Project Thoughts, pgraf

1 Simplified summary of implicit sampling (IS)

GOAL OF THIS SECTION: concise summary of implicit sampling so there are some equations and an algorithm to talk about.

Implicit sampling (IS) of Chorin et al is a method to combine data and computational models, e.g. in the context of "data assimilation" or "filtering" applications. For state x and data (observations) b over time steps n, we assume the model (process)

$$x^{n+1} = f(x^n) + noise,$$

and the observations

$$b^{n+1} = h(x^{n+1}) + noise.$$

(this is a simplification, no reason that noise is necessarily additive). The goal is to best estimate $P(x^{n+1}|b^{n+1})$, i.e., the state x^{n+1} given a new observation b^{n+1}. Bayes rule allows us to write

$$P(x^{n+1}|b^{n+1}) \propto P(x^{n+1}|x^n)P(b^{n+1}|x^{n+1}),$$

which I'll shorten to

$$p \propto p_1 p_2.$$

The term p_1 is from the model, the term p_2 is from the observation. Sampling this distribution is the goal. The usual method ("importance sampling") samples p_1, then uses p_2 to weight the sample. The problem is that likely samples from p_1 may be very *unlikely* w.r.t. p_2 and thus have very small weights, essentially wasting the sample. Implicit sampling is an attempt to overcome this problem. The idea is to define a direct map from a "simple" distribution we *can* sample to the target distribution p. Let the simple distribution's pdf, e.g. a Gaussian, be $g(\xi)$. Now let

$$F(x^{n+1}) = -\ln(p), \qquad G(\xi) = -\ln(g).$$

Note that F depends parametrically on x^n and b^{n+1} through p_1 and p_2, but the "free variable", e.g. the one with respect to which we will optimize, is x^{n+1}. (We now drop the superscript "$n+1$" on the free variable). Now we define an equation, the solution of which defines the mapping from the simple (g) distribution to the target one (p). For this, let

$$\phi_F = \min F, \tag{1}$$

and let $\phi_G = \min G$. The map is defined by declaring that

$$F(x) - \phi_F = G(\xi) - \phi_G. \tag{2}$$

Note how high probability ξ's will map to high probability x's: Such ξ will have $G(\xi) \sim \phi_G$, i.e. the equation becomes, roughly,

$$F(x) \sim \phi_F.$$

The solution will be an x that is near the minimum of F, which is near the maximum of p, which was the goal.

So, the procedure to assimilate a new data point b^{n+1} consists of executing, for each of a set of "particles" (the particles represent an empirical sample of p from which desired statistics are then derived):

1. solve (1) to get ϕ_F.

2. sample g to get ξ.

3. solve (2) to find x^{n+1} (for this particle)

A very simple example: Suppose "noise" above is $N(0, \sigma_1)$ for the model, $N(0, \sigma_2)$ for the observations. Then

$$p_1 \sim N(f(x^n), \sigma_1), \qquad p_2 \sim N(h(x^{n+1}), \sigma_2),$$

so

$$p_1 \propto \exp[-(x^{n+1} - f(x^n))^2/\sigma_1], \qquad p_2 \propto \exp[-(b^{n+1} - h(x^{n+1}))^2/\sigma_2].$$

Thus

$$F = -\ln(p_1 p_2) = -\ln(p_1) - \ln(p_2) = (x^{n+1} - f(x^n))^2/\sigma_1 + (b^{n+1} - h(x^{n+1}))^2/\sigma_2.$$

This makes the above procedure clear: For each new observation b^{n+1}, for each particle x (there should be a particle index on x, but I want to keep the indices to a minimum), we have to (eqn (1)) minimize this explicit equation w.r.t. x^{n+1}, then (eqn (2)) solve a nonlinear equation using this explicit equation, in both cases with "known" (quotes because these may only be know via a simulation) functions f and h. These may be hard tasks, but there is no mystery about what needs to be done.

Final thought on the method itself: Often in UQ one is interested in more than just high probability samples. For example, in reliability analysis very often one is interested in outliers and tails of distributions. It would be interesting to see if modifications of the algorithm (e.g., the right modification to eqn (2) would allow us to target what "types" of samples from p we get.

2 Implicit Sampling for exascale science

GOAL OF THIS SECTION: To outline what we will need to do to use IS in an extreme scale computational setting.

Chorin et al have developed a number of methods to implement this method and demonstrated its effectiveness in simple cases. Part of our proposal is to develop the machinery necessary to implement it for extreme scale problems. An exascale science problem will presumably involve model function f and perhaps observation function h that are simulations of arbitrary complexity. I can think of two levels to consider, depending on what one's conception of an "exacscale science" computation is.

I) Assuming the simulation is one large simulation (e.g., highly highly resolved CFD), a variety of technical issues that we can propose to address emerge:

Solving (1) and (2) are in general high dimensional nonlinear problems (though the "random map" implementation avoids this for (2)). It is very advantageous to use derivatives in such problems. For the optimization problem (1), a straightforward approach would utilize adjoint codes that provide such derivatives. Then a parallel-optimizer such as TAO (from Argonne) could be used, among others. Limited memory (e.g. BFGS) algorithms would be used if, as is likely, Hessians are not available. Then the main issues become parallel communication of large vectors during the optimization. The issue of parallel communication and load balancing during the solution of (1) and (2) is relevant in any case.

If no such adjoint code is available, we will need to use derivative free optimization (DFO) methods. These methods fall into 3 categories: finite difference gradients, direct search, and surrogate methods. The most interesting and likely most successful of these for large scale problems

is surrogate methods. In such a method, some number of model executions are utilized to make a computational model of the simulation; this model is then used for some number of optimization steps; further refinement of the model goes hand in hand with its use for seeking the optimum. This is an active research area, and we could probably tailor a proposal to the context of the overall framework. Especially: the optimal tradeoff of sampling and searching is a very interesting and unsolved problem. We are already talking about using a "hierarchy of models", which fits nicely with this line of thought; the surrogates are just reduced order models, i.e. they qualify as lower fidelity simulations in this context.

How exactly their use in optimization meshes with the overall multi-fidelity program is not clear, but would be an interesting thing to think about. In particular, two ideas come to mind. First, it should be possible to design DFO methods specifically for multicore architectures, because many of them rely on performing the same operations many times on slightly different data. In fact, the whole IS algorithm is probably conducive to this type of programming. Which leads to the second idea, that if one uses a "population based" DFO to perform the optimizations (1), then they start to look very similar to the "particle based" IS algorithms. There would appear to be an opportunity to combine them.

For the nonlinear solves (2), we can use jacobian-free Newton-Krylov (JFNK) methods. Such methods are very powerful because they solve large scale problems with a convergence like Newton's method (which by itself requires Jacobians) without ever computing the Jacobian. Recent advances in directly using the JFKN idea in optimization problems, known as Lagrange-Newton-Krylov methods, should also be considered.

II) If the conception of "exascale science" assumes instead that the simulation is actually a coupled set of simulations (as many of the DOE documents appear to, though this is apparently not in line with the actual funded exascale projects): multi-scale, multi-physics, multi-domain. I do not know how the IS framework fits into this paradigm. But it would be very interesting to consider variations of the method that explicitly exploit the structure of the solver(s).

In particular, we have stipulated that the data comes in a hierarchy of scales. We can further stipulate that the models come in a hierarchy of scales as well. Here, "scale" can mean fidelity as well, so a hierarchy of reduced order models qualifies. Given data and a model that "match" in scale, we can perform IS on this scale. This produces "posteriors" (state estimates *after* incorporating the data) from "priors" (state estimates based only on running the model). Now, we have also mentioned that "posteriors" on one scale become "priors" on the next. How do we reconcile these notions? Perhaps we can think of finer (higher fidelity) scales as "conditioned by" all coarser scales. The machinery to do IS on a single scale is clear (well, sort of...). We will develop the machinery to treat the posteriors on all coarser scales, each of which is the outcome of applying IS on that scale, as the prior on a given scale.

3 Applications to consider

GOAL OF THIS SECTION: throw out some applications that might be relevant, either as "featured" or "mentioned".

1) systems biology: Many formulations exist, but the heart of the problem, in extreme scale sense I) above, is a reaction diffusion system for an entire cell. The exascale documents from DOE talk about systems biology as a multi-physics problem that "accurately represents processes such as cell growth, metabolism, locomotion, and sensing." But a simplified system capable of running on a big machine would be a box containing a variety of reacting and diffusing chemicals. This is much like the combustion problem, but (as I understand it) somewhat complementary; in biology, the chemistry is *extremely* complicated, whereas the transport is fairly mundane (diffusion). Perhaps this could be folded into the combustion problem as part of a spectrum of reaction-transport

problems that our "novel UQ methodology" can address. I was part of a SciDAC project (that ended about 3 years ago) that took small steps forward on this problem.

2) batteries: The Li-ion battery (and all the alternative chemistries) proposed is hugely multi-scale, with important phenomena spanning, say, 10 orders of magnitude. This is a system that has lots of morphological structure: the pack is arranged into cells; the cells are, for example, wound into spirals or folded into stacks of electrodes; the electrodes have a cathode, separator, and anode; the cathode and anode are heterogenous mixtures of Li-hosting particles and electrolyte. Simulation to date has focused either on one scale, or on multi-scale models that involve totally separate models of different scale that are coupled through boundary conditions and source terms. An extreme scale battery simulation, in this context, would serve as a great example of how to apply "our methods" (whatever they turn out to be) on a hierarchy of models at different scales. I am currently working on two different projects with the NREL battery simulation group.

3) Wind energy: Another manifestly multi scale problem (as you CFD people of course appreciate), with the added complexity of a highly complex engineered system "in the loop". I am currently working on a project in "systems engineering for wind energy" that is less focused on the CFD and atmospheric physics, and more focused on writing software to support the complex multi-physics multi-scale engineering of the wind farm. The emphasis is on incorporating real physics models (over a diverse range of scales and areas, e.g. from "nuts and bolts" of blade structure all the way up to operations and maintenance costs of the entire wind farm) into the computation of overall cost.