# ABC-NS: a new computational inference method applied to parameter estimation and model selection in structural dynamics

**A. BEN ABDESSALEM, N. DERVILIS, D. WAGG, K. WORDEN**

Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom

## Abstract

*The inference of dynamical systems is a challenging issue, particularly when the dynamics include complex phenomena such as the existence of bifurcations and/or chaos. In this situation, the likelihood function formulated based on time-series data may be complex with several local minima and as a result not suitable for parameter inference. In the most challenging scenarios, the likelihood function may not be available in an analytical form, so a standard statistical inference is impossible to carry out. To overcome this problem, the inclusion of new features/invariants less sensitive to small variations from either the time or frequency domains seems to be potentially a very useful way to make Bayesian inference. The use of approximate Bayesian computation (ABC) or likelihood-free algorithms is an appropriate option as they offer the flexibility to use different metrics for parameter inference. However, most variants of the ABC algorithm are inefficient due to the low acceptance rate. In this contribution, a new ABC algorithm based on an ellipsoidal nested sampling technique is proposed to overcome this issue. It will be shown that the new algorithm performs perfectly well and maintains a relatively high acceptance rate through the iterative inference process. In addition to parameter estimation, the new algorithm allows one to deal with the model selection issue. To demonstrate its efficiency and robustness, a numerical example is presented.*

**Keywords: Approximate Bayesian computation, ellipsoidal sampling, system identification, model selection, Duffing oscillator**

## 1 Introduction

Model selection and parameter estimation still remain challenging issues for dynamicists, particularly for chaotic systems when small perturbations could cause significant changes in the system response. As can be seen in literature, in parameter estimation, the Bayesian approach has been widely used for linear and non-linear dynamical systems. Compared with the optimisation techniques (gradient-based methods, evolutionary algorithms, etc) which have been widely applied in the past, the Bayesian approach is more informative in the sense that one could get the full distributions of the unknown parameters. Moreover, the Bayesian method is very well suited for model selection. For more details about the implementation of the Bayesian method for parameter estimation and model selection, the reader is referred to [1, 2, 3, 4].

The application of the Bayesian approach requires the definition of a likelihood function to measure the level of agreement between the observed and simulated data. However, in some circumstances the likelihood function might not be available in a closed form. To overcome this issue and make possible the inference of complex systems, approximate Bayesian computation (ABC) has been introduced. ABC offers the possibility of using different kinds of metric to measure the similarity between simulated and observed data to infer the system. Therefore, it has attracted attention in a wide variety of applied disciplines (e.g., biology, psychology, genetics to mention a few [5, 6]) and recently in structural dynamics [7, 8]. Several variants of the ABC algorithm have been proposed in the literature, including ABC based on Markov chain Monte Carlo sampling [9] and ABC based on sequential Monte Carlo (proposed by Sisson et al. [10]), which has proven to be more efficient than [9]. It should be noted that ABC was introduced initially to infer model parameters and then was extended to deal with model selection. One common issue with the existing ABC algorithms is that the acceptance rate dramatically decreases along the iterations, which exponentially increases the computational requirements. In this paper, a new ABC algorithm based on the idea of a nested ellipsoidal sampling technique [11] is proposed; it has been named ABC-NS. In the proposed algorithm, instead of removing one particle, as in the traditional nested sampling algorithm [12], a proportion of particles are removed in each iteration (called the population in the ABC jargon) based on assigned weights. In this study, two numerical examples have been proposed to demonstrate the efficiency and robustness of the new algorithm. The first example is focused on the parameter estimation of a chaotic system. Then, the application of the algorithm is extended to model selection.

The rest of this paper is structured as follows. Section 2 provides a summary of likelihood-free Bayesian algorithms and gives details of the ABC-NS algorithm and its implementation. Two numerical examples are then given in Section 3 to illustrate the main results and the efficiency of the proposed algorithm. Finally, conclusions are given in Section 4.

# 2 Approximate Bayesian computation

## 2.1 Basic theory

In the Bayesian method, the posterior probability density, $p(\boldsymbol{\theta}|\mathcal{D})$ given observed data $\mathcal{D}$ and a model $\mathcal{M}$, can be computed using Bayes' Theorem:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta})\mathcal{L}(\mathcal{D}|\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(\boldsymbol{\theta})\mathcal{L}(\mathcal{D}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}} \propto p(\boldsymbol{\theta})\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}) \tag{1}$$

where $p(\boldsymbol{\theta})$ is the prior probability of $\boldsymbol{\theta}$ and $\mathcal{L}(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood function.

However, as mentioned earlier, explicit forms for likelihood functions are rarely available. The ABC methods approximate the likelihood through evaluating the discrepancy between the observed data and the data generated by a simulation using a given model, yielding an appoximate form of the Bayes' Theorem:

$$p(\boldsymbol{\theta}|\Delta(\mathcal{D},\mathcal{D}_s) < \varepsilon_1) \propto p\left(\Delta(\mathcal{D},\mathcal{D}_s) < \varepsilon_1|\boldsymbol{\theta}\right) p(\boldsymbol{\theta}) \tag{2}$$

where $\mathcal{D}_s \sim f(\mathcal{D}|\boldsymbol{\theta})$ is the simulated data, $\Delta$ is a discrepancy metric, and $\varepsilon_1 > 0$ is a tolerance threshold.

The most simple implementation of ABC is ABC rejection sampling (RS) as illustrated in Algorithm 1. While ABC-RS is simple to implement, it can be computationally prohibitive in practice. In the next section, the new ABC algorithm is presented. It should be noted that in the ABC algorithms in general, the identification strategy starts at a coarse resolution (higher initial $\varepsilon_1$ value), which then is adaptively refined, giving a gradually finer representation of the model parameters.

**Algorithm 1** ABC-RS

```
Require: $\mathcal{D}$: observed data, $\mathcal{M}$: model, $\varepsilon_1$
while $i \leq N$ do
    repeat
        generate $\theta^*$ from the prior distribution $\pi(\cdot)$
        simulate $\mathcal{D}_s$ using the model $\mathcal{M}(\cdot)$
    until $\Delta(\mathcal{D}_s, \mathcal{D}) < \varepsilon_1$
    set $\boldsymbol{\theta} = \theta^*$
end for
```

## 2.2 ABC-NS implementation

In this section, a brief description of the ABC-NS implementation is given, the ABC-NS algorithm broadly works following the same scheme as the ABC-SMC algorithm in [13]. The main novelties are in (i) the way of sampling, (ii) the weighting technique adopted from [5] and (iii) instead of dropping one particle per iteration, a proportion of particles is dropped, which speeds-up the algorithm without compromising the precision on the posterior estimates. The iterative process is detailed in Algorithm 2. The algorithm starts by generating $N$ particles from the prior satisfying the constraint $\Delta(u, u^*) < \varepsilon_1$ (here, $u$ for observed data and $u^*$ for simulated data). The accepted particles are then weighted (see, Step 9, Algorithm 2) and the next tolerance threshold is defined based on discrepancy values ranked in descending order (highest on top, see, step 11) as the $(\alpha_0 N)^{\text{th}}$ value where $\alpha_0$ is the proportion of dropped particles defined by the user. Then, one assigns a weight of zero to the dropped particles. After that, the weights of the remaining particles are normalised. From the remaining particles, so-called "alive" particles, one selects $\beta_0 N$ particles based on the updated weight values. The remaining particles are then enclosed in an ellipsoid in which the mass center $\mu_1$ and covariance matrix $\mathcal{C}_1$ are estimated based on those particles; denote this ellipsoid by $\mathcal{E}_1 = (\mu_1, \mathcal{C}_1)$. It should be noted that the generated ellipsoid could be enlarged by a factor $f_0$ to ensure that the particles on the borders are inside (in a similar fashion with minimum volume enclosed ellipsoid method). Ellipsoidal sampling was firstly proposed in [11] to improve the efficiency of the nested sampling algorithm which has been widely used for Bayesian inference, mainly in cosmology [14]. Finally, the population is replenished by resampling $(1 - \beta_0)N$ particles inside the enlarged ellipsoid [14] and a re-weighting step is carried out. The procedure is repeated until a stopping criterion defined by the user is met.

In the considered examples, the number of samples is set to 1000, $\alpha_0$, $\beta_0$ and $f_0$ are set to 0.3, 0.6 and 1.1, respectively. In the examples, these tuning parameters work quite well to maintain relatively high acceptance rates and they work well for the considered examples. Of course, more investigations are required to select the optimum hyperparameters defined below, this is outside the scope of this study.

**Algorithm 2** ABC-NS SAMPLER

```
Require: $u$: observed data, $\mathcal{M}(\cdot)$: model, $\varepsilon_1, N$, $\alpha_0$, $\beta_0$, $f_0$
 1: set $t = 1$
 2: for $i = 1, \dots, N$ do
 3:     repeat
 4:         Sample $\theta^*$ from the prior distributions $\pi(\cdot)$
 5:         Simulate $u^*$ using the model $\mathcal{M}(\cdot)$
 6:     until $\Delta(u, u^*) < \varepsilon_1$
 7:     set $\Theta_i = \theta^*$, $e_i = \Delta(u, u^*)$
 8: end for
 9: Associate a weight to each particle: $\omega_i \propto \frac{1}{\varepsilon_1}\left(1 - (\frac{e_i}{\varepsilon_1})^2\right)$
10: Sort $e_i$ in descending order and store them in $e^t$.
11: Define the next tolerance threshold $\varepsilon_2 = e^t(\alpha_0 N)$
12: Drop particles with $\Delta(u, u^*) \geq \varepsilon_2, \quad \omega_{j=1:\alpha_0 N} = 0$
13: Normalise the weights such that $\sum_{i=1}^{(1-\alpha_0)N} \omega_i = 1$
14: Select $\mathcal{A}^t = \beta_0 N$ particles from the remaining based on the weights
15: Define the ellipsoid by its centre of the mass and covariance matrix $\mathcal{E}_t = \{\mu_t, \mathcal{C}_t\}$
16: Enlarge the ellipsoid by $f_0$    ▷ For simplicity the same notation for the updated ellipsoid is kept
17: for $t = 2, \dots, T$ do
18:    for $j = 1, \dots, (1 - \beta_0)N$ do
19:     repeat
20:       Sample one particle $\theta^*$ inside $\mathcal{E}_{t-1}$
21:       Simulate $u^*$ using the model $\mathcal{M}(\cdot)$
22:     until $\Delta(u, u^*) < \varepsilon_t$
23:     set $\Theta_j = \theta^*$, $e_j = \Delta(u, u^*)$
24:     end for
25:       Store the new particles in $\mathcal{S}_t$
26:       Obtain the new particle set, $\mathcal{N}_{new} = [\mathcal{A}_{t-1}; \mathcal{S}_t]$ with their correspondent distance values $e^t$
27:       Sort $e^t$ and define $\varepsilon_{t+1} = e^t(\alpha_0 N)$
28:       Associate a weight to each particle as in step (9)
29:       Define the new set of selected particles $\mathcal{A}^t$ as in step (14)
30:       Update the ellipsoid hyperparameters using $\mathcal{A}^t$, $\mathcal{E}_t = \{\mu_t, \mathcal{C}_t\}$    ▷ The enlargement factor is kept constant
31: end for
```

# 3 Numerical examples

## 3.1 Parameter estimation of the Duffing oscillator using ABC-NS

In this example, one is interested in the identification of a Duffing oscillator without linear stiffness given by:

$$\ddot{z} + c\dot{z} + k_3 z^3 = f(t) \tag{3}$$

where $c$ is the damping coefficient, $k_3$ is the nonlinear stiffness and $f(t)$ is a Gaussian input force with mean zero and standard deviation of 10.

Denote by $z_0$ the initial conditions ($z(0) = 3$, $\dot{z}(0) = 4$), the system is sensitive to small variations that may affect the initial conditions. If one changes the initial conditions only by $0.1\%$ ($z_0' = [3.003, 4.004]$), the trajectories of the displacement diverge from each other as can be seen in Fig. 1. For simplicity, the initial conditions are denoted by $z_0 = (\xi_0, \xi_1) = [3, 4]$. The training data were synthetically generated by integrating numerically the above equation using the fourth-fifth order Runge-Kutta method.

Through this example, one aims to investigate the potential of the ABC-NS algorithm to deal with complex scenarios by choosing a suitable feature and a corresponding metric to carry out Bayesian inference.

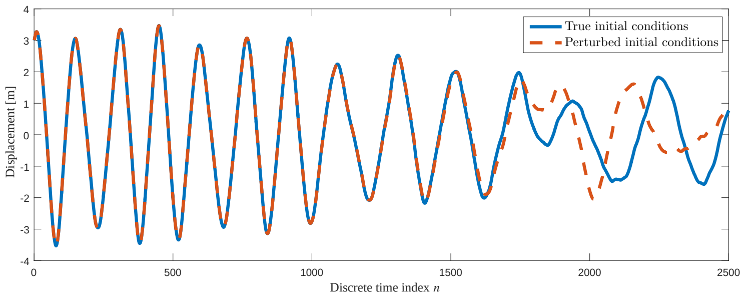

Figure 1: Comparison between the Duffing oscillator responses using the true and perturbed initial conditions.

One is interested now in implementing ABC-NS to infer model parameters and the initial conditions. Table 1 summarises the unknown parameters with the assumed priors. Fig. 2 shows in the logarithmic scale, the probability density functions (PDF)s of the acceleration obtained using the responses from the true and perturbed initial conditions for the same set of model parameters. One can see a perfect agreement between the responses which means that the PDF of the acceleration is almost entirely insensitive to small variations. In [15], it has been shown that the spectra remains nearly invariant, which means that it could potentially be used to infer model parameters.

| Parameter | True value | Lower bound | Upper bound |
|---|---|---|---|
| $c$ | 0.05 | 0.005 | 0.5 |
| $k_3$ | 50 | 5 | 500 |
| $\xi_0$ | 3 | 2.9 | 3.1 |
| $\xi_1$ | 4 | 3.9 | 4.1 |

Table 1: Parameter ranges of the cubic model.

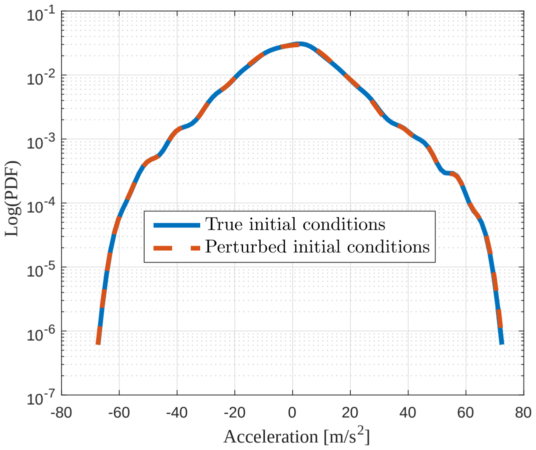

Figure 2: Comparison between the PDFs of the acceleration using the true and perturbed initial conditions in the logarithmic scale.

In this example, the model parameters and the initial conditions are treated as the parameters to be inferred. In the ABC-NS settings, $\alpha_0, \beta_0, f_0$ are set to 0.3, 0.6 and 1.1 respectively. The initial tolerance $\varepsilon_1$ is set to 100 and a population of size 1000 is used. To infer the system, the Euclidean distance between the observed and simulated PDFs, given by Eq. (4), is used to measure the discrepancy between the observed and simulated data:

$$\Delta(p_f^l, \hat{p}_f^l) = \left[ \sum_{l=1}^{n} \left( \log(p_f^l) - \log(\hat{p_f}^l) \right)^2 \right]^{\frac{1}{2}} \tag{4}$$

where $p_f^l$ and $\hat{p}_f^l$ are the probabilities associated with the observed and simulated data, respectively.

## 3.2 Results and discussion

Fig. 3 shows the histograms of the inferred parameters; from which one can observe that the obtained histograms are well peaked on the true parameters. Clearly, the PDF of the acceleration can be considered as a good feature to infer the model parameters and the initial conditions.

Fig. 4 shows the model response and the 99$\%$ confidence bounds over a few populations. One can see how by decreasing the tolerance threshold values, the model prediction is improved and one gains confidence in the results. The algorithm is stopped when negligible change on the posterior distributions is noticed.

In conclusion, the obtained results show the effectiveness and efficiency of the ABC-NS algorithm to infer the chaotic system in question. It has been shown that the ABC-NS is a convenient way to recover the model parameters, the initial conditions and the associated uncertainties precisely. In addition, the algorithm is able to achieve a low tolerance threshold value within a reasonable computational time. The ABC-NS algorithm is now applied for model selection.

## 3.3 Model selection using ABC-NS

The performance of the ABC-NS algorithm is investigated here for model selection by considering two candidate models: the cubic and cubic-quintic Duffing oscillators denoted by $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively.

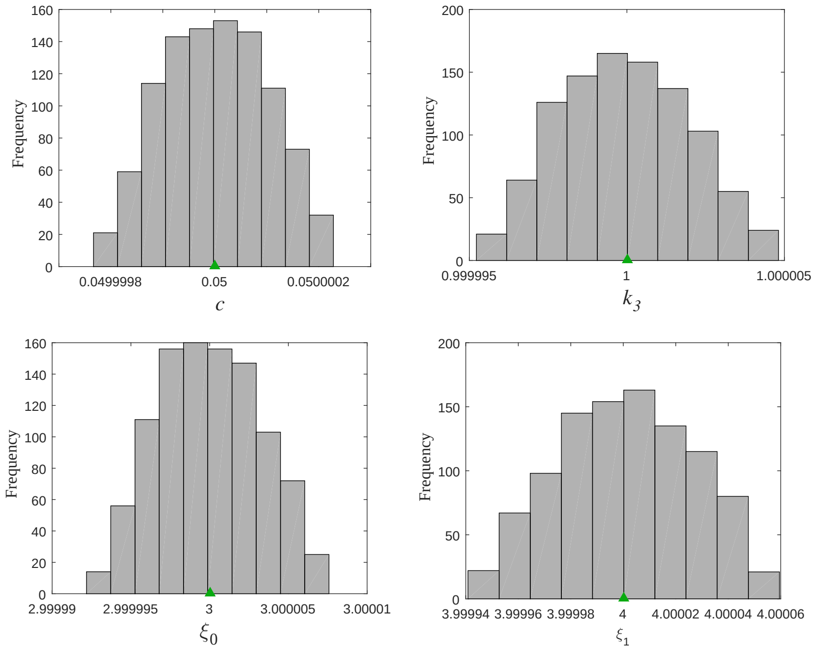

Figure 3: Histograms of the inferred model parameters (the green triangles show the true values).

The equation of motion associated to each model is given by:

$$\mathcal{M}_1 \quad : \quad \ddot{z} + c\dot{z} + kz + k_3 z^3 = f(t) \tag{5}$$

$$\mathcal{M}_2 \quad : \quad \ddot{z} + c\dot{z} + kz + k_3 z^3 + k_5 z^5 = f(t) \tag{6}$$

where $c$ is the damping, $k$ is the linear stiffness, $k_3$ and $k_5$ are the non-linear stiffness coefficients. $z$, $\dot{z}$ and $\ddot{z}$ are displacement, velocity and acceleration responses, respectively. The excitation $f(t)$ is a Gaussian sequence with mean zero and standard deviation of 10.

To make Bayesian inference, a noise-free and noisy training data sets were generated from the cubic-quintic model and shown in Fig. 5 (the first $n = 500$ data points). The noisy training data set has been corrupted with Gaussian noise of standard deviation $1\%$. To evaluate the model predictability, a set of testing data has been generated and is shown in Fig. 5. Table 2 summarises the prior lower and upper bounds associated to each unknown parameter of the models.

| Parameter | True value | Lower bound | Upper bound |
|---|---|---|---|
| $c$ | 0.05 | 0.005 | 0.5 |
| $k$ | 50 | 10 | 100 |
| $k_3$ | $10^3$ | 500 | 1500 |
| $k_5$ | $10^5$ | $0.8 \times 10^4$ | $1.5 \times 10^5$ |

Table 2: Parameter ranges of competing models.

For ABC-NS implementation, the same scheme shown in Algorithm 2 is followed by considering the candidate models as an additional hyperparameter. One sets the prior probabilities of each model to be equal, i.e., $p(\mathcal{M}_1) = p(\mathcal{M}_2) = \frac{1}{2}$. The convergence criterion used here is when the difference between two successive tolerance values is less than $10^{-7}$. For the rest, the same hyperparameters defined for Example 1 have been used. It should be noted that the number of the dropped and remaining particles

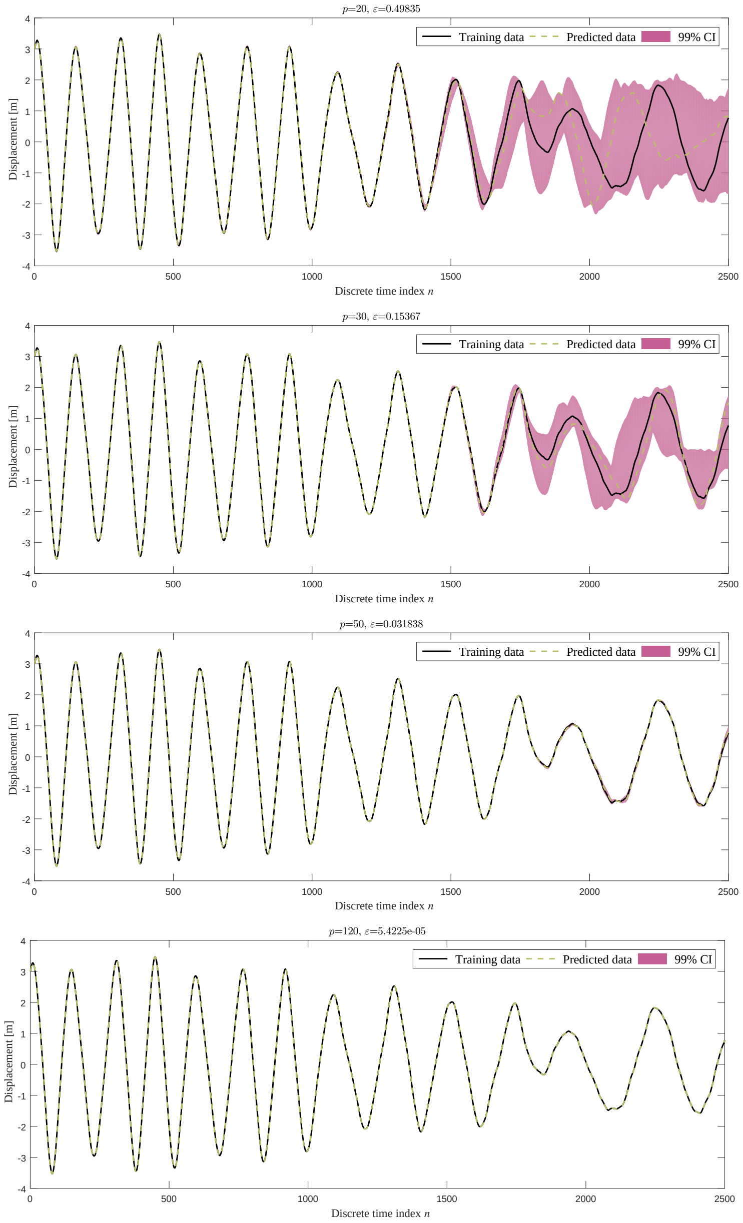

Figure 4: Predicted response and the 99% confidence bounds for different tolerance values.

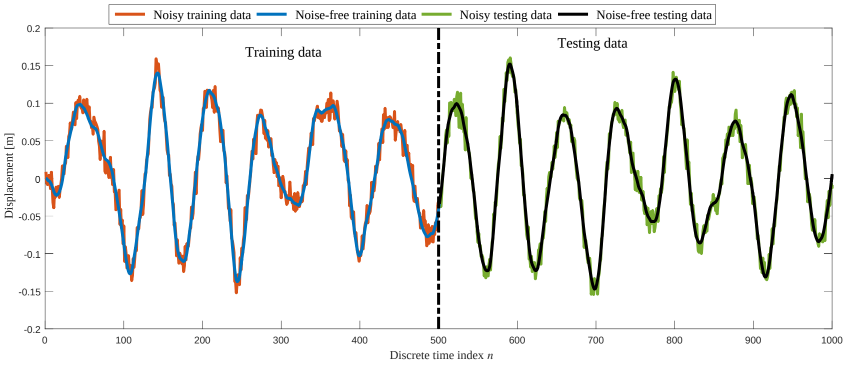

Figure 5: Training and testing data sets.

are rounded to the nearest integer in each step of the algorithm. The reader is referred to [16] for more details and examples. Finally, the normalised mean square error (MSE) given by Eq. (7) is selected as a metric to measure the discrepancy between the observed and simulated data:

$$\Delta(z^*, z) = \frac{100}{n\sigma^2_{z^*}} \sum_{i=1}^{n} (z^*_i - z_i)^2 \tag{7}$$

where $n$ is the size of the training data, $\sigma^2_{z^*}$ is the variance of the observed displacement; $z^*$ and $z$ are the observed and simulated displacements given by the model, respectively.

## 3.4 Results and discussion

Fig. 6 shows the model posterior probabilities over a selected number of populations. One can see how the ABC-NS algorithm oscillates between the competing models and finishes by converging to the correct model at population 21 (The same tendency has been shown with the noisy training data, not shown here for brevity). From the same figure, one can see that the algorithm clearly tries first to favour the cubic model, this can be seen from populations 3 to 11. Then, when the cubic model is no longer able to match the data quite well, the ABC-NS algorithm jumps to the complex model to accommodate the nonlinearity coming from the quintic term. This proves that the parsimony principle [17] is well embedded in the ABC-NS algorithm by favouring simpler models. It should be noted that in the classical Bayesian inference methods, overly-complex models are penalised through an *ad-hoc* penalty term while the ABC algorithm naturally favours simpler models as shown here, which is a major advantage.

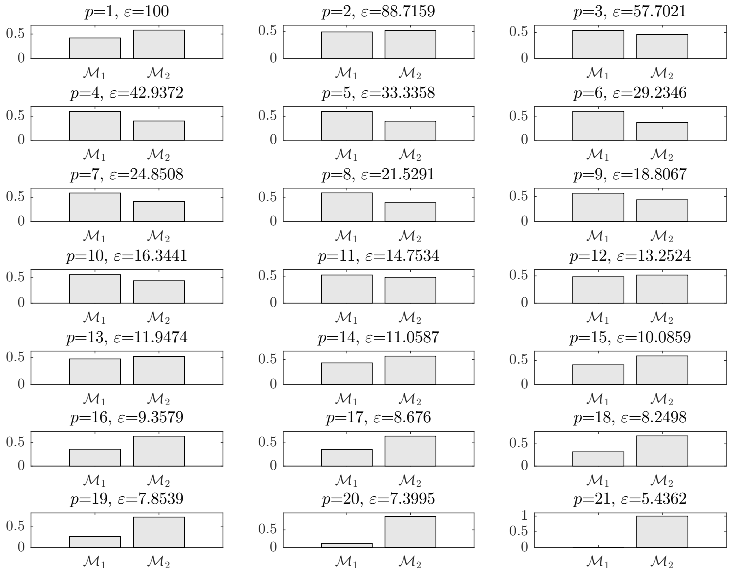

Figure 6: Model posterior probabilities.

Figs. 7 and 8 show the histograms of the unknown parameters obtained using the ABC-NS algorithm for both training data sets. For the noise-free training data set, one can see how the histograms are well peaked on the true values which proves the ability of the algorithm to correctly recover the model parameters.

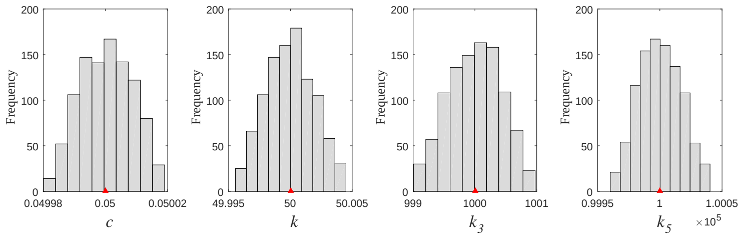

Figure 7: Histograms for the model parameters using free-noise training data set.

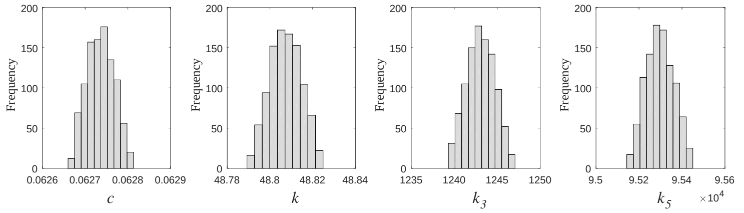

Figure 8: Histograms for the model parameters using noisy training data set.

Fig. 9 shows the acceptance rates over the populations for both cases. One can see how the ABC-NS algorithm maintains a relatively high acceptance rate over the populations considering the cubic and cubic-quintic Duffing oscillators with 4 and 5 parameters, respectively. At early populations, the acceptance rates decrease because the parameter space is large, then steadily rise as the the volume of the research space shrinks down and because one of the competing models has been already eliminated. At population 60, the acceptance rates stabilise around an average value higher than 50 per cent.

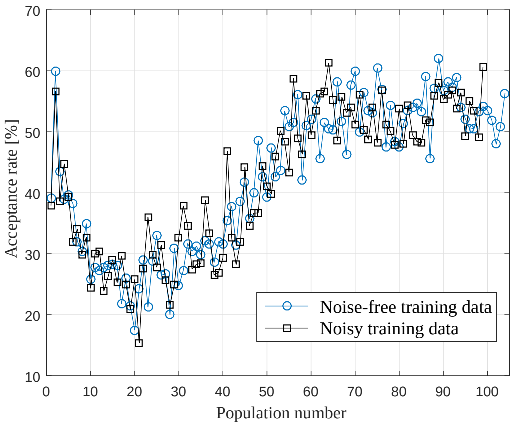

Figure 9: Acceptance rates over the populations.

Tables 3 and 4 summarise the statistics coming from the approximate Bayesian inference analysis using noise-free and noisy training data sets. The estimated model parameters are then used to evaluate the model predictability. As one can see from Figs. 10 and 11, the training data are well predicted 500 points ahead for both training data sets. The normalised MSE associated to each training data set estimated on the first 500 data points is equal to $5.02 \times 10^{-10}$ and 2.12, respectively. The prediction remains excellent based on the model parameter inferred using free-noise training data (see, Fig. 10, normalisd MSE is equal to $8.10 \times 10^{-7}$). However, the predicton based on the model parameters inferred using the noisy training data shows that beyond 865 data point the model is no longer valid (normalised MSE>5) since it is unable to provide a good prediction based on the normalised MSE values shown in Fig. 12. Based on [1], the discrepancy measure formulated in Eq. (7) has shown that a value greater than 5 generally indicates a poor model.

| Parameter | True value | Posterior estimates | | |
|---|---|---|---|---|
| | | Mean, $\mu$ | Std. Dev, $\sigma$ | [5th, 95th] percentiles |
| $c$ | 0.05 | 0.05 | $3.3893 \times 10^{-5}$ | $[4.9945 \times 10^{-2}, 5.0055 \times 10^{-2}]$ |
| $k$ | 50 | 49.9999 | $7.9975 \times 10^{-3}$ | $[49.9868, 50.0132]$ |
| $k_3$ | $10^3$ | 1000.0017 | $1.7377 \times 10^{-00}$ | $[9.9703 \times 10^2, 1.0029 \times 10^3]$ |
| $k_5$ | $10^5$ | $1.0000 \times 10^5$ | 68.1184 | $[9.9885 \times 10^4, 1.0011 \times 10^5]$ |

Table 3: Posterior estimates for the cubic-quintic model parameters, free-noise training data.

| Parameter | True value | Posterior estimates | | |
|---|---|---|---|---|
| | | Mean, $\mu$ | Std. Dev, $\sigma$ | [5th, 95th] percentiles |
| $c$ | 0.05 | 0.0627 | $3.3890 \times 10^{-5}$ | $[6.2682 \times 10^{-2}, 6.2792 \times 10^{-2}]$ |
| $k$ | 50 | 48.8073 | $8.3118 \times 10^{-3}$ | $[4.8793 \times 10^1, 4.8821 \times 10^1]$ |
| $k_3$ | $10^3$ | 1243.0963 | 1.8271 | $[1.2401 \times 10^3, 1.2460 \times 10^3]$ |
| $k_5$ | $10^5$ | $9.5297 \times 10^4$ | $7.1604 \times 10^1$ | $[9.5183 \times 10^4, 9.5415 \times 10^4]$ |

Table 4: Posterior estimates for the cubic-quintic model parameters, noisy training data.

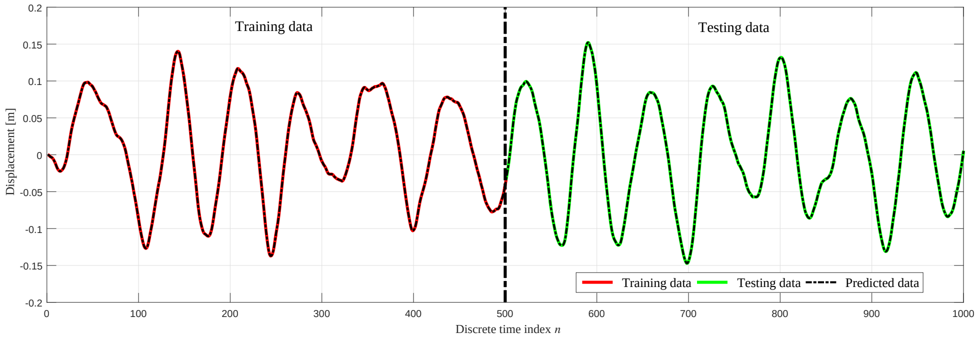

Figure 10: Model prediction using noise-free training data.

Finally, in order to check the repeatability of the model posterior probabilities, the ABC-NS algorithm is run 20 times. From Fig. 13, one can see how the ABC-NS produces repeatable results with small variations. Clearly the algorithm tries first to favour the simpler model (see the model posterior probabilities at populations 3, 6 and 12) and when a higher predictive performance is required the algorithm switches to the complex model to justify the increase in complexity.

# 4 Conclusion

A new approximate Bayesian computation algorithm named ABC-NS has been proposed in this paper for parameter estimation and model selection. It is shown through two numerical examples how the ABC-NS overcomes the low efficiency of the existing ABC algorithms by employing a nested ellipsoidal sampling method. ABC-NS maintains a high acceptance rate over the populations, which speeds

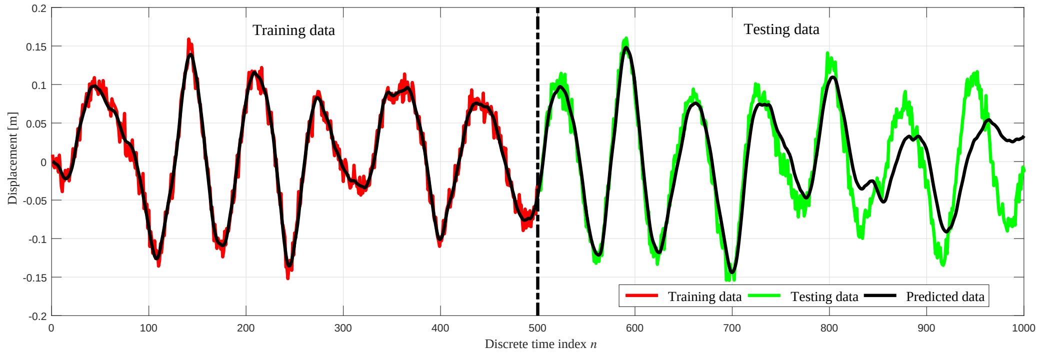

Figure 11: Model prediction using noisy training data.

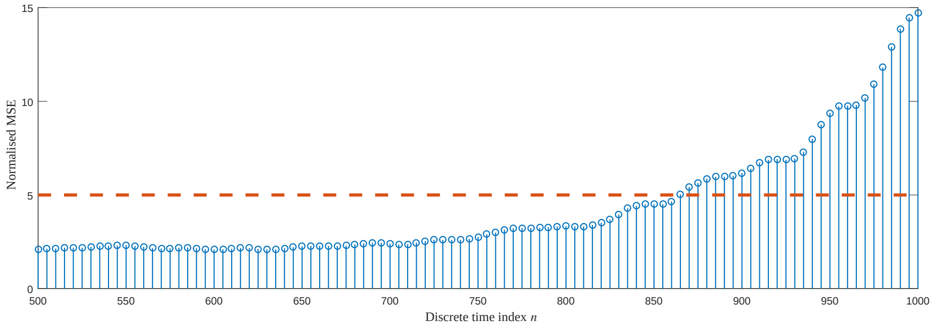

Figure 12: Fidelity of the model prediction over the testing data test.

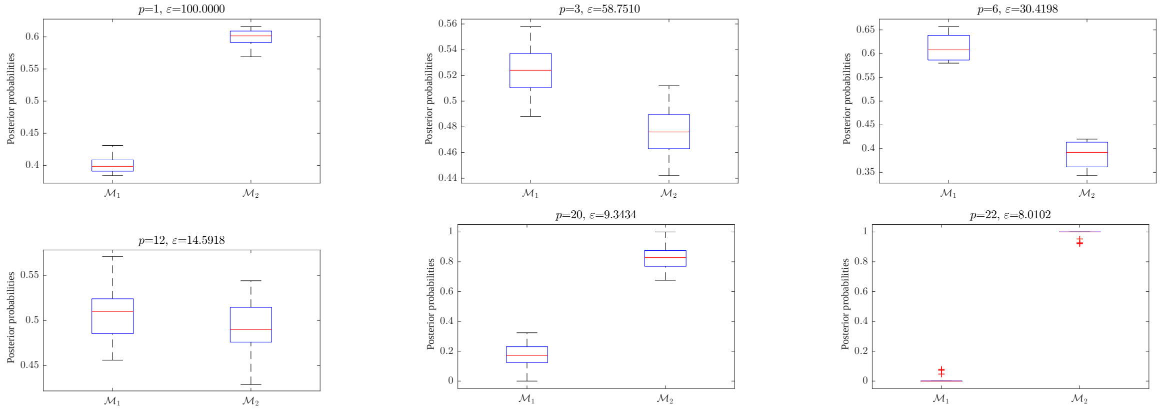

Figure 13: Boxplots of the model posterior probabilities over some selected populations (20 simulations have been performed).

up considerably the algorithm without compromising the precision of the posterior estimates. As a result, significant savings in computational effort can be achieved. Moreover, it has been shown that ABC-NS deals perfectly well with the model selection issue and that the parsimony principle is naturally embedded in it. The ABC tells one which models are supported by the data in a straightforward way without the need to evaluate any *ad-hoc* penalty terms as in the traditional Bayesian methods. A comparison of the ABC-NS algorithm with traditional ABC-SMC is left on an upcoming journal paper.

# Acknowledgement

The support of the UK Engineering and Physical Sciences Research Council (EPSRC) through grant reference no. EP/K003836/1 is greatly acknowledged.

# References

[1] K. Worden, J.J. Hensman, Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference, Mech. Syst. Signal. Process, 32 (2012) 153–169.

[2] Ph. Bisaillon. R. Sandhu. M. Khalil, C. Pettit, D. Poirel, A. Sarkar, Bayesian parameter estimation and model selection for strongly nonlinear dynamical systems, Nonlinear Dyn. 82 (2015) 1061–1080.

[3] J.L. Beck, K.V. Yuen, Model selection using response measurements: Bayesian probabilistic approach, J. Eng. Mech. 130(2): (2004) 192–203

[4] R. Sandhu, M. Khalil, A. Sarkara, D. Poirelc, Bayesian model selection for nonlinear aeroelastic systems using wind-tunnel data, Comput. Methods Appl. Mech. Eng, 282 (2014) 161–183.

[5] M.A. Beaumont, W.Zhang, D.J. Balding, Approximate Bayesian computation in population genetics, Genetics, 162 (4) (2002) 2025–2035.

[6] B.M. Turner, T. Van Zandt, A tutorial on approximate Bayesian computation, Journal of Mathematical Psychology, 56 (2012) 69–85.

[7] M. Chiachio, J.L. Beck, J. Chiachio, G. Rus, Approximate Bayesian computation by Subset Simulation, SIAM Journal on Scientific Computing, 36(3):A1339–A1338, 2014.

[8] A. Ben Abdessalem, N. Dervilis, D. Wagg, K. Worden, Identification of nonlinear dynamical systems using approximate Bayesian computation based on a sequential Monte Carlo sample,. International Conference on Noise and Vibration Engineering, September 19-21, 2016, Leuven (Belgium).

[9] P. Marjoram, J. Molitor, V. Plagnol, S. Tavare, Markov chain Monte Carlo without likelihoods, Proc. Natl. Acad. Sci. USA, 100, 15324–15328.

[10] S. Sisson, Y. Fan, M, Tanaka, Sequential Monte Carlo without likelihoods, Proc. Natn. Acad. Sci. USA, 104 (2007), 1760–1765.

[11] P. Mukherjee, D. Parkinson, A. R. Liddle, 2006, ApJ, 638, L51.

[12] J. Skilling, Nested sampling for general Bayesian computation, Bayesian Analysis, (2006) 1(4), 833-860, doi:10.1214/06-BA127.

[13] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, M.P.H. Stumpf, Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems, J. of the Royal Society Interface, 6 (2009) 187–202.

[14] F. Feroz, M.P. Hobson, M. Bridges, MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics, Mon Not R Astron Soc 398 (4):(2009) 1601–1614.

[15] K. Worden, Some thoughts on model validation for nonlinear systems, IMAC-XIX, 19$^{th}$ International Modal Analysis Conference in Orlando, Florida 2001.

[16] A. Ben Abdessalem, K. Worden, N. Dervilis, D. Wagg, Recent advances in approximate Bayesian computation methodology: application in structural dynamics, ENL Workshop, January 2017, Bristol, United Kingdom.

[17] H. Walach, Ockham's razor, in: Wiley Interdisciplinary Reviews Computational Statistics, volume 2, Sage Publications, 2007.