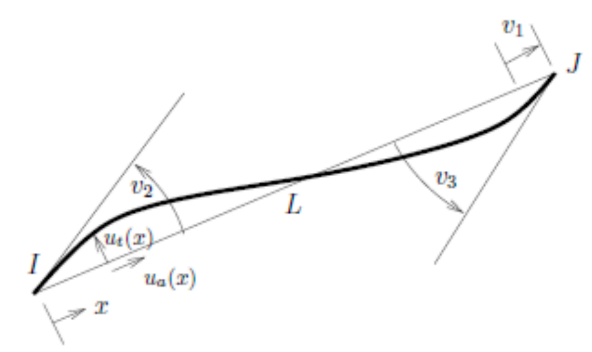
$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \rightarrow \begin{bmatrix} \epsilon_a(x) \\ \kappa(x) \end{bmatrix} = Bv \rightarrow \{\epsilon_i\} \rightarrow \sigma_i = \sigma_i(\epsilon_i; E) \rightarrow p(x_i) = \begin{bmatrix} N(x_i) \\ M(x_i) \end{bmatrix} \rightarrow \int_0^L B^t(x)p(x)dx$$

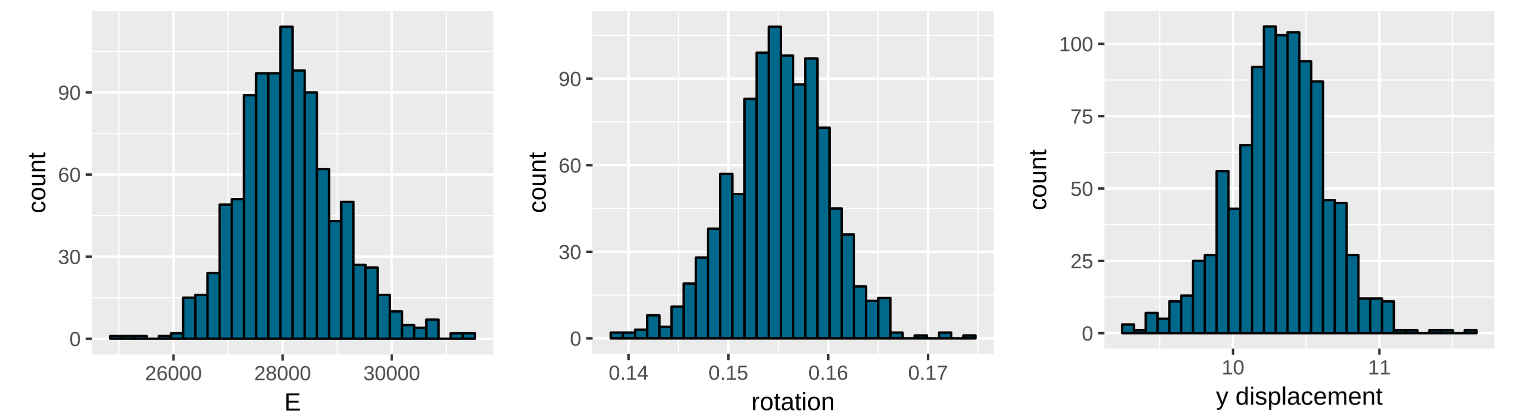
- Inference of Young's modulus $E$ from rotation $v_3$ of a cantilever beam.

$$\hat{\mathcal{J}} = v_3(E) + \mathcal{N}(0,\sigma^2). \qquad (10)$$

- Use $E$ to predict the $y$ displacement $v_2$ of another beam.

$$s = dv/d\theta, \quad \theta = E,$$
$$M(\theta)\ddot{s}(t,\theta) + C(\theta)\dot{s}(t,\theta) + \widetilde{K}_T(v(t,\theta))s(t,\theta) = \frac{\partial F}{\partial\theta} - \frac{\partial M}{\partial\theta}\ddot{v}(t,\theta) - \frac{\partial C}{\partial\theta}\dot{v}(t,\theta) - \frac{\partial R(u(t,\theta),\theta)}{\partial\theta}\Big|_v$$

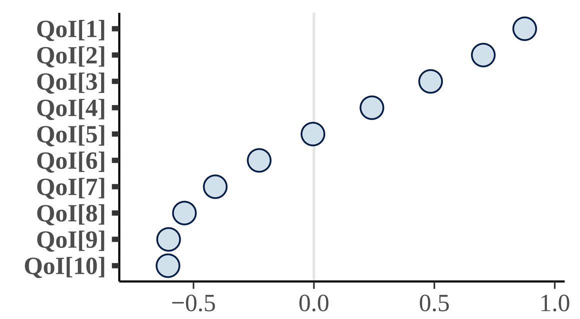



## Example 4: heat equation

- FEM2D_HEAT [Bur]:
- sensitivity solution: complex step derivative approximation [LM67].

$$\frac{du}{dt} - \nabla^2 u(x,t) = f(x,t;\theta) \text{ on } [0,1]\times[0,1],$$
$$u_{\text{exact}} = \cos(2\pi t)\sin(\pi x)\sin(\pi y)e^{-\theta t},$$
$$\hat{\mathcal{J}}(t_i) = u(x_0,t_i) + \mathcal{N}(0,\sigma^2).$$




```
real (kind = c_double), intent(in) :: theta_r ! parameter
real (kind = 8), parameter :: h = 1.D-20;     ! complex step
complex ( kind = 8 ) :: theta                 ! complex parameter
theta = cmplx(theta_r, h, kind = 8)
! ...
u_sensitivity = aimag(u)/h                    ! complex step derivative approximation
```

## What's next?

- Inverse problem using GP emulation [CMMY91] /ROM [ZY18].
- Biomedical applications such as tumor growth [XVG16].

## References

[Bur] John Burkardt. FEM2D HEAT: Finite element solution of the heat equation on a triangulated region. https://people.sc.fsu.edu/~jburkardt/f_src/fem2d_heat/fem2d_heat.html.

[CMMY91] Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments. *Journal of the American Statistical Association*, 86(416):953, 1991.

[CVM03] J. P. Conte, P. K. Vijalapura, and M. Meghella. Consistent Finite-Element Response Sensitivity Analysis. *Journal of Engineering Mechanics*, 129(12):1380–1393, 2003.

[KPSC06] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. `libMesh`: A C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3–4):237–254, 2006.

[LM67] J. N. Lyness and C. B. Moler. Numerical Differentiation of Analytic Functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, 1967.

[mfe] MFEM: Modular finite element methods library. mfem.org.

[MSF10] Frank McKenna, Michael H. Scott, and Gregory L. Fenves. Nonlinear Finite-Element Analysis Software Architecture Using Object Composition. *Journal of Computing in Civil Engineering*, 24(1):95–107, 2010.

[XVG16] Jiangping Xu, Guillermo Vilanova, and Hector Gomez. A Mathematical Model Coupling Tumor Growth and Angiogenesis. *PLOS ONE*, 11(2), 2016.

[ZY18] Yi Zhang and Solomon C. Yim. Low-Dimensional components of flows with large Free/MovingSurface motion. *Journal of Offshore Mechanics and Arctic Engineering*, to appear, 2018.

yiz@metrumrg.com

The first and second author would like to thank Office of Naval Research for funding this research under grant N6833518C0110.