Continuous-Time Emulation of Large Distributed Parameter Dispersion Models.

Peter C. Young

Systems and Control Group, Lancaster Environment Centre, Lancaster University, U.K (Tel: +44 1524 33558; e-mail: p.young@lancaster.ac.uk); and Integrated Catchment Assessment and Management Centre, Fenner School of Environment & Society, Australian National University College of Medicine, Biology & Environment, Canberra, Australia.

Abstract: The paper discusses the emulation of large, distributed parameter, computer models by low order, continuous-time, transfer function models obtained using the SRIVC method of identification and estimation for continuous-time models. This yields a minimally parameterized, reduced order, 'nominal' emulation model that often reproduces the dynamic behavior of the large model to a remarkable degree. In full *Dynamic Model Emulation* (DEM), the objective is to emulate the high order model over a whole, user-defined range of parameter values, so that it can act as a surrogate for the high order model in applications that demand fast, repeated solution, as in Monte Carlo simulation and sensitivity analysis, or be used as a low order model in automatic control system design and adaptive forecasting applications. Most of the paper deals with the 'stand-alone' emulation of two high order, distributed parameter, computer models for the transport and dispersion of solutes in water systems.

1. INTRODUCTION

This paper briefly reviews the concept of a *Dynamic Emulation Model* (DEM; also called a 'meta-model') and shows how the development of a DEM is facilitated by the use of continuous-time system identification and estimation. In particular, it shows how the *Simplified Refined Instrumental Variable* algorithm for identifying and estimating continuous-time models from discrete-time, sampled data (SRIVC) has been used in DEM synthesis, where it serves as the basis for model reduction using *Dominant Mode Analysis* (DMA) [Young, 1999]. Most of the paper is concerned with describing the successful low-order emulation of two high order, distributed parameter, computer models for simulating solute transport and dispersion in water systems. The paper also draws a distinction between 'nominal' DEM models that emulate the large simulation model for a specified, single set of its parameters; and the 'full' DEM that maps the relationship between the two models in a more complete manner involving a whole, user-specified range of parameter values.

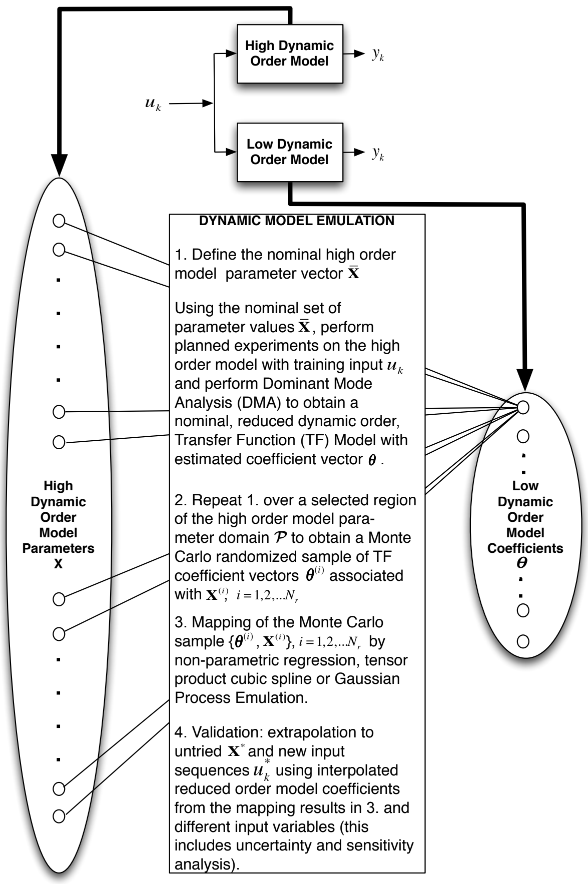
2. EMULATION MODELLING

The idea of full emulation modeling was suggested by Oakley and O'Hagan [2004] and Li et al. [2006], but this was restricted mainly within the context of static models and associated sensitivity analysis. Other recent references on this topic that address the idea of dynamic emulation modeling are Conti et al. [2007] and Bayarri et al. [2007a,b]. The present paper outlines the partly heuristic *Data-Based Mechanistic* (DBM) approach to dynamic emulation modeling that has been described and evaluated in detail by Young and Ratto [2009, 2011], to which the reader is directed for further information on its strengths and limitations. Here, the reduced order dominant mode model is obtained by DMA applied to the large simulation model and the mapping between the parameters of the large model and the coefficients of this reduced order model is in the form of a non-parametric (graphical) hyper-surface. Note that such an emulation model is limited to the defined domain of simulation model parameter values used in this analysis (although extrapolation outside this domain may be possible in certain circumstances if the mapping is parameterized).

The complete dynamic emulation model behaves like the high order simulation model and so it can usefully replace it in certain applications. For example, it can assist in (or even replace) conventional sensitivity analysis, which is normally required to investigate which of the many parameters that characterize the high order simulation model are most important in defining the models dynamic behaviour. And it can take the place of the simulation model in hydrological applications such as real-time flood forecasting [Beven et al., 2008] or the modeling of solute transport and dispersion in water bodies [Young, 2011]. It is the use of emulation modeling in this latter context that is considered in the present paper.

The four stages of the DBM emulation model development, as illustrated diagrammatically in Figure 1, are as follows:

(1) **Identification and Estimation of a Nominal Transfer Function (TF) Model**: On a nominal set of large model parameters ($\bar{\mathbf{X}} = (\bar{X}_1, \ldots, \bar{X}_p)$), *Dominant Mode Analysis* (DMA: see Young [1999]) is used to provide an identifiable, reduced order TF model that reproduces the response $\mathbf{y}(t)$ of the large model as well as possible. This large model response is obtained using a 'reference' set of forcing input data


High Dynamic Order Model
y_k
u_k
Low Dynamic Order Model
y_k
High Dynamic Order Model Parameters X
Low Dynamic Order Model Coefficients $\boldsymbol{\theta}$

DYNAMIC MODEL EMULATION

1. Define the nominal high order model parameter vector $\bar{\mathbf{X}}$

Using the nominal set of parameter values $\bar{\mathbf{X}}$, perform planned experiments on the high order model with training input u_k and perform Dominant Mode Analysis (DMA) to obtain a nominal, reduced dynamic order, Transfer Function (TF) Model with estimated coefficient vector $\boldsymbol{\theta}$.

2. Repeat 1. over a selected region of the high order model parameter domain $\mathcal{P}$ to obtain a Monte Carlo randomized sample of TF coefficient vectors $\boldsymbol{\theta}^{(i)}$ associated with $\mathbf{X}^{(i)}$; $i = 1,2,...N_r$

3. Mapping of the Monte Carlo sample $\{\boldsymbol{\theta}^{(i)}, \mathbf{X}^{(i)}\}$, $i = 1,2,...N_r$ by non-parametric regression, tensor product cubic spline or Gaussian Process Emulation.

4. Validation: extrapolation to untried $\mathbf{X}^*$ and new input sequences u_k^* using interpolated reduced order model coefficients from the mapping results in 3. and different input variables (this includes uncertainty and sensitivity analysis).

Fig. 1. A diagrammatic interpretation of full model emulation.

$\mathbf{u}(t)$ that sufficiently excites the model to allow for identification of the dominant dynamic modes, while $\bar{\mathbf{X}}$ is typically set at the mean of the distributions used for the parameters of the large simulation model. This reduced order TF model constitutes the *Nominal Emulation Model* (NEM).

(2) **MC Randomization**: *Monte Carlo* (MC) simulation analysis is carried out, by taking a training set $\mathbf{X}^{(i)}$, $i = 1, \ldots, N_r$, of independent, randomly chosen parameters from the large model, and computing the output response $\mathbf{y}(t)^{(i)}$ of the original large order model, always keeping the same set of forcing input data $\mathbf{u}(t)$ used in the first step. Then, for each MC computer experiment $\{\mathbf{u}(t),\ \mathbf{y}(t)^{(i)}\}$, a dynamic TF model with the same reduced order obtained in step 1. above is estimated, obtaining a MC set $\boldsymbol{\theta}^{(i)}$ of TF coefficients.

(3) **Mapping**: in order to emulate the large simulator at any untried sample $\mathbf{X}^*$, a mapping between the TF models estimated in step 2. and the $\mathbf{X}$ parameters is required. This mapping can be achieved in two different ways: *'stand-alone parameter mapping'* and *'response mapping'* modes.

(4) **Validation, uncertainty and sensitivity analysis**: the previous three steps yield the *Full Emulation Model* (FEM). This FEM can be validated by generating dynamic simulations $\mathbf{y}(\mathbf{u}^*(t)|\mathbf{X}^*, t)$ at untried $\mathbf{X}^*$, for any new sequence of forcing inputs $\mathbf{u}^*(t)$, and then comparing this with the FEM output response $\hat{\mathbf{y}}(t)$ to the same forcing inputs $\mathbf{u}^*(t)$. A complete statistical uncertainty estimation can be derived for the surrogate dynamic behaviour, as well as a comprehensive sensitivity analysis.

Note that 'response mapping' is a version of emulation approach where, for a given set of inputs and a set (grid) of large model parameters, a set of outputs is generated, and the emulator's response to this set of inputs is obtained by mapping using the stored outputs directly. This approach is not considered in the present paper but further details are available in other published papers: Young and Ratto [2011], Higdon et al. [2007], Bayarri et al. [2007b].

The NEM is quite simple since it relies mainly on the well-tried ability [e.g. Young et al., 1996, Young, 1999] of the TF identification and estimation routines in the CAPTAIN Toolbox[1] for Matlab$^{\mathrm{TM}}$ to perform the model reduction that is the core of DMA. As a result, the dynamic behaviour of the nominal emulation model is normally almost identical to that of a large linear computer simulation model and, depending on the nature of nonlinear dynamics, can reproduce well the behaviour of large nonlinear models using *State-Dependent Parameter* (SDP) estimation and models [e.g. see Young et al., 2009, Young, 2011, and the previous references therein]. On the other hand, the development of the FEM presents a much more complex computational problem because it requires the manipulation of data in multivariable parametric spaces of different, and potentially high, dimensions. And, as Young and Ratto Young and Ratto [2011] point out, it includes approximations that can reduce the emulation accuracy when compared with that of the NEM.

3. DBM EMULATION OF DISTRIBUTED PARAMETER SOLUTE TRANSPORT AND DISPERSION MODELS

In this paper, the emulation methodology outlined in the previous section is applied to the two large computer simulation models described below.

3.1 The Quasi-Distributed ADZ Model

The *Aggregated Dead Zone* (ADZ) model has been used for modelling solute dispersion in rivers for many years [Beer and Young, 1983, Young and Wallis, 1993]. The *Semi-Distributed* ADZ (SDADZ) model is constructed rather simply by a chain of suitably small ADZ elements connected in series, parallel or even feedback (should this relate to a physically meaningful situation). For example, if the concentration of conservative solute at the i^{th} reach of a river is denoted by $C_i(t)$, the equation for the transport and dispersion of the solute, over a river having n identical reaches, takes the form [Young, 2011]:

$$T\frac{dC_i(t)}{dt} = -C_i(t) + GC_{i-1}(t) + G_dC_{i+1}(t) \quad i = 1, 2, \ldots\ n \tag{1}$$

Alternatively, this model can be represented in TF form:

[1] The CAPTAIN Toolbox is freely available via the Internet: see `http://captaintoolbox.co.uk/Captain_Toolbox.html`.

$$C_i(t) = \frac{G}{1+Ts}C_{i-1}(t) + \frac{G_d}{1+Ts}C_{i+1}(t) \qquad i = 1, 2, \ldots\ n \tag{2}$$

or as the following 3^{rd} order *element* in a n dimensional state space model:

$$\begin{bmatrix} \dfrac{dC_{i-1}(t)}{dt} \\ \dfrac{dC_i(t)}{dt} \\ \dfrac{dC_{i+1}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{T} & \dfrac{G_d}{T} & 0 \\ \dfrac{G}{T} & -\dfrac{1}{T} & \dfrac{G_d}{T} \\ 0 & \dfrac{G}{T} & -\dfrac{1}{T} \end{bmatrix} \begin{bmatrix} C_{i-1}(t) \\ C_i(t) \\ C_{i+1}(t) \end{bmatrix} \tag{3}$$

where the output is defined by the state variable $C_i(t)$. The input to this whole system is then the input to the farthest upstream reach, denoted by $u(t) = C_0(t)$, and the output is that of the farthest downstream reach, denoted by $x(t) = C_n(t)$. If it is assumed that any pure advective time delays are lumped into a single time delay τ at the input, the complete deterministic model can be represented in the following general TF form:

$$x(t) = \frac{b_0 s^{m-1} + b_1 s^{m-2} + \ \ldots\ + b_m}{s^n + a_1 s^{n-1} + \ \ldots\ + a_n} u(t-\tau) \tag{4}$$

If G and G_d are selected so that the system is conservative, then $b_m = a_n$ and the overall steady state gain is unity.

The model (3) for large n and the time constant T small, is a 'quasi-distributed' model, in the sense that it describes a distributed system but is formulated using ordinary, rather than partial differential equations. However, its behaviour is very similar to partial differential equation equivalents, such as the well known *Advection Dispersion Equation* (ADE). Indeed, if the SDADZ model is discretized it has similarity with the finite difference solution of the ADZ model and has response characteristics similar to the 'Transient Storage' model described in section 3.2. Because of its form, the SDADZ model is easy to code in Matlab and this was used in the emulation study described in the next section 3.1.1.

3.1.1 SDADZ Emulation Modelling and Results Here, the NEM analysis is carried out using an input signal in the form of two repeated pulses of period 1000 hours and amplitude 180 entering at reach 1. Although this is certainly not an optimal input from a statistical identification and estimation standpoint, it is sufficient to produce good TF emulation models. For instance, the continuous-time SRIVC identification results for reach 5, as produced by the CAPTAIN rivcbjid routine, are as follows (shown verbatim):

```
den   num   del       YIC        Rt2
 5     6     1     -25.7517   1.000000
 5     6     0     -18.6377   0.999998
 5     5     0     -22.4703   0.999998
 4     5     1     -19.1426   0.999976
 4     5     0     -11.6862   0.999976
 4     4     0     -18.1096   0.999976
```

Here, YIC is the order identification criterion [Young, 1989] (most negative value identifies the best model); and R_T^2 is the coefficient of determination based on the simulated output (unity indicates a perfect fit). Although this suggests that the [5 6 1] model is marginally better, the [5 5 0] model is not only almost as good but it has one less parameter and, most importantly, it proves superior in the case of reach 40, where the best identified model is [5 5 79].

To summarize, therefore, this initial, NEM analysis, which examines the extremes of the large simulation model response characteristics, shows that a [5 5 τ] continuous-time TF model of the form;

$$\hat{x}(t) = \frac{\hat{b}_0 s^4 + \hat{b}_1 s^3 + \hat{b}_2 s^2 + \hat{b}_3 s + \hat{b}_4}{s^5 + \hat{a}_1 s^4 + \hat{a}_2 s^3 + \hat{a}_3 s^2 + \hat{a}_4 s + \hat{a}_5} u(t-\tau) \tag{5}$$

provides very good emulation at both extremes. For reach 5, $\tau = 0$ and the parameter estimates, obtained using the SRIVC option in the CAPTAIN rivcbj routine, are (note that $a_5 = b_4$ and mass is conserved):

$$\begin{aligned} &\hat{a}_1 = 0.5003;\ \hat{a}_2 = 0.0772;\ \hat{a}_3 = 0.00335 \\ &\hat{a}_4 = 4.6033 \times 10^{-5};\ \hat{a}_5 = 1.7265 \times 10^{-7} \\ &\hat{b}_0 = 7.0278 \times 10^{-4};\ \hat{b}_1 = -4.5845 \times 10^{-4} \\ &\hat{b}_2 = 1.0403 \times 10^{-3};\ \hat{b}_3 = 3.1639 \times 10^{-5} \\ &\hat{b}_4 = 1.7265 \times 10^{-7} \end{aligned} \tag{6}$$

while for reach 40, $\tau = 79$ and the estimates are:

$$\begin{aligned} &\hat{a}_1 = 0.06116;\ \hat{a}_2 = 0.001475;\ \hat{a}_3 = 1.6720 \times 10^{-5} \\ &\hat{a}_4 = 8.7469 \times 10^{-8};\ \hat{a}_5 = 1.6736 \times 10^{-10} \\ &\hat{b}_0 = 1.2088 \times 10^{-5};\ \hat{b}_1 = 1.0735 \times 10^{-6} \\ &\hat{b}_2 = 9.9738 \times 10^{-8};\ \hat{b}_3 = 2.8188 \times 10^{-9} \\ &\hat{b}_4 = 1.6736 \times 10^{-10} \end{aligned} \tag{7}$$

In both cases, the explanation of the large model response, when validated with a single impulse input of 70 for two hours, is almost perfect, as shown in Figure 2, with $R_T^2 = 0.999999$ in both cases. Similarly good validation results are obtained for other inputs.

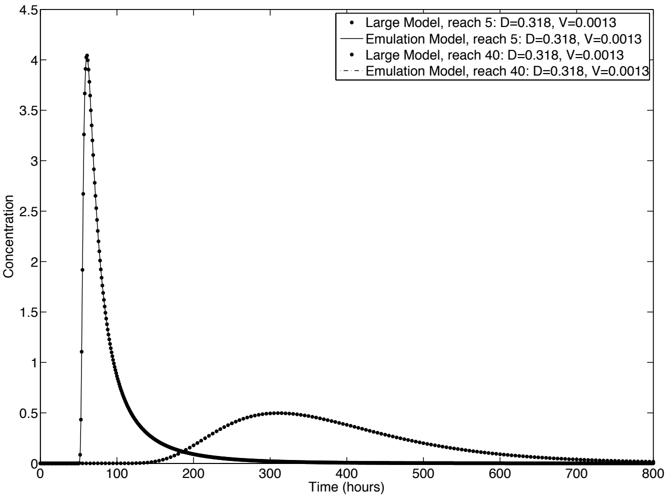


Fig. 2. Nominal emulation validation for the SDADZ model at reaches 5 and 40 by continuous-time RIVC estimated 5^{th} order TF models.

The FEM mapping analysis is carried out using the same input forcing function $u(t)$, which is shown in Figure 3 together with three typical simulation model responses measured at reach 5 for different values of the *Dispersion Coefficient D* and and *Velocity V*. It involves 850 separate TF model identification and estimation runs, using 50 equally spaced values of D over the range $\{0.030 -$

1.5} m^2/sec and 17 equally spaced values of V over the range $\{0.00045 - 0.0014\}$ m/sec. Because the computational burden is not too large in this case, it is possible to carry out the mapping over this grid of simulation model parameter values and so ensure good mapping coverage without having to resort to MC randomization. More specifically, the TF model identification at each combination of D and V values is based on the $[5\ 5\ \tau]$ model with τ considered in the range $\{0 - 2\}$ sec. These 3 calls to the rivcbj routine in CAPTAIN take about 10 seconds on a quad-core Mac Pro computer, so that the overall computation time for the mapping analysis is about 2.3 hours.

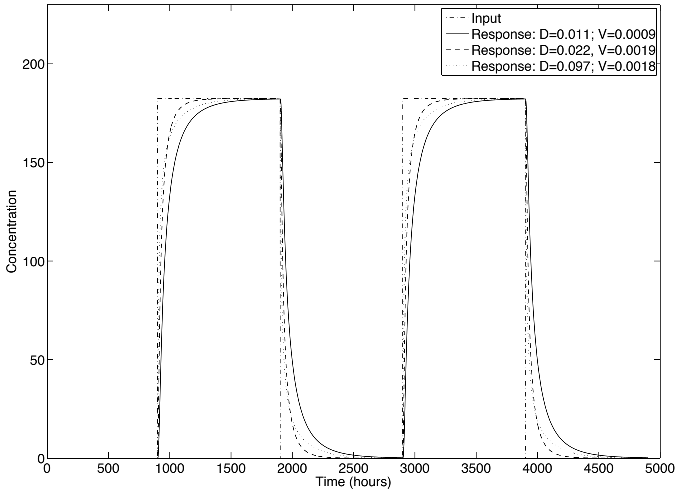


Fig. 3. Emulation experimental data at reach 5 for 3 combinations of the dispersion coefficient, D, and velocity, V.

Given the results of these mapping experiments, the mapping relationships are obtained using the interp2 routine in Matlab, with the 'spline' option. Figure 4 is a three dimensional plot of the resulting mapping surface for the five TF denominator parameters $a_i, i = 1, 2, \dots\ 5$. Note that the mapping surface in Figure 4 is quite smooth for $D > 0.1 m^2/sec$ but there is a quite sharp change at smaller values than this, suggesting that a finer grid might be necessary in this region. However this region is not important in the present case and this has not been investigated further.

The final stage of the full emulation analysis is validation on interpolated values of the parameters and Figure 5 shows the validation results for a forcing function of the more general type that might be expected if the model was being employed for the evaluation of pollutant transport and dispersion. In both cases, the emulation is exceptional, with R_T^2 values greater than 0.999.

3.2 The Transient Storage (TS) model

The TS model is a development of the ADE model and is expressed as two coupled equations that describe the transport and dispersion of a solute in a river channel: a standard, partial differential ADE for the concentration $C(t, x)$ of solute in the main channel, linked by an exchange coefficient α to an ordinary differential equation that describes the variation of the solute concentration $C_s(t)$ in the storage zone:

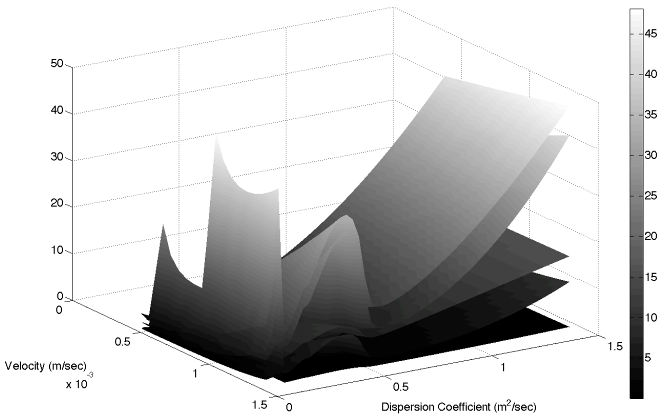


Fig. 4. DBM stand-alone emulation: parameter mapping for reach 5: 3 dimensional plot of the five TF denominator parameters $a_i, i = 1, 2, \dots\ 5$, as functions of the dispersion coefficient, D, and the velocity, V.

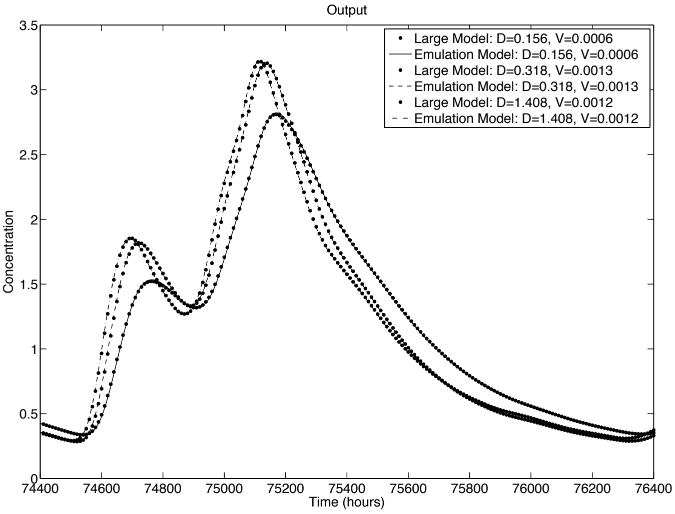


Fig. 5. Typical validation result for the SDADZ emulation model.

$$\begin{aligned}\frac{\partial C(t,x)}{\partial t} &= -\frac{Q}{A}\frac{\partial C(t,x)}{\partial x} + \frac{1}{A}\frac{\partial}{\partial x}\left(AD\frac{\partial C(t,x)}{\partial x}\right)\\ &\qquad + \alpha\left(C_s(t,x) - C(t,x)\right) \qquad (8)\\ \frac{dC_s(t,x)}{dt} &= -\alpha\frac{A}{A_s}\left(C_s(t,x) - C(t,x)\right)\end{aligned}$$

Here t denotes time; x is the distance along the main channel; Q is the volumetric flow; A is the main channel cross-sectional area; α is the exchange coefficient; A_s is the storage zone cross-sectional area; and D is the longitudinal dispersion coefficient.

The OTIS model developed by Runkel et al. [1996] expands the original TS model by including the solute's sorption and release, lateral inflow etc. It has been selected to illustrate the application of the proposed emulation software environment because it is a well known and very popular simulation model that provides a reasonable example for the full emulation of a linear, distributed parameter system. Crucially for the present study, however, the developers of OTIS provide a Fortran simulation module and full documentation, including Fortran source code and several binaries (see `http://water.usgs.gov/software/OTIS/`).

3.2.1 DBM Emulation Software Development using the OTIS Model Since the OTIS model is very popular in practical applications, it has been used in the continuing development of a software system within Matlab to automate the DBM emulation process [Tych and Young, 2011]. The emulation framework uses a p-dimensional grid defined by the user, where each of the dimensions represents one of the parameters of the large model, and is split into a number of points defining the nodes. In the current version, the grid defines a hyper-cube in p dimensions, uniformly divided along each of its dimensions.

The uniformity of this large model parameter grid is dictated by the requirement for flexible visualisation of the estimated low order model parameters that are embedded within the DBM emulation approach. However, the large model parameters can be sampled within the large model parameter space in a variety of ways, and the uniform grid required for visualisation can be 'filled in' after the simulation and estimation stages, during the mapping and DBM identification stage. For example the user could use a (potentially fine) uniform grid, subsampled using any available strategy: e.g. a version of Monte Carlo sampling, as in Step 2 (section 2) and then, during Step 3, apply an interpolation procedure, such as the Smoothing Spline *Analysis of Variance* (ANOVA) procedure used in Young and Ratto [2011], to fill the remaining grid nodes.

Every stage of the emulation procedure has its associated p-dimensional data structure constructed on the same p-dimensional grid of the large model parameters. There are three data structures, which contain the results of the subsequent stages in the emulation process:

(i) The *ParSpace* structure contains the parameters $\mathbf{X}$ of the large model: each node I includes the values $\mathbf{X}^{(I)}$, where I defined above is the set of indices spanning the entire p-dimensional grid-space.

(ii) The *OutSpace* structure contains outputs of the large model $\mathbf{y}(t)^{(i)}$ given its chosen input(s), obtained from simulation of the large model for the selected subset $\mathbf{X}^{(i)}$; $i \subseteq I$ of its parameter sets - the *ParSpace* nodes.

(iii) The *EstSpace* structure of the same dimensionality contains in its nodes, indexed by i, the low order TF models (model objects, potentially multiple), including parameter vectors $\boldsymbol{\theta}^{(i)}$ and their uncertainty information, as well as other quantities contained in the standard Matlab Identification Toolbox *idpoly* objects (see for example `http://www.mathworks.com/help/toolbox/ident/ref/idpoly.html`), which are also returned by the Captain Toolbox TF identification (rivcbjid) and estimation (rivcbj) routines. These employ the input-output data contained in the *OutSpace* structure. Since in general the *OutSpace* structure's grid may be sub-sampled at the nodes $i \subseteq I$, the estimation is carried out at the nodes i where simulations were carried out; subsequent p-dimensional interpolation (or approximation/extrapolation, as appropriate), expanding the mapping to the whole grid I, can then be used at the mapping stage and so complete the grid for visualisation purposes.

These structures can be linked with the steps in the DBM emulation modelling procedure outlined in section 2. In particular, stage 1. relates not only to the sampling of the large model parameter space, but also to the identification and estimation of the low order TF model. There is also an unavoidable preparatory step of experimenting with the feasible TF model structures over the full large model parameter range in order to ensure that the entire parameter sub-space of the model can be adequately mapped into its TF counterpart. Mapping is necessarily limited to the user-specified range of large model parameters that are considered adequate for the required modelling task. Naturally, the selected TF model structure needs to emulate the large model well over the whole of this range.

If it is required, the most time consuming processes of 'filling' the, potentially subsampled, *OutSpace* and *EstSpace* structures (large model simulations and estimation of the TF models) can be parallelised quite easily, either by manually splitting the *ParSpace* into 'slices' run as separate processes, or through the use of the Matlab parallel tools. The linking of the *ParSpace* and *EstSpace* structures is the inherent part of the mapping step of DBM emulation. The full mapping visualisation requires TF model parameter values being available at all the nodes on the grid as indexed by I. TF parameters will be either estimated from the Input-Output data (over the $i \subseteq I$ subset of the grid), or interpolated (or approximated) over the remainder of the full grid: $I \setminus i$ during the mapping process.

The validation as well as uncertainty and sensitivity analysis are facilitated by the existing data structures, as shown in the example below. Each of the plots, including annotations, is produced automatically by a single call to a visualisation function, so that the whole framework would easily lend itself to being absorbed into a full GUI-based package.

3.2.2 OTIS Emulation Modelling and Results Figures 6 and 7 illustrate the emulation of the OTIS model using the software system described in the previous sub-section (further details are available in Tych and Young [2011]). Figure 6 shows the 3D projection of the mapping of the a_1 parameter of the emulating TF from α (termed ALPHA in the OTIS parameters list) and D (termed DISP in OTIS) parameters of the full OTIS model, with A_s (AREA2 in OTIS) fixed. Note that in this and subsequent automatically generated figures, the original names of OTIS parameters are used, and clarified in the captions. All the plots generated by the software are labelled automatically, using the OTIS configuration file nomenclature.

Finally, Figure 7 shows a typical validation result for the the DBM emulation of the OTIS model. The shaded area is the 95% uncertainty band and we see that the full model simulation line is practically indistinguishable from the emulator output.

4. CONCLUSIONS

This paper has introduced the concept of dynamic emulation modelling (DEM) and shown how the data-based

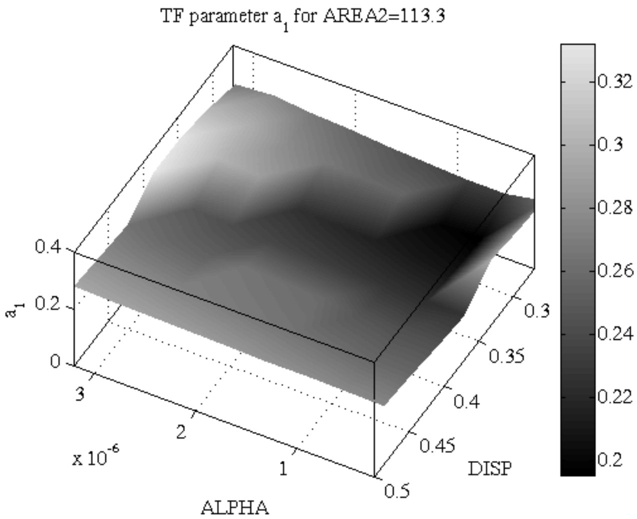


Fig. 6. 3D projection of the a_1 parameter mapping from α (ALPHA) and D (DISP).

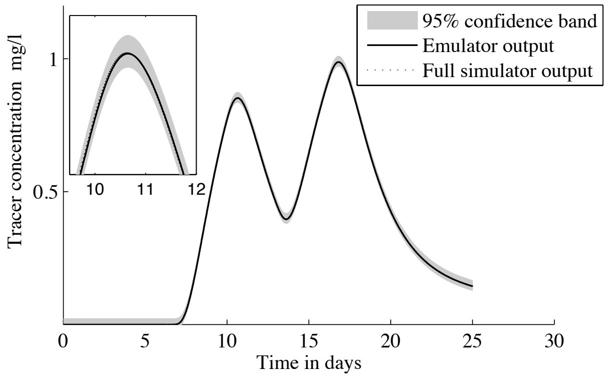


Fig. 7. A typical validation result for the the OTIS emulation model.

mechanistic (DBM) modelling approach to DEM is able to exploit the simplified refined instrumental variable method of continuous-time transfer function model identification and estimation (SRIVC) for dominant mode analysis (DMA) and the ‘stand-alone’ emulation of large, distributed parameter transport and dispersion models. Other methods of model reduction could be used but the use of the SRIVC algorithm in DMA offers some advantages: e.g. it can out-perform existing methods, such as balanced model reduction and proper orthogonal projection, as illustrated by its application to the ‘Gasifier’ model reduction demonstration example in Matlab [see Young, 2011, pages 362-365] and, unlike these methods, it does not require direct access to the model equations.

REFERENCES

M. J. Bayarri, J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, J. Sacks R. J. Parthasarathy, R. Paulo, and D. Walsh. Computer model validation with functional output. *The Annals of Statistics*, 35: 1874–1906, 2007a.

M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49:138–154, 2007b.

T. Beer and P. C. Young. Longitudinal dispersion in natural streams. *Am. Soc. of Civil Eng., Jnl. Env. Eng.*, 109:1049–1067, 1983.

K. J. Beven, P. C. Young, and D. Leedal. Computationally efficient flood water level prediction (with uncertainty). In *Proceedings European Conference on Flood Risk Management*, Oxford, 2008.

S. Conti, J.P. Gosling, J. Oakley, and A. O’Hagan. Gaussian process emulation of dynamic computer codes. 2007. mimeo (submitted).

D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model validation using high-dimensional outputs. In J. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 8*. Oxford Univ. Press, 2007.

G. Li, S. Azarm, A. Farhang-Mehr, and A. R. Diaz. Approximation of multiresponse deterministic engineering simulations: a dependent metamodeling approach. *Structural and Multidisciplinary Optimization*, 31:260–269, 2006.

J. E. Oakley and A. O’Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal Royal Statistical Society, Series B*, 66:751–769., 2004.

R. L. Runkel, K. E. Bencala, R. E. Broshears, and S. C. Chapra. Reactive solute transport in small streams: I. development of an equilibrium-based simulation model. *Water Resources Research*, 32(2):409–418, 1996.

W. Tych and P. C. Young. A Matlab software framework for dynamic model emulation. *Environmental Modelling & Software*, (in press), 2011.

P. C. Young. Recursive estimation, forecasting and adaptive control. In C. T. Leondes, editor, *Control and Dynamic Systems*, pages 119–166. Academic Press: San Diego, 1989.

P. C. Young. Data-based mechanistic modelling, generalised sensitivity and dominant mode analysis. *Computer Physics Communications*, 117:113–129, 1999.

P. C. Young. *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner.* Springer-Verlag, Berlin., 2011.

P. C. Young and M. Ratto. A unified approach to environmental systems modeling. *Stochastic Environmental Research and Risk Assessment*, 23:1037–1057, 2009.

P. C. Young and M. Ratto. Statistical emulation of large linear dynamic models. *Technometrics*, 53:29–43, 2011.

P. C. Young and S. G. Wallis. Solute transport and dispersion in channels. In K. J. Beven and M. J. Kirkby, editors, *Channel Network Hydrology*, pages 129–173, Chichester, 1993. J. Wiley.

P. C. Young, S. Parkinson, and M. J. Lees. Simplicity out of complexity: Occam’s razor revisited. *Journal of Applied Statistics*, 23:165–210, 1996.

P. C. Young, D. Leedal, K. J. Beven, and C. Szczypta. Reduced order emulation of distributed hydraulic models. In *Proceedings 15th IFAC Symposium on System Identification SYSID09, St Malo, France*, 2009.