Evaluation of Bayesian Filters Applied to Heat Conduction Problems

Wellington B. Silva, Helcio R. B. Orlande, Marcelo J. Colaço
Department of Mechanical Engineering, PEM/COPPE
Federal University of Rio de Janeiro, UFRJ
wellingtonuff@yahoo.com.br, {helcio, colaço}@mecanica.ufrj.br

Abstract
In this paper we applied Bayesian filters to a state estimation problem involving a one-dimensional, transient, heat conduction problem. The main objective of this paper is to discuss and compare the performance of the several types of filters, namely: Sampling Importance Re-sampling Filter (SIR filter), Auxiliary Sampling Importance Re-sampling Filter (ASIR filter), Sequential Monte Carlo Samplers (SMC), Combined Parameter and State Estimation in Simulation Based Filtering, and Sequential Monte Carlo without Likelihoods. Results obtained with simulated experiments show a good performance of some of the filters, as applied to inverse heat conduction problems.

Keywords: Bayesian inference, Sequential Monte Carlo, Importance sampling, Approximate Bayesian Computation.

1. Introduction
Sequential Monte Carlo (SMC) or Particle Filter Methods, which have been originally introduced in the beginning of the 50’s, became very popular in the last few years in the statistical and engineering communities. Such methods have been widely used to deal with sequential Bayesian inference problems in fields like economics, signal processing, and robotics, among others. SMC Methods are an approximation of sequences of probability distributions of interest, using a large set of random samples, named particles. These particles are propagated along time with a simple Sampling Importance distribution, SI [1], and re-sampling techniques as well.

Hammersley and Handscomb [2] presented a technique that used recursive Bayesian filters, together with Monte Carlo simulations, known as Sequential Importance Sampling (SIS). In such approach, the key idea was to represent the posterior probability function as a set of random samples associated with some weights, in order to calculate the estimates based on such samples and weights. Gordon et al [3] added an extra step, named re-sampling, into the Sequential Importance Sampling method, to avoid the problem known as degeneration of particles. Such filter is known as Sampling Importance Re-sampling (SIR) Filter. In 2008, Orlande et al [4] presented an application of the SIR Filter to linear and non-linear heat conduction problems.

In order to overcome some difficulties of SIR filter, Pitt and Shepard [5] introduced the Auxiliary Particle Filter (APF). In 2006, Del Moral et al [6] presented also modifications to improve the SIR Filter, and named the resulting algorithm as Sequential Monte Carlo Samplers, introducing a method for the evolution of the particles and also an artificial delayed kernel.

Liu and West [7] have introduced an algorithm for the simultaneous estimation of state variables and constant parameters, which uses a combination of the artificial evolution method (where the problem related with the loss of information is avoided) and the smoothness kernel proposed by West [8], which improves the choice of the particles. In 2007, Sisson and Fran [9] presented a new filter technique, where the Sequential Monte Carlo Method was coupled with the Approximate Bayesian Computation, named Sequential Monte Carlo without Likelihoods.

In this paper we will apply the above referenced methods to a linear, one-dimensional, transient heat conduction problem, in order to estimate a transient heat flux applied to one of the surfaces of a plate. Simulated temperature measurements are used in the inverse analysis. The methods are compared in terms of accuracy of the recovered unknown boundary heat flux.

2. Physical Problem
Consider heat conduction in a semi-infinite one-dimensional medium, initially at the uniform temperature $T_0$. The boundary at $x = 0$ is subjected to heat flux $q(t)$, while the boundary at $x = L$ exchanges heat by convection with some medium at temperature $T_e$ (Robin boundary condition), with a heat transfer coefficient $h$. Physical properties are constant and there is no heat generation in the medium. The formulation for this problem is given by:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T(x,t)}{\partial t} \quad \text{in} \quad 0 < x < L \quad \text{and} \quad t > 0$$

(1.a)
By assuming that state variables \( \text{In order to define the state estimation problem, consider a model for the evolution of the vector \( x(t) \) in the form
\[
 x_k = f_k(x_{k-1}, v_{k-1}) \quad (4.a)
\]
where the subscript \( k = 1, 2, \ldots \) denotes a time instant \( t_k \) in a dynamic problem. The vector \( x \in \mathbb{R}^{n_x} \) is called the state vector and contains the variables to be dynamically estimated. This vector advances in accordance with the state evolution model given by equation (4.a), where \( f \) is, in the general case, a non-linear function of the state variables \( x \) and of the state noise vector \( v \in \mathbb{R}^{n_v} \).

Consider also that measurements \( z \in \mathbb{R}^{n_z} \) are available at \( t_k \), \( k = 1, 2, \ldots \). The measurements are related to the state variables \( x \) through the general, possibly non-linear, function \( h \) in the form
\[
z_k = h_k(x_k, v_k) \quad (4.b)
\]
where \( n \in \mathbb{R}^{n_z} \) is the measurement noise. Equation (4.b) is referred to as the observation (measurement) model.

The state estimation problem aims at obtaining information about \( x_k \) based on the state evolution model (4.a) and on the measurements \( z_{1:k} = \{z_i, i = 1, \ldots, k\} \) given by the observation model (4.b).

The evolution-observation model given by equations (4.a,b) are based on the following assumptions:

(i) The sequence \( x_k \) for \( k = 1, 2, \ldots \) is a Markovian process, that is,
\[
\pi(x_k|x_0, x_1, \ldots, x_{k-1}) = \pi(x_k|x_{k-1}) \quad (5.a)
\]

(ii) The sequence \( z_k \) for \( k = 1, 2, \ldots \) is a Markovian process with respect to the history of \( x_k \), that is,
\[
\pi(z_k|x_0, x_1, \ldots, x_k) = \pi(z_k|x_k) \quad (5.b)
\]

(iii) The sequence \( x_k \) depends on the past observations only through its own history, that is,
\[
\pi(x_k|x_{k-1}, z_{1:k-1}) = \pi(x_k|x_{k-1}) \quad (5.c)
\]

where \( \pi(a|b) \) denotes the conditional probability of \( a \) when \( b \) is given.

By assuming that \( \pi(x_0|x_0) = \pi(x_0) \) is available, the posterior probability density \( \pi(x_k|z_{1:k}) \) is then obtained with Bayesian filters in two steps [10]: prediction and update, as illustrated in Figure 1.
importance weight of each particle can be calculated and normalized, given the transition equation

\[
\pi(x_{t+1} | x_t) = \pi(x_{t+1} | x_t) \cdot \pi(x_t | x_{t-1})
\]

where the particles candidates to represent the posterior distribution at time instant \( t \) are obtained, and called \( x_t^i \). Then, the likelihood \( \pi(z_t | x_t) \) of each auxiliary particle given \( x_t \) is calculated, thus obtaining the importance of each particle, by means of their normalized weights, \( \tilde{w}_t^i \). Particles with higher weights represent the regions more prone to contain correct values of the variable being estimated. After this step, the selection procedure is done, where the re-sampling of the anterior particles is performed, using the weights \( \tilde{w}_t^i \). The new particles at the time instant \( t \), which represent the posterior distribution, are then obtained, and called \( x_t^i \). At the end, there is the evolution step where, starting from the equation of transition \( \pi(x_t | x_{t-1}) \), new particles are generated in order to represent the posterior distribution at time \( t \). The entire procedure is repeated until the final step. Figure 2 shows the iterative procedure of SIR Filter algorithm.

### 3.1. The SIR Filter

The SIR filter starts by initially sampling \( N \) random particles from an initial distribution \( \pi(x_0) \) which are samples of the distribution at time instant \( t = 0 \). In order to advance the particles from time \( t_{k-1} \) to time \( t_k \), \( k = 1, 2, \ldots \) an observed value \( z_k \) is used at the time instant \( t \). The proximity of such values is verified and its likelihood \( \pi(z_k | x_k) \) is calculated, thus obtaining the importance of each particle, by means of their normalized weights, \( \tilde{w}_k^i \). Particles with higher weights represent the regions more prone to contain correct values of the variable being estimated. After this step, the selection procedure is done, where the re-sampling of the anterior distribution \( \pi(x_k | x_{k-1}) \) of the \( N \) particles is performed, using the weights \( \tilde{w}_k^i \). The new particles at the time instant \( t \), which represent the posterior distribution, are then obtained, and called \( x_t^i \). At the end, there is the evolution step where, starting from the equation of transition \( \pi(x_t | x_{t-1}) \), new particles are generated in order to represent the posterior distribution at time \( t \). The entire procedure is repeated until the final step. Figure 2 shows the iterative procedure of SIR Filter algorithm.

### 3.2. The ASIR Filter

One of the weaknesses of the SIR Filter is related to its approximation based on the particles to represent the probability density, since the number of particles is finite and then the tail of the density distribution is not very well captured. Such problem is even worst when there are no outliers \([12]\). To solve this issue, Pitt and Sheppard \([5]\) introduced the so called Auxiliary Particle Filter (APF). The idea of this method is to increase the number of "good particles" \( \{x^{(i)}_t\} \), considering that the probabilities \( \pi(z_t | x^{(i)}_t) \) assume higher values for such "good particles". The idea is to sample \( N \) auxiliary random particles \( u^{(i)}_{t-1} \) from the initial distribution \( \pi(x_0) \), and calculate the likelihood of each auxiliary particle given \( \pi(z_t | u^{(i)}_{t-1}) \). After that, \( N \) other particles are randomly taken from the initial distribution \( \pi(x_0) \). Such new particles are candidates to represent the posterior distribution at time instant \( t - 1 \), and named \( x_{t-1}^i \). Then, the likelihood \( \pi(z_t | x_{t-1}^i) \) of each particle is calculated. With these information, the importance weight of each particle can be calculated and normalized, given \( w_t^i \). The last step is the evolution step, where the particles candidates to represent the posterior distribution at time instant \( t \), named \( x_t^i \), are obtained, by using the transition equation \( \pi(x_t | x_{t-1}) \). The entire procedure is repeated until the final time, as shown in Figure 3.
1. **Initialization**
   a. Sample a set of particles from an initial distribution \( \pi(x_0) \) and obtain \( \{(x^{(i)}_1, w^{(i)}_0); i = 1, ..., N\} \);
   b. Set \( t=1 \);
2. **Weight evaluation**
   a. Calculate the new weights: \( w^{(i)}_t = w^{(i)}_{t-1} \pi(x_t | x^{(i)}_t) \)
   b. Normalize the weights: \( \tilde{w}^{(i)}_t = \frac{w^{(i)}_t}{\sum_{i=1}^{N} w^{(i)}_t} \)
3. **Re-sampling / Selection**
   a. Resample the particles as follows:
      3.1.1 Construct the cumulative sum of weights (CSW) by computing \( c_i = c_{i-1} + w^{(i)}_t \) for \( i = 1, ..., N \), with \( c_0 = 0 \).
      3.1.2 Let \( i = 1 \) and draw a starting point \( u_1 \) from the uniform distribution \( U[0, N^{-1}] \).
      3.1.3 For \( j = 1, ..., N \)
         - Move along the CSW by making \( u_j = u_1 + N^{-1} (j - 1) \).
         - While \( u_j > c_i \) make \( i = i + 1 \).
         - Assign sample \( x^{(i)}_t = x^{(i)}_t \).
         - Assign weight \( w^{(i)}_t = N^{-1} \).
4. **Calculate the estimate of the actual state**
   \[ \pi(x_t | z_t) = \sum_{i=1}^{N} x_t(i) \cdot \tilde{w}^{(i)}_t \]
5. **Perform a model evolution**
   a. Set \( t = t + 1 \).
   b. Advance the states from time \( t-1 \) to time \( t \) using the following equation of state:
      \[ x_t = \pi(x_t | x_{t-1}) \] for \( i = 1, ..., N \).
   c. If \( t = t_{\text{final}} + 1 \), then stop.
6. Return to step 2 with the new particles.

Figure 2: SIR Filter Algorithm

1. **Initialization**
   a. Sample a set of particles from the initial distribution \( \pi(x_0) \) to obtain \( \{(x^{(i)}_1); i = 1, ..., N\} \);
   b. Set \( t=1 \);
2. **Calculate samples for the state of system for the auxiliary variable**
   a. \( u_t = \pi \left( x_t | \{x^{(i)}_t\}_{i=1}^{N-1} \right) \)
3. **Calculate the weights**
   a. Calculate the likelihood function \( \pi \left( x_t | x^{(i)}_t \right) \)
4. **Model evolution**
   a. Calculate the particles, given as: \( x_t = \pi \left( x_t | x^{(i)}_t \right) \) for \( i = 1, ..., N \).
5. **Calculate the weights**
   a. Calculate the likelihood function \( \pi \left( x_t | x^{(i)}_t \right) \)
   b. Calculate the new weights: \( w^{(i)}_t = \frac{\pi(x_t | x^{(i)}_t)}{\pi \left( x_t | x^{(i)}_t \right)} \)
   c. Normalize the weights: \( \tilde{w}^{(i)}_t = \frac{w^{(i)}_t}{\sum_{i=1}^{N} w^{(i)}_t} \)
6. **Calculate the estimate of the actual state**
   a. \( \pi(x_t | z_t) = \sum_{i=1}^{N} x_t(i) \cdot \tilde{w}^{(i)}_t \)
   b. Set \( t = t + 1 \).
   c. If \( t = t_{\text{final}} + 1 \), then stop.
7. With the new particles, return to step 2.

Figure 3: ASIR Filter Algorithm
3.3. The SMC Samplers Filter

The SMC Sampler Filter [6] is a generalization of the Sequential Monte Carlo and Particle Filters. The SMC Samplers algorithm involves three steps: (i) The importance sampling (mutation) step, where the particles are moved from \( x_{t-1} \) to \( x_t \) using a kernel of transition or mutation \( K(x_{t-1}, x_t) \) [13,6]; (ii) The correction step, where the weights of the particles are calculated and normalized; (iii) The selection step, where the weights can be used in a re-sampling process, in order to selected the most important particles [14,15].

The filter starts by sampling \( N \) random particles from an initial distribution \( \pi(x_0) \), and call them \( x_{-1} \), which are candidates to represent the posterior distribution at the first time instant. Then, it calculates the likelihood \( \pi(y_t|x_{t-1}) \), and its normalized weights by importance, named \( \tilde{w}_t^{(i)} \). Particles with higher weights represent regions more important in the distribution. Resampling is then applied to obtain the particles that represent the posterior distribution. In the initial time, this algorithm is very similar to the SIR. When \( t > 1 \), however, the new particles are generated using the kernel of mutation MCMC \( K_t(x_{t-1}, x_t) \), given by the random walk. Thus, new particles are created, and the best ones are selected. Following that, it calculates \( L_{t-1}(x_t|x_{t-1}) \) using Eq. (6) [6], where \( \pi_t(x_t) \) is a sequence of distributions that describes a generic iterative/sequential (IS) method to sample the new weights \( \tilde{w}_t \).

The algorithm then calculates the weights of the particles and normalizes them. It follows a resampling process, obtaining new particles and finding the posterior distribution related with the actual time. The process return to the generation of new particles using the kernel of mutation and the process is repeated until the final time, according to Figure 4.

\[
L_{t-1}(x_t|x_{t-1}) = \frac{\pi_t(x_{t-1}) w_t(x_{t-1}, x_1)}{\pi_t(x_t)}
\]  

(6)

1. Initialization
   a. Set \( t = 1 \)
   b. For \( i = 1, ..., N \), sample \( x_1^{(i)} = \pi_x(x_1|x_0) \)
   c. Weights evaluation
      i. Calculate the likelihood \( \pi(x_1|x_1^{(i)}) \)
      ii. Calculate the new weights: \( W_1^{(i)} = \pi(x_1|x_1^{(i)}) \)
      iii. Normalize the weights: \( W_1^{(i)} = \frac{W_1^{(i)}}{\sum_{i=1}^{N} W_1^{(i)}} \)
   d. Re-sampling / Selection
      i. Re-sample
      ii. If \( \tilde{N}_{eff} < \frac{N}{2} \), then calculate:
         \[
         W_t^{(i)} = \frac{1}{\tilde{N}}, \text{ where } i = 1, ..., N
         \]
   e. Calculate an estimate of the actual state
      \[
      \pi(x_1|y_1) \approx \sum_{i=1}^{N} x_1^{(i)} * W_1^{(i)}
      \]

2. Calculate the sampling importance
   a. Set \( t = t + 1 \). If \( t = t_{final} + 1 \), then stop.
   b. Generate a particle population
      i. For \( j = 1, ..., M \); For \( i = 1, ..., N \), sample \( x_t^{(i,j)} = K_t(x_{t-1}^{(i,j)}) \)
   c. Calculate the delayed kernel \( L_{t-1}(x_t|x_{t-1}) \)
   d. Weight evaluation
      i. Calculate the new weights:
      \[
      \tilde{w}_t^{(i)}(x_{t-1}|x_t) = \frac{y_t(x_t|x_{t-1})}{L_{t-1}(x_t|x_{t-1})}\tilde{w}_t^{(i-1)}(x_{t-1}|x_{t-1})
      \]
      ii. Normalize the weights: \( W_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{i=1}^{N} \tilde{w}_t^{(i)}} \)

3. Re-sampling / Selection
   i. Re-sample
      ii. If \( \tilde{N}_{eff} < \frac{N}{2} \), then calculate: \( W_t^{(i)} = \frac{1}{\tilde{N}}, \text{ where } i = 1, ..., N \)

4. Calculate the estimate of the actual state
   a. \( \hat{x}_t = \sum_{i=1}^{N} x_t^{(i)} * W_t^{(i)} \)
   b. Return to step 2.

Figure 4: SMC Samplers Algorithm
3.4. The Combiner Parameter and State Estimation in Simulation Based Filtering (CPSE)

The algorithm developed by Liu and West [7] starts by randomly sampling \( N \) auxiliary particles for the constant parameters \( \theta_{i}^{t} \) from the distribution \( \pi(x_{i}|x_{i-1}^{t}) \). Then, it finds the values of \( u_{i}^{t} \) and \( m_{i}^{t} \), and calculate the likelihood for each auxiliary particles, given by \( \pi(x_{i}|u_{i}^{t}, m_{i}^{t}) \). The next step is to sample \( N \) random particles \( \theta_{i}^{t} \), using the normal distribution with average \( m_{i}^{t} \) and variance \( h^2V_{t} \). Then, it calculates the likelihood \( \pi(x_{i}|x_{i}^{t}, \theta_{i}^{t}) \) of each particle, and their normalized weights \( W_{i}^{t} \). The last step is the evolution step. By using the distribution \( \pi(x|t, x_{t-1}) \), which is the equation of transition, the algorithm obtains the particles candidate to represent the posterior distribution at instant \( t \) and call them \( x_{t} \). The procedure is then repeated until the final time, according to Figure 5.

1. Initialization
   a. Let \( \delta \in [0.95, 0.99] \)
   b. Let \( a = \frac{(3\delta - 1)}{2\delta} \) and \( h^2 = 1 - a^2 \)
   c. Set \( t = 1 \)
2. Calculate the localization kernel
   a. Generate a set of particles \( \theta_{i}^{t(i)} \) with \( \pi(x_{i}|x_{i}^{t}) \), where \( i = 1, ..., N \).
   b. Obtain \( \{ (u_{i}^{t}, m_{i}^{t}) ; i = 1, ..., N \} \), where:
      \[
      u_{i}^{t} = \frac{E}{\pi(x_{i}^{t(i)}, \theta_{i}^{t(i)})} \]
      \[
      m_{i}^{t} = a\theta_{i}^{t(i)} + (1 - a)\theta_{i}^{t}
      \]
      \( \theta_{i}^{t} \) is the average value of all particles \( \theta_{i}^{t(i)} \)
   c. Calculate the auxiliary sample
      i. Calculate the likelihood \( \pi(x_{i}|u_{i}^{t}, m_{i}^{t}) \)
3. Model evolution
   a. For \( j = 1, ..., N \):
      i. Generate the new particles \( \theta_{j}^{t} \sim N(m_{j}^{t}, h^2V_{t}) \), where \( V_{t} \) is the variance matrix.
      ii. Sample \( x_{j}^{t} = \pi(x_{i}|x_{i}^{t-1}, \theta_{j}^{t}) \)
      iii. Calculate the weights, \( W_{j}^{t(j)} \) as:
         \[
         W_{j}^{t(j)} = \frac{\pi(x_{j}|x_{j}^{t(i)}, \theta_{j}^{t(j)})}{\pi(x_{j}|u_{j}^{t(i)}, m_{j}^{t(j)})}
         \]
4. Normalize the weights
   a. \( \tilde{\omega}_{j}^{t(j)} = \frac{W_{j}^{t(j)}}{\sum_{i=1}^{N} W_{i}^{t(j)}} \)
5. Calculate an estimate of the actual state
   a. \( \pi(x_{t}|x_{t}) = \sum_{j=1}^{N} \theta_{j}^{t(i)} \in \tilde{\omega}_{j}^{t(i)} \)
   b. Set \( t = t + 1 \)
6. If \( t = t_{\text{final}} + 1 \), then stop. Otherwise, return to step 2.

Figure 5: Combiner Parameter and State Estimation in Simulation Based Filtering Algorithm

3.5. The Sequential Monte Carlo Filter without Likelihoods

This method is a combination of Monte Carlo Sequential Method with the Approximate Bayesian Computation, in order to present an alternative to Bayesian filters. Such method is known as Sequential Monte Carlo without Likelihood [9], and the major difference among the others is that the particles are generated one by one and they are only accepted if the Euclidian distance between the observed and simulated data for each particle is less than some pre-specified tolerance. Only after the particle is accepted, its weight is calculated and the process is repeated until \( N \) particles have been generated. Once all particles have been generated, the normalization and re-sampling steps are done and then the posterior distribution of the variable \( x_{t} \) at the instant \( t \) is calculated. The entire process is repeated until the final time, as shown in Figure 6.
4. Results and Discussion

The physical problem defined by Eqs. (2.a-d) was solved by the implicit finite difference method, for a semi-infinite plate with 50 mm length and a initial temperature of 20 °C. The heat flux was supposed to have a maximum value equal to 2000 W/m$^2$ and the Biot number was considered unitary. In this work, the simulated noisy measurements were uncorrelated, additive, Gaussian, with zero mean and constant standard deviation of 0.45 °C, which represents 5% of the maximum temperature.

Two different kinds of heat flux variation were examined: the first one was discontinuous, in the form of a step function, while the second one with discontinuities in its first derivative, in the form of a triangular function. Figures 7 and 8 show the transient temperature profiles obtained after applying such heat fluxes, for measurements with and without errors, respectively.

![Figure 7: Simulated temperature measurements obtained with the step heat flux](a) errorless and (b) Gaussian with standard deviation of 5% of the maximum temperature)
Figure 8: Simulated temperature measurements obtained with the triangular heat flux (a) errorless and (b) Gaussian with standard deviation of 5% of the maximum temperature.

The vector of state variables is composed of the temperatures at each of the finite-differences nodes used in the discretization, as well as by the heat flux component, for each time step. The evolution model for temperatures is given by the finite-difference discrete equations, while a random walk was used for the heat flux components, as given by Eq. (7). In this equation, $\sigma$ is the standard deviation used to advance the heat flux in time, taken to be equal to 0.7 W/m$^2$ for the step heat flux and 0.5 W/m$^2$ for the triangular heat flux, while $W_i$ are random numbers with normal distribution, zero mean, and unitary standard deviation.

$$Q_t = x_{t-1} + \sigma W_i$$

All filters previously presented were applied to the problem of estimating the two heat fluxes described above. Figures 9.a,b show the results for all filters used for the step and triangular heat fluxes, respectively. Five thousand particles were used for the SIR, ASIR, SMC and CPSE filters, while only 100 particles were used for the Sequential Monte Carlo filter without evaluation of the likelihood distribution.

The SIR filter required a computational time of 1.53 minutes for the step heat flux and 3.01 minutes for the triangular heat flux. From Figure 9.a, it is observed that SIR filter had a great difficulty in capturing the step heat flux around its discontinuities. Such filter was not effective even when the standard deviation used in the evolution model given by Eq. (7) was increased. The ASIR filter presented better results than those for the SIR, as shown in Figs. 9.a and 9.b, for both heat fluxes. The ASIR filter also required a computational time lower than the SIR filter, since it used less particles to obtain results with lower RMS - root mean squared - errors (see Tables 1 and 2).

The SMC Samplers filter presented a RMS error lower than the SIR and ASIR filters for the squared heat flux, as well as a lower computational time when compared to the previous filters, as shown in Table 1. However, for the triangular heat flux, the SMC Samplers filter presented higher values of both the RMS error and the computational time, as shown in Table 2.

The SMC Method without Likelihood presented the lowest RMS error for the squared heat flux and also the lowest computational times for both heat fluxes, as shown in Tables 1 and 2.

The Combined Parameter and State Estimation in Simulation Based Filtering presented good results when compared to the other methods. From the analysis of Figs. 9.a and 9.b, it can be seen that it presented the best estimate compared to all filters. Also, this method can estimate simultaneously the state variables and the model parameters.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Particles Number</th>
<th>Time</th>
<th>Flux Type</th>
<th>RMS Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIR</td>
<td>3000</td>
<td>1.5288 Min.</td>
<td>Step</td>
<td>0.813511</td>
</tr>
<tr>
<td>ASIR</td>
<td>2000</td>
<td>1.4630 Min.</td>
<td>Step</td>
<td>0.596332</td>
</tr>
<tr>
<td>SMC Samplers</td>
<td>100</td>
<td>1.2940 Min.</td>
<td>Step</td>
<td>0.540161</td>
</tr>
<tr>
<td>CPSE</td>
<td>3000</td>
<td>4.5024 Min.</td>
<td>Step</td>
<td>0.597597</td>
</tr>
<tr>
<td>SMC without Likelihoods</td>
<td>100</td>
<td>0.4209 Min.</td>
<td>Step</td>
<td>0.354163</td>
</tr>
</tbody>
</table>
Table 2. Triangular Heat Flux

<table>
<thead>
<tr>
<th>Filter</th>
<th>Particles Number</th>
<th>Time</th>
<th>Flux Type</th>
<th>RMS Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIR</td>
<td>5000</td>
<td>3.0082 Min.</td>
<td>Triangular</td>
<td>0.589826</td>
</tr>
<tr>
<td>ASIR</td>
<td>1000</td>
<td>0.6422 Min.</td>
<td>Triangular</td>
<td>0.115927</td>
</tr>
<tr>
<td>SMC Samplers</td>
<td>200</td>
<td>4.3139 Min.</td>
<td>Triangular</td>
<td>0.340977</td>
</tr>
<tr>
<td>CPSE</td>
<td>2000</td>
<td>2.2292 Min.</td>
<td>Triangular</td>
<td>0.125009</td>
</tr>
<tr>
<td>SMC without Likelihoods</td>
<td>100</td>
<td>0.1830 Min.</td>
<td>Triangular</td>
<td>0.253169</td>
</tr>
</tbody>
</table>

Figure 9: Estimated (a) squared and (b) triangular heat fluxes by several Bayesian filters (NP means number of particles).
5. Conclusions

In this work we applied five different Bayesian filters to a linear heat conduction problem, with the objective to recover an unknown heat flux having two different functional forms.

The following filters were used: Sampling Importance Re-sampling Filter (SIR), Auxiliary Sampling Importance Re-sampling Filter (ASIR), Sequential Monte Carlo Samplers (SMC), Combined Parameter and State Estimation in Simulation Based Filtering, and Sequential Monte Carlo without Likelihood.

The SIR filter presented the worst results among all methods. The SMC Samplers and the SMC without Likelihoods presented good results, although the Kernel of transition of the SMC Samplers method and the tolerances of the SMC without Likelihood method need to be studied in future works in order to obtain better results.

The ASIR and the CPSE filters presented the best results, being the CPSE filter capable of simultaneously estimate the state variables and the model parameters.

Acknowledgements

The authors are grateful to the financial support provided by CNPq (a fostering agency of the Brazilian Ministry of Science and Technology), CAPES (an agency of the Brazilian Ministry of Education), and FAPERJ (a fostering agency of the Rio de Janeiro State), which made this work possible.

References

[12] Z. Chen, Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond, IEEE Transactions on Signal Processing, 50(2), 2002.