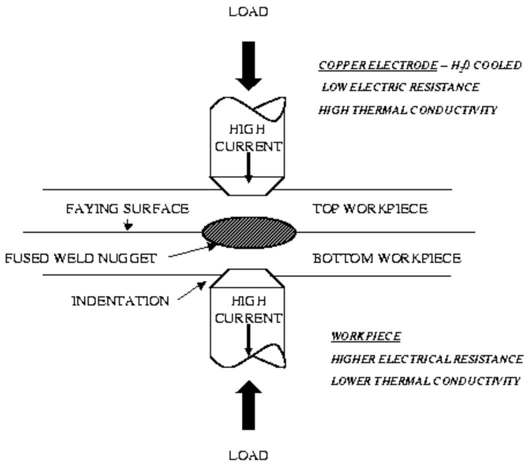
Calibration and Validation of Computer Models: a Bayesian Approach
Lecture 2: Computer models; generalities and emulation

Rui Paulo

ISEG/CEMAPRE Technical University of Lisboa, Portugal

June 29 2011


LOAD
COPPER ELECTRODE – H2O COOLED
LOW ELECTRIC RESISTANCE
HIGH THERMAL CONDUCTIVITY
HIGH CURRENT
FAYING SURFACE
TOP WORKPIECE
FUSED WELD NUGGET
BOTTOM WORKPIECE
INDENTATION
HIGH CURRENT
WORKPIECE
HIGHER ELECTRICAL RESISTANCE
LOWER THERMAL CONDUCTIVITY
LOAD

Bayarri et al. (2007) Technometrics: “A Framework for the validation of computer models.”

- Output of computer model for input vector $\mathbf{z}$ is $y^M(\mathbf{z})$
- $\mathbf{z} = (\mathbf{x}, \mathbf{u})$ where

  $\mathbf{x}$ are *controllable* inputs

  $\mathbf{u}$ are *uncertain* inputs or parameters, which can be *tuning* or *calibration* parameters
- Computer model aims at reproducing some real phenomenon which we denote by $y^R(\mathbf{x})$

Validation of computer models

- Question of interest: *Does the computer model adequately represent reality?*
- The answer to the question "Is the model correct?" is almost always "No"
- In general, people are interested in whether the model produces results that are accurate enough for the intended use

Tolerance bounds

- We respond to that question by producing statements like

$$\text{Pr}\{|\text{reality} - \text{model}| < \tau\} > \gamma$$

  for some tolerance τ and probability γ
- Example: $5.76 \pm .44$ — there's a specified chance (say 90%) that the true underlying process at certain input value lies within this specified range

Tolerance bounds—Why?

- Accuracy of model predictions varies over the range of inputs
- The degree of accuracy may differ from one application of the model to another
- Tolerance bounds account for model bias

Difficulties

- Uncertainty in $\mathbf{u}$ (or $\mathbf{x}$) arising from multiple sources
- Limited model runs
- Field data limited and/or noisy
- Model runs and field data at different $\mathbf{x}$
- Simultaneously "tune" $\mathbf{u}$ and validate model, based on the same set of field data
- y^M highly non-linear and biased
- Validation as dynamic process

Different problem?

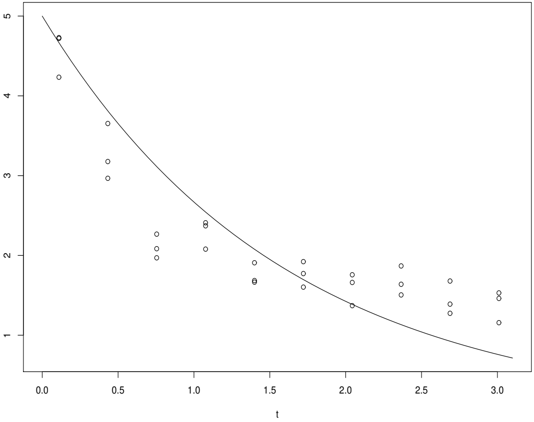
How is this different from *statistical* model validation?

- Limited data
- Expense of running computer model — construction of emulators
- What is to be done with a “rejected” computer model
- Example: $y(t_i) = g(t_i) + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

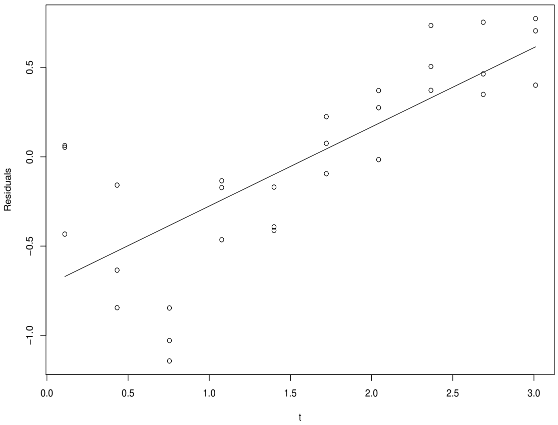
$$H_0: \; g(t) = 5 \; \exp(-ut)$$

- $\hat{u} = 0.63$

Maximum likelihood fit of H0



Residuals and linear fit



One might be tempted to think that the additional structure found in the residuals is real, but

- If the hypothesized model is incorrect then "over-fitting" will typically occur; u is compensating for the lack of fit
- The over-fitting makes it problematic to believe any structure found in the residuals

From a statistical perspective, one would postulate another H_0, but

- If g is the computer model that is not viable, unless this analysis has suggested possible improvements, which are then implemented
- Computer models have science built in and are crucially needed for extrapolation beyond the range of the data; statistical models are typically not as trustworthy for such an extrapolation

Kennedy and O'Hagan 2001; Craig et al 1996

- Formally introduce a bias function $b(t)$:

$$y(t_i) = 5\exp(-ut_i) + b(t_i) + \varepsilon_i$$

  where $b(t)$ is an unspecified function
- One then attempts to jointly estimate $b(\cdot)$ and u — prevent over-fitting and account for all uncertainties
- u and $b(\cdot)$ are severely confounded
- Bias and resulting confounding are more common in Statistics than might be thought (Gustafson 2006, Goldstein 2010)
- model inadequacy: computer model is an imperfect representation of reality (Goldstein 2010 for a recent account)

- Bayesian analysis is the most straightforward way of dealing with such confounding — prior distribution on u and $b(\cdot)$ containing as much expert knowledge as possible:
  - u may have physical meaning or at least physical limits
  - prior on $b(\cdot)$ which “encourages” the function to be 0
- Some inferences will tend to be sensitive to prior specification, others not so much
- *modularization* techniques to reduce confounding

The Framework

- Specify model inputs and parameters with associated uncertainties
- Determine evaluation criteria
- Data collection and design of experiments

⇒ Construct fast approximation to the computer model

⇒ Analysis of model output; comparing model output with field data

- Feedback and feed-forward

- Some computer models take several hours to compute $y^M(\mathbf{z})$ for a single $\mathbf{z}$
- Fast approximation to the output of the computer model (and associated measure of uncertainty) — the emulator
- Other uses: uncertainty & sensitivity analysis, optimization
- We use it in the MCMC as a surrogate for the computer model; we have to evaluate $y^M(\mathbf{x}, \mathbf{u}_j)$ many times
- Gaussian processes as priors for unknown functions, O'Hagan (1978); use in the context of computer models, Sacks et al. (1989)

Gaussian Process

The stochastic process $\{Y(\mathbf{x}) : \mathbf{x} \in \mathcal{S} \subset \mathrm{IR}^p\}$ is called Gaussian if, for all finite $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{S}$, the random vector

$$\mathbf{Y} = (Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n))'$$

has a Multivariate Gaussian distribution.

In order to characterize a Gaussian process all one has to do is specify the mean function and the covariance function, which is usually done in a parametric fashion.

Prior

- *a priori* $y^M(\cdot) \mid \boldsymbol{\theta} \sim \mathrm{GP}(\mu(\cdot), C^M(\cdot,\cdot))$
- common choice for the mean and covariance functions:

$$\mu(\mathbf{z}) = \boldsymbol{\Psi}(\mathbf{z})'\boldsymbol{\theta}^L$$
$$C(\mathbf{z},\mathbf{z}^\star) = \frac{1}{\lambda^M}\, c(\mathbf{z},\mathbf{z}^\star \mid \boldsymbol{\theta}^M) = \frac{1}{\lambda^M}\prod_{i=1}^{p} \exp[-\beta_i^M |z_i - z_i^\star|^{\alpha_i^M}]$$

- Separable power exponential correlation function
- Separability criticized
- The power exponential correlation function has several limitations (Stein 1999)
- Nugget or jitter (Gramacy and Lee 2010)

Construction of the emulator

- Construct a design set $D^M = \{\mathbf{z}_i,\ i = 1, \ldots, m\}$ — very important research area
- Observe model runs $\mathbf{y}^M = y^M(D^M) = \{y^M(\mathbf{x}_i, \mathbf{u_i})\}$,
- *A posteriori*, given $\boldsymbol{\theta} = (\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$, $\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$

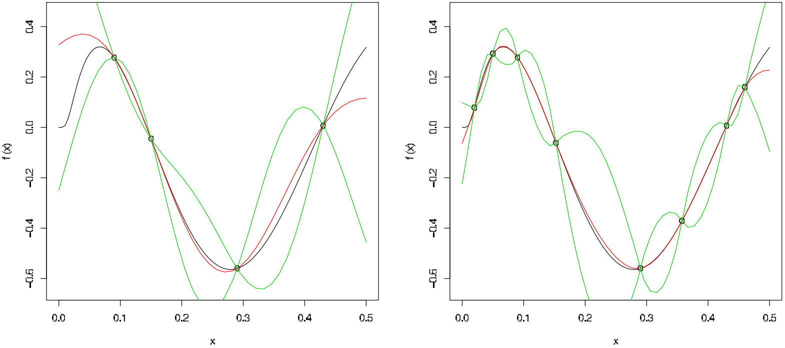
$$y^M(\cdot) \mid \mathbf{y}^M, \boldsymbol{\theta} \sim \mathrm{GP}(\mu^\star(\cdot), C^\star(\cdot, \cdot))$$

where $\mu^\star$ and $C^\star$ have closed form expressions:

$$\mu^\star(\mathbf{z}) = \boldsymbol{\Psi}'\boldsymbol{\theta}^L + \mathbf{r}_\mathbf{z}'(\boldsymbol{\Gamma})^{-1}(\mathbf{y}^M - \mathbf{X}\boldsymbol{\theta}^L)$$
$$C^\star(\mathbf{z}, \mathbf{z}^\star) = \frac{1}{\lambda^M} c^M(\mathbf{z}, \mathbf{z}^\star) - \mathbf{r}_\mathbf{z}(\boldsymbol{\Gamma})^{-1}\mathbf{r}_\mathbf{z}$$

- $\boldsymbol{\Gamma} = c^M(D^M, D^M)$, $\mathbf{r}_\mathbf{z}' = c^M(\mathbf{z}, D^M)/\lambda^M$, $\mathbf{X}$ is a matrix with rows $\boldsymbol{\Psi}'(\mathbf{z}_i)$

- With $\boldsymbol{\theta}$ known, this posterior predictive distribution acts as the emulator; the point prediction is the mean, the variance measures the uncertainty
- In particular, $\mathrm{Var}(y^M(\mathbf{z}_i) \mid \boldsymbol{\theta}, \mathbf{y}^M) = 0$—that is, the GP approximation is an interpolator of the original function at the observed points $\mathbf{z}_i$
- $\boldsymbol{\theta}$ is typically unknown
  - plug-in $\hat{\boldsymbol{\theta}}$, an estimate of $\boldsymbol{\theta}$ — underestimate variability
  - integrate $\boldsymbol{\theta}$ wrt its posterior in a full Bayesian analysis



0.4
0.2
0.0
-0.2
-0.4
-0.6
f(x)
0.0
0.1
0.2
0.3
0.4
0.5
x
0.4
0.2
0.0
-0.2
-0.4
-0.6
f(x)
0.0
0.1
0.2
0.3
0.4
0.5
x

- If we want to obtain a prediction of $\mathbf{y}^M_{\text{new}} = \mathbf{y}^M(D^M_{\text{new}})$, $D^M_{\text{new}} = \{\mathbf{z}^\star_j\}$, all we have to do is draw from

$$f(\mathbf{y}^M_{\text{new}} \mid \mathbf{y}^M) = \int f(\mathbf{y}^M_{\text{new}} \mid \mathbf{y}^M, \boldsymbol{\theta})\ \pi(\boldsymbol{\theta} \mid \mathbf{y}^M)\ d\boldsymbol{\theta}$$

  where $\pi(\boldsymbol{\theta} \mid \mathbf{y}^M)$ is either an actual posterior or degenerated at $\hat{\boldsymbol{\theta}}$
- For each element of a sample from $\pi(\boldsymbol{\theta} \mid \mathbf{y}^M)$, $\boldsymbol{\theta}_j$, draw a random vector from the corresponding Gaussian distribution $f(\mathbf{y}^M_{\text{new}} \mid \mathbf{y}^M, \boldsymbol{\theta}_j)$
- numerical instabilities; nugget effect or jitter $y^M(\cdot) = GP + \varepsilon$

Full Bayes or plug-in MLE?

- The full Bayesian approach is superior: it takes into account uncertainty in θ when assessing accuracy
- We are interested not in the emulator itself but rather in integrating it in the validation/calibration process
- Often, the uncertainty in the calibration parameters and in the bias tend to overwhelm the uncertainty in θ; hence, full Bayes or plug-in mle give essentially the same results
- plug-in mle allows implementation of the validation/calibration in more complicated settings

Full Bayes

- prior on $\boldsymbol{\theta}$ is difficult to specify from a subjective perspective
- need for automatic procedure
- the derivation of objective priors in this setting is an important problem
- Berger et al. (2001): many objective priors used in spatial statistics produce improper posterior; unusual asymptotic behavior of the likelihood
- Paulo(2005) derives objective priors (known $\boldsymbol{\alpha}$)
- in general

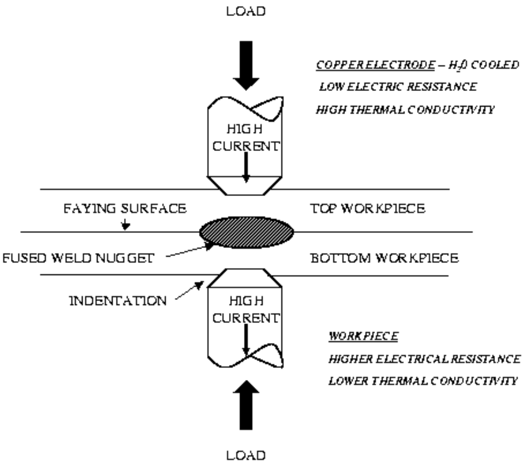
$$\pi(\boldsymbol{\beta}^M, \boldsymbol{\theta}^L, \lambda^M) \propto \frac{\pi(\boldsymbol{\beta}^{\mathbf{M}})}{(\lambda^M)^a}$$

- priors are computationally demanding
- as we explore the parameter space in MCMC, we get to regions where the correlation matrix is close to singular
- proposal can use explicit formulae for the Fisher info
- active area of research (De Oliveira 07, Ren et al 2010 etc)
- empirical Bayes solution which works well: placing exponential priors centered at a multiple of the marginal MLE

Plug-in MLE

- Likelihood surfaces arising from GP have surprising properties (Warnes and Ripley 1987 etc)
- Maximum likelihood estimators have unusual asymptotic behavior (Ying 1993, van der Vaart 1996, Chen et al 2000)
- Bottom line: computing $\hat{\theta}$ is not trivial and requires specialized software
- Alternatives: marginal MLE; posterior modes with objective priors; penalized likelihood (Li and Sudjianto 2005)

The Spotweld Example



The Spotweld Example

- Controllable inputs are Load (L), Current (C), Gauge (g);
- Tuning parameter is u. The contact resistence between the sheets of metal and the electrode is critical to the model. Yet, this function is not known. The modeller introduced a parametric family of functions indexed by u.
- The evaluation criteria:
  1. Weld diameter after eight cycles—primary use of the model;
  2. Weld diameter as a function of the number of cycles—possible aid in reducing the number of cycles;

  The second had to be eliminated because the code was not producing reliable computer runs at earlier times than eight cycles

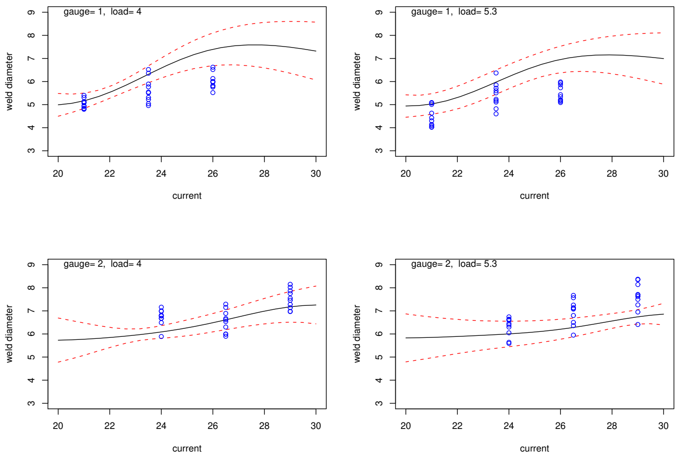
Data collection

- Model data: computer model takes 30 minutes to run; 35 runs chosen according o Latin hypercube design
- Field data: 10 replicates for $L \in \{4, 5.3\}$, $g \in \{1, 2\}$ and three values for $C \in \{21, 23.5, 24, 26.5, 29\}$

Emulation exercise

- Predict output of computer model at the values of L and g observed in the field experiments but as a function of C
- Take C varying in a 20-point equally spaced grid in the interval $(20, 30)$
- Value for the calibration parameter u? Let's pick $u = 3.0$

Pure−model predictions − Spotweld Data



Details — how do we produce each of the panels?

1. Obtain $\hat{\theta}$ by maximum likelihood
2. Let $L_0 = 4$, $g_0 = 1$, $u_0 = 3.0$ and denote by $\{C_j,\ j = 1, \ldots, 20\}$ an equally-spaced grid in $(20, 30)$
3. $D^M_{\mathrm{new}} = \{(C_j, L_0, g_0, u_0),\ j = 1, \ldots, 20\}$; $\mathbf{y}^M_{\mathrm{new}} = y^M(D^M_{\mathrm{new}})$
4. Draw $T = 50000$ random vectors from the multivariate normal

$$f(\mathbf{y}^M_{\mathrm{new}} \mid \mathbf{y}^M, \hat{\theta})$$

Denote those draws by $\mathbf{y}^{M(t)}_{\mathrm{new}}$, $t = 1, \ldots, T$

Details — cont.

5. The solid line is obtained as

$$\bar{\mathbf{y}}^M_{\rm new} = \frac{1}{M}\sum_{t=1}^{M} \mathbf{y}^{M(t)}_{\rm new}$$

6. At each C_j, the upper and lower uncertainty limits (a_j, b_j) are obtained as the 5% and 95% sample quantile of

$$\{y^{M(t)}_{\rm new}(C_j, L_0, g_0, u_0), t = 1, \dots, M\}$$

so that we can state that

$$P(a_j < y^M(C_j, L_0, g_0, u_0) < b_j \mid \hat{\theta}, \mathbf{y}^M) = 0.90$$

Observations

- This can all be done analytically for fixed $\hat{\theta}$
- The simulation-based calculation makes it easy to predict any function of $y^M(\cdot)$
- If we have a sample $\boldsymbol{\theta}^{(t)}$, $t = 1, \ldots, T$ from the posterior $\pi(\theta \mid \mathbf{y}^M)$, then $\mathbf{y}_{\text{new}}^{M(t)}$ should be drawn from

$$f(\mathbf{y}_{\text{new}}^M \mid \mathbf{y}^M, \boldsymbol{\theta}^{(t)})$$

- If u_0 was an estimate of u, then $\bar{\mathbf{y}}_{\text{new}}^M$ would be called the pure-model prediction of $(\mathbf{y}^R(C_j, L0, g_0),\ j = 1, \ldots, 20)$ associated with u_0
- At this point, we do not have yet the information to evaluate the quality of that estimate as an estimate of y^R

Use of derivative information

- Some computer models correspond to the implementation of systems of PDE's
- Along with $y^M(\mathbf{z})$ the software may also provide information about derivatives of $y^M(\mathbf{z})$
- Typically that information is ignored, but it may utilized in the construction of the emulator

Derivatives of Gaussian Processes

- *a priori*, $y^M(\cdot) \sim \mathrm{GP}(\mu(\cdot), \frac{1}{\lambda^M} c^M(\cdot, \cdot))$

$$\mu(x, u) = \Psi(x)'\theta^L$$
$$c^M((x, u), (z, v)) = \exp(\beta_1 |x - z|^{\alpha_1}) \exp(\beta_2 |u - v|^{\alpha_2})$$

- $\partial y^M(x, u) \equiv \frac{\partial y^M}{\partial u}(x, u)$
- If $\alpha_2 = 2$, ∂y^M is still a Gaussian process and

$$\mathrm{E}(\partial y^M(x, u)) = \frac{\partial \mu}{\partial u}(x, u)$$
$$\mathrm{Cov}(\partial y^M(x, u), y^M(z, v)) = \frac{1}{\lambda^M} \frac{\partial c^M}{\partial u}((x, u), (z, v))$$
$$\mathrm{Cov}(\partial y^M(x, u), \partial y^M(z, v)) = \frac{1}{\lambda^M} \frac{\partial^2 c^M}{\partial u \partial v}((x, u), (z, v))$$

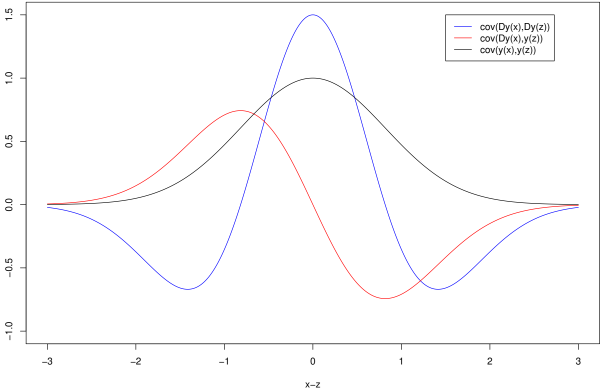
- Which in the case at hand turn out to be

$$E(\partial y^M(x,u)) = 0$$

$$\mathrm{Cov}(\partial y^M(x,u), y^M(z,v)) = -\frac{2}{\lambda^M}\beta_2(u-v)c^M((x,u),(z,v))$$

$$\mathrm{Cov}(\partial y^M(x,u), \partial y^M(z,v)) = \frac{2}{\lambda^M}\beta_2[1-2\beta_2|u-v|^2]c^M((x,u),(z,u$$

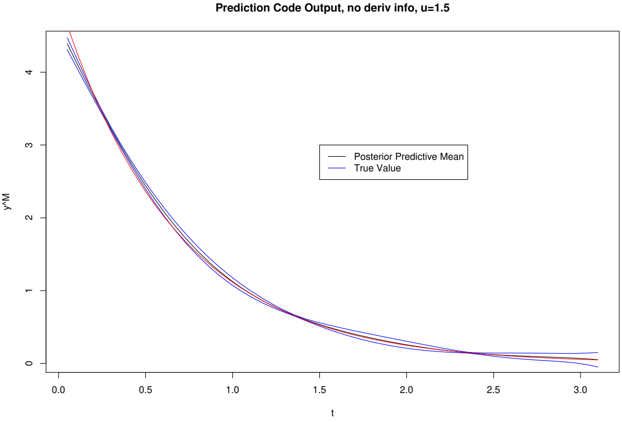
Covariance functions



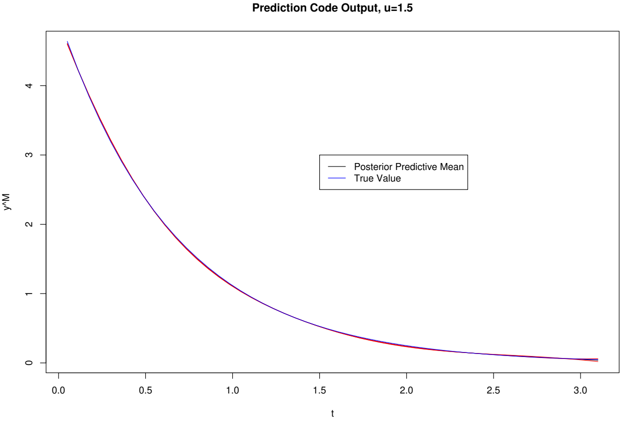
Example

- Math model:
$$\frac{dy(t)}{dt} = -uy(t)$$
with $y(0) = y_0$
- Solution
$$y^M(t, u) = y_0 \exp(-ut)$$
is treated as a slow computer model
- Model and its derivatives with $y_0 \equiv 5$ are exercised at a 15-point Latin hypercube design in $[0.5, 2] \times [0.1, 3.0]$ in the (u, t) space.
- The plots that follow have been produced by computing estimates of the parameters of the model using code data only

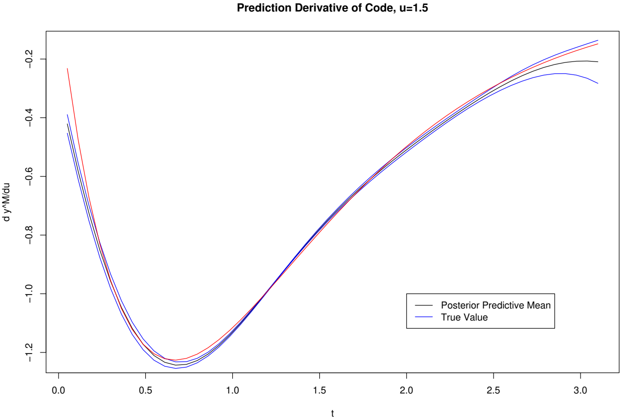
Prediction using model data only



Prediction using model and derivative data



Prediction of derivatives



Incorporating derivative information

- O'Hagan (1992), Morris et al. (1993)
- Increase in sample size; numerical instabilitites
- Inference depends on finner properties of the GP prior
- May be a problem-specific decision

Summary

- Overview of the framework for the validation of computer models
- Emulators: Gaussian process priors; inference and predicition
- The Spotweld example
- Incorporating derivative information