Proceedings Paper:
Hristov, PO, DiazDelaO, FA, Kubiak, KJ orcid.org/0000-0002-6571-2530 et al. (1 more author) (2016) Reliability analysis and surrogate modelling of biodiesel filters. In: Risk, Reliability and Safety: Innovating Theory and Practice. ESREL 2016: 26th European Safety and Reliability Conference, 25-29 Sep 2016, Glasgow, UK. CRC Press , pp. 390-395. ISBN 978-1-138-02997-2

Reliability Analysis and Surrogate Modelling of Biodiesel Filters

P.O. Hristov & F.A. DiazDelaO
Institute for Risk and Uncertainty, University of Liverpool, United Kingdom

K.J. Kubiak
School of Computing and Engineering, University of Huddersfield, United Kingdom

U. Farooq
Parker Hannifin Manufacturing (UK) Ltd.

This paper was published in 'Risk, Reliability and Safety: Innovating Theory and Practice - Proceedings of the 26th European Safety and Reliability Conference, ESREL 2016, held at the University of Strathclyde in Glasgow, Scotland (25–29 September 2016).

ABSTRACT: Biodiesel is a biofuel that offers a number of advantages, but is also prone to water contamination. Free water in the fuel can, both decrease engine performance and cause damage. Small droplets are especially hard to separate once they form emulsion in the fuel. In this paper the sparation processes are simulated via lattice Boltzmann modelling. These types of models have a variety of advantages, but, as most complex codes, suffer from high computational cost. This causes problems when various statistical analyses are needed, since these procedures often require many runs. In order to ameliorate this problem, a Gaussian process emulator is used with the time consuming code. To estimate the probability of failure, caused by abnormal operation conditions, Subset Simulation is used. A single fibre lattice Boltzmann model was created and run with three input variables - droplet and fibre diameter, and fibre surface energy. The emulator was successfully used to create an inexpensive approximation of the code which was then sampled via subset simulation to efficiently calculate the probability of filter failure due to excessive water volume escaping the filter. The results of this work will benefit the improved performance of engines using biodiesel, through the numerical modelling of an optimum coalescer filter structure and by estimating and minimising filter failure mode arising due to an anomaly in the multiple variables affecting the water separability.

1 INTRODUCTION

Studies indicate that at the present usage rates, some of the main fossil fuel resources will have run out by 2042 (Shafiee and Topal (2009)). Biodiesel has a number of advantages, the chief of which are its *renewability* and *carbon neutrality*. The former means that, unlike fossil fuels, the raw materials needed for the production of biodiesel can be cultivated and grown to respond to demands. This reduces the cases where arable land, forests or the sea bottom are destroyed by oil drilling and coal quarrying. Carbon neutrality means that all the carbon that is produced by burning the fuel to generate power, has already been consumed in the process of photosynthesis while the raw materials have been grown. Biodiesel can also be used in both its pure form or as a part of a blend with normal petroleum diesel. However, because of its chemical composition, biodiesel is very prone to water contamination, which forms a stable emulsion in the fuel medium. Free water in the fuel promotes biological growth in the fuel tanks, which results in corrosion and the forming of slime which could block and damage the fuel injection system (Fregolente et al. (2014)). The majority of the contaminant can be separated with the help of mechanical (centrifuges) or static devices (coalescers). However separating water droplets on the microscopic scale (5 - 10 μm) is a challenging endeavour (Schweitzer (1997)). This is due to the relatively low interfacial tension between biodiesel and water (Yang et al. (2007)). Special additives (surfactants) are mixed in the fuel to improve the performance of the engine at low temperatures (Dwivedi et al. (2011)). These chemicals deteriorate the separability of water even further. The process of efficiently separating the two liquids with porous fibrous filters involves the investigation of the role of many different variables. These could be *chemical properties* (interfacial and surface tension, density, viscosity) of both water and fuel, *contaminant properties* (droplet size distribution), *filter medium characteristics* (fibre diameter distribution, surface energy, orientation) and *ambient properties* (pressure, temperature, velocity) to name but a few. The research

focuses on identifying optimal filter media structures that will improve water separation efficiency and recognizing critical variables that can adversely affect the filtration performance under a given set of conditions.

In order to study the relationship between all variables and filtration efficiency, a numerical model is built in place of physical experiments. The lattice Boltzmann model (LBM) (also referred to as *the simulator*) places the simulation domain on grid (or a lattice) and associates local particle velocities and densities with each node of the grid. The process then simulates the interaction between the particles in two separate steps - *collision* and *streaming* (Sukop and Thorne (2006)). It can be shown that under some general assumptions the Navier-Stokes equations can be fully recovered from the model, making it a valid tool for simulating processes involving fluids (Wolf-Gladrow (2000)).

Given the complex nature of the process being simulated, it is expected that the simulator will be computationally expensive. This is a problem, since various tests and validation processes, among which reliability analysis, require thousands of evaluations of the code under different input combinations. Therefore, *surrogate modelling* is used extensively in this project. Surrogate modelling and in particular Gaussian process emulation (GPE) is a powerful technique for approximating the output of expensive computer codes. Once constructed and validated, the surrogate can be used independently of the simulator, thus saving an appreciable amount of time and allowing different analyses of the code to be performed (O'Hagan (2006)).

There are many possible regimes under which a filter could fail. For a typical well engineered system, however the *probability of failure* is small, which means that a lot of testing (under the form of simulations) needs to be done in order for it to be reliably estimated. In addition, the regions of the input space resulting in failure could be disjoint. Subset simulation (SuS)(Au and Beck (2001)) can be used to tackle both of these problems. The algorithm relies on the generation of a series of nested, less-rare events of fixed probability, which could be easily populated. Instead of trying to sample the whole input space, once the current less-rare region is populated, the corresponding samples are used to seed a Markov Chain Monte Carlo (MCMC) algorithm which populates the next conditional level. This operation is repeated until a predefined number of samples from the true failure region is obtained.

The present paper has two main goals. Firstly, to investigate the use of emulation techniques with lattice Boltzmann models and secondly to test the ability to combine the emulator results with reliability tools (SuS). Combining the emulator with subset simulation will provide an affordable reliability estimate and hence means to minimise filter failure mode in the event of an anomaly amongst different variables or any combinations thereof. In this work the failure of the separator due to its saturation with water is discussed. The remainder of the paper is organized as follows: Section 2 offers a theoretical overview of lattice Boltzmann modelling. Section 4 briefly discusses the details behind Gaussian process emulation. Section 3 summarizes the idea of subset simulation. Selected results are presented in Section 5. Finally, Section 6, provides a discussion and some conclusions.

2 LATTICE BOLTZMANN MODELS

LBM is an approach which stems from methods initially developed for research on dilute gases (Sukop and Thorne (2006)). Consider a gas which consists of hard spherical particles. These particles can only interact with each other via elastic collisions. If one is able to determine the position and momentum of each individual particle at a particular instant in time, one can fully characterize the behaviour of the gas under observation. That is, a gas with N number of particles can be characterized by constructing the *particle distribution function* $f^N(\mathbf{x}^N, \mathbf{p}^N, t)$, such that $f^N(\mathbf{x}^N, \mathbf{p}^N, t)\delta\mathbf{x}\delta\mathbf{p}$ represents the probable number of particles in a infinitesimal volume, centred around $\mathbf{x}, \mathbf{p}$ in the phase space. However, dealing with real fluids means that at atmospheric temperature and pressure, a mole ($\sim 10^{23}$)of particles is contained in just several litres of gas. It becomes clear that tracking individual particles is not feasible. In such situation, the tools and principles of statistical mechanics can be used, that is the macroscopic properties which are important for the fluid analysis can be obtained from the average behaviour of a large number of copies of single particle. Thus, the fluid is represented by the single particle distribution function - $f^{(1)}(\mathbf{x}, \mathbf{p}, t)$, which gives the probability of finding a certain particle at a given position with a given momentum. Using the single particle distribution function, the interaction between particles could be modelled. This model is the Boltzmann Transport Equation (BTE):

$$\mathbf{v} \cdot \nabla_x f^{(1)} + \mathbf{F} \cdot \nabla_p f^{(1)} + \frac{\partial f^{(1)}}{\partial t} = \Omega \tag{1}$$

In (1) $\mathbf{v}$ is the fluid velocity; $\cdot\nabla_x f^{(1)}$ is the spatial gradient of the single particle distribution function;$\mathbf{F}$ is the forcefield acting on the fluid particles $\cdot\nabla_p f^{(1)}$ is the momentum gradient of the single particle distribution function; $\partial f^{(1)}/\partial t$ is the time change in $f^{(1)}$ and Ω is the collision integral. The BTE is well documented and studied equation and will not be discussed here. A special discretization of the BTE is the Bhatnagar-Gross-Krook lattice Boltzmann Equa-

tion (BGK LBE)(Wolf-Gladrow (2000)):

$$f_i^{(1)}(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = f_i^{(1)}(\mathbf{x},t) - \\ -\frac{1}{\tau}[f_i^{(1)}(\mathbf{x},t) - f_i^{(eq)}(\mathbf{x},t)] \quad (2)$$

$$f_i^{(eq)}(x,t) = w_i\rho\left[1+\frac{3\mathbf{c}_i\cdot\mathbf{u}}{c^2}+\frac{(\mathbf{c}_i\cdot\mathbf{u})^2}{2c^4}-\frac{3\mathbf{u}^2}{2c^2}\right] \quad (3)$$

$$\rho = \sum_i f_i \quad (4)$$

and

$$\mathbf{u} = \rho\sum_i f_i\mathbf{c}_i \quad (5)$$

where $f_i^{(eq)}$ is the equilibrium distribution function (given by (3)); τ is the single relaxation time to equilibrium; $\mathbf{c}_i$ are the discrete velocities associated with the lattice and w_i are appropriate lattice weights. Additionally, in (3), ρ and $\mathbf{u}$ are the macroscopic density (4) and velocity (5). In order to simulate immiscible liquids the single component multiphase or the multi component multiphase (SCMP; MCMP) LB models could be used. The SCMP (Shan and Chen (1993)) which models the two liquids as the two phases (liquid and vapour) of the same chemical component is used in this work. The idea behind SCMP is that some interaction potential among particles is introduced (6), which governs the separation between vapour (light phase) and liquid (heavy phase).

$$\mathbf{F}(\mathbf{x},t) = -G\psi(\mathbf{x},t)\sum_i w_i\psi(\mathbf{x}+\mathbf{c}_i,t)\mathbf{c}_i \quad (6)$$

where:

$$\psi(\rho) = \rho_0\left[1-exp\left(\frac{-\rho}{\rho_0}\right)\right]$$

furthermore ρ_0 is the initial density - commonly taken as 1 and G is the interaction strength parameter (negative for attraction). The LB models readily allow for a full specification of complex geometries in the flow field, which is why they are particularly suitable for simulating flow through filters. The mesh structures can be artificially created or microscope imaging can be used.

3 SUBSET SIMULATION

One very important problem in engineering is the estimation of the probability of failure, p_F of a system. In the context of numerical simulations failure can be defined as the scenario where a response variable (output) of the model, exceeds some threshold of acceptable system behaviour. The output, y is related to the input variables, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, via some mapping provided by the model,

$$y = \eta(\mathbf{x}) \quad (7)$$

thus the failure domain is defined as the values of $\mathbf{x}$ which cause the system response, y to exceed some critical value y^*

$$F = \{\mathbf{x} : \eta(\mathbf{x}) > y^*\} \quad (8)$$

Estimating p_F is associated with sampling from F. Additionally the complementary CDF (CCDF) (Au and Wang (2014)) of the output distribution of the model can also be constructed, to present a fuller picture of the failure regime. Usually, for a well designed system the true value of p_F is very small, that is, it is a *rare event*. Also, a typical model has a high dimensional input space and often the failure domain of that space is disjoint and sampling from it poses a significant challenge. Subset simulation (Au and Beck (2001)) aims to divide the rare event into a series of nested less-rare events, (9).

$$F \subset F_m \subset F_{m-1} \subset \ldots \subset F_1 \quad (9)$$

In (9) F_1 is a relatively frequent event. Given that sequence, it can be shown that the probability of the rare event F could be expressed as a product of larger probabilities:

$$\mathbb{P}(F) = \mathbb{P}(F_1)\cdot\mathbb{P}(F_1|F_2)\cdot\ldots\cdot\mathbb{P}(F|F_m) = p_F \quad (10)$$

In (10) $\mathbb{P}(\cdot|\cdot)$ denotes the conditional probability. Beginning from the unconditional level F_1, the algorithm "probes" the input space $\mathcal{X}$ via direct Monte Carlo sampling. Then, based on the values of y in (7) it constructs the first intermediate failure threshold, $y_1^* < y^*$, defining a "relaxed failure domain", F_1. SuS then populates F_1 using a MCMC algorithm (Metropolis, Modified Metropolis, Metropolis-Hastings). The generation of intermediate levels continues until a predefined number of samples lie in the true failure domain F. At the end of the algorithm an estimate of the CCDF of the response function (simulator) is generated. The summarised procedure for subset simulation as presented by (Zuev (2015)) is given in Algorithm 1. The modified Metropolis algorithm which generates a new sample $\tilde{x}$ from a proposal distribution $q(\cdot|x)$, where $x \sim \pi(\cdot|F_l)$ and $\pi(\cdot|F_l)$ is the input PDF at the current level F_l, is given in Algorithm 2. In Algorithm 1 $p \in (0,1)$ is the conditional probability of each level; N is the total number of samples; $Nc = p_1 N$ is the number of Markov chains in each level; $Ns = p_1^{-1}$ is the number of samples per chain. There, the values of p and N are appropriately chosen by the user. In Algorithm 2 ϕ is the normal distribution PDF and d is the number of dimensions of $\mathcal{X}$.

Algorithm 1 Subset simulation

1: Generate samples $\{\mathbf{X}_k^{(1)} : k = 1, \ldots, N\}$ from the input PDF, π
2: **for** k in $\{\mathbf{X}_k^{(1)}\}$ **do**
3: $y_k^{(1)}$=$\eta(\mathbf{X}_k^{(1)})$
4: **end for**
5: Set $n_F(1) = \sum_{k=1}^{N} \mathbb{I}_F(y_k^{(1)} > y^*)$
6: **while** $n_F^{(l)} < Nc$ **do**
7: Sort y in descending order
8: Select the first Nc samples as "seeds" for populating F_{l+1}
9: Set $y_l^* = \frac{y_{Nc}^{(l)} + y_{Nc+1}^{(l)}}{2}$
10: Populate F_{l+1} using **Algorithm 2**
11: Calculate $n_F^{(l)}$
12: Set $l = l + 1$
13: **end while**
14: Calculate $\hat{p}_F^{SuS} = p^l \frac{n_F^{(l)}}{N}$

Algorithm 2 Modified Metropolis algorithm

1: **for** i in $1 \ldots d$ **do**
2: Sample $x_i' \sim q_i(\cdot|x_i)$
3: Take $\xi_k = x_i'$ with probability $\min\{1, \frac{\phi(x_i')}{\phi(x_i)}\}$
4: **end for**
5: **if** $\xi \in F_1$ **then**
6: $\tilde{x} = \xi$
7: **else**
8: $\tilde{x} = x$
9: **end if**

4 GAUSSIAN PROCESS EMULATION

Simulators used to model complex scientific phenomena are usually very computationally expensive. This is to say that a single evaluation of the code's output at a given set of input values takes sufficiently long time, as to prohibit any type of analysis which requires a large number of model runs. The LBM code is no exception. Depending on the resolution of the flow field a single run of the SCMP LBM used for this work takes between 60 seconds and just short of 8 minutes. The code was executed on a dedicated server computer with 20 Intel® Xeon® E5-2670 CPUs. Even though SuS requires fewer samples than DMC to estimate the failure probability, it still relies on such amounts of simulation runs which result in very large computational times. Clearly, the analysis cannot be carried out using the code directly. In such cases it is common to use a less expensive approximation of the code output. These approximations are widely known as *metamodels* or *emulators*. There is a number of existing metamodelling techniques, but for the purposes of this study, GPE is used. Formally, the model structure is expressed as:

$$\eta(\mathbf{x}) = h(\mathbf{x})^T \boldsymbol{\beta} + Z(\mathbf{x}) \tag{11}$$

where $\eta(\mathbf{x})$ is the simulator output as a function of its inputs, $h(\mathbf{x})^T$ is a known function of the inputs, $\boldsymbol{\beta}$ is a vector of unknown coefficients and $Z(\mathbf{x})$ is a Gaussian process with zero mean and covariance - $\sigma^2 c(\mathbf{x}, \mathbf{x}'; \psi)$. The function $h(\mathbf{x})$ should express any expert opinion about the form of the simulator output and together with the parameter $\boldsymbol{\beta}$ reflects its overall trend. In practice, however, the trend is often taken to be constant as $h(\mathbf{x}) = 1$, charging the Gaussian process in (11) with the responsibility of capturing the behaviour of the underlying function. In the formulation above, σ^2 is a scale parameter and ψ is a parameter specifying the behaviour of the correlation function.

Using Gaussian process emulation, a *posterior* probability distribution for the mean of the computer code's output can be constructed, conditional on a relatively small number of simulator runs, $\mathbf{y}$ and the parameters, $\boldsymbol{\theta} = \{\beta, \sigma^2, \psi\}$. It can be shown that this distribution has the form:

$$\eta(\mathbf{x}^*)|\mathbf{y}, \hat{\boldsymbol{\theta}}, \sim \mathcal{N}(m(\cdot), C(\cdot, \cdot)) \tag{12}$$

with posterior predictive mean (also called a*surrogate*):

$$m(\mathbf{x}^*) = \hat{\beta} + \mathbf{t}(\mathbf{x}^*)^T \boldsymbol{C}^{-1}(\mathbf{y} - \mathbf{1}\hat{\beta}) \tag{13}$$

and posterior predictive variance:

$$C(\mathbf{x}^*, \mathbf{x}'^*) = \hat{\sigma}^2(c(\mathbf{x}^*, \mathbf{x}^*) - \boldsymbol{t}(\mathbf{x}^*)^T \boldsymbol{C}^{-1} \boldsymbol{t}(\mathbf{x}'^*)) \tag{14}$$

In (13) and (14), $\mathbf{C} \in \mathbb{R}^{n \times n}$ is such that $\mathbf{C}_{ij} = c(\mathbf{x}_i, \mathbf{x}_j)$; $\mathbf{t}(\mathbf{x}^*) \in \mathbb{R}^n$ such that $\mathbf{t}(\mathbf{x}^*) = (c(\mathbf{x}^*, \mathbf{x}_1), \ldots, c(\mathbf{x}^*, \mathbf{x}_n))^T$; $c(\mathbf{x}_i, \mathbf{x}_j)$ is called the correlation function; and $\mathbf{1} \in \mathbb{R}^n$ such that $\mathbf{1} = (1, \ldots, 1)^T$.The process of estimating $\boldsymbol{\theta}$ from observed data is referred to as *training* and is well described in (Forrester et al. (2008)) from a classical prospective or in Oakley (1999); Becker (2011) from Bayesian standpoint. Once the emulator is trained, its posterior distribution can be sampled many times at an affordable cost to provide data for various analyses.

5 NUMERICAL EXPERIMENTS

In order to better understand the complex nature of the filter environment, a simplified version of the real mesh was simulated and tested initially. As seen in the panels of Figure 5, the simulation represents a single water droplet impacting onto a fibre placed in the diesel medium. This particular fibre has low surface energy (hydrophobic) represented by the wettability coefficient, $\mu = 0.08$. In panels $k)$ and $l)$ of Figure 5 some of the water volume can be seen to "escape" from the fibre and go to the outlet of the domain, which represents the downstream portion of the fuel line. The escape volume denoted with V_e is of interest in this simulation since its presence signifies that

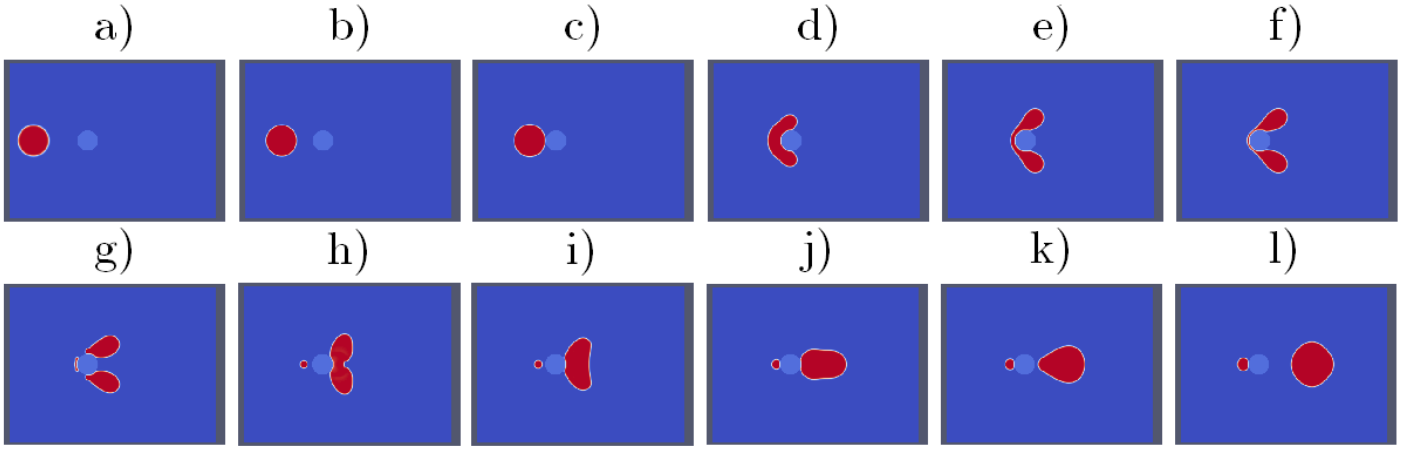

Figure 1: Sample from a run of SCMP LBM - a droplet of radius 30 lu (lattice units) impacts on a hydrophobic fibre of radius 20 lu. Density ratio between the two phases is 1:35.

the filter has failed to accomplish its task. The model as shown previously depends on three input variables, namely *droplet diameter*, *fibre diameter* and *fibre surface energy* denoted with d_d, d_f, μ, respectively. That is the relationship between inputs and outputs is given by $V_e = \eta(d_d, d_f, \mu)$. As seen in Figure 5 and in particular panels $e)$ to $h)$ the droplet exhibits highly non-trivial deformation upon impacting the fibre, which determines its coalescence and separation dynamics. The simulation domain pictured here is 400 lattice units long and 300 lattice units high. The average simulation time was found to be 62 seconds. In order to perform a reliability analysis on the environment represented by the code, the GPE had to be used to construct a more computationally efficient approximation to the code. For the purpose of training the emulator, the code was ran at 50 points selected via *Latin hypercube sampling* (McKay et al. (1979)). It is commonly accepted that $10d$ samples are sufficient for the purpose, where d is the number of dimensions of $\mathcal{X}$ (Loeppky et al. (2008)). Another 50 data points were sampled for validating the meta model. Figure 5 shows the predictions of the GPE from the 50 validation runs, plotted against the observations from the LBM simulator. Most of the data points lie close to the 45^o line which indicates perfect correspondence between prediction and observation. There are a few outliers which represent possible local problems with the emulator and could be rectified by obtaining more samples in that vicinity. One of the main diagnostics of a GP emulator is the analysis of individual prediction errors (IPE) (Bastos and OHagan (2009)) given in (15). Each IPE gives the error between the simulator output, y_i and the mean of the emulator (13), normalised by the variance of the emulator's posterior distribution - the diagonal of C in (14). For a well working emulator the IPE have a Student-t distribution and therefore, 95% of them should lie in the interval $[-2, 2]$. Figure 5 depicts the distribution of errors for the emulator used on the LBM code. Again, most of the points lie in the desired region and are evenly distributed in it.

$$D_i^I = \frac{y_i - \mathbb{E}[\eta(\mathbf{x_i}^*)|\mathbf{y}]}{\sqrt{\mathbb{V}[\eta(\mathbf{x_i}^*)|\mathbf{y}]}} \tag{15}$$

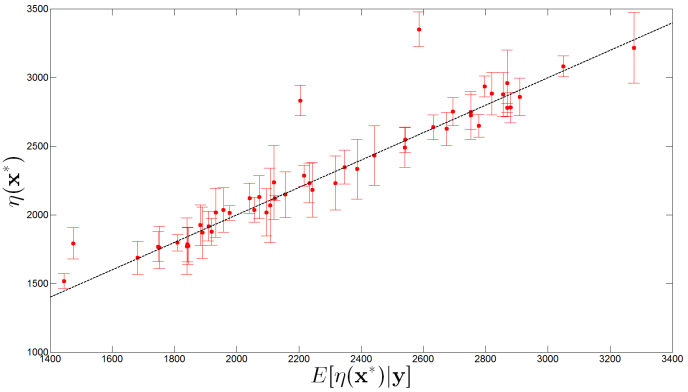

Figure 2: Predictions against observed values of escape volume of water. Error bars represent 95% credible interval.

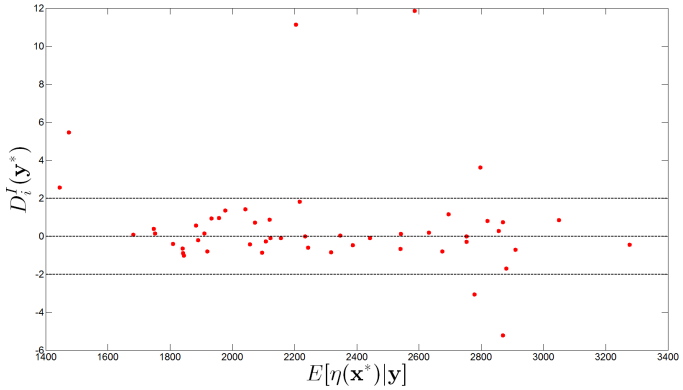

Figure 3: Predictions against individual prediction errors.

Overall, it is accepted that the emulator represents the simulator appropriately and can be used in its place. Most importantly a single prediction from the emulator is instantaneous and thus all desired statistical analysis can be performed. Subset simulation can also be run with the emulator. A failure level was set at $V_e \geq 3270$. Following (Au and Wang (2014)) the

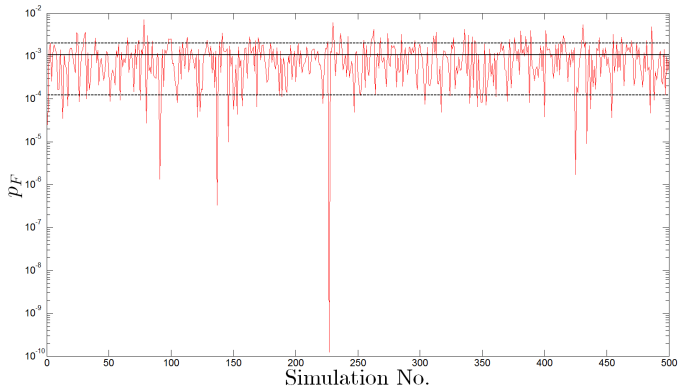

Figure 4: Failure probability estimated from 500 runs of SuS. The mean $p_F \approx 0.00107$ is plotted with black solid line. The two dashed lines report $\pm$ one standard deviation.

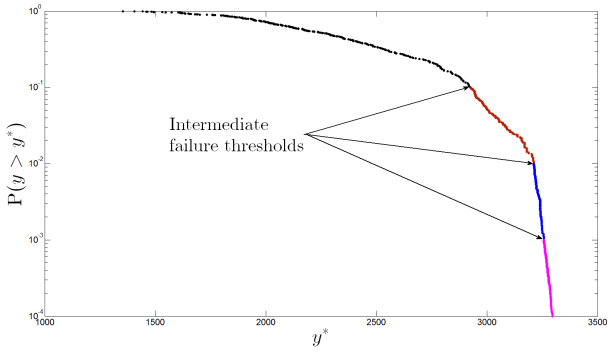

Figure 5: Complementary CDF estimate of escape volume. The colours indicate different conditional failure levels.

level probability was chosen as $p = 0.1$. The proposal PDF, $q(\cdot|\cdot)$ (Algorithm 2) was chosen to be the standard normal distribution - a popular choice for the method. Each level was populated with 300 samples from q. Figure 5 shows the estimation of the CCDF via SuS with the aforementioned characteristics. Another 500 SuS runs were performed, which resulted in a mean probability of failure, $p_F = 0.00107$. The estimations of p_F are shown in Figure 5. The mean CCDF, together with the first and third quartiles are shown in Figure 5. It can be seen that the estimation deviate very little from the mean value. Each simulation had between 3 and 4 conditional levels, resulting in a total of either 900 or 1200 samples. A reliable estimation of probability of the order given above, would require approximately 100000 samples using direct Monte Carlo methods.

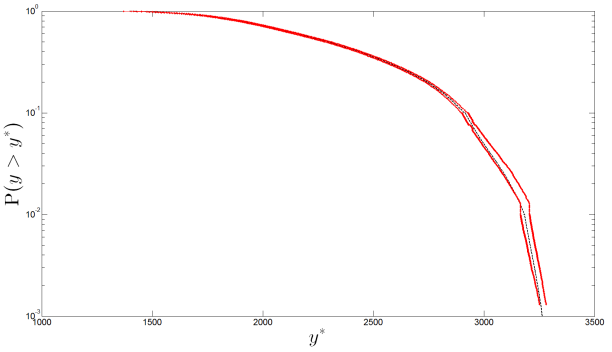

Figure 6: Complementary CDF estimates of escape volume for 500 runs of SuS. The black dashed line shows the mean estimate. The red dotted lines represent the 25$^{\text{th}}$ and the 75$^{\text{th}}$ percentiles.

6 DISCUSSION

In this paper, a combination between Gaussian process emulation and Subset simulation was presented for the estimation of a rare event in Lattice Boltzmann simulation of a filter. The efficiency gains in computational time due to the use of the emulator instead of the LBM code are significant. The reduction in number of samples required by SuS as compared to DMC is appreciable. Overall, the performance of the tools is promising and can be used to a satisfactory result. In the future the project will focus on modelling the filter with increasingly higher accuracy. This will introduce more dimensions to the input space and possibly create more complex failure domains. Other causes of filter failure will also be addressed.

REFERENCES

Au, S. K. and Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277.

Au, S. K. and Wang, Y. (2014). *Engineering Risk Assessment with Subset Simulation*. Wiley.

Bastos, L. S. and OHagan, A. (2009). Diagnostics for Gaussian Process Emulators. *Technometrics*, 51(4):425–438.

Becker, W. (2011). *Uncertainty Propagation through Dependability Models*. PhD thesis.

Dwivedi, G., Jain, S., and Sharma, M. P. (2011). Impact analysis of biodiesel on engine performance - A review. *Renewable and Sustainable Energy Reviews*, 15(9):4633–4641.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley.

Fregolente, P. B. L., Fregolente, L. V., and Maciel, M. R. W. (2014). Water Content in Biodiesel, Diesel, and BiodieselDiesel Blends. *Igarss 2014*, (1):1–5.

Loeppky, J. L., Sacks, J., Welch, W., and Welch, W. J. (2008). Choosing the Sample Size of. (170).

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Oakley, J. (1999). Bayesian uncertainty analysis for complex computer codes. *Thesis*.

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, 91(10-11):1290–1300.

Schweitzer, P. A. (1997). *Handbook of Separation Techniques for Chemical Engineers*. Chemical engineering books. McGraw-Hill.

Shafiee, S. and Topal, E. (2009). When will fossil fuel reserves be diminished. *Energy Policy*, 37:181–189.

Shan, X. and Chen, H. (1993). Lattice Boltzmann model for simulating flows with multi phases and components. *Physical Review E*, 47(3):1815–1819.

Sukop, M. C. and Thorne, D. T. (2006). *Lattice Boltzmann Modeling*, volume 79.

Wolf-Gladrow, D. a. (2000). Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction. *PoLAR*, page 308.

Yang, C., Larsen, S., and Wagner, S. (2007). Understanding Emulsified Water Filtration from Diesel Fuels. *8th International Filtration Conference*, pages 1–5.

Zuev, K. M. (2015). Understanding the Subset Simulation Method for Rare Event Estimation. 1(2):1–25.