SIAM/ASA J. Uncertainty Quantification
Vol. 8, No. 1, pp. 114–138

Multifidelity Approximate Bayesian Computation*

Thomas P. Prescott† and Ruth E. Baker†

Abstract. A vital stage in the mathematical modeling of real-world systems is to calibrate a model's parameters to observed data. Likelihood-free parameter inference methods, such as approximate Bayesian computation (ABC), build Monte Carlo samples of the uncertain parameter distribution by comparing the data with large numbers of model simulations. However, the computational expense of generating these simulations forms a significant bottleneck in the practical application of such methods. We identify how simulations of corresponding cheap, low-fidelity models have been used separately in two complementary ways to reduce the computational expense of building these samples, at the cost of introducing additional variance to the resulting parameter estimates. We explore how these approaches can be unified so that cost and benefit are optimally balanced, and we characterize the optimal choice of how often to simulate from cheap, low-fidelity models in place of expensive, high-fidelity models in Monte Carlo ABC algorithms. The resulting early accept/reject multifidelity ABC algorithm that we propose is shown to give improved performance over existing multifidelity and high-fidelity approaches.

Key words. Bayesian inference, likelihood-free methods, stochastic simulation, multifidelity methods

AMS subject classifications. 62F15, 65C20, 65C60, 93B30, 92C42

DOI. 10.1137/18M1229742

1. Introduction.

Throughout all scientific domains, predictive models of complex dynamical systems require calibration against experimental data. Approximate Bayesian computation (ABC) is a popular likelihood-free method of parameter inference for complex models in the biomedical sciences [29]. Rather than calculating the likelihood of the data for any given parameter, the predictive model is simulated using that parameter. The likelihood is then estimated based on how close, in some sense, the observed data is to the simulated data. A classical technique is known as rejection sampling, where the likelihood is approximated with a randomly assigned value of 1 (accept) or 0 (reject), where the probability of acceptance is larger for simulations that are close to the data. The prior parameter distribution is explored by repeatedly evaluating this accept/reject decision for a large number of parameter values sampled from the prior. Therefore, ABC sampling typically requires a large number of simulations, which can form a bottleneck if the computational cost of each simulation is prohibitively high.

The efficiency of ABC can be improved using parallelization [17], or with alternative sam-

*Received by the editors November 29, 2018; accepted for publication (in revised form) October 21, 2019; published electronically January 16, 2020.
https://doi.org/10.1137/18M1229742
Funding: This work was supported by Biotechnology and Biological Sciences Research Council grant BB/R000816/1. The work of the second author was also supported by a Royal Society Wolfson Research Merit Award and a Leverhulme Research Fellowship.

†Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Andrew Wiles Building, Woodstock Road, Oxford, OX2 6GG, UK (prescott@maths.ox.ac.uk, baker@maths.ox.ac.uk).

pling strategies that reduce the number of required simulations by a more efficient exploration of the prior distribution. These include Markov chain Monte Carlo (MCMC) [20] and sequential Monte Carlo (SMC) [22, 32] approaches, which ensure that simulated parameters are sampled more often from high-likelihood regions of parameter space. A wider discussion of these sampling strategies can be found in [29, Chap. 4]. Although the parameter space is explored more efficiently with these methods, there remains a high computational burden from a large number of repeated simulations. Rather than focusing on exploring parameter space efficiently, this paper instead focuses on reducing the computational burden of the Monte Carlo sampling approach by using models that can be simulated more cheaply.

In this work, we consider a model as a map from a parameter vector to a distribution on an output space. To simulate a model is to draw from the output distribution, the computational burden of which is the simulation cost. Note that our use of "model" includes domain-specific modeling choices and numerical implementation. Many ways to approximate a given model with one that can be simulated more cheaply have been proposed and investigated, such as model reduction [1, 2, 3, 30], discretization [13], surrogate modeling [27], and early stopping [18]. Recent work [23, 24] unifies these approaches in the context of *multifidelity* methods, which integrate information from many models of the same system to accelerate tasks such as optimization, inference, and uncertainty quantification. Here, we use the terminology of Peherstorfer, Willcox, and Gunzburger [24], denoting the model being calibrated as the high-fidelity model, and other models as low-fidelity models. Simulations from low/high-fidelity models are termed low/high-fidelity simulations; we assume that low-fidelity simulations are cheaper than high-fidelity simulations.

Multilevel Monte Carlo (MLMC) [10, 11] is one example of a multifidelity estimation approach. In its original formulation, continuous-time stochastic differential equations are simulated using progressively finer, more accurate discretizations. For a given computational budget, the statistical error of a Monte Carlo estimate can be reduced by using variance reduction techniques that combine estimates built from simulations at different discretizations with common input noise. The key aim of MLMC implementation is to optimize the number of simulations using each of the different discretizations to reduce the estimator's variance.

Previous work has exploited multifidelity approaches to parameter inference [6, 7]. A multilevel approach to ABC is considered in [33], where a set of ABC samples of increasing simulation cost is produced by using progressively stricter rejection sampling thresholds, chosen to optimize the efficiency of building the overall sample. In approximate ABC (aABC) [4] a small number of simulations are used to create a low-fidelity statistical surrogate of the model output across parameter space, to which ABC is applied. Other examples include lazy ABC [25] and delayed acceptance ABC [5], where low-fidelity simulations are used to decide whether the parameter can be rejected, without necessarily needing to simulate from a high-fidelity model.

In this paper we apply multifidelity model management ideas to the specific case of rejection sampling ABC. We present a new method that allows a reject/accept decision to be made for a parameter sample using a low-fidelity simulation alone, i.e., without necessarily requiring a corresponding high-fidelity simulation. Section 2 introduces ABC and the motivation for multifidelity approaches. We develop these into a new multifidelity rejection sampling algorithm in section 3. In section 4 we describe how to analyze the performance of this algorithm

and optimize its inputs. The theoretical work is illustrated by applying the multifidelity rejection sampling algorithm to a stochastic synthetic biology model in section 5, which is also used in section 6 to illustrate practical issues around implementation. We consider a second example in section 7 and conclude with a view of potential future developments in section 8. Our code, implemented in Julia, is available at https://github.com/tpprescott/mf-abc.

2. ABC and estimators. The goal of Bayesian parameter estimation is to update prior beliefs about model parameters, θ, encoded in a prior distribution $\pi(\theta)$. The updates depend on experimental observations, y_{obs}, subject to stochasticity such as measurement and environmental noise. The parameterized model is denoted $p(\cdot \mid \theta)$, which defines a likelihood, $p(y_{\text{obs}} \mid \theta)$. The likelihood is combined with the prior distribution to give the posterior distribution, $p(\theta \mid y_{\text{obs}}) \propto p(y_{\text{obs}} \mid \theta)\pi(\theta)$. We assume that the likelihood is not available, and that we need to use ABC to estimate the posterior distribution.

The simplest version of ABC approximates the likelihood, $p(y_{\text{obs}} \mid \theta)$, of observing y_{obs} under the model, based on simulations of the model being in some sense *close enough* to y_{obs}. We denote the output of a model simulation by y, often taking values in a low-dimensional space. These model outputs are known as *summary statistics* and are an important design choice for ABC methods [9], albeit often constrained by the experimental data y_{obs} that can be gathered practically. We can thus write the approximate posterior,

$$p_{\text{ABC}}(\theta \mid y_{\text{obs}}) = \frac{p(d(y, y_{\text{obs}}) < \epsilon \mid \theta)\pi(\theta)}{Z} = \frac{p(y \in \Omega(\epsilon) \mid \theta)\pi(\theta)}{Z}, \tag{2.1}$$

where the normalization constant Z ensures the distribution has unit integral, and $\Omega(\epsilon) = \{y \mid d(y, y_{\text{obs}}) < \epsilon\}$ is the ϵ-close neighborhood of y_{obs}, where $d(y, y_{\text{obs}})$ is a distance measure between the observations, y_{obs}, and model outputs, y. The approximate posterior also induces an expectation,

$$\mathbb{E}_{\text{ABC}}(F(\theta) \mid y_{\text{obs}}) = \int F(\theta)\, p_{\text{ABC}}(\theta \mid y_{\text{obs}})\, \mathrm{d}\theta,$$

which is the ABC approximation to the posterior expectation of an arbitrary function F.

The value of $p(y \in \Omega(\epsilon) \mid \theta)$ is typically estimated using simulation. Given $\theta \sim \pi(\cdot)$ sampled from the prior distribution, we simulate $y \sim p(\cdot \mid \theta)$ from the model and calculate a weight $w(\theta) = \mathbb{I}(y \in \Omega(\epsilon))$. If we consider $w(\theta) = 0$ as rejection and $w(\theta) = 1$ as acceptance of θ, the parameter is accepted (resp., rejected) if it generates summary statistics that are close to (resp., far from) the observed data. Taking the expectation over $y \sim p(\cdot \mid \theta)$ gives $\mathbb{E}(w(\theta) \mid \theta) = p(y \in \Omega(\epsilon) \mid \theta)$. Thus, $w(\theta)$ is an unbiased estimator of the ABC approximation to the likelihood.

The weights $w(\theta)$ can be used in a Monte Carlo algorithm to build a weighted sample $\{w_i, \theta_i\}$. The simplest approach, the ABC rejection sampler (Algorithm 2.1), involves independently generating $\theta_i \sim \pi(\cdot)$ for $i = 1, \dots, N$ and setting $w_i = w(\theta_i)$. The estimator calculated by Algorithm 2.1 is

$$\mu_{\text{ABC}}(F) = \frac{\sum_i w(\theta_i)F(\theta_i)/N}{\sum_j w(\theta_j)/N} \approx \frac{Z\mathbb{E}_{\text{ABC}}(F(\theta) \mid y_{\text{obs}})}{Z} = \mathbb{E}_{\text{ABC}}(F(\theta) \mid y_{\text{obs}}). \tag{2.2}$$

Algorithm 2.1. Rejection sampling ABC.

Input: observed measurements y_{obs}; prior $\pi(\cdot)$; function $F(\theta)$; model $p(\cdot \mid \theta)$; distance function $d(\cdot, y_{\text{obs}})$; threshold ϵ; Monte Carlo sample size N.

for $i = 1, \ldots, N$ **do**
  Generate $\theta_i \sim \pi(\cdot)$.
  Simulate $y_i \sim p(\cdot \mid \theta_i)$ from the model.
  Calculate $w_i = w(\theta_i) = \mathbb{I}(d(y_i, y_{\text{obs}}) < \epsilon)$.
end for
Calculate $\mu_{\text{ABC}}(F) = \sum_{i=1}^{N} w_i F(\theta_i) / \sum_{j=1}^{N} w_j$.
return $\{w_i, \theta_i\}$ and $\mu_{\text{ABC}}(F)$.

The numerator and denominator of μ_{ABC} are each unbiased estimators of $Z\mathbb{E}_{\text{ABC}}(F(\theta) \mid y_{\text{obs}})$ and Z, respectively. Although the ratio is not an unbiased estimator of $\mathbb{E}_{\text{ABC}}(F \mid y_{\text{obs}})$, the bias of $\mu_{\text{ABC}}(F)$ vanishes as the sample size N becomes large [29].

The key issue with Algorithm 2.1 is that a large number, N, of simulations $y_i \sim p(\cdot \mid \theta_i)$ are required to generate an accurate approximation of $\mathbb{E}_{\text{ABC}}(F(\theta) \mid y_{\text{obs}})$. Rather than aiming to reduce the number, N, of simulations [20, 32], this paper considers the use of computationally cheap approximations to $w(\theta) = \mathbb{I}(y \in \Omega(\epsilon))$. The goal is to reduce the computational burden of producing $\mu_{\text{ABC}}(F)$ for any fixed number, N, of Monte Carlo sample points.

3. Multifidelity approximate Bayesian computation.

To reduce the computational cost of rejection sampling ABC, we will exploit the concept of multifidelity modeling [23]. The high-fidelity "ground truth" model, $p(\cdot \mid \theta)$, is assumed to be a computationally expensive, accurate representation of the observed system. We consider the model to be a map from parameter sample θ to a distribution on an output space containing the observations, y_{obs}. A simulation from the high-fidelity model (i.e., a high-fidelity simulation) for a particular θ is a draw $y \sim p(\cdot \mid \theta)$ from this distribution, the computational cost of which is denoted by $c(\theta)$.

We also consider a *low-fidelity* model, $\tilde{p}(\cdot \mid \theta)$, which is an alternative map from the parameter sample θ to a distribution on an output space. Note that the output space of the low-fidelity model may be different from that of the high-fidelity model; we assume that the output space is induced by taking potentially different measurements $\tilde{y}_{\text{obs}}$ from the same experiment generating the measurements comprising y_{obs}. A simulation from the low-fidelity model (i.e., a low-fidelity simulation) is a draw $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$, the computational cost of which is denoted by $\tilde{c}(\theta)$. We will assume that low-fidelity simulations are, on average, much cheaper than high-fidelity simulations, such that $\mathbb{E}(\tilde{c}(\theta)) \ll \mathbb{E}(c(\theta))$. In direct analogy with Algorithm 2.1, we define a distance function $\tilde{d}(\tilde{y}, \tilde{y}_{\text{obs}})$, measuring how close the simulated data is to the observed data, and a threshold $\tilde{\epsilon}$. These define a weight $\tilde{w}(\theta) = \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$, where we write $\tilde{\Omega}(\tilde{\epsilon}) = \{\tilde{y} \mid \tilde{d}(\tilde{y}, \tilde{y}_{\text{obs}}) < \tilde{\epsilon})\}$ for the neighborhood of the data.

The sample $\{\tilde{w}(\theta_i), \theta_i\}$ will be built more quickly than $\{w(\theta_i), \theta_i\}$, for a fixed N. However, this computational speedup comes at the cost of bias, which arises because the likelihood of the low-fidelity model does not equal that of the high-fidelity model. The ABC approximations to each likelihood are also not identical, since $\mathbb{E}(\tilde{w}(\theta)) = \tilde{p}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}) \mid \theta) \neq p(y \in \Omega(\epsilon) \mid \theta) =$

$\mathbb{E}(w(\theta))$. The bias is compounded by the fact that the observations y_{obs} and $\tilde{y}_{\text{obs}}$, distance functions d and $\tilde{d}$, and thresholds ϵ and $\tilde{\epsilon}$ may be specified independently of one another.

The goal of the remainder of this section is to consider how best to use the information generated by the low-fidelity model to reduce the reliance on the high-fidelity model in estimating $p_{\text{ABC}}(\theta \mid y_{\text{obs}})$. We aim to produce an unbiased estimate of the ABC approximation to the likelihood generated by the high-fidelity model, $p(y \in \Omega(\epsilon) \mid \theta) \approx p(y_{\text{obs}} \mid \theta)$.

3.1. Early rejection ABC. As a starting point, we will describe an existing approach that uses the low-fidelity model, $\tilde{p}(\cdot \mid \theta)$, to reduce the cost of calculating an unbiased estimator of $p(y \in \Omega(\epsilon) \mid \theta)$. A version of this approach is used in *lazy ABC* [25] and is also the key idea of *delayed acceptance MCMC* [5], but here we will refer to it as *early rejection* ABC. Recall that the weight $w(\theta)$ is an unbiased estimator of the ABC approximation to the likelihood, $p(y \in \Omega(\epsilon) \mid \theta)$, and requires a simulation of the high-fidelity model. The early rejection ABC approach generates an alternative unbiased estimator, $w_{\text{er}}(\theta)$, which saves computational costs by using the result of the low-fidelity simulation to decide whether to simulate the high-fidelity model or reject the parameter early.

For a sample θ from the prior, we first simulate $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$ from the low-fidelity model at a cost $\tilde{c}$. A continuation probability $\eta(\tilde{y}) \in (0, 1]$ is then defined, dependent on the result of the low-fidelity simulation. With probability $1 - \eta(\tilde{y})$, the parameter is *rejected early*; without simulating $y \sim p(\cdot \mid \theta)$, and therefore avoiding simulation cost c, the weight is set to $w_{\text{er}} = 0$. Otherwise, the high-fidelity simulation $y \sim p(\cdot \mid \theta)$ is generated and the parameter is accepted or rejected according to $\mathbb{I}(y \in \Omega(\epsilon))$, as before. If accepted, however, the weight is set to $w_{\text{er}} = 1/\eta(\tilde{y})$ rather than 1. For the uniform random variable $U \sim \text{Unif}(0, 1)$, we can write

$$w_{\text{er}}(\theta) = \frac{\mathbb{I}(U < \eta(\tilde{y}))}{\eta(\tilde{y})} \mathbb{I}(y \in \Omega(\epsilon)). \tag{3.1}$$

Taking the expectation with respect to U recovers $w(\theta)$, and hence $\mathbb{E}(w_{\text{er}}(\theta)) = \mathbb{E}(w(\theta)) = p(y \in \Omega(\epsilon) \mid \theta)$. Thus the early rejection estimate, $w_{\text{er}}(\theta)$, is unbiased.

The improved performance of early rejection ABC relies on the low-fidelity simulation output, $\tilde{y}$, being informative about the high-fidelity simulation output, y, and on the careful definition of the continuation probabilities $\eta(\tilde{y})$. First, as we will show in subsection 4.1, the expected time taken to compute $w_{\text{er}}(\theta)$ is less than for $w(\theta)$ if $\mathbb{E}(\eta(\tilde{y})) < 1 - \mathbb{E}(\tilde{c})/\mathbb{E}(c)$. Furthermore, suppose that $\tilde{y}$ is such that, with high probability, $y \notin \Omega(\epsilon)$ and hence θ will be rejected. Rather than generate $\mathbb{I}(y \in \Omega(\epsilon))$ at cost c, it would be preferable to reject θ early. For such $\tilde{y}$, this is achieved by ensuring $\eta(\tilde{y})$ is small. Conversely, if $\tilde{y}$ is such that, with high probability, $y \in \Omega(\epsilon)$, then θ is more likely to be accepted, corresponding to a positive value of $w_{\text{er}}(\theta)$. It follows that $\eta(\tilde{y})$ should be larger, allowing a positive weight, meaning that y is more likely to be generated. However, the converse uncovers an important asymmetry underlying the early rejection approach. If $\tilde{y}$ is such that $y \in \Omega(\epsilon)$ with high probability, then an efficient approach could be to assign a positive weight to θ without simulating $y \sim p(\cdot \mid \theta)$ from the high-fidelity model. However, such an *early acceptance* is not possible within the framework of early rejection.

3.2. Early decision ABC. Instead of using $\tilde{y}$ to determine whether or not to simulate the high-fidelity model, we now assume that this decision is independent of $\tilde{y}$. We can instead

use $\tilde{y}$ to determine the weight for θ if the high-fidelity model is not simulated. As with early rejection, for a given θ we first simulate $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$ from the low-fidelity model. Now suppose a continuation probability $\eta \in (0, 1]$ is fixed (independently of $\tilde{y}$). Then, with probability $1-\eta$, the parameter θ is accepted or rejected based on the early decision, $\mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$, without simulating the high-fidelity model and thus avoiding cost c. Otherwise, with probability η, we simulate $y \sim p(\cdot \mid \theta)$ from the high-fidelity model and calculate $\mathbb{I}(y \in \Omega(\epsilon))$ to determine acceptance or rejection, as before. The appropriate weight for θ is

$$w_{\mathrm{ed}}(\theta) = \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})) + \frac{\mathbb{I}(U < \eta)}{\eta} \left[\mathbb{I}(y \in \Omega(\epsilon)) - \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))\right], \tag{3.2}$$

where, again, taking the expectation over $U \sim \mathrm{Unif}(0, 1)$ recovers $w(\theta)$. Thus, $w_{\mathrm{ed}}(\theta)$ is another unbiased estimator for $p(y \in \Omega(\epsilon) \mid \theta)$. Note that we can consider $w_{\mathrm{ed}}(\theta)$ as a multilevel weight, since it is a randomized multilevel estimator for $p(y \in \Omega(\epsilon) \mid \theta)$ [28].

Note that, by allowing early acceptance, $w_{\mathrm{ed}}(\theta)$ can take negative values. In particular, if we simulate $U \leq \eta$, $\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})$, and $y \notin \Omega(\epsilon)$, then $w_{\mathrm{ed}} = 1 - 1/\eta \leq 0$. This is a necessary consequence of early acceptance, which may overestimate the posterior weight on θ where $\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})$. Negative weights mean that the constructed set $\{w_{\mathrm{ed}}(\theta_i), \theta_i\}$ cannot be interpreted as a weighted sample from the ABC posterior. Nevertheless, it is still valid to use $\{w_{\mathrm{ed}}(\theta_i), \theta_i\}$ in the estimator $\mu_{\mathrm{ABC}}(F)$.

3.3. Multifidelity ABC: Early acceptance and early rejection. We are now in a position to introduce early accept/reject multifidelity ABC. The approaches discussed in subsections 3.1 and 3.2 use the low-fidelity simulation output, $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$, in different ways. The early rejection weight $w_{\mathrm{er}}(\theta)$ uses $\tilde{y}$ to determine whether to simulate the high-fidelity model. In contrast, when calculating the early decision weight $w_{\mathrm{ed}}(\theta)$, we determine whether to simulate the high-fidelity model independently of $\tilde{y}$. However, w_{ed} uses $\tilde{y}$ to determine the early decision that is to be made (i.e., accept or reject θ) if the high-fidelity model is not simulated. The following expression combines these ideas in a more general, *multifidelity* weight,

$$w_{\mathrm{mf}}(\theta) = \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})) + \frac{\mathbb{I}(U < \eta(\tilde{y}))}{\eta(\tilde{y})} \left[\mathbb{I}(y \in \Omega(\epsilon)) - \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))\right], \tag{3.3}$$

where the continuation probability and early decision both depend on the output of the low-fidelity simulation.

As with early rejection ABC, the choice of continuation probability $\eta(\tilde{y})$ is important to the performance of w_{mf}. A natural form of continuation probability, and the one we consider here, is

$$\eta(\tilde{y}) = \eta_1 \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})) + \eta_2 \mathbb{I}(\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})). \tag{3.4}$$

This choice of $\eta(\tilde{y})$ allows both early acceptance and early rejection with constant probabilities $1 - \eta_1$ and $1 - \eta_2$, respectively. We will therefore refer to using w_{mf} and $\eta(\tilde{y})$ given by (3.3) and (3.4) as *early accept/reject multifidelity ABC*. Note that constraining $\eta_1 = \eta_2$ makes η independent of $\tilde{y}$ and recovers the early decision weight w_{ed}. Fixing $\eta_1 = 1$ means that there is no early acceptance and recovers the early rejection weight w_{er}. Finally, putting $\eta_1 = \eta_2 = 1$ recovers the original ABC rejection sampling weight w.

Using $\eta(\tilde{y})$ in (3.4) means that $w_{\text{mf}}(\theta)$ can take one of only four possible values:

$$w_{\text{mf}}(\theta) = \begin{cases} 1, & \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}) \cap U \geq \eta_1 \text{ (early accept)}, \\ 0, & \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}) \cap U \geq \eta_2 \text{ (early reject)}, \\ 1, & \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}) \cap y \in \Omega(\epsilon) \cap U < \eta_1 \text{ (checked true positive)}, \\ 0, & \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}) \cap y \notin \Omega(\epsilon) \cap U < \eta_2 \text{ (checked true negative)}, \\ 1 - 1/\eta_1, & \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}) \cap y \notin \Omega(\epsilon) \cap U < \eta_1 \text{ (checked false positive)}, \\ 0 + 1/\eta_2, & \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}) \cap y \in \Omega(\epsilon) \cap U < \eta_2 \text{ (checked false negative)}. \end{cases} \tag{3.5}$$

These cases imply the implementation, Algorithm 3.1, of a Monte Carlo algorithm to estimate $\mathbb{E}_{\text{ABC}}(F(\theta) \mid y_{\text{obs}})$. They also have the interesting consequence that, in addition to computational speedup, the performance of Algorithm 3.1 will be dependent on the receiver operating characteristics (ROCs) [21] of the cheap, biased binary classifier $\tilde{w}(\theta) = \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ as an approximation of the expensive binary classifier $w(\theta) = \mathbb{I}(y \in \Omega(\epsilon))$.

Algorithm 3.1. Early accept/reject multifidelity ABC.

Input: observations y_{obs} and $\tilde{y}_{\text{obs}}$ from a common experiment; prior $\pi(\cdot)$; function $F(\theta)$; low- and high-fidelity models $\tilde{p}(\cdot \mid \theta)$ and $p(\cdot \mid \theta)$; distance functions $\tilde{d}(\cdot, \tilde{y}_{\text{obs}})$ and $d(\cdot, y_{\text{obs}})$; thresholds $\tilde{\epsilon}$ and ϵ; continuation probabilities η_1 and η_2; Monte Carlo sample size N.

for $i = 1, \ldots, N$ **do**
  Generate $\theta_i \sim \pi(\cdot)$ and $U \sim \text{Unif}(0, 1)$.
  Simulate $\tilde{y}_i \sim \tilde{p}(\cdot \mid \theta_i)$ from low-fidelity model.
  Calculate $\tilde{w} = \mathbb{I}(\tilde{d}(\tilde{y}_i, \tilde{y}_{\text{obs}}) < \tilde{\epsilon})$.
  Set $w_i = \tilde{w}$.
  Set $\eta = \eta_1 \tilde{w} + \eta_2(1 - \tilde{w})$.
  if $U < \eta$ **then**
    Simulate $y_i \sim p(\cdot \mid \theta_i)$ from the high-fidelity model.
    Calculate $w = \mathbb{I}(d(y_i, y_{\text{obs}}) < \epsilon)$.
    Update $w_i = w_i + (w - w_i)/\eta$.
  end if
end for
Calculate $\mu_{\text{ABC}}(F) = \sum_{i=1}^{N} w_i F(\theta_i) / \sum_{i=1}^{N} w_i$.
return $\{w_i, \theta_i\}$ and $\mu_{\text{ABC}}(F)$.

In common with many rejection-sampling approaches, this algorithm is embarrassingly parallel: the for-loop can be implemented across many independent workers. Furthermore, rejection sampling ABC often relies on a threshold value being specified a posteriori to ensure a specific acceptance rate; the distances are ranked and ϵ is chosen so that the parameter proposals corresponding to the smallest quantile of distances are taken into the sample. In this setting, we could adapt the algorithm above into two serial components (each of which can still be parallelized). The first component applies the a posteriori thresholding approach to the low-fidelity model alone, giving weights 0 or 1 to each proposed parameter. In the second,

the high-fidelity model is simulated using a random subset of the parameter proposals, chosen based on the continuation probabilities. The weights are then corrected to give w_{mf}; at this point ϵ can be chosen to achieve a desired effective sample size (introduced in the next section). For simplicity, we only consider the case of fixed $\tilde{\epsilon}$ and ϵ in the following.

4. Performance of early accept/reject multifidelity ABC. This section considers the performance of Algorithm 3.1 in constructing the Monte Carlo sample $\{w_{\text{mf}}(\theta_i), \theta_i\}$. We discuss how to define the sample quality, and thus how to choose the inputs (η_1, η_2) to optimize performance. We will show that the multifidelity approach provides improved performance over rejection sampling ABC, and that early acceptance adds to the benefit of early rejection.

4.1. Effective sample size and efficiency. Consider a weighted sample $\{w_i, \theta_i\}$ output from an importance sampling algorithm. The weights w_i correspond to any weighting, for example, $w(\theta_i)$ or $w_{\text{mf}}(\theta_i)$. We denote the random variable taking values w_i by W. A common measure of the quality of such a sample is its effective sample size (ESS), defined as

$$\text{ESS} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2} = N\frac{(\sum_i w_i/N)^2}{\sum_i w_i^2/N} \approx N\frac{\mathbb{E}(W)^2}{\mathbb{E}(W^2)}, \tag{4.1}$$

where the approximation is taken in the limit as $N \to \infty$, and the expectations are across the proposal distribution—in this case, the prior parameter distribution, $\theta \sim \pi(\cdot)$. Note that ESS is inversely proportional to a first order approximation of the variance of $\mu_{ABC}(F)$ output by Algorithm 3.1, for any F; see section SM1 in the supplementary material for more details. Hence, we will use ESS without requiring $w_i \geq 0$ for all i.

Proposition 4.1. *Assume one or both of the following hold:*

1. $\eta_1 < 1$ *and the false positive probability,* $\mathbb{P}\big(\{\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})\} \cap \{y \notin \Omega(\epsilon)\}\big) > 0$*;*
2. $\eta_2 < 1$ *and the false negative probability* $\mathbb{P}\big(\{\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})\} \cap \{y \in \Omega(\epsilon)\}\big) > 0$.

In the limit as $N \to \infty$*, the* ESS *of the weighted sample* $\{w_{\text{mf}}(\theta_i), \theta_i\}$ *is smaller than the* ESS *of the weighted sample* $\{w(\theta_i), \theta_i\}$.

Proof. The conditional expectations $\mathbb{E}(w(\theta)) = \mathbb{E}(w_{\text{mf}}(\theta)) = p(y \in \Omega(\epsilon) \mid \theta)$ are equal and unbiased. It follows that $\mathbb{E}(w) = \mathbb{E}(w_{\text{mf}}) = p(y \in \Omega(\epsilon))$, and hence that the numerators of the limiting value of the ESS in (4.1) are equal for $w_i = w(\theta_i)$ and $w_i = w_{\text{mf}}(\theta_i)$.

It can be shown that $\mathbb{E}(w^2) = Z = p(y \in \Omega(\epsilon))$. Using (3.5) and taking expectations, we find

$$\begin{aligned}\mathbb{E}(w_{\text{mf}}^2) = \mathbb{E}(w^2) &+ \left(\frac{1}{\eta_1} - 1\right)\mathbb{P}\left(\left\{\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \notin \Omega(\epsilon)\}\right) \\ &+ \left(\frac{1}{\eta_2} - 1\right)\mathbb{P}\left(\left\{\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \in \Omega(\epsilon)\}\right).\end{aligned} \tag{4.2}$$

Assuming at least one of the two conditions in the statement gives $\mathbb{E}(w_{\text{mf}}^2) > \mathbb{E}(w^2)$, the result follows from this inequality. ∎

The goal of the multifidelity approach to rejection sampling is to build a sample more efficiently than with standard rejection sampling ABC. The smaller ESS produced by Algorithm 3.1 is the cost of early acceptance and early rejection. Equation (4.2) shows that the

marginal cost of decreasing either η_1 or η_2 is dependent on the probability of either a false positive or false negative, respectively. Clearly, if the approximation $\tilde{y}$ is a good one for y (in terms of the set membership $\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})$ as a predictor of $y \in \Omega(\epsilon)$), then the cost of early acceptance or early rejection is reduced.

Having shown that a smaller ESS is the cost of early acceptance and early rejection, we can now show how this is balanced against the intended benefit of reducing computational burden. Suppose that T_i is the time it takes to generate the weight w_i, with total simulation time $T_{\text{tot}} = \sum_i T_i$. A measure of the efficiency of building the sample $\{w_i, \theta_i\}$ is the ratio of ESS to total simulation time,

$$\frac{\text{ESS}}{T_{\text{tot}}} = \frac{(\sum_i w_i/N)^2}{(\sum_i w_i^2/N)(\sum_i T_i/N)} \approx \frac{\mathbb{E}(W)^2}{\mathbb{E}(W^2)\mathbb{E}(T)}, \tag{4.3}$$

where we have considered the limit as $N \to \infty$ and the expectations are taken across $\theta \sim \pi(\cdot)$.

The expected cost $\mathbb{E}(T)$ of computing $w_{\text{mf}}(\theta)$ over $\theta \sim \pi(\cdot)$ is

$$\mathbb{E}(T) = \mathbb{E}(\tilde{c}) + \eta_1 \mathbb{E}(c \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))\mathbb{P}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})) + \eta_2 \mathbb{E}(c \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))\mathbb{P}(\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})),$$

where $\tilde{c}(\theta)$ is the simulation cost of $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$ and $c(\theta)$ is that of $y \sim p(\cdot \mid \theta)$. If $\eta_1, \eta_2 < 1 - (\mathbb{E}(\tilde{c})/\mathbb{E}(c))$, then the expected simulation time $\mathbb{E}(T)$ to calculate w_{mf} is less than the expected cost $\mathbb{E}(c)$ of calculating w. The computational cost of calculating $w_{\text{mf}}(\theta)$ is decreased for smaller values of η_1, η_2, to a lower bound of $\tilde{c}(\theta)$. Hence, the benefit of decreasing η_1 and η_2 is a saving in computational cost, traded off against a decrease in the ESS.

4.2. Optimal continuation probabilities. Algorithm 3.1 takes the continuation probabilities (η_1, η_2) as an input, producing a sample $\{w_{\text{mf}}(\theta_i), \theta_i\}$. We now consider the choice of (η_1, η_2) that optimally balances the benefit of reducing the simulation time against the cost of reducing the ESS. Our approach is to choose (η_1, η_2) to maximize the limiting efficiency of the algorithm, defined in (4.3) as the ratio $\text{ESS}/T_{\text{tot}}$ as $N \to \infty$.

The numerator, $\mathbb{E}(w_{\text{mf}})^2 = p(y \in \Omega(\epsilon))^2$, in (4.3) is independent of η_1 and η_2. Therefore the efficiency is maximized when the denominator, $\phi(\eta_1, \eta_2) = \mathbb{E}(w_{\text{mf}}^2)\mathbb{E}(T)$, is minimized. We define

$$p_{tp} = \mathbb{P}\left(\left\{\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \in \Omega(\epsilon)\}\right), \tag{4.4a}$$

$$p_{fp} = \mathbb{P}\left(\left\{\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \notin \Omega(\epsilon)\}\right), \tag{4.4b}$$

$$p_{fn} = \mathbb{P}\left(\left\{\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \in \Omega(\epsilon)\}\right), \tag{4.4c}$$

$$c_p = \mathbb{E}(c \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))\mathbb{P}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})), \tag{4.4d}$$

$$c_n = \mathbb{E}(c \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))\mathbb{P}(\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})) \tag{4.4e}$$

to write the objective function

$$\phi(\eta_1, \eta_2) = \left((p_{tp} - p_{fp}) + \frac{1}{\eta_1} p_{fp} + \frac{1}{\eta_2} p_{fn}\right)\left(\mathbb{E}(\tilde{c}) + \eta_1 c_p + \eta_2 c_n\right). \tag{4.5}$$

The false positive and false negative probabilities, p_{fp} and p_{fn}, respectively, are the average rates at which simulations from the high- and low-fidelity models are different, defined in terms of being close to the data. The average computation time, $\mathbb{E}(c) = c_p + c_n$, to simulate the high-fidelity model is partitioned conditionally on the value of $\mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$.

Lemma 4.2. *The denominator $\phi(\eta_1, \eta_2)$ has a unique minimizer on $[0, \infty)^2$ if and only if $\mathbb{E}\big((w - \tilde{w})^2\big) < \mathbb{E}(w^2)$. The values of $\eta_1, \eta_2 \geq 0$ that minimize ϕ over $[0, \infty)^2$ are*

$$(\eta_1^\star, \eta_2^\star) = \left(\sqrt{\frac{R_p}{R_0}}, \sqrt{\frac{R_n}{R_0}}\right), \tag{4.6}$$

where

$$\begin{aligned} R_p &= \frac{p_{fp}}{c_p/\mathbb{E}(\tilde{c})} = \frac{\mathbb{P}(y \notin \Omega(\epsilon) \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))}{\mathbb{E}(c \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))/\mathbb{E}(\tilde{c})}, \\ R_n &= \frac{p_{fn}}{c_n/\mathbb{E}(\tilde{c})} = \frac{\mathbb{P}(y \in \Omega(\epsilon) \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))}{\mathbb{E}(c \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))/\mathbb{E}(\tilde{c})}, \\ R_0 &= p_{tp} - p_{fp} = \mathbb{E}(w^2) - \mathbb{E}\left((\tilde{w} - w)^2\right). \end{aligned}$$

If $R_0 \leq 0$, then $\nabla\phi \neq 0$ globally.

Lemma 4.3. *The value of $\eta_1 \in (0, 1]$ that minimizes $\phi(\eta_1, 1)$ is*

$$\bar{\eta}_1 = \min\left\{1, \eta_1^\star \Big/ \sqrt{\frac{1 + p_{fn}/R_0}{1 + c_n/\mathbb{E}(\tilde{c})}}\right\}.$$

The value of $\eta_2 \in (0, 1]$ that minimizes $\phi(1, \eta_2)$ is

$$\bar{\eta}_2 = \min\left\{1, \eta_2^\star \Big/ \sqrt{\frac{1 + p_{fp}/R_0}{1 + c_p/\mathbb{E}(\tilde{c})}}\right\}.$$

Corollary 4.4. *If $\max\{R_p, R_n\} \leq R_0$, then the continuation probabilities $\hat{\eta}_1, \hat{\eta}_2 \in (0, 1]$ that maximize the efficiency, $\mathrm{ESS}/T_{\mathrm{tot}}$, of the sample $\{w_{\mathrm{mf}}(\theta_i), \theta_i\}$ built by Algorithm 3.1 are equal to $\eta_1^\star, \eta_2^\star \in (0, 1]$ in (4.6). Conversely, if $\max\{R_p, R_n\} > R_0$, then at least one of $\eta_1^\star, \eta_2^\star > 1$, and the values of $\hat{\eta}_1, \hat{\eta}_2 \in (0, 1]$ that maximize the efficiency, $\mathrm{ESS}/T_{\mathrm{tot}}$, are*

$$(\hat{\eta}_1, \hat{\eta}_2) = \begin{cases} (1, \bar{\eta}_2), & \phi(1, \bar{\eta}_2) \leq \phi(\bar{\eta}_1, 1), \\ (\bar{\eta}_1, 1) & \text{else}, \end{cases}$$

where $\bar{\eta}_1, \bar{\eta}_2$ are given in Lemma 4.3.

Proof. The proofs of Lemmas 4.2 and 4.3 and Corollary 4.4 are sketched in the supplementary material, section SM2. ∎

The optimal continuation probabilities can be interpreted in terms of the ROC analysis of the quality of the low-fidelity classifier $\tilde{w}(\theta) = \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ as an approximation of $w(\theta) = \mathbb{I}(y \in$

$\Omega(\epsilon)$), and the computational saving of the low-fidelity model over the high-fidelity model. The false discovery rate, $\mathbb{P}(y \notin \Omega(\epsilon) \mid \tilde{y} \in \tilde{\Omega}(\epsilon))$, and the false omission rate, $\mathbb{P}(y \in \Omega(\epsilon) \mid \tilde{y} \notin \tilde{\Omega}(\epsilon))$, are conditional versions of the false positive and false negative rates in (4.4b) and (4.4c). The ratio R_p is the false discovery rate, divided by the expected time to simulate the high-fidelity model when $\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})$, expressed in units of the low-fidelity simulation cost. Smaller values of R_p occur when the false discovery rate is small, and where the low-fidelity model is much cheaper than the high-fidelity model. A similar interpretation exists for R_n, defined as the false omission rate divided by the expected time taken to simulate the high-fidelity model when $\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})$, again in units of the low-fidelity simulation cost.

The smallest values of $\eta_1^\star, \eta_2^\star$ are found when p_{fp} and p_{fn} are as small as possible, that is, where the accuracy of $\tilde{w}$ as an approximation to w is greatest. As the accuracy decreases, the benefit to the efficiency of putting $\eta_1, \eta_2 < 1$ becomes progressively less, until the optimal choice is for one or both of η_1, η_2 to be unity. In such cases, $\tilde{w}$ is not a good enough approximation to w to recommend any early acceptance and/or rejection at all.

The optimal continuation probabilities make it clear that the early accept/reject multifidelity approach relies on (i) the false discovery rate $\mathbb{P}(y \notin \Omega(\epsilon) \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ and the false omission rate $\mathbb{P}(y \in \Omega(\epsilon) \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))$ being suitably small, and (ii) the simulation costs $\mathbb{E}(c \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ and $\mathbb{E}(c \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))$ of the high-fidelity model being suitably large in comparison to the average simulation time, $\mathbb{E}(\tilde{c})$, of the low-fidelity model.

5. Example: Stochastic repressilator model.

We now illustrate the multifidelity approach to rejection sampling by its application to a stochastic model of a synthetic genetic network known as the repressilator [8]. This synthetic genetic network consists of three genes, G_1, G_2, and G_3, which are transcribed and translated into proteins P_1, P_2, and P_3, respectively. Transcription of G_2 is repressed by P_1, transcription of G_3 is repressed by P_2, and transcription of G_1 is repressed by P_3. This cycle of repression is known to cause oscillatory behavior.

5.1. Model.

The specific form of the model is adapted from that used in [32]. The chemical reaction description of the model is

$$\xrightarrow{\alpha_0+\alpha f(p_j)} \; m_i \xrightarrow{1} \emptyset \qquad \text{for } (i,j) = (1,3),\ (2,1), \text{ and } (3,2), \tag{5.1a}$$

$$m_i \xrightarrow{\beta} m_i + p_i \qquad \text{for } i = 1,2,3, \tag{5.1b}$$

$$p_i \xrightarrow{\beta} \emptyset \qquad \text{for } i = 1,2,3, \tag{5.1c}$$

where the decreasing function $f(p) = K_h^n/(K_h^n + p^n)$ models the repression of mRNA transcription by protein. The goal of parameter identification will be to identify the parameters n and K_h. For the purposes of this example, the observed data y_{obs} will be synthetic, generated by simulating the model in (5.1) using the "real" parameter values: $\alpha_0 = 1$, $\beta = 5$, $\alpha = 1000$, $n = 2$, and $K_h = 20$, to a final time of $T_{\text{final}} = 10$. For the parameter inference task, the values of α_0, α, and β are fixed at these nominal values. The remaining parameters, n and K_h, are uncertain with prior distributions $n \sim U(1,4)$ and $K_h \sim U(10,30)$. The initial conditions are fixed at $(m_1, m_2, m_3) = (0,0,0)$ and $(p_1, p_2, p_3) = (40,20,60)$.

5.2. Data generation. We used Gillespie's stochastic simulation algorithm (SSA) [12] to generate the observed data y_{obs} for the nominal parameter values. Then, for each of $N = 5 \times 10^6$ sample points (n, K_h) from the uniform prior, we generated (i) a simulation $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$ from a low-fidelity tau-leap [13] implementation of (5.1), and (ii) a simulation $y \sim p(\cdot \mid \theta)$ from the high-fidelity SSA implementation of (5.1). For more details of the stochastic simulations, we refer the reader to the tutorial [16] and to section SM3 in the supplementary material.

For each fidelity, the summary statistics are vectors of each species' molecule count at integer time-points $t = 0, 1, \ldots, 10$, such that $y_{\text{obs}} = \tilde{y}_{\text{obs}}$ from the synthetic data. The distances $\tilde{d}(\tilde{y}_{\text{obs}}, \tilde{y})$ and $d(y_{\text{obs}}, y)$ are Euclidean distances normalized by the time horizon, T_{final}, and the threshold values are $\tilde{\epsilon} = \epsilon = 50$, common to both fidelities. Figure 1 shows, for a subset of $N_{\text{sample}} = 10^4$ pairs of simulations, how the values of d vary with $\tilde{d}$ (left) and n (right). The left panel shows that the distances from the data of the high- and low-fidelity simulations are correlated. The quadrants in the left panel also show the correlation between w and $\tilde{w}$. We map the false positive and false negative simulations from the left to the right panel, where the orange points show parameter samples where $\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})$ but $y \notin \Omega(\epsilon)$, while conversely the green points show parameter values where $\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})$ but $y \in \Omega(\epsilon)$.

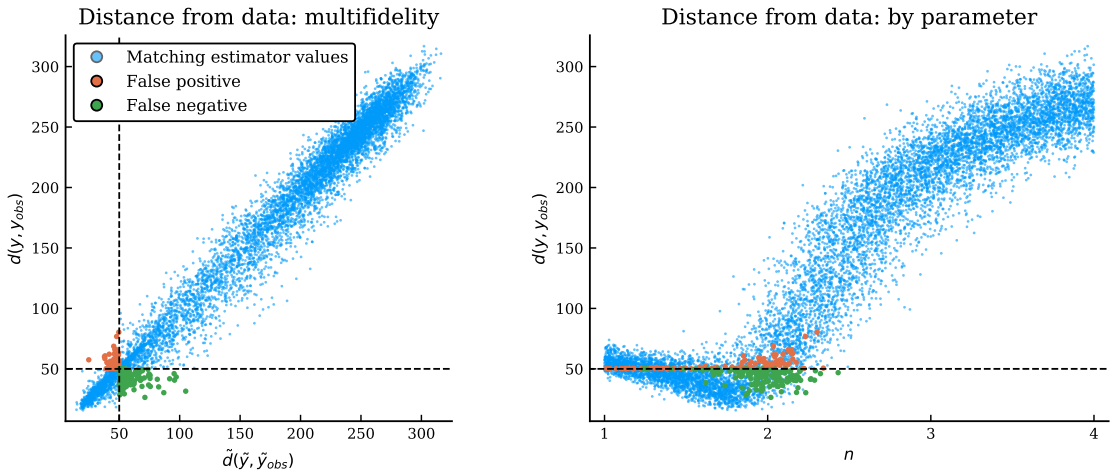

Figure 1. *Left: distances between observed data and low-fidelity (x-axis) and high-fidelity (y-axis) simulations for $N = 10^4$ sample points generated from the uniform prior. Simulations of low- and high-fidelity models are coupled by use of a common noise input (see subsection 6.1 for details). Quadrants correspond to the four possible values of $(\tilde{w}, w) \in \{0, 1\}^2$. Right: distances between observed data and high-fidelity simulations, plotted against values of n for the same $N = 10^4$ sample points of (n, K_h) generated from the uniform prior.*

5.3. Applying early accept/reject multifidelity ABC. We use the set of $N = 5 \times 10^6$ simulations as a benchmark dataset and assume that the values of the expectations and probabilities in (4.4) are given by the empirical expectations and probabilities observed in this dataset. These values can then be used to calculate the optimal continuation probabilities $(\hat{\eta}_1, \hat{\eta}_2) = (0.25, 0.12)$. In order to demonstrate the optimality of these continuation probabilities, we will compare the efficiency of Algorithm 3.1 using the values of (η_1, η_2) shown in Figure 2. We consider the following: *early rejection*, using $\eta_1 = 1$ and $\bar{\eta}_2 = 0.16$; *early*

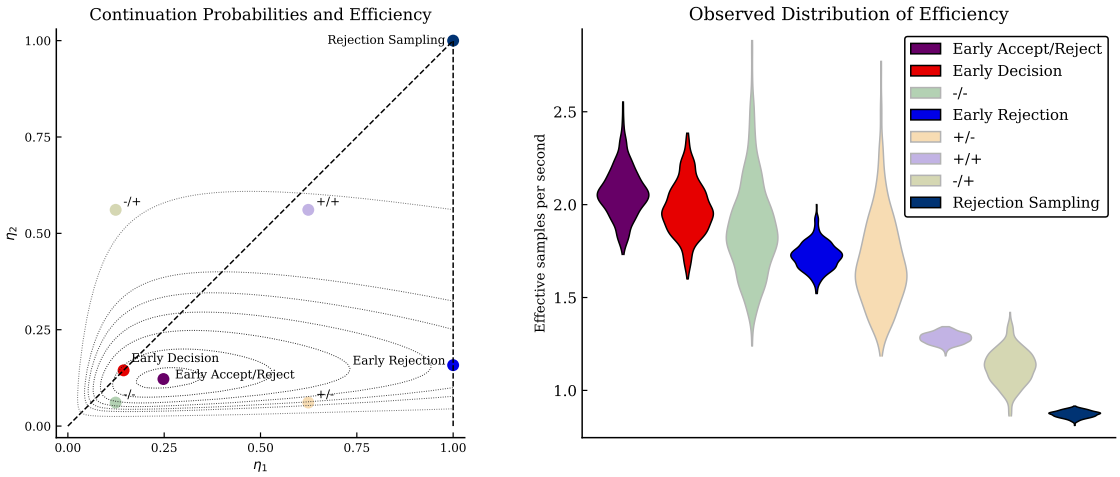

Figure 2. *Left: values of* (η_1, η_2) *used in comparison. Level sets of* $\phi(\eta_1, \eta_2)$ *are depicted, corresponding to* (η_1, η_2) *giving* 99%, 95%, 90%, 85%, 80%, 75%, *and* 60% *of the maximum theoretical efficiency* $\phi(\hat{\eta}_1, \hat{\eta}_2)$. *The broken vertical and diagonal lines give the constrained spaces in which efficiency is maximized for* early rejection *and* early decision, *respectively. Right: observed efficiencies* $\mathrm{ESS}/T_{\mathrm{tot}}$ *across* 500 *realizations of Algorithm* 3.1 *for each* (η_1, η_2), *ordered by theoretical efficiency* $\phi(\eta_1, \eta_2)$. *Lighter colors correspond to the nonoptimized* $(\hat{\eta}_1^{\pm}, \hat{\eta}_2^{\pm})$ *pairs; darker colors correspond to rejection sampling and the three optimized values of* (η_1, η_2).

Table 1

The observed probability (across 500 *samples built using each* (η_1, η_2)*) that the efficiency of a realization using* (η_1, η_2) *given by a row exceeds that using* (η_1, η_2) *given by a column. The values of* (η_1, η_2) *and the distribution of efficiencies are depicted in Figure* 2.

$\mathbb{P}$ (row exceeds column)	Early decision	$-/-$	Early rejection	$+/-$	$+/+$	$-/+$	Rejection
Early accept/reject	0.67	0.77	0.99	0.90	1.00	1.00	1.00
Early decision		0.67	0.95	0.85	1.00	1.00	1.00
$-/-$			0.71	0.70	1.00	1.00	1.00
Early rejection				0.58	1.00	1.00	1.00
$+/-$					0.99	1.00	1.00
$+/+$						0.94	1.00
$-/+$							0.99

decision, using $\eta_1 = \eta_2 = 0.14$; *rejection sampling*, using $\eta_1 = \eta_2 = 1$; and four additional nonoptimized values $(\eta_1^{\pm}, \eta_2^{\pm})$, midway between $(\hat{\eta}_1, \hat{\eta}_2)$ and each corner of $(0, 1]^2$.

To create the distributions shown in Figure 2, we partitioned the benchmark dataset into 500 subsamples of size $N_{\mathrm{sample}} = 10^4$. For each value of (η_1, η_2), we applied Algorithm 3.1 to each of the 500 subsamples and recorded the value of $\mathrm{ESS}/T_{\mathrm{tot}}$.[1] Figure 2 supports the optimality of $(\hat{\eta}_1, \hat{\eta}_2) = (0.25, 0.12)$ for maximizing the efficiency of Algorithm 3.1. Table 1 quantifies the pairwise comparisons between all eight continuity probability pairs, in terms of how many observed realizations have higher efficiency. While early decision and early rejection continuation probabilities do improve performance over rejection ABC, the theoretically optimal continuation probabilities $(\hat{\eta}_1, \hat{\eta}_2)$ give the highest efficiency across 500 realizations.

[1] Using $(\eta_1, \eta_2) = (1, 1)$ in Algorithm 3.1 to give the rejection sampling baseline efficiency is slightly unfair: Algorithm 2.1 is faster, because no low-fidelity simulations are generated. However, in subsection 6.1 we will justify this using $(\eta_1, \eta_2) = (1, 1)$ in Algorithm 3.1 as the rejection sampling baseline.

For example, we observe that 99% of realizations built using $(\hat{\eta}_1, \hat{\eta}_2)$ (early accept/reject) were more efficient than using early rejection alone. If we enable early acceptance but do not treat the continuation probabilities separately (i.e., early decision), then we still observe that 95% of such realizations are built more efficiently than using early rejection. However, 67% of samples built treating the early accept/reject continuation probabilities separately are built more efficiently than using a single continuation probability for both.

In summary, we have shown that allowing early acceptance improves performance over early rejection alone. Furthermore, this benefit is increased by treating early acceptance and early rejection separately, by optimizing the continuation probabilities such that $\eta_1 \neq \eta_2$.

6. Implementation and performance optimization.

In this section we discuss the practical issues involved in defining and optimizing the performance of Algorithm 3.1, and illustrate them in the context of the example introduced in section 5.

6.1. Variance reduction by coupling.

In Algorithm 3.1, for each $\theta_i \sim \pi(\cdot)$, we first simulate $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$ from the low-fidelity model. If $U < \eta(\tilde{y})$, we then simulate $y \sim p(\cdot \mid \theta)$ from the high-fidelity model. In the simplest case, none of the information from the low-fidelity simulation is used to simulate the high-fidelity model; however, this is not optimal. Consider the specific multifidelity approach of early stopping; in this case, the low-fidelity model replicates the high-fidelity model but only over $[0, t]$ for the stopping time $t < T_{\text{final}}$. To generate y, rather than simulating the model afresh over $[0, T_{\text{final}}]$, we can instead restart the simulation used to generate $\tilde{y}$ from its state at t and generate the trajectory over $(t, T_{\text{final}}]$. The high-fidelity model is thus simulated conditional on the low-fidelity simulation.

We can apply this concept to the more general multifidelity setting by simulating the high-fidelity model conditional on the low-fidelity simulation. Consider a model, $p(\cdot \mid \tilde{y}, \theta)$, which we will term a *coupling* between the high-fidelity and low-fidelity models, defined such that

$$\int p(\cdot \mid \tilde{y}, \theta)\tilde{p}(\tilde{y} \mid \theta)\, \mathrm{d}\tilde{y} = p(\cdot \mid \theta). \tag{6.1}$$

Given a low-fidelity simulation, $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$, consider a simulation, $y \sim p(\cdot \mid \tilde{y}, \theta)$, from the coupling, which we will term a *coupled simulation*. Then the preceding theory still holds, since (6.1) implies that the coupled simulation is a simulation of the high-fidelity model, after marginalizing out $\tilde{y}$.

One consequence of the coupled simulation being conditional on the low-fidelity simulation is that y and $\tilde{y}$, and thus the estimators w and $\tilde{w}$, will (by a judicious choice of coupling) be correlated. In the context of Algorithm 3.1 this, in turn, means that the false discovery and false omission rates $\mathbb{P}(y \notin \Omega(\epsilon) \mid \tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ and $\mathbb{P}(y \in \Omega(\epsilon) \mid \tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon}))$, respectively, can be reduced. This subsequently reduces the variance of $w_{\text{mf}}(\theta)$ as an estimator of the ABC approximation to the likelihood. A second consequence is that the time c taken to simulate $y \sim p(\cdot \mid \tilde{y}, \theta)$ from the coupling may be smaller than when simulating $y \sim p(\cdot \mid \theta)$ from the uncoupled high-fidelity model. If the reuse of information from $\tilde{y}$ means that the high-fidelity simulation time is smaller, then the optimal rate of early acceptance/rejection is lower.[2]

[2] Using $(\eta_1, \eta_2) = (1, 1)$ in Algorithm 3.1 as the baseline rejection sampler performance can be justified when the time c to simulate $y \sim p(\cdot \mid \theta)$ from the high-fidelity model is equal to the time taken to simulate both $\tilde{y}$ from the low-fidelity model and the coupled simulation $y \sim p(\cdot \mid \tilde{y}, \theta)$.

The key problem in this approach is how to define the coupling, $p(\cdot \mid \tilde{y}, \theta)$. The appropriate choice of coupling is usually specific to the details of the low- and high-fidelity models [11, 19, 23, 33]. The results presented in sections 5 and 7 are based on a coupling between low- and high-fidelity models using a common noise input, as described in section SM3.

6.2. Parameter estimation. Recall that the output of Algorithm 3.1 is a set of weights and parameter pairs $\{w_i, \theta_i\}$ that are used in the estimator

$$\mathbb{E}_{\mathrm{ABC}}(F(\theta)) \approx \frac{1}{N}\sum_i w_i F(\theta_i) \Big/ \frac{1}{N}\sum_j w_j = \mu_{\mathrm{ABC}}(F).$$

Section 4 considered the value of the continuation probabilities to optimize the efficiency $\mathrm{ESS}/T_{\mathrm{tot}}$. However, the ESS is independent of the function F being estimated by the sample. We can instead measure the performance of Algorithm 3.1 by trading off the variance of the Monte Carlo estimate $\mu_{\mathrm{ABC}}(F)$ against simulation time, a performance metric that is closer to that typically used in multilevel estimation algorithms [10].

Lemma 6.1. *The variance of μ_{ABC} can be expressed in terms of the weights $w_{\mathrm{mf}}(\theta_i)$ as approximately equal to*

$$\mathbb{V}\left(\mu_{\mathrm{ABC}}\left(F\right)\right) \approx \frac{1}{NZ^2}\mathbb{V}\left(w_{\mathrm{mf}}\left(F-\bar{F}\right)\right) = \frac{1}{N}\frac{\mathbb{E}\left(w_{\mathrm{mf}}^2\left(F-\bar{F}\right)^2\right)}{\mathbb{E}\left(w_{\mathrm{mf}}\right)^2},$$

where $\bar{F} = \mathbb{E}_{\mathrm{ABC}}(F \mid y_{\mathrm{obs}})$ is the ABC posterior expectation of F estimated by $\mu_{\mathrm{ABC}}(F)$.

Proof. This expression is derived in the supplementary material, section SM1. ∎

Corollary 6.2. *In the limit as $N \to \infty$, the product $\mathbb{V}\left(\mu_{\mathrm{ABC}}\left(F\right)\right) T_{\mathrm{tot}}$ of the estimator variance and the total simulation time can be approximated by*

$$\mathbb{V}\left(\mu_{\mathrm{ABC}}\left(F\right)\right) T_{\mathrm{tot}} \approx \frac{\mathbb{E}\left(w_{\mathrm{mf}}^2\left(F-\bar{F}\right)^2\right)\mathbb{E}\left(T\right)}{\mathbb{E}\left(w_{\mathrm{mf}}\right)^2} = \frac{\phi\left(\eta_1,\eta_2;F\right)}{\mathbb{E}\left(w_{\mathrm{mf}}\right)^2} \tag{6.2}$$

for the random time T taken to generate $w_{\mathrm{mf}}(\theta)$.

Note that the reciprocal of this approximation has a form similar to that of the limiting value of $\mathrm{ESS}/T_{\mathrm{tot}}$ and can therefore be thought of as an estimator-specific efficiency. As $\mathbb{E}(w_{\mathrm{mf}})^2$ is independent of (η_1, η_2), the optimal trade-off is where $\phi(\eta_1, \eta_2; F)$ is minimized.

The expected computation time, $\mathbb{E}(T)$, is given in section 4. However, the factor corresponding to the second moment is now F-dependent, such that

$$\begin{aligned}\mathbb{E}\left(w_{\mathrm{mf}}^2\left(F-\bar{F}\right)^2\right) &= \int \left(F(\theta)-\bar{F}\right)^2 \mathbb{E}\left(w_{\mathrm{mf}}^2 \mid \theta\right)\pi(\theta)\,\mathrm{d}\theta \\ &= \left(p_{tp}(F) - p_{fp}(F)\right) + \frac{1}{\eta_1}p_{fp}(F) + \frac{1}{\eta_2}p_{fn}(F),\end{aligned}$$

where

$$p_{tp}(F) = \int \mathbb{P}\left(\left\{\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \in \Omega(\epsilon)\} \mid \theta\right)(F(\theta) - \bar{F})^2 \pi(\theta)\, \mathrm{d}\theta, \tag{6.3a}$$

$$p_{fp}(F) = \int \mathbb{P}\left(\left\{\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \notin \Omega(\epsilon)\} \mid \theta\right)(F(\theta) - \bar{F})^2 \pi(\theta)\, \mathrm{d}\theta, \tag{6.3b}$$

$$p_{fn}(F) = \int \mathbb{P}\left(\left\{\tilde{y} \notin \tilde{\Omega}(\tilde{\epsilon})\right\} \cap \{y \in \Omega(\epsilon)\} \mid \theta\right)(F(\theta) - \bar{F})^2 \pi(\theta)\, \mathrm{d}\theta. \tag{6.3c}$$

In these coefficients, values of θ generating false positives and false negatives are now penalized based on how much they contribute to the variance. The optimal continuation probabilities $(\hat{\eta}_1, \hat{\eta}_2)$ specific to a given estimator $\mu_{\mathrm{ABC}}(F) \approx \mathbb{E}_{\mathrm{ABC}}(F(\theta)) = \bar{F}$ can now be found by replacing p_{tp}, p_{fp}, p_{fn} in Lemmas 4.2 and 4.3 and Corollary 4.4 with the respective F-dependent parameters in (6.3).

To illustrate the impact of this alternative performance metric on the continuation probabilities, we return to the repressilator example of section 5. We consider three functions of the uncertain parameter n to estimate the following: $F_1(n) = \mathbb{I}(n \in (1.9, 2.1))$; $F_2(n) = \mathbb{I}(n \in (1.2, 1.4))$; and $F_3(n) = n$. The optimal pairs $(\hat{\eta}_1, \hat{\eta}_2)_i$ for each function are $(\hat{\eta}_1, \hat{\eta}_2)_1 = (0.44, 0.28)$, $(\hat{\eta}_1, \hat{\eta}_2)_2 = (0.23, 0.06)$, and $(\hat{\eta}_1, \hat{\eta}_2)_3 = (0.38, 0.20)$. These clearly deviate, to different degrees, from the optimal continuation probabilities for maximizing $\mathrm{ESS}/T_{\mathrm{tot}}$ of $(\hat{\eta}_1, \hat{\eta}_2) = (0.25, 0.12)$. Importantly, although the computational time saved by early rejection or acceptance does not change, the contribution of false positives and false negatives to increasing the variance is different enough to change the optimal continuation probabilities.

To demonstrate the efficiency of each pair of continuation probabilities, we partition the benchmark data into 5000 subsamples of size 1000 and run Algorithm 3.1 on each. Furthermore, we stop Algorithm 3.1 early once the total simulation time reaches 30 seconds; in this example, every subsample is of size less than 1000. For each subsample we estimate $\mu(F_i)$; Table 2 then records the variance across the resulting 5000 estimates. By fixing the computational cost for every subsample, the variances in the estimator $\mu_{\mathrm{ABC}}(F_i)$ across 5000 subsamples for each value of (η_1, η_2) can be directly compared. Using any $(\eta_1, \eta_2) \neq (1, 1)$ seems to outperform rejection sampling, but the early accept/reject continuation probabilities are the best performing. By also showing the values of $\phi(\eta_1, \eta_2; F_i)$, we can see that the observed sample variances remain approximately proportional to this objective function. The percentage values in Table 2 show that the expected benefits of optimizing $\phi(\eta_1, \eta_2; F)$ over $(\eta_1, \eta_2) \in (0, 1]^2$ may be marginal, depending on the function, F, being estimated. Furthermore, using $(\eta_1, \eta_2) = (0.25, 0.12)$ chosen to optimize ESS clearly does not produce the lowest variances across all functions F_i.

6.3. Estimating optimal continuation probabilities. The values of $(\hat{\eta}_1, \hat{\eta}_2)$ depend on the values given in (4.4) and (6.3), which are based on the times taken to generate $\tilde{y}$ and y, together with the ROC values of $\tilde{w}$ as an approximation of w. Thus, in the absence of any initial information about computation times and ROC values, the optimal continuation probabilities cannot be known in advance. Before applying Algorithm 3.1 we therefore need a burn-in period to enable reasonable estimates of the values in (4.4) and (6.3).

Suppose that, at iteration m of Algorithm 3.1, both $\tilde{y}_i$ and y_i have been generated for k of the m sampled parameter values, θ. The other $m - k$ values of θ have only generated

Table 2

Continuation probabilities, objective function values (multiplied by 10^3), and the sample variance (multiplied by 10^3) of 5000 estimates of $\mu_{\mathrm{ABC}}(F_i)$, $i = 1, 2, 3$, each built with Algorithm 3.1 for a fixed simulation budget of 30 seconds. Optimal ESS *refers to (η_1, η_2) chosen to minimize* ESS*, independently of F. Percentages are reductions relative to rejection sampling (first column).*

$\times 10^{-3}$	Rejection	Early rejection	Early decision	Early accept/reject	Optimal ESS
$F_1 = \mathbb{I}(n \in (1.9, 2.1))$					
(η_1, η_2)	$(1, 1)$	$(1, 0.35)$	$(0.30, 0.30)$	$(0.44, 0.28)$	$(0.25, 0.12)$
$\phi(\eta_1, \eta_2;\ F_1)$	6.14	4.68 (24%)	4.50 (27%)	4.41 (28%)	5.09 (17%)
Sample variance	4.17	3.16 (24%)	3.04 (27%)	2.85 (32%)	3.26 (22%)
$F_2 = \mathbb{I}(n \in (1.2, 2.4))$					
(η_1, η_2)	$(1, 1)$	$(1, 0.08)$	$(0.10, 0.10)$	$(0.23, 0.06)$	$(0.25, 0.12)$
$\phi(\eta_1, \eta_2;\ F_2)$	8.14	3.16 (61%)	2.85 (65%)	2.53 (69%)	2.70 (67%)
Sample variance	5.63	2.03 (64%)	1.97 (65%)	1.69 (70%)	1.82 (68%)
$F_3 = n$					
(η_1, η_2)	$(1, 1)$	$(1, 0.25)$	$(0.23, 0.23)$	$(0.38, 0.20)$	$(0.25, 0.12)$
$\phi(\eta_1, \eta_2;\ F_3)$	4.08	2.64 (35%)	2.51 (38%)	2.41 (41%)	2.55 (38%)
Sample variance	2.82	1.72 (39%)	1.68 (40%)	1.59 (44%)	1.66 (41%)

$\tilde{y}$ and have been accepted or rejected early. We denote the index sets $I_m = \{1, \dots, m\}$ and $I_k = \{i \in I_m \text{ s.t. both } \tilde{y}_i, y_i \text{ generated}\}$ and write $\rho_m = \sum_{I_m} \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon}))/m$ and $\rho_k = \sum_{I_k} \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon}))/k$. Natural estimates of the computation times are

$$\mathbb{E}(\tilde{c}) \approx \frac{1}{m} \sum_{i \in I_m} \tilde{c}(\theta_i), \tag{6.4a}$$

$$\bar{c}_p = \frac{\rho_m}{\rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} c(\theta_i) \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon})), \tag{6.4b}$$

$$\bar{c}_n = \frac{1 - \rho_m}{1 - \rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} c(\theta_i) \mathbb{I}(\tilde{y}_i \notin \tilde{\Omega}(\tilde{\epsilon})). \tag{6.4c}$$

The remainder of the values in (4.4) are similarly estimated by

$$\bar{p}_{tp} = \frac{\rho_m}{\rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} \mathbb{I}(y_i \in \Omega(\epsilon)) \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon})), \tag{6.5a}$$

$$\bar{p}_{fp} = \frac{\rho_m}{\rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} \mathbb{I}(y_i \notin \Omega(\epsilon)) \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon})), \tag{6.5b}$$

$$\bar{p}_{fn} = \frac{1 - \rho_m}{1 - \rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} \mathbb{I}(y_i \in \Omega(\epsilon)) \mathbb{I}(\tilde{y}_i \notin \tilde{\Omega}(\tilde{\epsilon})), \tag{6.5c}$$

while the F-dependent integrals in (6.3) are estimated through

$$\bar{p}_{tp}(F) = \frac{\rho_m}{\rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} (F(\theta_i) - \bar{\mu})^2 \mathbb{I}(y_i \in \Omega(\epsilon)) \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon})), \tag{6.6a}$$

$$\bar{p}_{fp}(F) = \frac{\rho_m}{\rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} (F(\theta_i) - \bar{\mu})^2 \mathbb{I}(y_i \notin \Omega(\epsilon)) \mathbb{I}(\tilde{y}_i \in \tilde{\Omega}(\tilde{\epsilon})), \tag{6.6b}$$

Algorithm 6.1. Adaptive early accept/reject multifidelity ABC.

Input: observations data y_{obs} and $\tilde{y}_{\text{obs}}$ from a common experiment; prior $\pi(\cdot)$; function $F(\theta)$; low- and high-fidelity models $\tilde{p}(\cdot \mid \theta)$ and $p(\cdot \mid \theta)$; distance functions $\tilde{d}(\cdot, \tilde{y}_{\text{obs}})$ and $d(\cdot, y_{\text{obs}})$; thresholds $\tilde{\epsilon}$ and ϵ; lower bounds $\eta_{1,0}$ and $\eta_{2,0}$; Monte Carlo sample size N; burn-in length $M < N$.

Initialize $(\eta_1, \eta_2) = (1, 1)$ and set $I_m = I_k = \emptyset$.
for $i = 1, \ldots, N$ **do**
  Generate $\theta_i \sim \pi(\cdot)$ and $U \sim \text{Unif}(0, 1)$.
  Generate $\tilde{y}_i \sim \tilde{p}(\cdot \mid \theta_i)$ from the low-fidelity model.
  Calculate $\tilde{w} = \mathbb{I}(\tilde{d}(\tilde{y}_i, \tilde{y}_{\text{obs}}) < \tilde{\epsilon})$.
  Set $w_i = \tilde{w}$.
  Set $\eta = \eta_1 \tilde{w} + \eta_2(1 - \tilde{w})$.
  if $U < \eta$ **then**
    Generate $y_i \sim p(\cdot \mid \theta_i)$ from the high-fidelity model.
    Calculate $w = \mathbb{I}(d(y_i, y_{\text{obs}}) < \epsilon)$.
    Update $w_i = w_i + (w - w_i)/\eta$.
    Update $I_k = I_k \cup \{i\}$.
  end if
  Update $I_n = I_n \cup \{i\}$.
  Set $n = |I_n|$ and $k = |I_k|$.
  if $k \geq M$ **then**
    Recalculate values in (6.4)–(6.6).
    Estimate optimal $\hat{\eta}_1$ and $\hat{\eta}_2$ according to Lemmas 4.2 and 4.3 and Corollary 4.4.
    Update $\eta_1 = \hat{\eta}_1$ and $\eta_2 = \hat{\eta}_2$.
    Update $\eta_1 = \max(\eta_1, \eta_{1,0})$ and $\eta_2 = \max(\eta_2, \eta_{2,0})$.
  end if
end for
Calculate $\mu_{\text{ABC}} = \sum_{i=1}^N w_i F(\theta_i) / \sum_{i=1}^N w_i$.
return μ_{ABC}

$$\bar{p}_{fn}(F) = \frac{1 - \rho_m}{1 - \rho_k} \cdot \frac{1}{k} \sum_{i \in I_k} (F(\theta_i) - \bar{\mu})^2 \mathbb{I}(y_i \in \Omega(\epsilon)) \mathbb{I}(\tilde{y}_i \notin \Omega(\epsilon)), \tag{6.6c}$$

where $\bar{\mu} = \sum_{i \in I_m} F(\theta_i) w_i / \sum_{j \in I_m} w_j$. In practical implementations of Algorithm 3.1, we propose beginning with a burn-in run by using $(\eta_1, \eta_2) = (1, 1)$ for a suitably large number $M < N$ of sample points θ_i. We can then estimate optimal continuation probabilities $(\hat{\eta}_1, \hat{\eta}_2)$ using the estimates given in (6.4)–(6.6) to use for subsequent iterations, $i = M + 1, \ldots, N$. Note that the values in (6.4)–(6.6) will continue to evolve over $i > M$. We can therefore adapt the continuation probabilities (η_1, η_2) used for subsequent iterations. Algorithm 6.1 combines a burn-in period of length M with an adaptation of continuation probabilities (η_1, η_2) toward an evolving estimate of the optimum, subject to lower bounds $\eta_{1,0}$ and $\eta_{2,0}$.

This algorithm is no longer "embarrassingly" parallel, although many copies of the for

loop could run independently to produce a sample, potentially exchanging information on an optimal (η_1, η_2). It also requires a priori fixed ϵ and $\tilde{\epsilon}$ for the optimal continuation probabilities to be well-defined, and therefore cannot target a particular acceptance rate. More importantly, there are no longer guarantees of the consistency of the resulting estimate, although the example in the following section shows good performance. To guarantee consistency, the adaptive phase may be followed by running Algorithm 3.1 with fixed continuation probabilities equal to (η_1, η_2) found by the end of the adaptive phase.

7. Example: Viral kinetics.

We conclude with a further example using a model of intracellular viral kinetics [15, 31].

7.1. Model.

A cell is initially infected with a single viral template. Templates hijack cellular processes to produce new viral genomes and structural protein, which combine to produce new viral vectors that are expelled from the cell. Alternatively, viral genomes can become new templates, and templates and structural protein can also decay. We denote the counts of each molecule at time t by the vector

$$(\text{template}(t), \text{genome}(t), \text{struct}(t), \text{virus}(t)) = (x_1(t), x_2(t), x_3(t), x_4(t)).$$

The six reactions are

$$\text{template} \xrightarrow{k_1} \text{template} + \text{genome}, \tag{7.1a}$$

$$\text{genome} \xrightarrow{k_2} \text{template}, \tag{7.1b}$$

$$\text{template} \xrightarrow{k_3} \text{template} + \text{struct}, \tag{7.1c}$$

$$\text{template} \xrightarrow{k_4} \emptyset, \tag{7.1d}$$

$$\text{struct} \xrightarrow{k_5} \emptyset, \tag{7.1e}$$

$$\text{genome} + \text{struct} \xrightarrow{k_6} \text{virus}. \tag{7.1f}$$

We use initial conditions $x(0) = (1, 0, 0, 0)$ and time horizon $[0, T_{\text{final}}] = [0, 200]$. One important characteristic of this system is that cells can randomly recover from small-scale infection, whenever the template decays before enough genome is produced to set off the positive feedback loop leading to viral infection. Even if not recovered, cells can stay latently infected for a randomly long period of time before $\text{virus}(t) > 0$.

7.2. Data generation.

The goal of parameter identification will be to identify the reaction rates k_i, $i = 1, \ldots, 6$, in (7.1). We first generate synthetic data y_{obs} using the exact Gillespie SSA [12] with nominal parameters $(1, 0.025, 100, 0.25, 1.9985, 7.5 \times 10^{-5})$. Ten independent simulations are produced, each corresponding to a cell in a population of size ten with a common, randomly selected parameter set. The prior distribution on each uncertain parameter k_i is log-uniform around its nominal value; that is, we multiply the nominal value of k_i by 1.5^{u_i} for $u_i \in U(-1, 1)$. The initial conditions are fixed at a single viral template, $x_1(0) = 1$.

The low-fidelity model is an adaptation of that given in [15]. For the parameter ranges considered in this example, the propensities of the reactions in (7.1c) and (7.1e) are extremely large relative to those of the other reactions in (7.1). Low-fidelity model simulations are

therefore generated using a hybrid stochastic/deterministic algorithm [15] that avoids the computational bottleneck arising from excessive firings of the fast reactions. We approximate these reactions by considering only their net effect on the mean molecule count, ignoring the fast stochastic fluctuations around the slowly evolving mean. In this example, we simulate the high-fidelity model conditional on the simulation of the low-fidelity model using a coupling, $p(\cdot \mid \tilde{y}, \theta)$, that shares the random noise input between the two simulations. For more details on the coupling approach, see the supplementary material, section SM3.

For $M = 10^5$ sample parameters generated from the prior distribution for $(k_1, \ldots, k_6)$, we produced ten simulations from the low-fidelity model with ten coupled simulations from the high-fidelity model, corresponding to populations of size-ten cells for each parameter vector. The summary statistics are defined as follows. First, a cell is considered infected if it has output a nontrivial number of virus replicates over the 200-day horizon, such that $x_4(200) > 3$. Each population thus has a number of infected cells: the three-dimensional summary statistics y and $\tilde{y}$ are (i) the infected percentage of the population, (ii) $\log_2$ of the average viral output of each infected cell by $t = 200$, and (iii) the average percentage along the time horizon that an infected cell first exceeds the detection threshold of 3. If there are zero infected cells, we use the zero vector. The distances $\tilde{d}(\cdot, \tilde{y}_{\text{obs}})$ and $d(\cdot, y_{\text{obs}})$ are both the Euclidean distance between summary statistics, shown in Figure 3 for $N = 10^4$ pairs of simulations.

Note that, in comparison to the repressilator example (Figure 1), the distances in this case are much more closely correlated. However, the relative speed-up in simulation times is not significantly different: the average cost of a low-fidelity simulation is 17.0% of an average high-fidelity simulation in the repressilator example, compared to 17.4% in this example. The improved accuracy in Figure 3 suggests that the optimal continuation probabilities $(\hat{\eta}_1, \hat{\eta}_2)$ should be smaller, as fewer corrections will be needed.

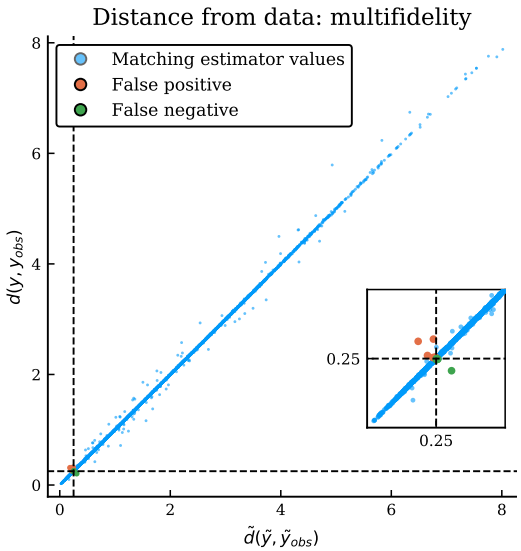

Figure 3. *Distances between observed data and low-fidelity (x-axis) and coupled high-fidelity (y-axis) simulations for $M = 10^4$ sample points generated from the six-dimensional log-uniform prior. Quadrants correspond to the four possible values of $(\tilde{w}, w) \in \{0, 1\}^2$. Inset: region where $\tilde{d}$ and d are close to $\tilde{\epsilon} = \epsilon = 0.25$.*

7.3. Applying early accept/reject multifidelity ABC. We return to measuring a sample's quality by $\mathrm{ESS}/T_{\mathrm{tot}}$. Taking the full set of 10^5 pairs of simulations implies optimal continuation probabilities of $(\hat{\eta}_1, \hat{\eta}_2) = (0.161, 0.048)$. We produced 100 independent runs of the adaptive phase of Algorithm 6.1, using this full set as the burn-in set each time; thus, the adaptive (η_1, η_2) values began at $(0.161, 0.048)$. The leftmost plot in Figure 4 shows the observed distribution of the efficiencies of the 100 samples built during the adaptive phase. This is clearly multimodal; some samples are built much less efficiently than others.

This is a consequence of $(\hat{\eta}_1, \hat{\eta}_2)$ being the optimal continuation probabilities only in the asymptotic limit. Due to the accuracy of the low-fidelity model, shown in Figure 3, observed misclassifications $w(\theta_i) \neq \tilde{w}(\theta_i)$ are relatively rare events within a finite sample. When these rare events do happen, they lead to a much smaller ESS. For example, assuming that the continuation probabilities stay approximately equal to $(\hat{\eta}_1, \hat{\eta}_2)$, then $w_{\mathrm{mf}} = -5.22$ for a false positive and $w_{\mathrm{mf}} = 20.82$ for a false negative. Each realization used to construct the leftmost plot in Figure 4 thus effectively contains a Poisson number of weights $w_{\mathrm{mf}}(\theta_i) \in \{-5.22, 20.82\}$, each of which significantly decreases the ESS, inducing a multimodal distribution for efficiency.

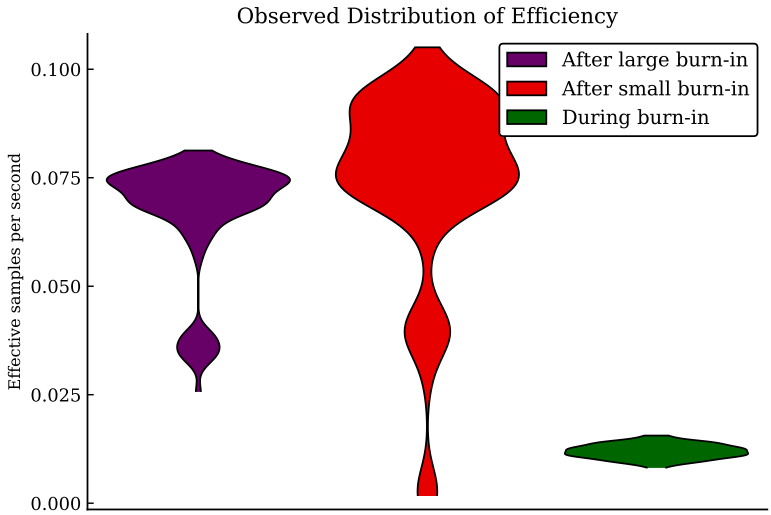

Figure 4. *The efficiency distribution of* 100 *samples built using Algorithm* 6.1. *"After large/small burn-in" depicts the efficiencies of the part of the samples built during the adaptive phase, with a starting value of* (η_1, η_2) *derived using a common burn-in sample of size* 10^5 *(large) or* 100 *independent burn-in samples of size* 10^3 *(small). "During burn-in" depicts the distribution of the efficiency for each of the* 100 *small burn-in phases.*

Recall that the objective functions $\phi(\eta_1, \eta_2)$ and $\phi(\eta_1, \eta_2; F)$ are the products of the limiting values of the second moment and computation time as the sample size, N, approaches infinity. This example demonstrates that when the false discovery rate and false omission rate, and hence the continuation probabilities, are particularly small, the effect of finite N becomes more important. We hypothesize that the sample size N can be considered large enough for an accurate estimate of p_{fp}, etc. (and hence of the optimal continuation probabilities) only when the number of weights $w_{\mathrm{mf}}(\theta_i) \notin \{0, 1\}$ is suitably large. Future work could potentially aim to further optimize the continuation probabilities by taking into account a fixed N or computational budget $\sum_i T_i < B$ more explicitly.

7.4. Shorter burn-in estimates. The burn-in set of 10^5 pairs of simulations took 132 hours of computation time. We created a further 100 independent samples using the adaptive phase of Algorithm 6.1, but this time also partitioned the burn-in set into 100 independent subsamples of size $M = 10^3$. The center and right plots in Figure 4 show the distributions of efficiency across the 100 samples using this shorter burn-in phase during the adaptive phase (red) and initial burn-in (green). Clearly, the portions of each sample built during the burn-in phase are built much less efficiently, on average, than the portions of the samples built during the adaptive phase. However, the small burn-in duration leads to an even more pronounced multimodal efficiency distribution during the adaptive phase, and the effective sample size of some samples has collapsed due to large-magnitude weights.

To observe how far the weights are from the optimum, Figure 5 shows the variability in the continuation probabilities used when applying Algorithm 6.1. The values of (η_1, η_2) used at the beginning of the adaptive phase are shown in blue, and in orange are the resulting values at the end of the adaptive phase. A point lies at each of the fixed lower boundaries $\eta_1 = 0.01$ or $\eta_2 = 0.01$ if no false positive or no false negative has been observed, respectively. The continuation probabilities may lie on a lower boundary at the start of the adaptive phase, but during the adaptive phase a false positive or false negative may be observed. The resulting sample will then include a weight of 100 or -99; these are the samples of extremely low efficiency shown in the red plot in Figure 4, as the effective sample size will be significantly decreased. However, the continuation probabilities that lie on a lower boundary at the end of the adaptive phase (i.e., the orange points) are those where no false positive or false negative has been observed during either the burn-in or the adaptive phase. These are the samples with extremely high efficiency in the red plot in Figure 4. Similarly to the case of a long burn-in phase, future development of the adaptive approach should focus on identifying corrections to (η_1, η_2) to account for these finite sample size effects.

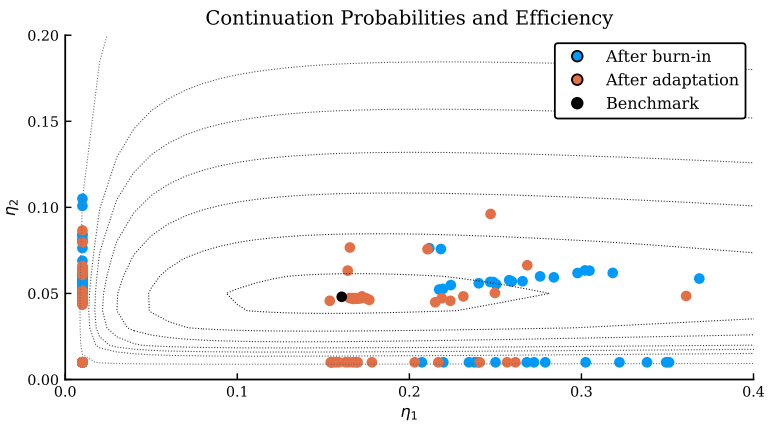

Figure 5. *Cloud of near-optimal estimates of* $(\hat{\eta}_1, \hat{\eta}_2)$ *output by Algorithm 6.1. The black point is the "true" value of* $(\hat{\eta}_1, \hat{\eta}_2)$ *estimated using the entire benchmark dataset. Contours are level sets corresponding to* 99%, 95%, 90%, 85%, 80%, 75%, *and* 60% *of the theoretical maximum efficiency achieved at* $(\hat{\eta}_1, \hat{\eta}_2)$.

8. Discussion and conclusions. In this work, we have considered the use of multifidelity methods to improve the efficiency of constructing ABC estimators by optimally combining high- and low-fidelity models. We combined the strengths of early rejection and early decision approaches to construct a multifidelity method with both early acceptance and early rejection of parameter samples, which treats the choice of whether to simulate the high-fidelity model differently, depending on the output of the low-fidelity simulation. One consequence of this is that parameter samples for which the high-fidelity model is simulated are not distributed across the parameter space according to the prior, π. The early accept/reject method can thus be interpreted as an importance sampling approach, with an importance distribution induced by the low-fidelity model.

The samples built in Algorithms 3.1 and 6.1 will contain negative weights whenever $\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon})$ and $y \notin \Omega(\epsilon)$. These negative weights mean that the resulting set of weights and parameters $\{w_i, \theta_i\}$ cannot be treated as a weighted sample from the ABC posterior. As a result, Algorithm 3.1 with $\eta_1 < 1$ cannot currently be adapted to methods reliant on resampling, such as SMC-ABC [20, 29, 32], or to the rejection approach of discarding the proposal used in MCMC-ABC [34]. An SMC approach would allow the acceptance thresholds ϵ and $\tilde{\epsilon}$ to be selected adaptively rather than be fixed a priori, and for the continuation probabilities to adapt with them. In future work we will combine the multifidelity approach with ABC-SMC by investigating how to sample from a weighted kernel density estimate with positive and negative weights.

Up to now, we have considered only a single low-fidelity model. There are often situations where there are multiple competing low-fidelity models, the accuracy and computational savings of which vary across parameter space. The low-fidelity models therefore do not necessarily form a hierarchy of progressively increasing accuracy or cost that is valid across all of parameter space, although such hierarchies may exist locally [24]. For example, both the accuracy and relative speed-up of the approximation in section 7 will vary with parameters k_3 and k_5. One strength of the multifidelity method proposed here is that the requirement for high-fidelity simulations varies across parameter and simulation space, without any analytical input. Hence, we expect that adapting the approach described in this work to situations with multiple low-fidelity models should focus computational effort toward models that give the greatest benefits, potentially uncovering local hierarchies in model fidelity in the process.

The continuation probability $\eta(\tilde{y})$ was chosen in (3.4) to depend on the value of $\mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ in order to implement early acceptance and early rejection with constant probabilities. However, there is no reason to constrain $\eta(\tilde{y})$ to this form. Further work in this area could also explore the potential of generalizations, such as $\eta(\tilde{y}) = \sum_{r=1}^{R} \eta_r \mathbb{I}(\tilde{y} \in \Omega_r)$, for any partition Ω_r of the output space of the low-fidelity model, $\tilde{p}(\cdot \mid \theta)$. For example, we could consider $\Omega_r = \{\tilde{d}(\tilde{y}, \tilde{y}_{\text{obs}}) \in (\tilde{\epsilon}_r, \tilde{\epsilon}_{r-1})\}$ for a decreasing sequence of thresholds $\{\tilde{\epsilon}_r\}$. Another option would be to also include explicit θ dependence into the continuation probability $\eta(\tilde{y}, \theta)$ to reflect, for example, the effect of θ on the times taken to simulate $\tilde{y} \sim \tilde{p}(\cdot \mid \theta)$ and $y \sim p(\cdot \mid \theta)$, or knowledge about $F(\theta)$.

Subsection 6.3 discusses one way of dealing with the lack of a priori knowledge on the ROC analysis of the cheap rejection sampler $\tilde{w} = \mathbb{I}(\tilde{y} \in \tilde{\Omega}(\tilde{\epsilon}))$ as an approximation to the expensive rejection sampler $w = \mathbb{I}(y \in \Omega(\epsilon))$ and hence of the optimal continuation probabilities. However, different application areas may provide low-fidelity models with known

error bounds relative to the high-fidelity models, such as standard results on balanced truncation [14], for example. It may be possible to use these bounds to reduce uncertainty in the ROC values more efficiently than in Algorithm 6.1. This approach is likely to be much more application-driven, as much error estimation theory for model reduction is based on specific model reductions and specific model outputs and summary statistics [3, 26].

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, 2005, https://doi.org/10.1137/1.9780898718713.

[2] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715.

[3] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, SIAM, 2017, https://doi.org/10.1137/1.9781611974829.

[4] E. O. Buzbas and N. A. Rosenberg, *AABC: Approximate approximate Bayesian computation for inference in population-genetic models*, Theoret. Popul. Biol., 99 (2015), pp. 31–42, https://doi.org/10.1016/j.tpb.2014.09.002.

[5] J. A. Christen and C. Fox, *Markov chain Monte Carlo using an approximation*, J. Comput. Graph. Statist., 14 (2005), pp. 795–810, https://doi.org/10.1198/106186005x76983.

[6] T. Cui, Y. M. Marzouk, and K. E. Willcox, *Data-driven model reduction for the Bayesian solution of inverse problems*, Internat. J. Numer. Methods Engrg., 102 (2015), pp. 966–990, https://doi.org/10.1002/nme.4748.

[7] T. Cui, Y. M. Marzouk, and K. E. Willcox, *Scalable posterior approximations for large-scale Bayesian inverse problems via likelihood-informed parameter and state reduction*, J. Comput. Phys., 315 (2016), pp. 363–387, https://doi.org/10.1016/j.jcp.2016.03.055.

[8] M. B. Elowitz and S. Leibler, *A synthetic oscillatory network of transcriptional regulators*, Nature, 403 (2000), pp. 335–338, https://doi.org/10.1038/35002125.

[9] P. Fearnhead and D. Prangle, *Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 74 (2012), pp. 419–474, https://doi.org/10.1111/j.1467-9868.2011.01010.x.

[10] M. B. Giles, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617, https://doi.org/10.1287/opre.1070.0496.

[11] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numer., 24 (2015), pp. 259–328, https://doi.org/10.1017/S096249291500001X.

[12] D. T. Gillespie, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem., 81 (1977), pp. 2340–2361, https://doi.org/10.1021/j100540a008.

[13] D. T. Gillespie, *Approximate accelerated stochastic simulation of chemically reacting systems*, J. Chem. Phys., 115 (2001), pp. 1716–1733, https://doi.org/10.1063/1.1378322.

[14] S. Gugercin and A. C. Antoulas, *A survey of model reduction by balanced truncation and some new results*, Internat. J. Control, 77 (2004), pp. 748–766, https://doi.org/10.1080/00207170410001713448.

[15] E. L. Haseltine and J. B. Rawlings, *Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics*, J. Chem. Phys., 117 (2002), pp. 6959–6969, https://doi.org/10.1063/1.1505860.

[16] D. J. Higham, *Modeling and simulating chemical reactions*, SIAM Rev., 50 (2008), pp. 347–368, https://doi.org/10.1137/060666457.

[17] N. Jagiella, D. Rickert, F. J. Theis, and J. Hasenauer, *Parallelization and high-performance computing enables automated statistical inference of multi-scale models*, Cell Systems, 4 (2017), pp. 194–206, https://doi.org/10.1016/j.cels.2016.12.002.

[18] H. Jiang, *Statistical properties of an early stopping rule for resampling-based multiple testing*, Biometrika, 99 (2012), pp. 973–980, https://doi.org/10.1093/biomet/ass051.

[19] C. Lester, C. A. Yates, M. B. Giles, and R. E. Baker, *An adaptive multi-level simulation algo-*

rithm for stochastic biological systems, J. Chem. Phys., 142 (2015), 024113, https://doi.org/10.1063/1.4904980.

[20] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré, *Markov chain Monte Carlo without likelihoods*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 15324–15328, https://doi.org/10.1073/pnas.0306899100.

[21] C. E. Metz, *Basic principles of ROC analysis*, Seminars Nuclear Med., 8 (1978), pp. 283–298, https://doi.org/10.1016/S0001-2998(78)80014-2.

[22] P. D. Moral, A. Doucet, and A. Jasra, *An adaptive sequential Monte Carlo method for approximate Bayesian computation*, Statist. Comput., 22 (2012), pp. 1009–1020, https://doi.org/10.1007/s11222-011-9271-y.

[23] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Survey of multifidelity methods in uncertainty propagation, inference, and optimization*, SIAM Rev., 60 (2018), pp. 550–591, https://doi.org/10.1137/16M1082469.

[24] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Optimal model management for multifidelity Monte Carlo estimation*, SIAM J. Sci. Comput., 38 (2016), pp. A3136–A3194, https://doi.org/10.1137/15M1046472.

[25] D. Prangle, *Lazy ABC*, Statist Comput., 26 (2016), pp. 171–185, https://doi.org/10.1007/s11222-014-9544-3.

[26] T. P. Prescott and A. Papachristodoulou, *Guaranteed error bounds for structured complexity reduction of biochemical networks*, J. Theoret. Biol., 304 (2012), pp. 172–182, https://doi.org/10.1016/j.jtbi.2012.04.002.

[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[28] C.-H. Rhee and P. W. Glynn, *Unbiased estimation with square root convergence for SDE models*, Oper. Res., 63 (2015), pp. 1026–1043, https://doi.org/10.1287/opre.2015.1404.

[29] S. A. Sisson, Y. Fan, and M. A. Beaumont, *Handbook of Approximate Bayesian Computation*, Chapman & Hall/CRC Press, 2018.

[30] T. J. Snowden, P. H. van der Graaf, and M. J. Tindall, *Methods of model reduction for large-scale biological systems: A survey of current methods and trends*, Bull. Math. Biol., 79 (2017), pp. 1449–1486, https://doi.org/10.1007/s11538-017-0277-2.

[31] R. Srivastava, L. You, J. Summers, and J. Yin, *Stochastic vs. deterministic modeling of intracellular viral kinetics*, J. Theoret. Biol., 218 (2002), pp. 309–321, https://doi.org/10.1006/jtbi.2002.3078.

[32] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf, *Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems*, J. Roy. Soc. Interface, 6 (2009), pp. 187–202, https://doi.org/10.1098/rsif.2008.0172.

[33] D. J. Warne, R. E. Baker, and M. J. Simpson, *Multilevel rejection sampling for approximate Bayesian computation*, Comput. Statist. Data Anal., 124 (2018), pp. 71–86, https://doi.org/10.1016/j.csda.2018.02.009, https://arxiv.org/abs/1702.03126v3.

[34] S. Wiqvist, U. Picchini, and J. L. Forman, *Accelerating Delayed-Acceptance Markov Chain Monte Carlo Algorithms*, preprint, https://arxiv.org/abs/1806.05982v1, 2018.