Physical Experimental Design in Support of Computer Model Development

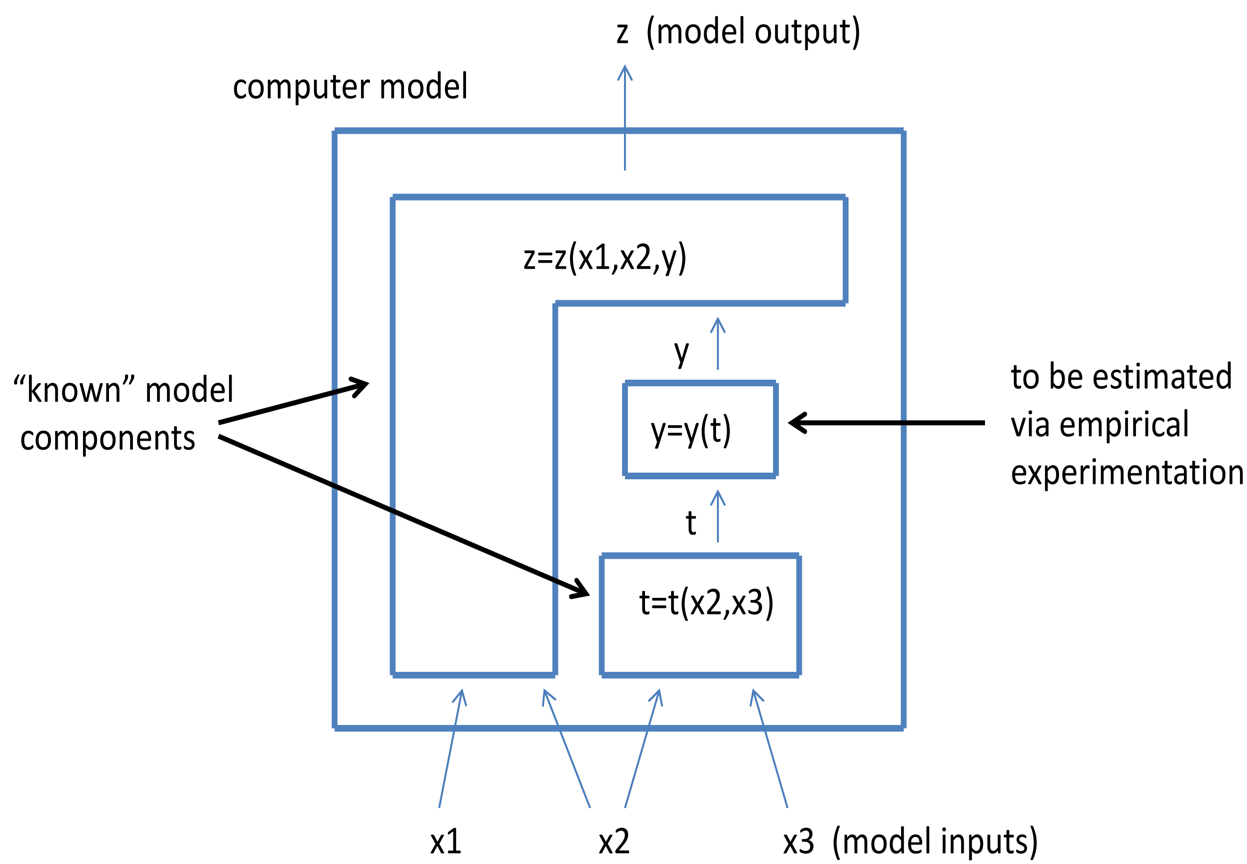
or

Experimental Design for RSM-Within-Model

Max D. Morris
Department of Statistics
Dept. of Industrial & Manufacturing Systems Engineering
Iowa State University

- Setting: Constructing a deterministic computer model of a complex physical process.
- Computer Model: Modular form, with the various components of the process represented by different subroutines, most of which are regarded as *known* based on theory or extensive experience.
- Problem: One module is not known and must be determined via physical experimentation.
- Example: Models of biofuel production: Different types of biomass (e.g. switchgrass, corn stover, wood chips ...) react differently to the initial reduction steps; physical experimentation is required to characterize for each. But material transport, economics, later-stage chemistry, et cetera, are relatively well-understood.

Basic System:



Notation:

- Model inputs: $x = (x_1, x_2, x_3)$
- Model outputs: z
- Ideal model: $z = f(x_1, x_2, x_3)$
- Known components: $z = z(x_1, x_2, y)$, $t = t(x_2, x_3)$
- Unknown component: $y = y(t)$
- Empirical experiment: $\hat{y} = \hat{y}(t)$
- Model to be constructed: $\hat{z} = z(x_1, x_2, \hat{y}(t(x_2, x_3)))$

For specificity, focus on:

- All functions and arguments are real-valued and continuous
- $\hat{y}(t)$ will be constructed via polynomial regression, Response Surface Methodology (RSM) (e.g. Myers, Montgomery, and Anderson-Cook)
- At specified t, observed response is:

$$y^* = y(t) + \epsilon, \quad y(t) = \mathbf{t}'\boldsymbol{\beta}, \quad \epsilon \sim N(0, \sigma^2)$$

- Select a design $D = \{t_1, t_2, t_3, ..., t_n\}$

"expanded" as $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, ..., \mathbf{t}_n$, the rows of $\mathbf{X}$

- For specified $\boldsymbol{\beta}$, σ^2, and t,

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}) \equiv BH(\boldsymbol{\beta}), \quad \hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$$

- What designs can be expected to work well?

Design for $\hat{y}$:

- Define a design space Δ_t
- Define a "weight" distribution Ω_t over Δ_t, with density ω_t
- Select D from some class to minimize:

$$\int_{t \in \Delta_t} Var(\hat{y}(t))\, \omega_t(t) dt = \sigma^2 \int_{t \in \Delta_t} \mathbf{t}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{t}\, \omega_t(t) dt,$$

i.e. I-optimality

Design for $\hat{z}$:

- Define a design space Δ_x
- Define a weight distribution Ω_x over Δ_x with density ω_x
- Select D from some class to minimize:

$$\begin{aligned}\phi &= \int_{\Delta_x} Var_{\hat{y}}(z(x_1, x_2, \hat{y}(t(x_2, x_3))))\ \omega_x(x)dx \\ &= E_x Var_{\hat{y}}(z(x_1, x_2, \hat{y}(t(x_2, x_3)))).\end{aligned}$$

- Problem: ϕ depends on on $\boldsymbol{\beta}$.
- Specify a prior distribution B (for purposes of design) on $\boldsymbol{\beta}$, and take the expectation of ϕ with respect to this distribution:

$$\phi_1 = E_{\boldsymbol{\beta},x} Var_{\hat{y}}(z(x_1, x_2, \hat{y}(t(x_2, x_3)))).$$

  – Preposterior Analysis, e.g. Bayarri & Berger, 2004, *Stat Sci*

Case 1: z and t are simple functions

- For a specified design D, evaluate ϕ_1 as:

1. Begin E-loop
   (a) Draw $\boldsymbol{\beta}$ from B
   (b) Draw (x_1, x_2, x_3) from Ω_x
   (c) Compute $t(x_2, x_3)$
   (d) Begin V-loop
      i. Draw $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$
      ii. Compute $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$
      iii. Compute $z = z(x_1, x_2, \hat{y})$
   (e) Compute $Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$.
2. Compute $E_{\boldsymbol{\beta},x} Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$.

<u>Case 2</u>: z and t are computationally demanding functions

- In (a) separate computational experiment(s), develop meta-models (surrogates) for each of z and t
- Here, assume Gaussian Process predictors (parametric kriging) ...

  predictive distributions $T(x_2, x_3)$ and $Z(x_1, x_2, y)$
- Then:

$$\phi_2 = E_{\boldsymbol{\beta},x,t,z} Var_{\hat{y}}(z(x_1, x_2, \hat{y}(t(x_2, x_3)))).$$

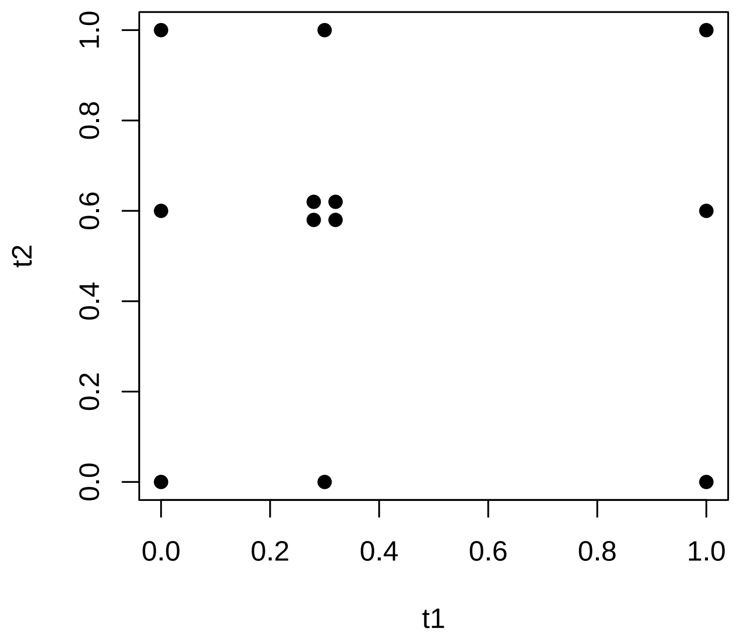
  and for a specified design D, evaluate ϕ_2 as:

1. Begin E-loop
   (a) Draw $\boldsymbol{\beta}$ from B
   (b) Draw (x_1, x_2, x_3) from Ω_x
   (c) Draw t from $T(x_2, x_3)$
   (d) Draw $\mathcal{Z}(y)$ from $Z(x_1, x_2, y)$, jointly for all values of y
   (e) Begin V-loop
      i. Draw $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$
      ii. Compute $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$
      iii. Compute $z = \mathcal{Z}(\hat{y})$
   (f) Compute $Var_{\hat{y}}(z|\boldsymbol{\beta}, x, t)$
2. Compute $E_{\boldsymbol{\beta},x,t,\mathcal{Z}} Var_{\hat{y}}(z|\boldsymbol{\beta}, x, t)$

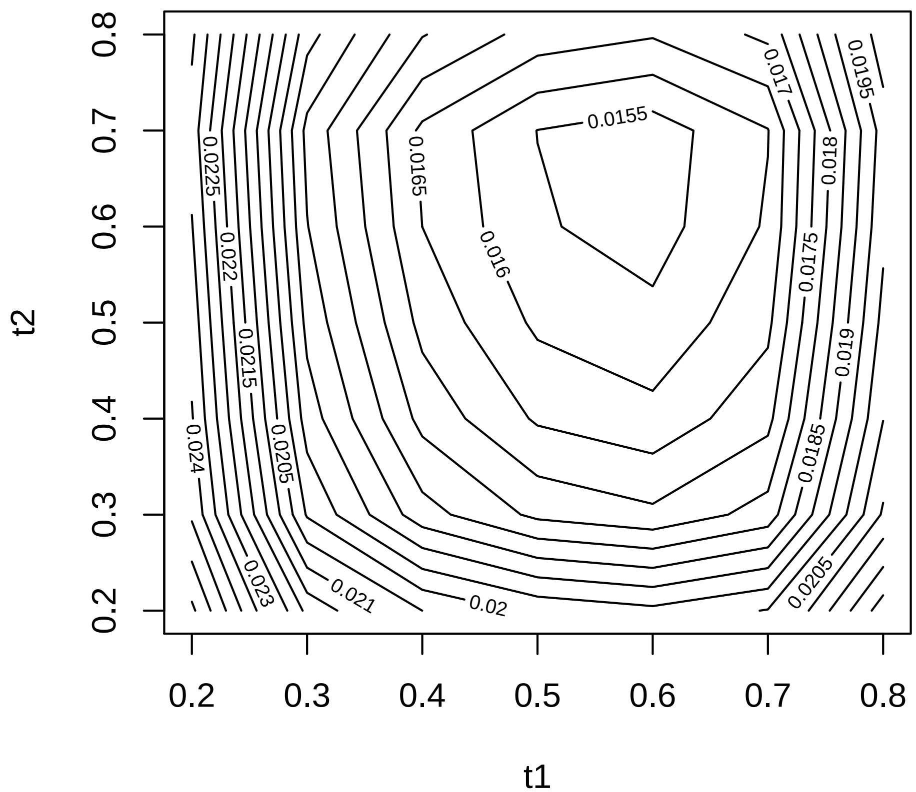
Example: Case 1

- $(x_1, x_2, x_3) \in [0,1]^3$, Ω_x = continuous uniform distribution
- $t_1 = log_2(1 + \frac{x_2+x_3}{2}), \quad t_2 = log_2(1 + x_2), \quad z = x_2 e^{-(x_1+y^2)}$
- $\rightarrow (t_1, t_2) \in [0,1]^2$
- $y(t_1, t_2) = \beta_0 + \beta_1 t_1 + \beta_2 t_2 + \beta_{11} t_1^2 + \beta_{22} t_2^2 + \beta_{12} t_1 t_2$
- $B = N(\mu_{\boldsymbol{\beta}}, \mathbf{V}_{\boldsymbol{\beta}})$
- $\sigma^2 = \frac{1}{100}, \quad \mu'_{\boldsymbol{\beta}} = (\frac{1}{2}, 0, 0, 0, 0, 0, 0), \quad \mathbf{V}_{\boldsymbol{\beta}} = \frac{1}{100}\mathbf{I}$

Designs: Complete 3^2 factorial designs for which 0 and 1 are the low and high levels for each of t_1 and t_2, the intermediate level of each is varied independently between 0.2 and 0.8, and 4 experimental runs are included at the run defined by the intermediate levels of t_1 and t_2 for replication ($n = 12$); for example:

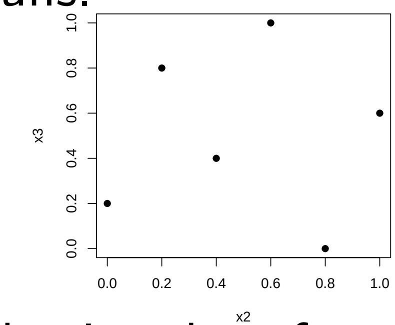


$\sqrt{\phi_1}$ as a function of t_1 and t_2 intermediate values

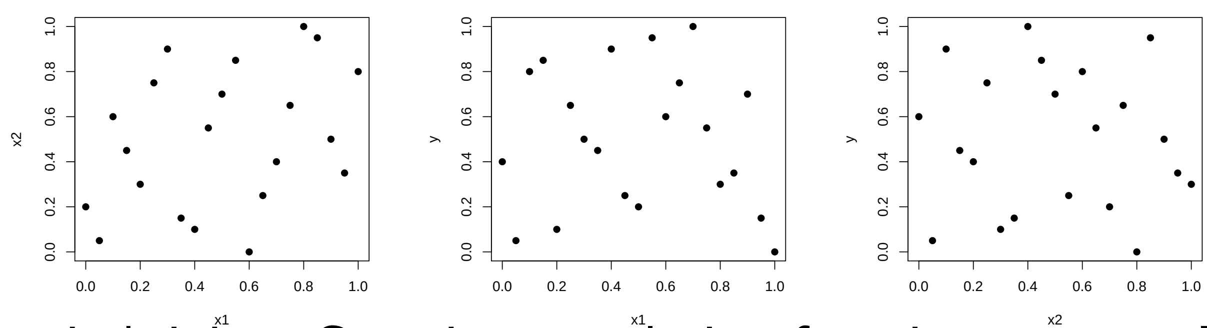


Example: Case 2

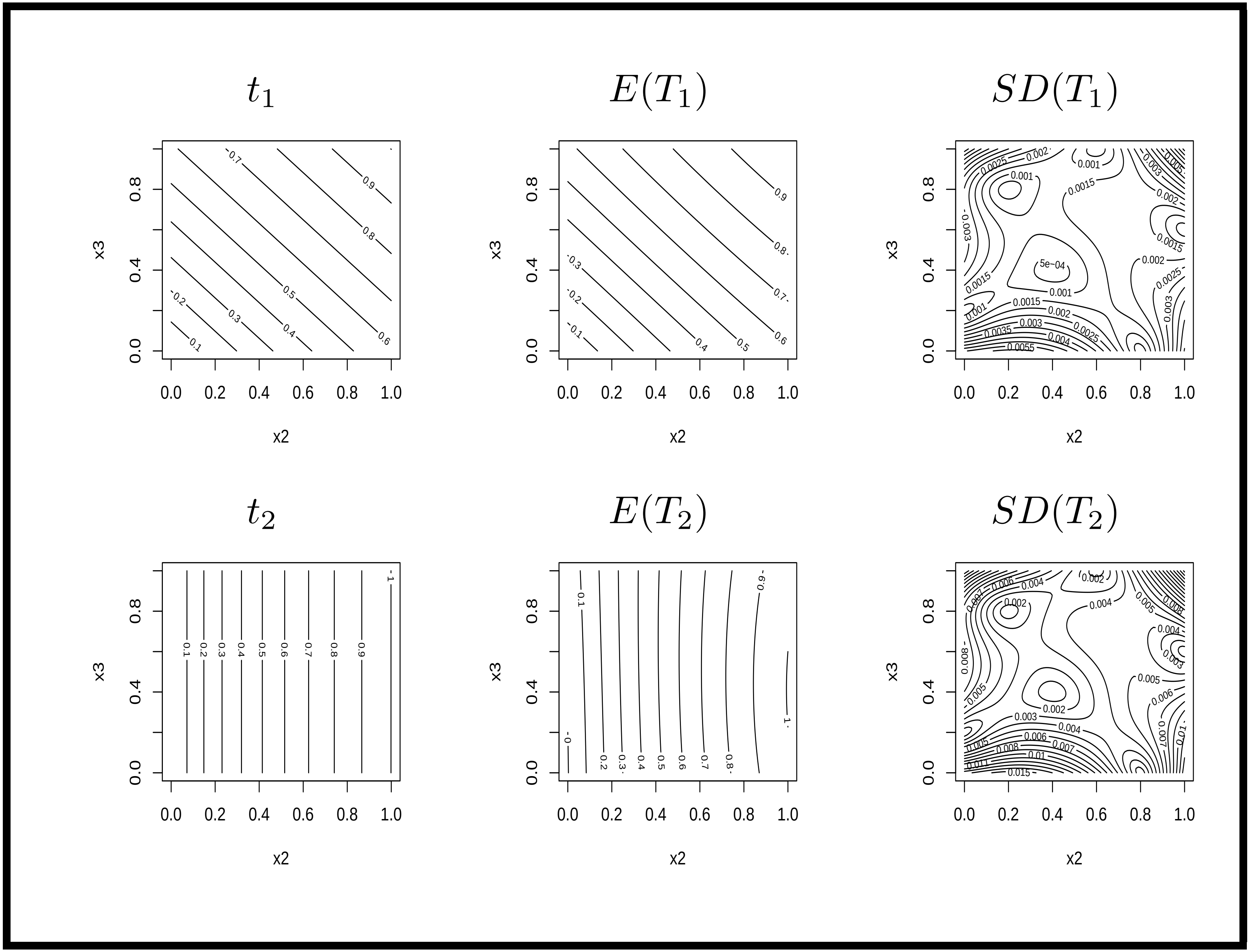
- $t_1(x_2, x_3)$ and $t_2(x_2, x_3)$ are estimated using data from a Latin hypercube design in 6 runs.

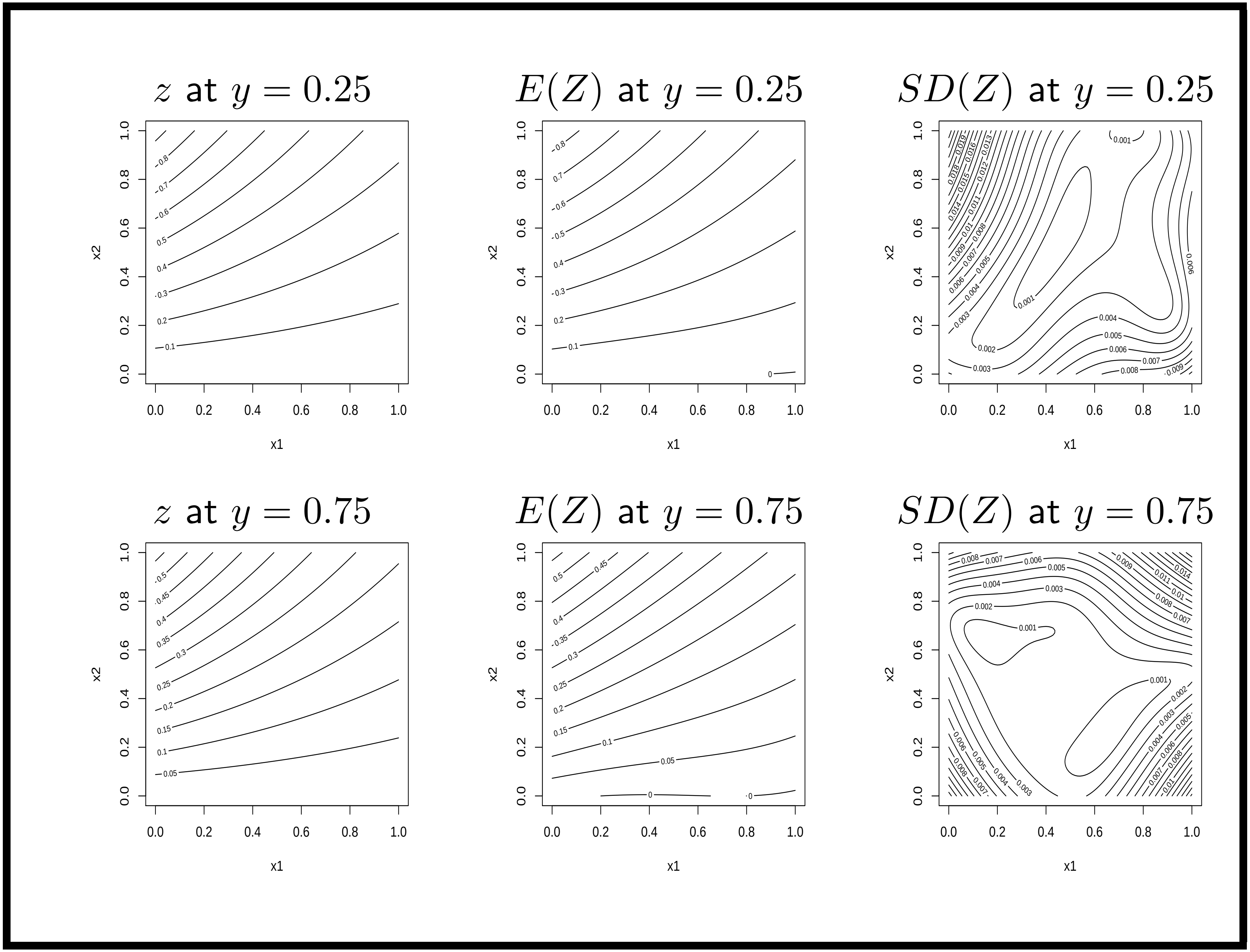


- $z(x_1, x_2, y)$ is estimated using data from a Latin hypercube design in 21 runs.

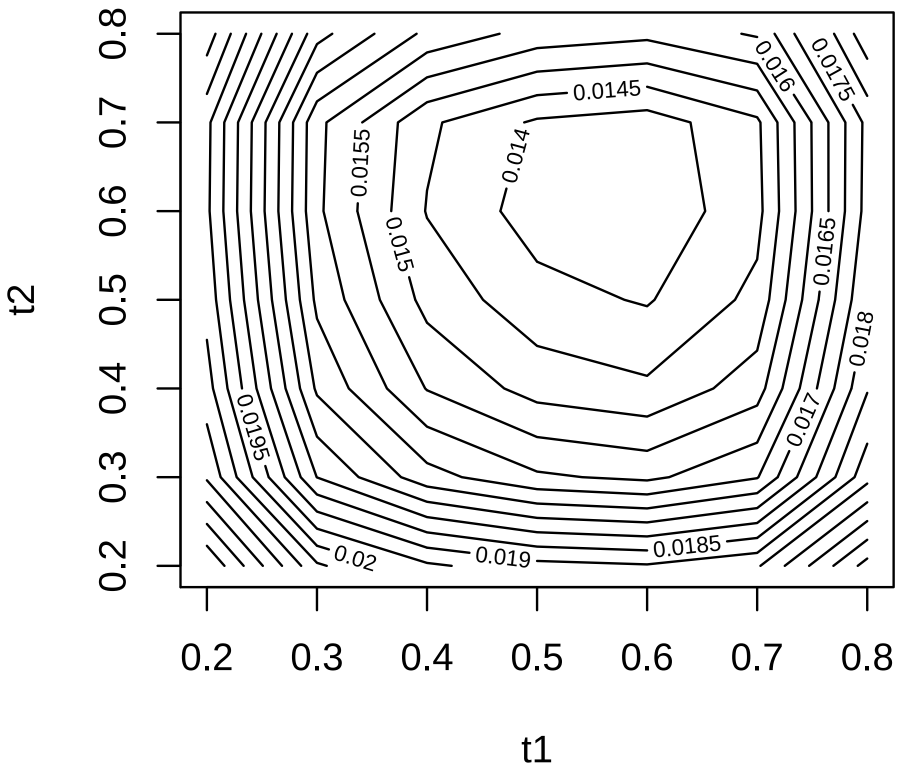


- Parametric kriging, Gaussian correlation function, process MLE's "plugged in" for corresponding parameters

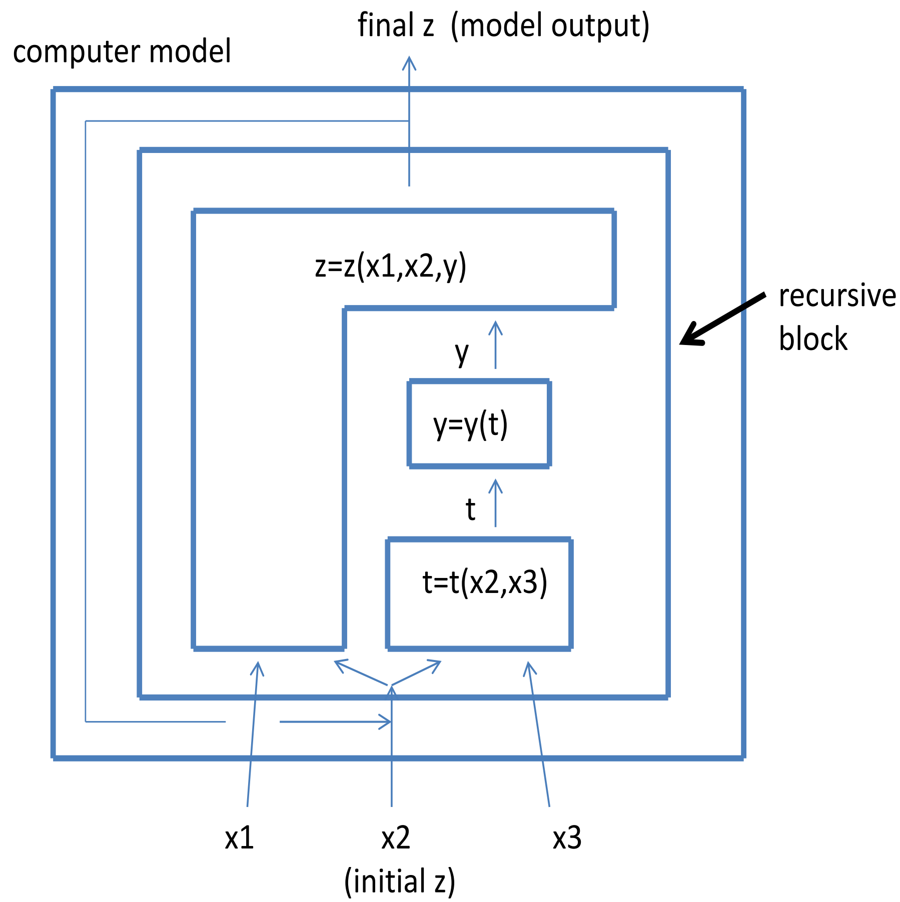

t_1
$E(T_1)$
$SD(T_1)$
t_2
$E(T_2)$
$SD(T_2)$
x2
x3


z at $y = 0.25$
$E(Z)$ at $y = 0.25$
$SD(Z)$ at $y = 0.25$
z at $y = 0.75$
$E(Z)$ at $y = 0.75$
$SD(Z)$ at $y = 0.75$
x1
x2

$\sqrt{\phi_2}$ as a function of t_1 and t_2 intermediate values

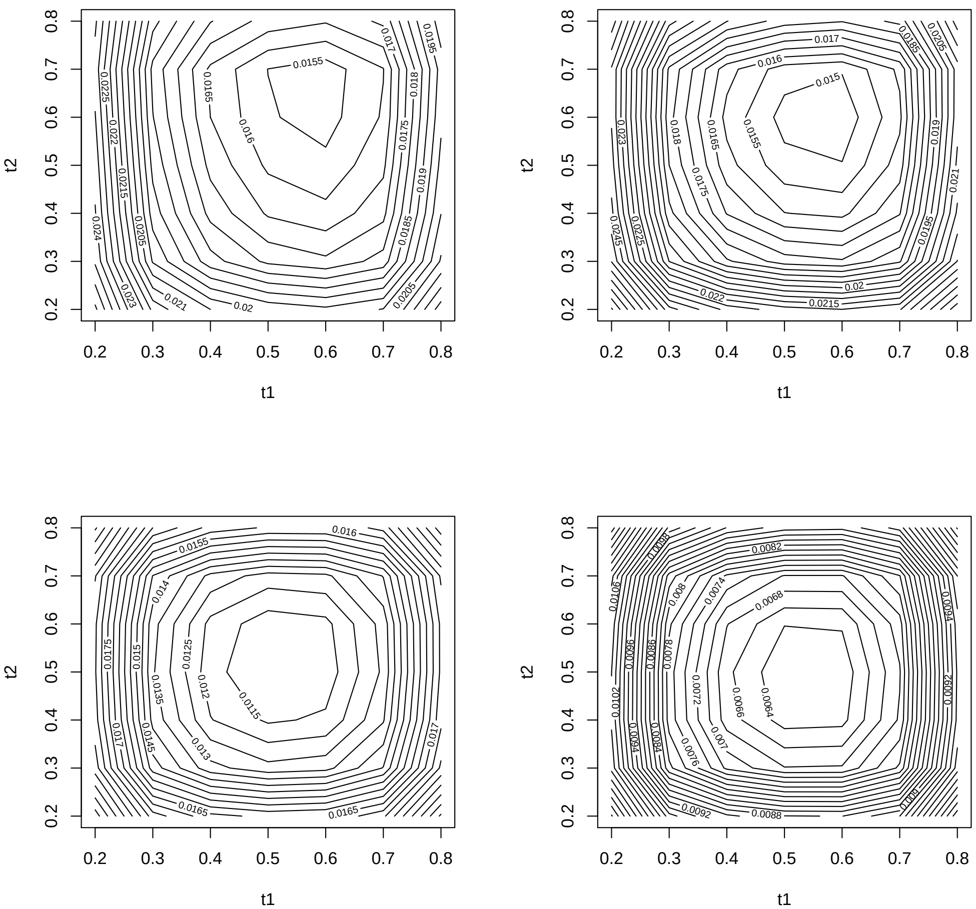


Iterative Systems: For example,

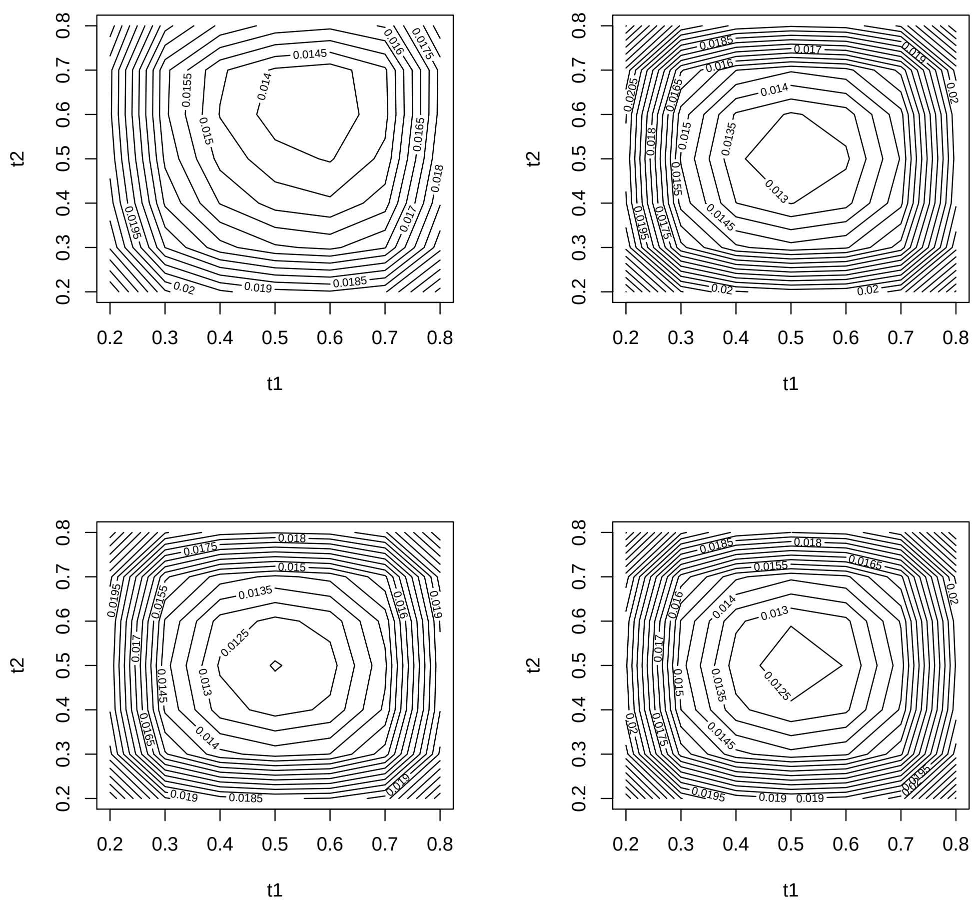


1. Begin E-loop
   (a) Draw $\boldsymbol{\beta}$ from B
   (b) Draw (x_1, x_2, x_3) from Ω_x
   (c) Begin V-loop
      i. Draw $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$
      ii. Begin model iteration
         A. Compute $t(x_2, x_3)$.
         B. Compute $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$
         C. Compute $z = z(x_1, x_2, \hat{y})$
         D. $x_2 \leftarrow z$
   (d) Compute $Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$
2. Compute $E_{\boldsymbol{\beta},x} Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$

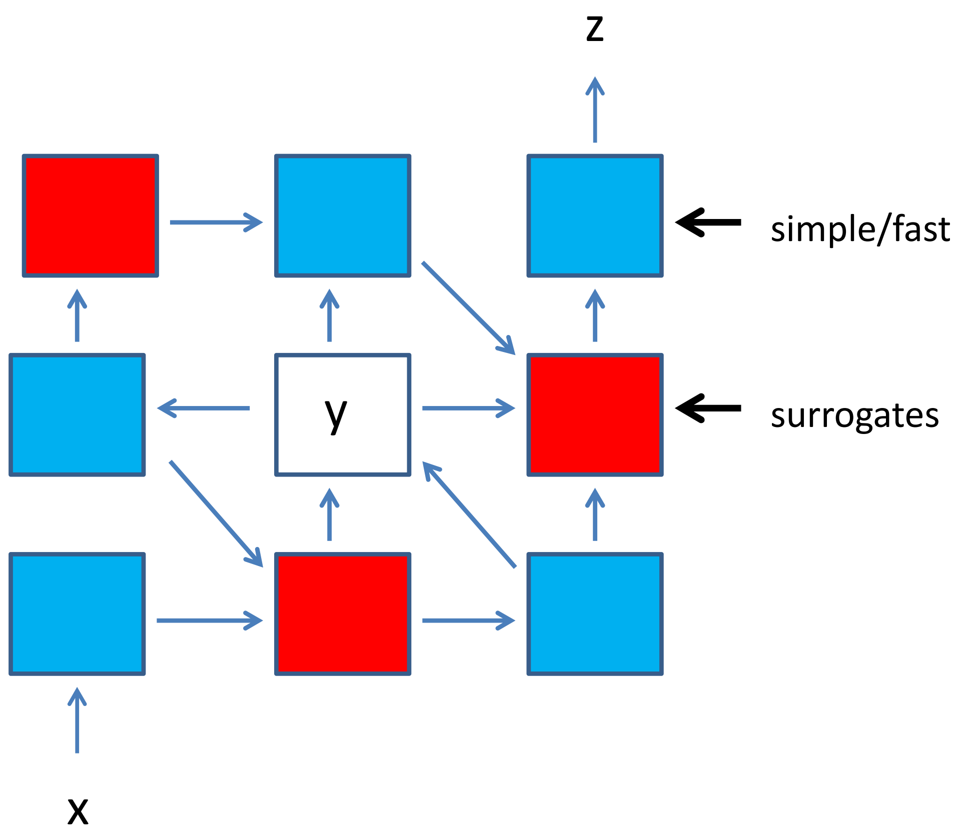
$\sqrt{\phi_1}$ as a function of t_1 and t_2 intermediate values, for the iterative version of the example model with 1 (for comparison), 2, 5 and 10 iterations



$\sqrt{\phi_2}$ as a function of t_1 and t_2 intermediate values, for the iterative version of the example model with 1 (for comparison), 2, 5 and 10 iterations



Iterative Systems: More generally,



1. Begin E-loop
   (a) Draw $\boldsymbol{\beta}$ from B
   (b) Draw (x_1, x_2, x_3) from Ω_x
   (c) Draw* surrogates
   (d) Begin V-loop
      i. Draw $\hat{\boldsymbol{\beta}}$
      ii. Execute $\rightarrow z$
   (e) Compute $Var_{\hat{y}}(z|...)$
2. Compute $E_{...}Var_{\hat{y}}(z|...)$

* Model structure determines the extent to which draws must be joint/functional (i.e. which arguments are constant throughout execution and which are not).

Some Major "Details":

- Cases 1 and 2
  - What kind of iterative design construction (analogous to point-exchange) is possible?
  - How should the experimental range of t be determined for iterative models?
- Case 2
  - How to best organize prior experiments to construct multiple metamodels?
  - What is the best way to represent functional draws (T, Z) for higher-dimensional arguments?
  - How is the computer model best "parsed" into sections to be represented by themselves or metamodels? (Not necessarily subroutine-by-subroutine as a modeler or programmer would view the code.)