# Extending Monolix to use models with Partial Differential Equations

**Emmanuel Grenier, Violaine Louvet, Paul Vigneaux**
INRIA, NUMED Team. Lyon, France

## Goal : Monolix & Complex models

If one wants to parametrize complex systems like :
- systems with a large number of ODEs
- Partial Differential Equations
- Multi-agent, Multi-physics , etc

**following problems arise** :
- computational time to obtain the solution of the model for a given set of parameters is very long ;
- up to now MLXTRAN can not implement these models.

Consequently, **parameters estimation of such models are not doable**, in practice.

We here recall existing solution and deliver (with codes available online) a generalization allowing to tackle more general complex problems[1]. This is done by means of coupling of Monolix with a computationally fast external solvers of the models.

## Existing coupling : Monolix/C++

Monolix allows to implement the model in a “Matlab script” (see page 19 of [2]). Recall that the goal of this (”main”) script is to compute, for a given set of parameters, the associated solution of the model. Consequently, it is possible :
- to call inside this script a Matlab function (ODE45, ...)
- to call another Matlab (”child”) script implemented by the user if the model is more complicated and not implementable with a built-in Matlab function
- to call a true “external” routine, compatible with Matlab ; for instance a C++ solver, *e.g.* this was done only for ODEs in [3]

This kind of approach is only doable in Monolix if associated solvers give the solution in a small computational time. In other words, it is limited to ODE or a small number of ODEs.

For truly complex models mentioned in the introduction above, a new conceptual breakthrough is needed.

## Precomputation–Interpolation

The breakthrough implemented here is a precomputation of the solution of the complex model (on a discretized grid of the parameters space) and its storing in a database. Then, the evaluation of the model is replaced by an interpolation in the database (instead of a full computation, as above).

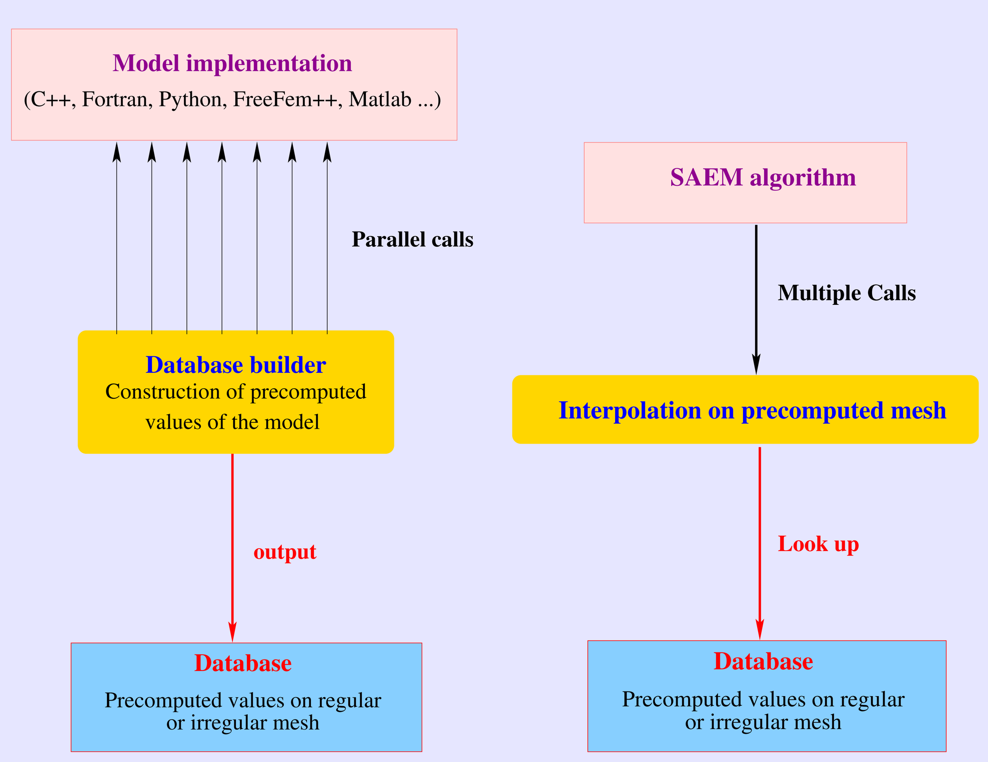


Figure 1. Step – Offline Building of the database (Left) and Step – SAEM with interpolation (Right).

## Step – Offline Building of the database

### Homogeneous database

We consider an hypercube $C_{init}$ which describes the space of parameters. Suppose for instance that the model has 3 parameters $P_1, P_2, P_3$, the hypercube is defined as ($P_i \in [x_{min,i}, x_{max,i}]$)

$$C_{init} = \Pi_{i=1}^{N}[x_{min,i}, x_{max,i}], \quad N = 3 \tag{1}$$

The database is filled with :
- the coordinates of summits of $C_{init}$
- the solution of the model associated to the summits

Then, $C_{init}$ is subdivided in a dyadic way in each direction (i.e. for each parameter). Every new summit and its associated solution are then added to the database.

This dyadic subdivision is then repeated iteratively up to a desired precision. Indeed, the more points there are in the database, the more precise will be the interpolation. Of note, this procedure :
- is fully parallelisable
- stores the final database in a ”.mat” binary Matlab format.

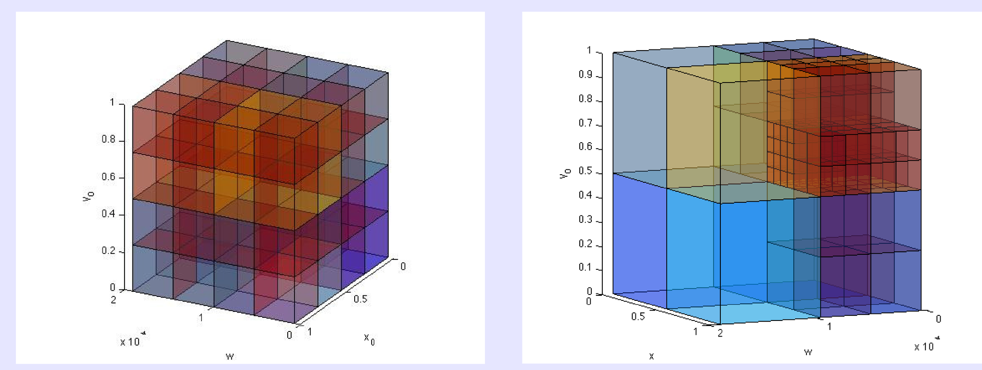

Figure 2. Homogeneous (Left) and heterogeneous (Right) databases.

### Heterogeneous database

In order to optimize the storage/computational cost and the parameter sensitivity, it is natural to improve the previous procedure by building an heterogeneous database. The idea is to refine the mesh in zones where the solution varies a lot with respect to parameters and, on the contrary, to have a coarser mesh where the sensitivity is low.

This ”heterogeneous” method results in the following gain : for a given precision, storage/computational cost of the heterogeneous database is much smaller than for an homogeneous database. This method is also fully parallelisable.

## Step – SAEM with interpolation

We are now in position to use Monolix with fast complex models.
The idea is to adopt aforementioned framework where a “main” Matlab script is used : this script calls a “child” Matlab script which does the computation of the solution by an interpolation in the database :
- search in the database the neighbours surrounding the “point” to evaluate
- compute the associated interpolation to obtain the solution of the model.

Examples of such scripts can be found on webpage [4].
Remark : such precomp./interpolation approach is also applied to a set of 29 ODEs, and presented in this conference : see [7]. It is a simpler case than the one presented below.

## Application : a PDE for invasion

We consider the following general model, the so called KPP equation [5] :

$$\partial_t u - \nabla.(D\nabla u) = Ru(1-u), \quad \forall t>0, \forall x \in \Delta \tag{2}$$

$$u(0,x) = 1_{|x-x_0|\leq\varepsilon}, \quad \text{with } \varepsilon \text{ fixed.} \tag{3}$$

Parameters : $R$ (reaction coefficient), $D$ (diffusion coefficient), $x_0$ (localisation of the initial “invaded zone”).
This model can be applied to numerous fields to better understand propagation phenomena (flame propagation, species invasion, tumour growth[6], etc) thanks to the existence of particular solutions called ”*travelling waves*”.
The unknown concentration is $u(x) \in [0;1]$ and we are interested in the size of the ”invaded zone” :

$$S(t) = \int_\Delta u(t,x)dx. \tag{4}$$

For sake of simplicity, we present the results in the one dimensional case ($\Delta = [0;1]$) but the same approach can be done with 2D or 3D images ($\Delta \in \mathbb{R}^2$ or $\mathbb{R}^3$, respectively).

Note that the equation is left unchanged if we multiply $D$ and $R$ by some constant and divide time by the same constant. This reduces by one the number of parameters.

We consider a virtual population of solutions of the KPP equation, assuming Gaussian distributions on its parameters, and adding noise. We then try to recover the distributions of the parameters using the approach database/Monolix-interpolation.
The homogeneous database can be freely downloaded on webpage [4].

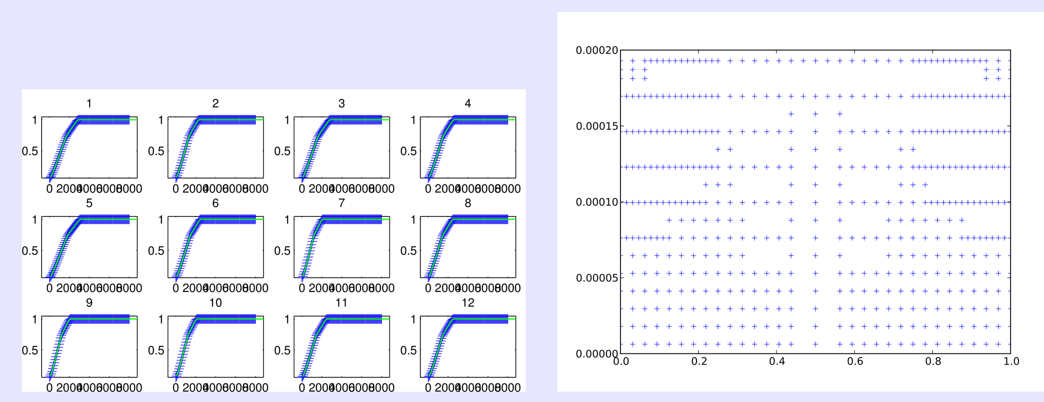

Figure 3. Left : digital (blue) and estimated (green) population. Right : inhomogeneous database ($x_0, w = D/R$)

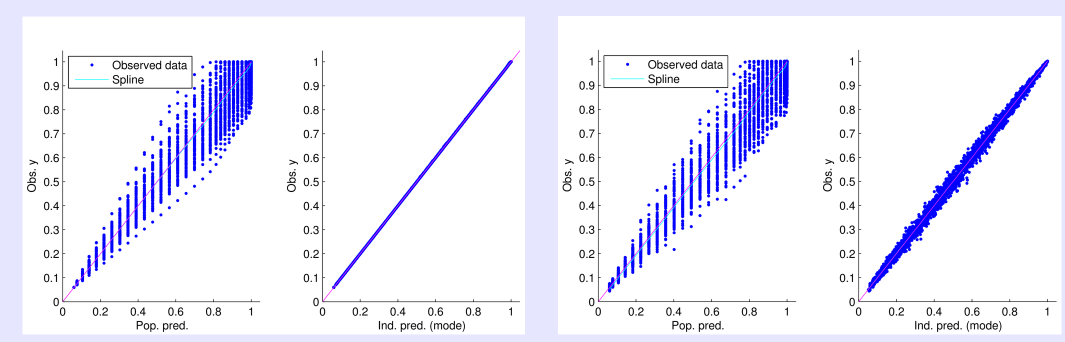

Figure 4. Monolix output of observed *v.s.* predicted population and individual data. Case with no noise (Left) and with 10% noise (Right).

| | ”Exact” case | Homog. DB | Heterog. DB |
|---|---|---|---|
| Offline cost | None | $38mn28s$ | $17mn40s$ |
| SAEM cost | $\sim 23d3h$ | $7mn30s$ | $8mn30s$ |
| Total cost | $\sim 23d3h$ | $45mn58s$ | $26mn10s$ |

Comparison of the CPU cost of the exact case (with the full EDP solver), with an homogeneous database and with an heterogeneous database (with the same precision but less points).

## Perspectives

This new method, combining SAEM algorithm and a precomputation step, is the first demonstration of parameters estimation of PDE thanks to a SAEM algorithm in Monolix. Specifically, we will use it to take into account the strong spatial behavior of some physical or biological phenomena, *e.g.* brain tumor growth as illustrated below. But more generally, it opens the path to the use of numerous new types of models and applications in population approaches.

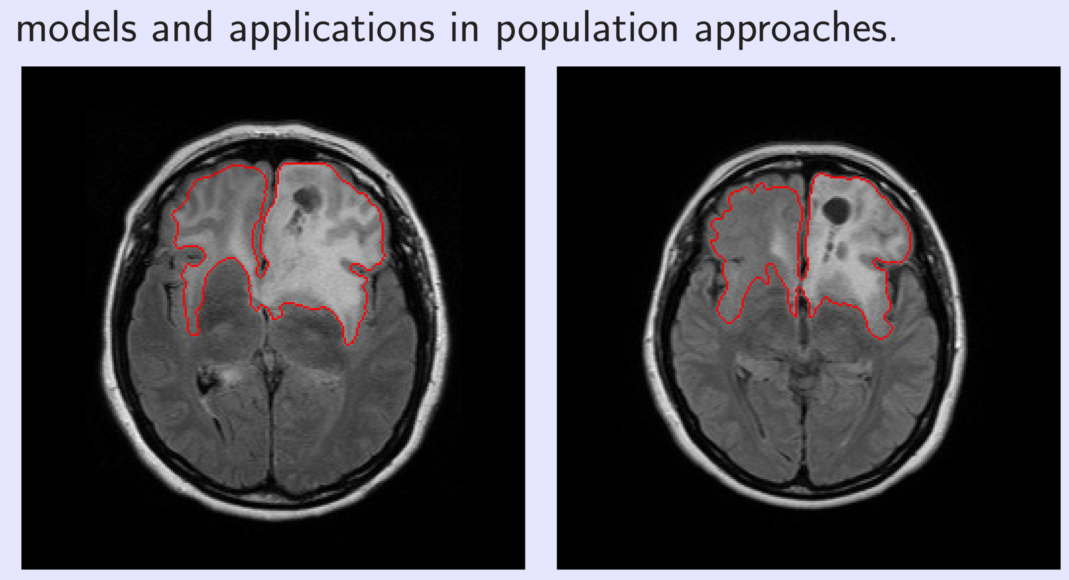

Figure 5. MRI of low grade glioma with strong spatial behavior, for the same patient at two differents dates : 2007 (Left) and 2008 (Right).

## Références

[1] E. Grenier, V. Louvet and P. Vigneaux. *Parameter estimation in non-linear mixed effects models with SAEM algorithm : extension from ODE to PDE. INRIA RR-8231 (Submitted 2012).*

[2] Lixoft Team. *Monolix Version 4.1.2 – User’s Guide*, March 2012

[3] B. Ribba et al.. *Monolix benefits from external modules to manage complex ODE.* Page Meeting 2008, Marseille. www.page-meeting.org/?abstract=1367

[4] Numed Team. *Software section*, http ://www.umpa.ens-lyon.fr/∼numed/software.html

[5] A. Kolmogoroff, I. Petrovsky, and N. Piscounoff. *Etude de l’equation de la diffusion avec croissance de la quantite de matiere et son application a un probleme biologique.*, Bulletin de l’universite d’Etat a Moscou. Section A, I(6)-1-26, 1937.

[6] K.R. Swanson et al.. *Virtual and real brain tumors : using mathematical modeling to quantify glioma growth and invasion*, J Neurol Sci, 2003.

[7] C. Barthelemy, M. Lavielle. *New methods for complex models defined by a large number of ODEs. Application to a Glucose/Insulin model*, Page Meeting 2013, Glasgow.