Reliability, sensitivity, and optimization-enabling technologies

Kevin Mackie – Central Florida
Michael Scott – Oregon State
Introduction

- OpenSees reaching maturity for dynamic, nonlinear, coupled SFSI simulation
- Research emphasis on PBEE evaluation of transportation network components

Challenges

- Sensitivity to model and design parameters, particularly for nonlinear response
- Reliability/risk assessment under prevailing uncertainties
- Extend design optimization to include more complex objectives
Illustrations

♦ Sensitivity
  ■ E.g., sensitivity of nodal displacements to constitutive model parameters:
    \[ \frac{\partial u}{\partial \Theta} \]

♦ Reliability
  ■ E.g., probability of nodal displacement exceeding threshold:
    \[ p_f = \int_\Omega f(\Theta)d\Theta \quad \Omega = \{ g(\Theta) \leq 0 \} \]
    \[ \Omega = \bigcup_{k} \bigcap_{j \in C_k} \{ g_j(\Theta) \leq 0 \} \]

♦ Optimization
  ■ E.g., structural properties minimizing nodal displacement \( p_f \):
    \[ y = \arg\min \left\{ p_f(\Theta) \left| \theta_1 > 0, a < \theta_2 < b, h > 2L \right. \right\} \]
Research Needs

- Development, maintenance, and extensibility of OpenSees framework
- Requires software abstractions for reliability/sensitivity/optimization consistent with OO design
- Interaction with FE and analysis domain
- Open-source and OO paradigm that will allow end-user contribution/extension
  - No code duplication
  - No changes to abstract and base classes
- Build on previous work (Der Kiureghian, Haukaas, etc.)
Proposed changes

- Correcting existing modules not functioning
  - Roadmap for re-implementing deprecated modules

- Syntax overhaul
  - Separation of Tcl implementation of LSF from base class

- Parameterization

- Gradient, Hessian, and LSF redesign

- Analysis module class hierarchy (e.g., findDesignPoint), data storage, others ...

- Coordinating with Conte/UCSD optimization framework
Parameterization

- Abstract Parameter class
- Concrete parameters can be positioned in FE, reliability, optimization, etc. domains
  - Pointers to FE model parameters, as well as FE response
  - Random variable and design variable parameters
  - Pointers to Reliability model (e.g., distribution parameters), as well as reliability results (e.g., pf)
  - Pointers to User/Scripting language variables
Syntax, Non-Tcl, etc.

- **PerformanceFunction** abstract or container class
  - LimitStateFunction, ObjectiveFunction, ConstraintFunction, GradientFunction are derived
  - Interfaces directly with FunctionEvaluator

- **FunctionEvaluators** no longer perform parsing
  - TclFun, MatlabFun, PythonFun, etc...

- For example, a Tcl PerformanceFunction:
  - limitStateFunction 3 "1.0+sqrt(\$par($parTag))"

- Parameter will dictate evaluator and gradient actions
Gradients

- No parsing in gradient classes
- No difference between OpenSeesGradG, FiniteDifferenceGradG now
- Only implicit and explicit gradients, as defined by parameters
- Option for analytic gradient, specified in GradientFunction
- Need similar abstraction for Hessian
Schedule

Phase I

- Immediate changes to get major existing reliability modules working
- Syntax and input file data flows revised for future class changes
- Document changes in cooperation with UCSD
- Original target date: OpenSees Days 2010
  - New target date: end of October

Phase II

- Re-engineer classes & inheritance as discussed (parameter, gradient, functions, etc.)
- Attempt to maintain methods from Phase I
- Target date: summer 2011
Thank You!

Please contact:

- Kevin Mackie: kmackie@mail.ucf.edu
- Michael Scott: michael.scott@oregonstate.edu