Lecture 2: Inverse Crimes, Model Discrepancy and Statistical Error Modeling

Erkki Somersalo

Case Western Reserve University

Cleveland, OH

Dynamic model

Forward model

$$\frac{dx}{dt} = f(t, x, \theta), \quad x(0) = x_0, \tag{1}$$

- $x = x(t) \in \mathbb{R}^n$ is the *state vector*,
- $\theta \in \mathbb{R}^k$ is the unknown, or poorly known *parameter vector*,
- $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$ is the *model function*
- x_0 possibly unknown, or poorly unknown *initial value*.

Observation model

Data: discrete noisy observations, may depend on the parameter vector:

$$b_j = g(x(t_j), \theta) + n_j, \quad t_1 < t_2 < \dots, \tag{2}$$

- $g : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ is the *observation function*
- n_j is the *observation noise*

The inverse problem: *Estimate the state vector and the parameter vector,* $(x(t), \theta)$*, based on the observations.*

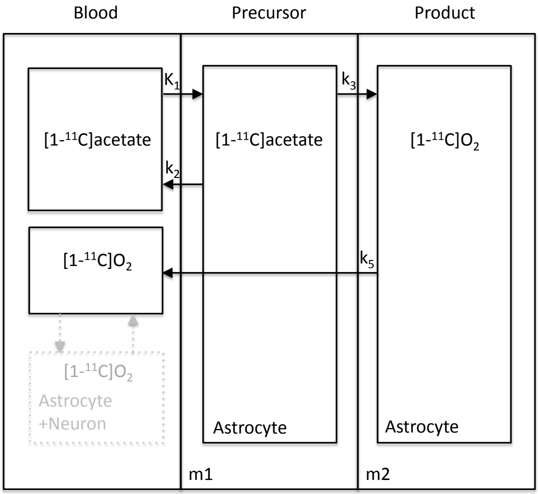
Two motivational problem

The dynamical system of acetate metabolism in brain by PET scan data:

$$\begin{aligned} \frac{dm_1}{dt}(t) &= K_1 c(t) - (k_2 + k_3) m_1(t) \\ \frac{dm_2}{dt}(t) &= k_3 m_1(t) - k_5 m_2(t) \end{aligned}$$

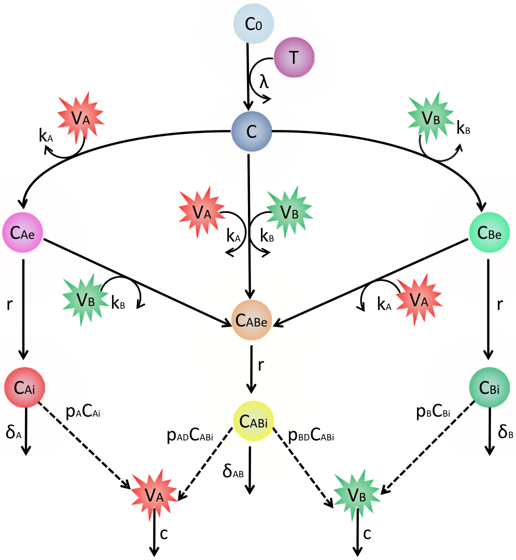
Observation: Noisy measurements of

$$c(t_j), \quad m(t_j) = V_0 c(t_j) + m_1(t_j) + m_2(t_j).$$



Blood
Precursor
Product
K1
[1-11C]acetate
[1-11C]acetate
k3
[1-11C]O2
k2
[1-11C]O2
k5
[1-11C]O2
Astrocyte
+Neuron
Astrocyte
Astrocyte
m1
m2

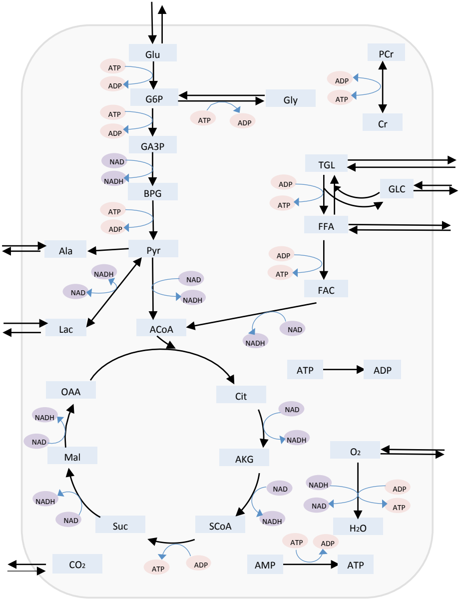
HIV-1 strain competition assay



HIV-1 strain competition assay

$$
\begin{aligned}
\dot{C} &= \lambda C_0 T - (k_{\mathrm{A}} V_{\mathrm{A}} + k_{\mathrm{B}} V_{\mathrm{B}} + \eta k_{\mathrm{A}} k_{\mathrm{B}} V_{\mathrm{A}} V_{\mathrm{B}}) C \\
\dot{C}_{\mathrm{Ae}} &= k_{\mathrm{A}} V_{\mathrm{A}} C - k_{\mathrm{B}} V_{\mathrm{B}} C_{\mathrm{Ae}} - r C_{\mathrm{Ae}} \\
\dot{C}_{\mathrm{Ai}} &= r C_{\mathrm{Ae}} - \delta_{\mathrm{A}} C_{\mathrm{Ai}} \\
\dot{C}_{\mathrm{Be}} &= k_{\mathrm{B}} V_{\mathrm{B}} C - k_{\mathrm{A}} V_{\mathrm{A}} C_{\mathrm{Be}} - r C_{\mathrm{Be}} \\
\dot{C}_{\mathrm{Bi}} &= r C_{\mathrm{Be}} - \delta_{\mathrm{B}} C_{\mathrm{Bi}} \\
\dot{C}_{\mathrm{ABe}} &= \eta k_{\mathrm{A}} k_{\mathrm{B}} V_{\mathrm{A}} V_{\mathrm{B}} C + k_{\mathrm{A}} V_{\mathrm{A}} C_{\mathrm{Be}} + k_{\mathrm{B}} V_{\mathrm{B}} C_{\mathrm{Ae}} - r C_{\mathrm{ABe}} \\
\dot{C}_{\mathrm{ABi}} &= r C_{\mathrm{ABe}} - \delta_{\mathrm{AB}} C_{\mathrm{ABi}} \\
\dot{V}_{\mathrm{A}} &= p_{\mathrm{A}} C_{\mathrm{Ai}} + p_{\mathrm{AD}} C_{\mathrm{ABi}} - c V_{\mathrm{A}} \\
\dot{V}_{\mathrm{B}} &= p_{\mathrm{B}} C_{\mathrm{Bi}} + p_{\mathrm{BD}} C_{\mathrm{ABi}} - c V_{\mathrm{B}} \\
\dot{C}_0 &= -\lambda C_0 T,
\end{aligned}
$$

Metabolic pathway of skeletal muscle



The discrete time Markov models framework

Evolution model:

$$X_{j+1} = F(X_j, \theta) + V_{j+1},$$

- F is a known propagation model
- V_{j+1} is an innovation process
- θ is a parameter: assumed known now, later to be estimated.

The observation model

$$Y_j = G(X_j) + W_j,$$

the observation noise W_j independent of X_j.
Update scheme for posterior densities given accumulated data:

$$\pi(x_j \mid D_j) \longrightarrow \pi(x_{j+1} \mid D_j) \longrightarrow \pi(x_{j+1} \mid D_{j+1})$$

Bayesian filtering

1. Propagation step: Chapman-Kolmogorov formula

$$\begin{aligned}\pi(x_{j+1} \mid D_j) &= \int \pi(x_{j+1} \mid x_j, D_j)\pi(x_j \mid D_j)dx_j \\ &= \int \pi(x_{j+1} \mid x_j)\pi(x_j \mid D_j)dx_j,\end{aligned}$$

2. Analysis step: Bayes' formula conditional on D_j

$$\begin{aligned}\pi(x_{j+1} \mid D_{j+1}) &= \pi(x_{j+1} \mid y_{j+1}, D_j) \\ &\propto \pi(y_{j+1} \mid x_{j+1}, D_j)\pi(x_{j+1} \mid D_j) \\ &= \pi(y_{j+1} \mid x_{j+1})\pi(x_{j+1} \mid D_j),\end{aligned}$$

Combining:

$$\pi(x_{j+1} \mid D_{j+1}) \propto \pi(y_{j+1} \mid x_{j+1}) \int \pi(x_{j+1} \mid x_j)\pi(x_j \mid D_j)dx_j.$$

Bayesian filtering

Assume that the current distribution is represented in terms of a sample

$$\mathscr{S}_j = \{(x_j^1, w_j^1), (x_j^2, w_j^2), \ldots, (x_j^N, w_j^N)\}.$$

The particle version (Monte Carlo integration):

$$\pi(x_{j+1} \mid D_{j+1}) \propto \pi(y_{j+1} \mid x_{j+1}) \int \pi(x_{j+1} \mid x_j)\pi(x_j \mid D_j) dx_j$$

is approximated by

$$\pi(x_{j+1} \mid D_{j+1}) \propto \pi(y_{j+1} \mid x_{j+1}) \sum_{n=1}^{N} w_j^n \pi(x_{j+1} \mid x_j^n).$$

Example: Gaussian innovation and noise

Assuming the propagation model

$$X_{j+1} = F(X_j) + V_{j+1}, \quad V_{j+1} \sim \mathscr{N}(0, \Gamma_{j+1}),$$

we have

$$\pi(x_{j+1} \mid x_j) \propto \exp\left(-\frac{1}{2}(x_{j+1} - F(x_j))^{\mathsf{T}} \Gamma_{j+1}^{-1} (x_{j+1} - F(x_j))\right),$$

Similarly, the observation model

$$Y_j = G(X_j) + W_j, \quad W_j \sim \mathscr{N}(0, \Sigma_j)$$

gives the likelihood

$$\pi(y_j \mid x_j) \propto \exp\left(-\frac{1}{2}(y_j - G(x_j))^{\mathsf{T}} \Sigma_j^{-1} (y_j - G(x_j))\right).$$

Sampling Importance Resampling (SIR)

Layered sampling: For $n = 1, 2, \ldots, N$,

1. Draw a candidate particle $\widetilde{x}_{j+1}^n$ from $\pi(x_{j+1} \mid x_j^n)$;
2. Compute the relative likelihood $g_{j+1}^n = \pi(y_{j+1} \mid \widetilde{x}_{j+1}^n)$;
3. Resample with replacement from

$$\left\{(\widetilde{x}_{j+1}^1, \widetilde{w}_{j+1}^1), (\widetilde{x}_{j+1}^2, \widetilde{w}_{j+1}^2), \ldots, (\widetilde{x}_{j+1}^N, \widetilde{w}_{j+1}^N)\right\},$$

   where the probability weights are defined as

$$\widetilde{w}_{j+1}^n = \frac{g_{j+1}^n}{\sum g_{j+1}^n}.$$

Sampling Importance Resampling (SIR)

Data thinning:

- Most particles $\widetilde{x}_{j+1}^{n}$ may have vanishingly small likelihood.
- Few candidate particles are sampled over and over: The new sample consists mostly copies of few candidate particles.
- The density is poorly sampled.

Improvement: Auxiliary particles

Before resampling, calculate an auxiliary predictor:

$$\overline{x}_{j+1}^n = F(x_j^n).$$

We write

$$\pi(x_{j+1} \mid D_{j+1}) \propto \sum_{n=1}^{N} \underbrace{w_j^n \, \pi(y_{j+1} \mid \overline{x}_{j+1}^n)}_{=g_{j+1}^n} \frac{\pi(y_{j+1} \mid x_{j+1})}{\pi(y_{j+1} \mid \overline{x}_{j+1}^n)} \pi(x_{j+1} \mid x_j^n),$$

The quantity g_{j+1}^n is a *predictor* of how well the auxiliary particle would explain the data.

Survival of the Fittest (SOF)

Given the initial probability density $\pi_0(x_0)$,

1. *Initialization:* Draw the particle ensemble from $\pi_0(x_0)$:

$$\mathcal{S}_0 = \left\{(x_0^1, w_0^1), (x_0^2, w_0^2), \ldots, (x_0^N, w_0^N)\right\},$$

$$w_0^1 = w_0^2 = \ldots = w_0^N = \frac{1}{N}.$$

   Set $j = 0$.

2. *Propagation:* Compute the predictor:

$$\overline{x}_{j+1}^n = F(x_j^n), \quad 1 \leq n \leq N.$$

Survival of the Fittest (SOF)

3. *Survival of the fittest:* For each n:
   (a) Compute the fitness weights

$$g_{j+1}^n = w_j^n \, \pi(y_{j+1} \mid \overline{x}_{j+1}^n), \quad g_{j+1}^n \leftarrow \frac{g_{j+1}^n}{\sum_n g_{j+1}^n};$$

   (b) Draw indices with replacement $\ell_n \in \{1, 2, \ldots, N\}$ using probabilities
   $\mathrm{P}\{\ell_n = k\} = g_{j+1}^k$;
   (c) Reshuffle

$$x_j^n \leftarrow x_j^{\ell_n}, \quad \overline{x}_{j+1}^n \leftarrow \overline{x}_{j+1}^{\ell_n}, \quad 1 \le n \le N.$$

Survival of the Fittest (SOF)

4. *Innovation:* For each n: Proliferate

$$x_{j+1}^n = \overline{x}_{j+1}^n + v_{j+1}^n.$$

5. *Weight updating:* For each n, compute

$$w_{j+1}^n = \frac{\pi(y_{j+1} \mid x_{j+1}^n)}{\pi(y_{j+1} \mid \overline{x}_{j+1}^n)}, \quad w_{j+1}^n \leftarrow \frac{w_{j+1}^n}{\sum_n w_{j+1}^n}.$$

6. If $j < T$, increase $j \leftarrow j + 1$ and repeat.

Estimating parameters: Sequential Monte Carlo

For the discrete time model, the propagation (and possibly the likelihood) may depend on the unknown θ,

$$x_{j+1} = F(x_j, \theta).$$

Monte Carlo integral for posterior update:

$$\pi(x_{j+1}, \theta \mid D_{j+1}) \propto \pi(y_{j+1} \mid x_{j+1}, \theta) \times \int \pi(x_{j+1} \mid x_j, \theta)\pi(x_j \mid \theta, D_j)\pi(\theta \mid D_j)dx_j,$$

Sample update:

$$\mathscr{S}_j \to \mathscr{S}_{j+1}, \quad \mathscr{S}_j = \left\{(x_j^n, \theta_j^n, w_j^n)\right\}_{n=1}^N.$$

where $\mathscr{S}_j$ is drawn from $\pi(x_j, \theta \mid D_j)$.

Auxiliary parameter particles

Given the current parameter sample,

$$(\theta_j^1, w_j^1), (\theta_j^2, w_j^2), \ldots, (\theta_j^N, w_j^N),$$

estimate the mean and covariance,

$$\overline{\theta}_j = \sum_{n=1}^{N} w_j^n \theta_j^n, \qquad \mathsf{C}_j = \sum_{n=1}^{N} w_j^n \left(\theta_j^n - \overline{\theta}_j\right)\left(\theta_j^n - \overline{\theta}_j\right)^{\mathsf{T}}.$$

Auxiliary parameter particles

Approximate the marginal probability density $\pi(\theta \mid D_j)$ of θ by a Gaussian mixture model,

$$\pi(\theta \mid D_j) \approx \sum_{n=1}^{N} w_j^n \mathcal{N}\left(\theta \mid \overline{\theta}_j^n, s^2 \mathsf{C}_j\right),$$

for which we define the auxiliary particle by

$$\overline{\theta}_j^n = a\theta_j^n + (1-a)\overline{\theta}_j,$$

where a is a shrinkage factor, $0 < a < 1$ and $a^2 + s^2 = 1$ to avoid artificial diffusion.

Auxiliary parameter particles

Left as an exercise:

$$\mathsf{E}\left\{\sum_{n=1}^{N} w_j^n \mathcal{N}\left(\theta \mid \overline{\theta}_j^n, s^2 \mathsf{C}_j\right)\right\} = \overline{\theta}_j,$$

and

$$\operatorname{cov}\left\{\sum_{n=1}^{N} w_j^n \mathcal{N}\left(\theta \mid \overline{\theta}_j^n, s^2 \mathsf{C}_j\right)\right\} = \mathsf{C}_j,$$

Auxiliary parameter particles

Approximate

$$\pi(x_{j+1}, \theta \mid D_{j+1}) \propto \sum_{n=1}^{N} w_j^n \pi(y_{j+1} \mid x_{j+1}, \theta) \pi(x_{j+1} \mid x_j^n, \theta) \mathcal{N}(\theta \mid \overline{\theta}_j^n, s^2 C_j),$$

which we write as

$$\pi(x_{j+1}, \theta \mid D_{j+1}) \propto \sum_{n=1}^{N} \underbrace{w_j^n \pi(y_{j+1} \mid \overline{x}_{j+1}^n, \overline{\theta}_j^n)}_{=g_{j+1}^n} \times \frac{\pi(y_{j+1} \mid x_{j+1}, \theta)}{\pi(y_{j+1} \mid \overline{x}_{j+1}^n, \overline{\theta}_j^n)} \pi(x_{j+1} \mid x_j^n, \theta) \mathcal{N}(\theta \mid \overline{\theta}_j^n, s^2 C_j),$$

where the coefficient g_{j+1}^n is the fitness of the predictor

$$(\overline{x}_{j+1}^n, \overline{\theta}_j^n) = (F(x_{j+1}^n, \overline{\theta}_j^n), \overline{\theta}_j^n).$$

Survivial of the Fittest – Sequential Monte Carlo (SOF – SMC)

1. *Initialization:* Draw the particle ensemble from $\pi_0(x_0, \theta)$:

$$\mathcal{S}_0 = \left\{(x_0^1, \theta_0^1, w_0^1), (x_0^2, \theta_0^2, w_0^2), \dots, (x_0^N, \theta_0^N, w_0^N)\right\},$$
$$w_0^1 = w_0^2 = \dots = w_0^N = \frac{1}{N}.$$

Compute the parameter mean and covariance:

$$\overline{\theta}_0 = \sum_{n=1}^{N} w_0^n \theta_0^n, \quad \mathsf{C}_0 = \sum_{n=1}^{N} w_0^n \left(\theta_0^n - \overline{\theta}_0\right)\left(\theta_0^n - \overline{\theta}_0\right)^{\mathsf{T}}.$$

Set $j = 0$.

Survivial of the Fittest – Sequential Monte Carlo (SOF – SMC)

2. *Propagation:* Shrink the parameters

$$\overline{\theta}_j^n = a\theta_j^n + (1-a)\overline{\theta}_j, \quad 1 \leq n \leq N,$$

by a factor $0 < a < 1$. Compute the state predictor:

$$\overline{x}_{j+1}^n = F(x_j^n, \overline{\theta}_j^n), \quad 1 \leq n \leq N.$$

Survivial of the Fittest – Sequential Monte Carlo (SOF – SMC)

3. *Survival of the fittest:* For each n:
   (a) Compute the fitness weights

$$g_{j+1}^n = w_j^n \pi(y_{j+1} \mid \overline{x}_{j+1}^n, \overline{\theta}_j^n), \quad g_{j+1}^n \leftarrow \frac{g_{j+1}^n}{\sum_n g_{j+1}^n};$$

   (b) Draw indices with replacement $\ell_n \in \{1, 2, \ldots, N\}$ using probabilities $\mathrm{P}\{\ell_n = k\} = g_{j+1}^k$;
   (c) Reshuffle

$$(\overline{x}_{j+1}^n, \overline{\theta}_j^n) \leftarrow (\overline{x}_{j+1}^{\ell_n}, \overline{\theta}_j^{\ell_n}), \quad \overline{x}_{j+1}^n \leftarrow \overline{x}_{j+1}^{\ell_n}, \quad 1 \leq n \leq N.$$

Survivial of the Fittest – Sequential Monte Carlo (SOF – SMC)

4. *Proliferation:* For each n:
   (a) Proliferate the parameter by drawing

$$\theta_{j+1}^n \sim \mathcal{N}\big(\overline{\theta}_j^n, s^2 \mathrm{C}_j\big), \quad s^2 = 1 - a^2;$$

   (b) Repropagate and add innovation:

$$x_{j+1}^n = F(x_j^n, \theta_{j+1}^n) + v_{j+1}^n.$$

Survivial of the Fittest – Sequential Monte Carlo (SOF – SMC)

5. *Weight updating:* For each n, compute

$$w_{j+1}^n = \frac{\pi(y_{j+1} \mid x_{j+1}^n, \theta_{j+1}^n)}{\pi(y_{j+1} \mid \overline{x}_{j+1}^n, \overline{\theta}_j^n)}, \quad w_{j+1}^n \leftarrow \frac{w_{j+1}^n}{\sum_n w_{j+1}^n}.$$

6. If $j < T$, update

$$\overline{\theta}_{j+1} = \sum_{n=1}^{N} w_{j+1}^n \theta_{j+1}^n, \quad \mathsf{C}_{j+1} = \sum_{n=1}^{N} w_{j+1}^n \left(\theta_{j+1}^n - \overline{\theta}_{j+1}\right)\left(\theta_{j+1}^n - \overline{\theta}_{j+1}\right)^{\mathsf{T}},$$

increase $j \leftarrow j + 1$ and repeat.

Propagation and innovation

The problem we are addressing assumes

$$\frac{dx}{dt} = f(t, x, \theta), \quad x(0) = x_0,$$

while the discrete propagation is written as

$$x_{j+1} = F(x_j, \theta) + v_{j+1}.$$

Questions:

1. How do we propagate?
2. What is the innovation?

Propagation and innovation

Naive propagation scheme

$$x_{j+1} \approx x_j + \Delta t F(x_j, \theta),$$

which is used for SDE schemes (Euler-Maruyama) has several problems:

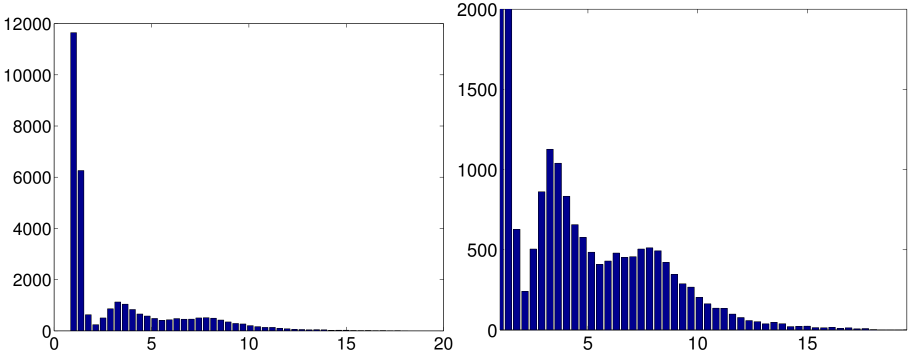
- The systems are typically stiff, leading to prohibitively small step size to guarantee stability
- The innovation needs to be related to approximation error.

Propagation and innovation

A more sophisticated solution: Use a standard stiff solver such as `ode15s`.
Error proportional to approximation accuracy.
Problem:

- Variable time step: some particles require longer integearation
- The slowest particle determines the propagation speed.

Histogram of propagation times: An example



Distribution of propagation times, $\sim 18\,000$ particles.

Stiffness and syncronization

For systems which are inherently stiff, we use a good stiff solver:

$$x_{j+1} = F^{\mathrm{exact}}(x_j, \theta) = F(x_j, \theta) + \text{approximation error},$$

where the approximation error is due to numerical integration.
If the stiffness of the systems varies a lot with the parameter values prescribing a fixed accuracy may be a problems because

- The time for the particles propagation may vary widely;
- The slowest particle determines the propagation speed
- We cannot take full advantage of parallel and vectorized computing environment.

Linear Multistep Methods

Given r past values,

$$u_n, u_{n+1}, \dots, u_{n+r-1},$$

write

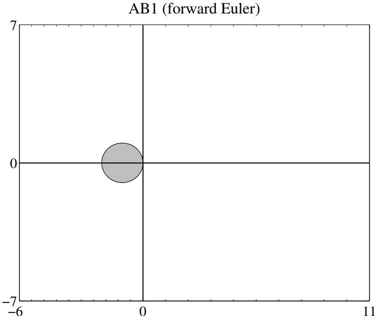
$$\sum_{j=0}^{r} \alpha_j u_{n+j} = h \sum_{j=0}^{r} \beta_j f(u_{n+j}, t_{n+j}),$$

and determine the coefficients α_j, β_j from a condition that the formula is accurate for a polynomial.

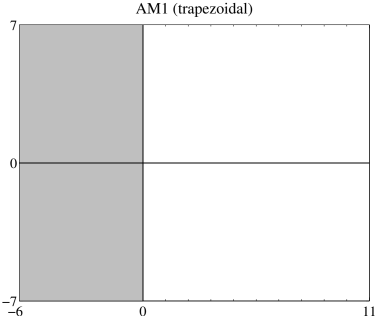
- Adams methods: $\alpha_r = 1$, $\alpha_{r-1} = -1$, $\alpha_j = 0$ for $j < r - 1$.
- Backwards Differentiation Formulas: $\beta_0 = \beta_1 = \dots = \beta_{r-1} = 0$.

For stiff problems: AM1–AM4, BDF1 – BDF4.

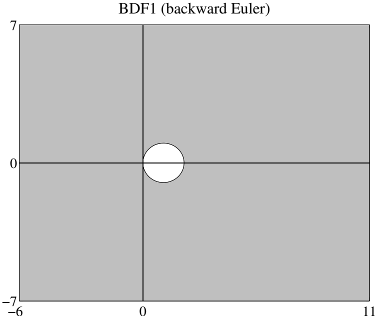
Stability regions:Adams-Bashford (explicit)



Stability regions:Adams-Moulton (implicit)



Stability regions: BDF (implicit)



Higher Order Method Error Control (HOMEC)

- u_{n+r} = candidate solution at time t_{n+r} computed by the LMM of order p,
- $\widehat{u}_{n+r}$ = solution at t_{n+r} given by the higher order method.
- u = exact solution with initial value $u(t_n) = u_n$

$$\begin{aligned} u_{n+r} &= u(t_{n+r}) + \ell h^{p+1} + \mathcal{O}(h^{p+2}) \\ \widehat{u}_{n+r} &= u(t_{n+r}) + \mathcal{O}(h^{p+2}) \end{aligned}$$

as $h \to 0$, where ℓ is some vector depending on the solution $u(t)$ of the ODE system but not on h.
Subtracting and neglecting higher order terms:

$$\ell h^{p+1} \approx u_{n+r} - \widehat{u}_{n+r}$$

Prescribe time, not accuracy

Thus, to improve the performance of the algorithm we

- Propagate each particle with fixed propagation time.
- Estimate the numerical accuracy for each particle
- Set the jth particle innovation variance proportional to the integration error.

This yields the innovation covariance matrix

$$V_{j+1} \sim \mathcal{N}(0, \Gamma_{j+1}),$$

where for $1 \leq i \leq d$,

$$\Gamma_{j+1} = \mathrm{diag}(\gamma) + \varepsilon \mathsf{I}, \quad \gamma_i = \tau^2 (u_{j+1} - \widehat{u}_{j+1})_i^2,$$

with $\tau > 1$.

SOF with error estimate innovation

Given the initial probability density $\pi_0(x_0)$,

1. *Initialization:* Draw the particle ensemble from $\pi_0(x_0)$:

$$\mathcal{S}_0 = \left\{(x_0^1, w_0^1), (x_0^2, w_0^2), \ldots, (x_0^N, w_0^N)\right\},$$

$$w_0^1 = w_0^2 = \ldots = w_0^N = \frac{1}{N}.$$

   Set $j = 0$.

2. *Propagation:* Compute the predictor using LMM:

$$\overline{x}_{j+1}^n = \Psi(x_j^n, h), \quad 1 \leq n \leq N.$$

3. *Survival of the fittest:* For each n:

   (a) Compute the fitness weights

$$g^n_{j+1} = w^n_j \, \pi(y_{j+1} \mid \overline{x}^n_{j+1}), \quad g^n_{j+1} \leftarrow \frac{g^n_{j+1}}{\sum_n g^n_{j+1}};$$

   (b) Draw indices with replacement $\ell_n \in \{1, 2, \ldots, N\}$ using probabilities
   $\mathrm{P}\{\ell_n = k\} = g^k_{j+1}$;

   (c) Reshuffle

$$x^n_j \leftarrow x^{\ell_n}_j, \quad \overline{x}^n_{j+1} \leftarrow \overline{x}^{\ell_n}_{j+1}, \quad 1 \leq n \leq N.$$

4. *Innovation:* For each n:
   (a) Using error estimate, estimate $\Gamma_{j+1}^n = \Gamma_{j+1}(x_j^n)$;
   (b) Draw $v_{j+1}^n \sim \mathcal{N}(0, \Gamma_{j+1}^n)$;
   (c) Proliferate
$$x_{j+1}^n = \overline{x}_{j+1}^n + v_{j+1}^n.$$
5. *Weight updating:* For each n, compute
$$w_{j+1}^n = \frac{\pi(y_{j+1} \mid x_{j+1}^n)}{\pi(y_{j+1} \mid \overline{x}_{j+1}^n)}, \quad w_{j+1}^n \leftarrow \frac{w_{j+1}^n}{\sum_n w_{j+1}^n}.$$
6. If $j < T$, increase $j \leftarrow j + 1$ and repeat from (ii).

The parameter estimation SMC can be also carried out concurrently.

Extensions to EnKF

Assuming that the current density $\pi\left(x_j, \theta \mid D_j\right)$ is represented in terms of an ensemble

$$\mathscr{S}_{j|j} = \left\{ \left(x_{j|j}^1, \theta_j^1\right), \left(x_{j|j}^2, \theta_j^2\right), \ldots, \left(x_{j|j}^N, \theta_j^N\right) \right\},$$

the state prediction ensemble is obtained by

$$x_{j+1|j}^n = F\left(x_{j|j}^n, \theta_j^n\right) + v_{j+1}^n, n = 1, 2, \ldots, N$$

following the state evolution equation and setting the innovation variance as for the particle filter.
Assume a linear observation model.

Combining state and parameter vectors

Compute the prediction ensemble statistics by defining the state and parameter vectors

$$z_{j+1|j}^{n} = \left[\begin{array}{c} x_{j+1|j}^{n} \\ \theta_{j}^{n} \end{array} \right], \qquad n = 1, 2, \ldots, N.$$

The prediction ensemble mean is

$$\overline{z}_{j+1|j} = \frac{1}{N} \sum_{n=1}^{N} z_{j+1|j}^{n}$$

and the prior covariance matrix is

$$\Gamma_{j+1|j} = \frac{1}{N-1} \sum_{n=1}^{N} \left(z_{j+1|j}^{n} - \overline{z}_{j+1|j}^{n} \right) \left(z_{j+1|j}^{n} - \overline{z}_{j+1|j}^{n} \right)^{\mathsf{T}}$$

When an observation y_{j+1} arrives, an observation ensemble is generated by parametric bootstrapping

$$y_{j+1}^n = y_{j+1} + w_{j+1}^n$$

where $w_{j+1}^n \sim \mathcal{N}(0, D)$ is a realization of the noise. In the case of a linear observation the combined posterior ensemble is obtained as

$$z_{j+1|j+1}^n = z_{j+1|j}^n + \mathsf{K}_{j+1} \left(y_{j+1}^n - G_{j+1} z_{j+1|j}^n \right), \qquad n = 1, 2, \ldots, N$$

where the Kalman gain K_{j+1} is

$$\mathsf{K}_{j+1} = \Gamma_{j+1|j} \mathsf{G}_{j+1}^{\mathsf{T}} \left(\mathsf{G}_{j+1} \Gamma_{j+1|j} \mathsf{G}_{j+1}^{\mathsf{T}} + D \right)^{-1}.$$

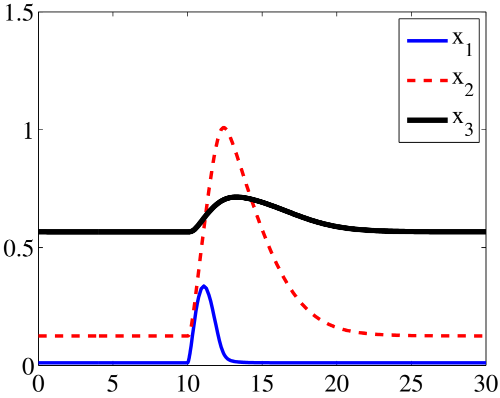
The posterior means and covariances for the states and parameters are computed using the posterior ensemble statistics.

A simple example motivated by acetate metabolism

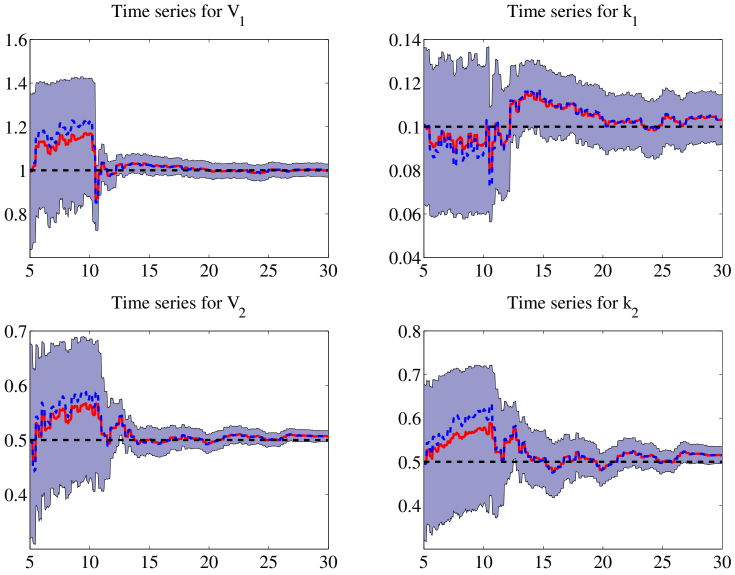
$$\begin{aligned}
\frac{dx_1}{dt} &= \Phi(t) - V_1 \frac{x_1}{x_1 + k_1} \\
\frac{dx_2}{dt} &= V_1 \frac{x_1}{x_1 + k_1} - V_2 \frac{x_2}{x_2 + k_2} \\
\frac{dx_3}{dt} &= V_2 \frac{x_2}{x_2 + k_2} - \lambda(x_3 - c_0)
\end{aligned} \tag{3}$$

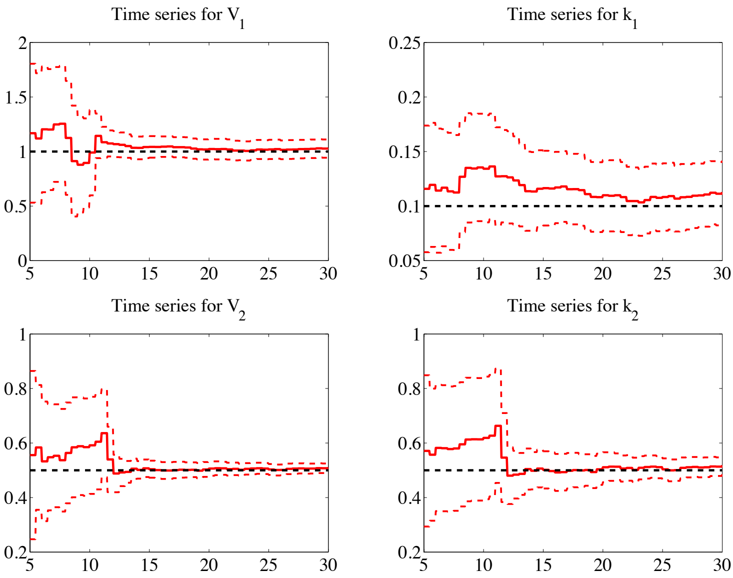
with known parameters λ and c_0 and input function

$$\Phi(t) = A_0 + A(t - t_0)\exp(-(t - t_0)/\tau)$$

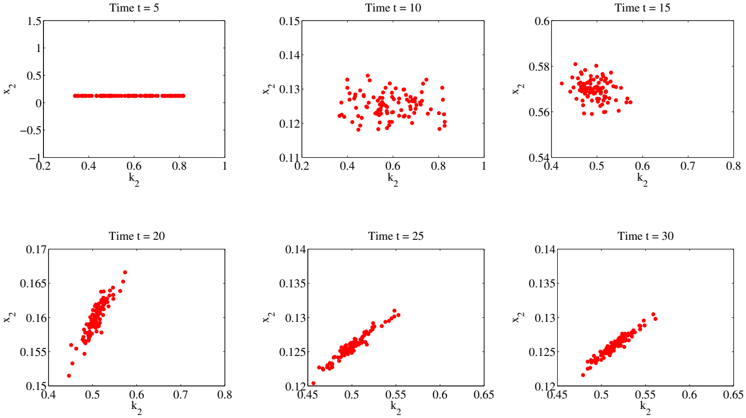


1.5
1
0.5
0
0
5
10
15
20
25
30
x1
x2
x3


Time series for V_1
Time series for k_1
Time series for V_2
Time series for k_2
1.6
1.4
1.2
1
0.8
0.14
0.12
0.1
0.08
0.06
0.04
0.7
0.6
0.5
0.4
0.8
0.7
0.6
0.5
0.4
5
10
15
20
25
30

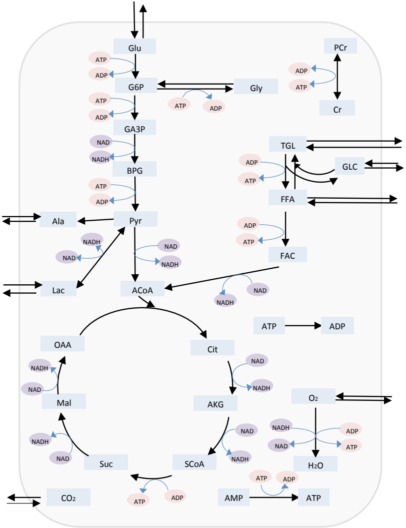


Time series for V_1
Time series for k_1
Time series for V_2
Time series for k_2
2
1.5
1
0.5
0
0.25
0.2
0.15
0.1
0.05
0.8
0.6
0.4
0.2
5
10
15
20
25
30


Time t = 5
1.5
1
0.5
0
−0.5
−1
0.2
0.4
0.6
0.8
1
x2
k2
Time t = 10
0.15
0.14
0.13
0.12
0.11
0.2
0.4
0.6
0.8
1
x2
k2
Time t = 15
0.6
0.58
0.56
0.54
0.4
0.5
0.6
0.7
0.8
x2
k2
Time t = 20
0.17
0.165
0.16
0.155
0.15
0.4
0.5
0.6
0.7
0.8
x2
k2
Time t = 25
0.14
0.135
0.13
0.125
0.12
0.45
0.5
0.55
0.6
0.65
x2
k2
Time t = 30
0.14
0.135
0.13
0.125
0.12
0.45
0.5
0.55
0.6
0.65
x2
k2

The humble and lowly skeletal muscle

and its metabolism



Stiffness and many unknown parameters

Problem: follow the time courses of metabolites and intermediates in skeletal model cellular metabolism model over 100 minutes.

- State vector: 38 concentrations, 30 in tissue, 8 in blood;
- Evolution model: nonlinear system of ODEs;
- Number of unknown parameters: 44+52.
- The stiffness of the underlying ODE system reflects the difference in time constants of the different reactions and transports.

Numerical test

- Data: Noisy measurements of 8 concentrations in blood at 11 time instances.
- Propagation: use implicit time integrators BDF2 (and BDF3) with fixed time step $h = 0.01$ or 0.6 seconds.
- Ensemble size: N=250.
- Fix 44 parameters corresponding to facilitators and estimate remaining 52 (max flux and affinity)
- Initial ensemble: cloud of parameters values not centered around true values and corresponding initial values which support steady state.
- At each time step we inflate the covariance matrix by 20%.

Spatio-temporal prior

- We want to favor solutions with moderate rates of change.
- Exponential a priori bound:

$$M^{-\Delta t_{j+1}}\overline{C}(t_j) \leq C(t_{j+1}) \leq M^{\Delta t_{j+1}}\overline{C}(t_j)$$

  for $M > 1$ and $\Delta t_{j+1} = t_{j+1} - t_j$, which implies

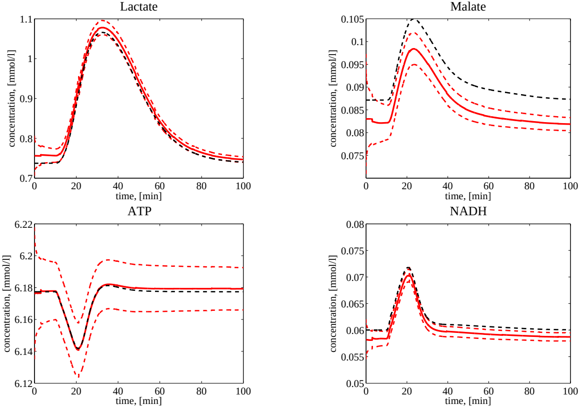
$$\| \log \left(\frac{C(t_{j+1})}{\overline{C}(t_j)}\right) \| = |x(t_{j+1}) - \overline{x}_{j|j}| \leq (\log M)\Delta t_{j+1}.$$

- We set

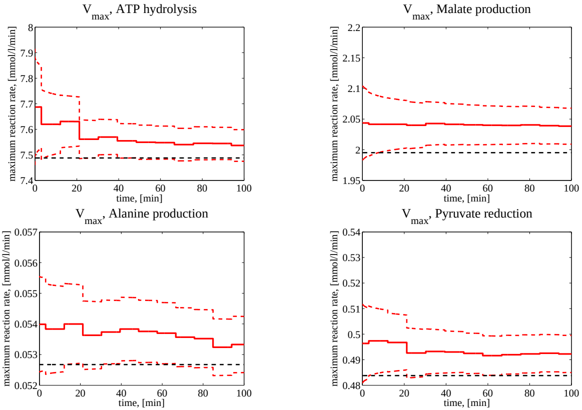
$$x(t_{j+1}) \sim \mathcal{N}\left(\overline{x}_{j|j}, \gamma^2 (\Delta t_{j+1})^2 I_d\right)$$

  with $M = 20$ and $\gamma = \log M/2$.

Estimation of blind components

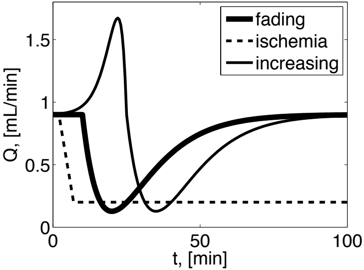


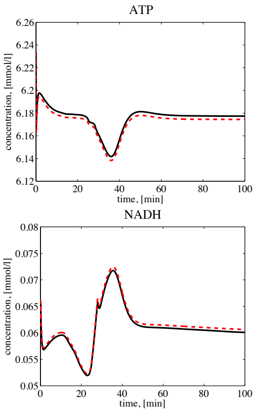
Estimation of parameters

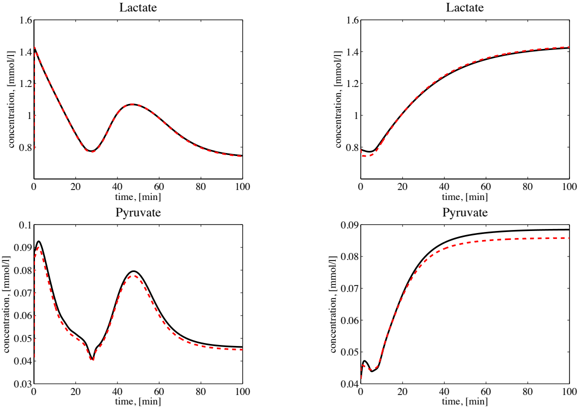


Predictive skill of the filter

Question: How well does a dynamical system identified by the estimated parameters describe predict other protocols?




ATP
concentration, [mmol/l]
time, [min]
ATP
concentration, [mmol/l]
time, [min]
NADH
concentration, [mmol/l]
time, [min]
NADH
concentration, [mmol/l]
time, [min]


Lactate
concentration, [mmol/l]
time, [min]
Lactate
concentration, [mmol/l]
time, [min]
Pyruvate
concentration, [mmol/l]
time, [min]
Pyruvate
concentration, [mmol/l]
time, [min]

References

- Arnold A, Calvetti D, and Somersalo E (2013) Linear multistep methods, particle filtering and sequential Monte Carlo. *Inverse Problems*, 29(8), 085007.
- Arnold A, Calvetti D, Gjedde A, Iversen P, and Somersalo E (2014) Astrocyte tracer dynamics estimated from $[1-^{11}C]$-acetate PET measuraments. *Math. Med. Biol.* doi: 10.1093/imammb/dqu021.
- Arnold A, Calvetti D, and Somersalo E (2014) Parameter estimation for stiff deterministic dynamical systems via Ensemble Kalman filter. *Inverse Problems* **30** : 105008.

References

- Cotter SL, Roberts GO, Stuart AM and White D (2013). MCMC methods for functions: modifying old algorithms to make them faster. Statistical Science, **28**, 424-446.
- Liu J and West M 2001 Combined parameter and state estimation in simulation-based filtering *Sequential Monte Carlo Methods in Practice* eds A Doucet, J F G de Freitas and N J Gordon (New York: Springer-Verlag) pp 197-223
- Pitt M and Shephard N 1999 Filtering via simulation: auxiliary particle filters *J. Amer. Statist. Assoc.* **94** 590-599