Adaptive Gaussian Particle Method for the Solution of the Fokker-Planck Equation

Moriz Dirk Scharpenberg[1,*] **and Maria Lukáčová-Medviďová**[2,**]

[1] Technische Universität Hamburg-Harburg,Institut für Numerische Simulation,Schwarzenbergstraße 95 E, 21073 Hamburg

[2] Johannes Gutenberg - Universität, Institut für Mathematik, Arbeitsgruppe Numerische Mathematik, Staudingerweg 9, 55099 Mainz

Received This is a pre-review version, see the Journal website for the final version, accepted

Key words uncertainty quantification, dynamic systems, particle methods, Fokker-Planck equation

MSC (2000) 00-xx

The Fokker-Planck equation describes the evolution of the probability density for a stochastic ordinary differential equation (SODE) or a deterministic ordinary differential equation (ODE) with stochastic initial values. A solution strategy for this partial differential equation (PDE) up to a relatively large number of dimensions is based on particle methods using Gaussians as basis functions. An initial probability density is decomposed into a sum of multivariate normal distributions and these are propagated according to the ODE. The decomposition as well as the propagation is subject to possibly large numeric errors due to the difficulty to control the spatial residual over the whole domain.
In this paper a new particle method is derived, which allows a deterministic error control for the resulting probability density. It is based on global optimization and allows an adaption of an efficient surrogate model for the residual estimation.

1 Error Controlled Uncertainty Analysis

For any numerical application knowledge about the accuracy of the applied method is essential. If we simulate a physical system, further knowledge about the effect of uncertainties in the input parameters is vital for the assessment and validation of the results. In [7] it is noted that uncertain input data does not only lead to an uncertain solution but to an uncertain solution error as well.
In the following both aspects will be covered for stochastic uncertainties within a system of ordinary differential equations (ODE)

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, t). \tag{1}$$

In this case every component of $\mathbf{x}$ is assumed to be subjected to stochastic uncertainties, i.e. the initial values follow a stochastic distribution. The ODE itself is considered deterministic. As shown in [13] this formulation also covers an ODE with stochastic parameters, which are treated as independent and constant additional states.

1.1 Stochastic Modelling

The most general way to describe the uncertain state vector $\mathbf{x}$ is to impose a stochastic distribution on them. Then any fixed combination of the states is mapped to a multivariate probability density function $\rho(\mathbf{x})$. Within ρ correlations between different components of $\mathbf{x}$ can be described.
If an initial distribution is defined, the Fokker-Planck equation [23] describes its time evolution of the density distribution. Its diffusion term describing white noise in the ODE is omitted here, since 1 is considered deterministic. Thus the Fokker-Planck equation

$$0 = \partial_t \rho(\mathbf{z}, t) + \sum_{i=1}^{D} \frac{\partial}{\partial z_i}\left[f_i(t, \mathbf{z})\rho(\mathbf{z}, t)\right] \tag{2}$$

* Corresponding author E-mail: ms@mds-line.de, Phone: +49 40 42878 3876, Fax: +49 40 42878 2696
** Second author E-mail:lukacova@mathematik.uni-mainz.de, Phone: +49 6131 39 22831, Fax: +49 6131 39 23331

describes the time evolution of the probability density with respect to the problem (1). It reformulates an ODE with stochastic initial values into a partial differential equation (PDE) with as many partial derivatives as there are states in the ODE.

1.2 Solution Strategies

As a solution for stochastic problems the Monte Carlo method (cf. [28], [24]) provides an obvious strategy. The non-intrusiveness makes the application rather straight forward. The proof of convergence is based on the central limit theorem (cf. [6]) and only guarantees stochastic convergence to zero for the deviation of the estimated probability from the accurate probability Δp for an event. Its order of convergence of $\Delta p \sim 1/\sqrt{n}$ is frequently sufficient for events with a large probability but not suitable for evaluating a probability density, where high accuracy is desired.
Finite volume schemes, e.g. described in [10] and [11], provide a solution for the PDE (2) and provide deterministic error bounds. Since equation (1) already determines the characteristic directions, an update step can be performed efficiently. Still the number of cells and with it memory consumption and computation time grows exponentially with the number of dimensions D of the problem. In practice this "curse of dimensionality", a term coined in [1], limits the application of finite volume methods to equation (2) to very small problems as a matter of principle.
Another approach to solve (2) stems from the class of particle methods, cf. [22], [12]. Applied to stochastic problems the algorithm was described in [8] proposing normal distributions as basis functions. Related are multiple spawning methods, cf. [2], which also use Gaussian functions to decompose a probability density. These methods were historically used to perform quantum mechanical calculations, e.g. to find the solution of the time dependent Schrödinger equation.
In [27] it was noted that a previous implementation in [8] made two related assumptions, which cannot always be fulfilled. Firstly Gaussian basis functions are not necessarily independent and secondly the variance of a Gaussian is not necessarily small compared to the length over which it potentially changes. The propagation of the Gaussian basis functions by minimizing methods, which avoid these restrictions, were introduced in [27].
In [15] and [16] a mechanism was proposed to increase and decrease the number of basis functions according to a predetermined tolerance and hence to adapt the decomposition to nonlinearities in the ODE. After applying this methodology to problems of molecular dynamics, in [29] it was also used to describe stochastic uncertainties in macroscopic fields.
The method relies on a stochastic estimation of the current global residual by an adapted Monte Carlo sampling. In [25] it was noted that since both the adaption and the error control rely on the same sampling procedure, relatively large residuals may stay undetected.
The goal of this paper is to describe a new particle method which cures these drawbacks by separating the error control and the adaption of the Gaussian basis function. Applying global optimization allows to obtain deterministic error predictions. We will propose an efficient estimator for the residual in the minimization step which is adapted according to the global information. As a consequence we will be able to control the error in a deterministic way.

2 Gaussian Based Particle Method with Deterministic Error Control

To represent the probability density at any point of time by mesh independent particles, the density is decomposed into a number N of multivariate Gaussian basis functions described by parameters $\mathbf{p}$, i.e.

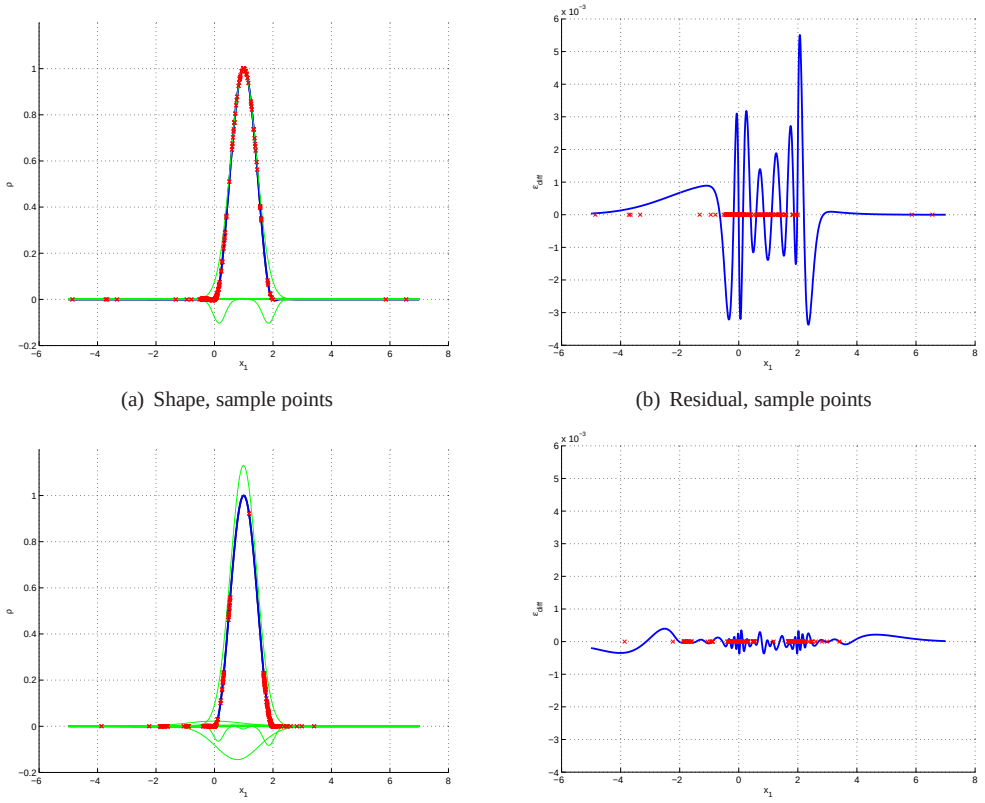
$$\rho_{\mathbf{p}}(\mathbf{x}) = \sum_{i=1}^{N} y_i e^{(\mathbf{x}-\mathbf{x}_i)^T G_i (\mathbf{x}-\mathbf{x}_i)}. \tag{3}$$

The shape matrices G_i may require a predefined structure to the single Gaussian basis function. The most simple form would be an identity matrix multiplied by a single standard deviation λ_i, as proposed in [16]. More parameters may be introduced into the shape matrices, e.g. proposed in [14], as long as the G_i stay positive definite. The longer the parameter vector becomes, the longer the optimization process will take. But if the overall density can be described by fewer basis functions, this may still speed up the overall process.
Typically different states of the problem have uncertainties of a different order of magnitude. Under this assumption individual inverted variances are necessary for each dimension of the problem. This leads to

$$G_i = \begin{pmatrix} \lambda_{i1} & & \\ & \ddots & \\ & & \lambda_{iD} \end{pmatrix}. \tag{4}$$

as definition for the shape matrix in use. Then the parameter vector becomes

$$\mathbf{p} = [y_1, \ldots, y_N, \mathbf{x}_1, \ldots, \mathbf{x}_N, \lambda_{11}, \ldots, \lambda_{1D}, \lambda_{21}, \ldots, \lambda_{2D}, \ldots, \lambda_{N1}, \ldots, \lambda_{ND}] \tag{5}$$



(a) Shape, sample points (b) Residual, sample points

(c) Shape, combined approach (d) Residual, combined approach

Fig. 1 A smooth cosine decomposed using sample points compared to a decomposition using a combined approach. On the left the shape of the cosine (black) is shown along with the decomposition (blue) and the shapes of the approximating basis functions (green). On the right the local residual (blue) is shown. The sample points are given as crosses.

For the initialization phase as well as for the time evolution the adaption of the Gaussian particles is based on minimization of a residual ϵ

$$\mathbf{p} : \min_{\mathbf{p}} \epsilon(\mathbf{p}), \tag{6}$$

where $\epsilon(\mathbf{p})$ describes the deviation from a density or the spatial error for a time step. The precise formulation depends on the corresponding step of the algorithm and will be given in what follows.

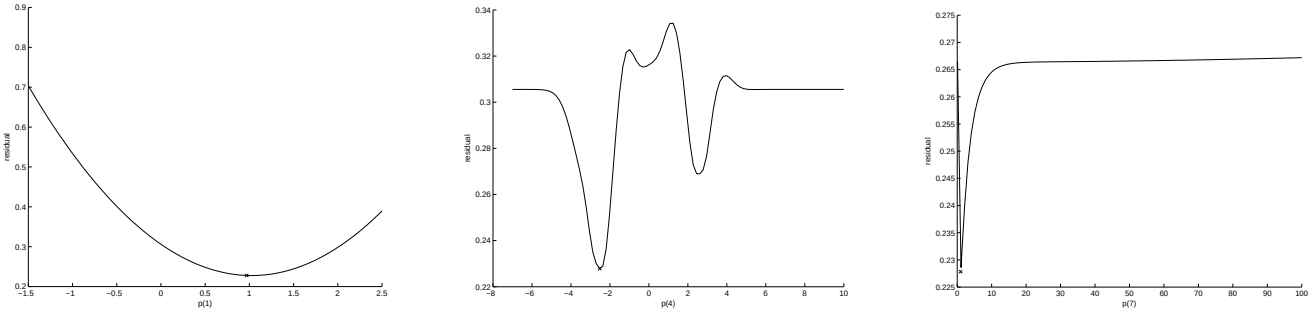
2.1 A Fully Error Controlled Initialization Process

To decompose the initial density ρ_0 into a sum of Gaussians (3), the residual ϵ_{ini} describes the deviation of (3) from the desired density distribution ρ_0

$$\epsilon_{\text{ini}}(\mathbf{p}) = \|\rho_{\text{approx}}(\mathbf{x}, \mathbf{p}) - \rho_0(\mathbf{x})\|. \tag{7}$$

The aim is to minimize it until a prescribed tolerance $\hat{\epsilon}_{\text{spatial}}$ is met by choosing a suitable parameter vector $\mathbf{p}$. New Gaussians are added if no combination of the components of $\mathbf{p}$ fulfils $\epsilon_{\text{ini}} < \hat{\epsilon}_{\text{spatial}}$. For this optimization process a Gauß-Newton method is proposed in [16] or a Nelder-Mead scheme (cf. [20]) in [14]. In our experiments we typically use a Nelder-Mead scheme. Even though this heuristic method does not guarantee a global minimum, it frequently recovers from local minima and turns out to be suitable for finding good parameter combinations $\mathbf{p}$.
The difficulty is to describe the norm for the whole domain appropriately. One approach is to consider the 2-norm, as described in [15], and to approximate it with a number of sample points $\mathbf{x}_i, i = [1, ..., N]$ following a water level distribution such that

$$\epsilon_{\text{ini,sampled}}(\mathbf{p}) = \sqrt{\sum_{i=1}^{N} \left(|\rho_{\text{approx}}(\mathbf{x}_i, \mathbf{p}) - \rho_0(\mathbf{x}_i)|\right)^2} \tag{8}$$



(a) Influence of the amplitude of the first Gaussian only

(b) Influence of the position of the first Gaussian only

(c) Influence of the inverted variance of the first Gaussian only

Fig. 2 Typical course of the residual estimated by a number of sample points due to a varied single dimension in the influence vector $\mathbf{p}$. The residual due to amplitude and inverted variance usually has a very distinct minimum, while the residual due to the center position of a Gaussian x_1 has a more complicated topology. Locally the effects are rather decoupled.

where $\rho_{\text{approx}}(\mathbf{x}_i, \mathbf{p}) = \rho_{\mathbf{p}}(\mathbf{x}_i)$. It gives a continuously differentiable topology for $\epsilon_{\text{ini}}(\mathbf{p})$, such that the optimization $\min_{\mathbf{p}} \epsilon_{\text{ini,sampled}}$ can be solved efficiently. However, if the initial sampling does not cover small regions with a large local residual, this will not be reflected in the optimization. Typically local residuals will become large in areas, where they are not measured but small at the sample points. Hence the overall residual will be much larger than the predicted one.
With a new Gaussian at the sample point with the largest local residual, additional sample points in its vicinity are proposed in [16]. This, however, leads to additional Gaussians and sample points in regions, where the measured residual is largest, leaving even larger residuals in other regions undetected.
Another drawback of using sample points chosen following a water level distribution is that to represent the domain well, i.e. to have at least a number n of sample points within a given radius, the number N of sample points is growing exponentially with respect to the number of dimensions D, as derived in [3]. Hence the calculation time for a method using evenly distributed sample points grows with the same order as for a grid based method.
A way out provides use of global optimization. An example of the deterministic algorithms is the multivariate method of divided rectangles (DIRECT, cf. [17], [5]) that belongs to the group of Lipshitzian optimization method (cf. [26]). As a non-deterministic but very reliable and highly efficient algorithm in [30] particle swarm optimization (PSO, cf. [18], [19], [21], [4]) was proposed. It can be observed that the PSO deals well even with difficult topologies and leads to a high detection rate, even if multiple distinct local minima are present. Then the residual is directly computed as

$$\epsilon_{\text{ini, max}}(\mathbf{p}) = \max_{\mathbf{x}} |\rho_{\text{approx}}(\mathbf{x}, \mathbf{p}) - \rho_0(\mathbf{x})|. \tag{9}$$

There are two drawbacks of this definition. Firstly the resulting topology may not be continuously differentiable any more and secondly there is still the possibility of the optimization algorithm to fail which may lead to distinct local minima. In addition global optimization is too expensive to be used within an optimization loop.
But the advantages of both ways can be combined. Global optimization is used to identify the location $\mathbf{x}_{\text{max}}$ of the maximum residual $\epsilon_{\text{ini, max}}$. Then sample points are added in the vicinity $\mathbf{x}_{\text{max}}$ and a new Gaussian is fitted in. The standard deviation σ_{estimate} of the new Gaussian as well as the radius in which to add new sample points is determined by searching the minima $\mathbf{x}_{j,\text{min}}$ surrounding $\mathbf{x}_{\text{max}}$ using a steepest decent method starting from points close to $\mathbf{x}_{\text{max}}$. For the search the residual

$$\epsilon_{\text{ini,diff}}(\mathbf{x}) = \begin{cases} \rho_{\text{approx}}(\mathbf{x}, \mathbf{p}) - \rho_0(\mathbf{x}) & \text{if } \rho_{\text{approx}}(\mathbf{x}_{\text{max}}, \mathbf{p}) - \rho_0(\mathbf{x}_{\text{max}}) > 0 \\ \rho_0(\mathbf{x}) - \rho_{\text{approx}}(\mathbf{x}, \mathbf{p}) & \text{if } \rho_{\text{approx}}(\mathbf{x}_{\text{max}}, \mathbf{p}) - \rho_0(\mathbf{x}_{\text{max}}) \leq 0 \end{cases} \tag{10}$$

is considered. Then

$$\sigma_{\text{estimate}} = \sum_{j=1}^{N_{\text{vicinity}}} \frac{1}{N_{\text{vicinity}}} \|\mathbf{x}_{\text{max}} - \mathbf{x}_{j,\text{min}}\|_2 \tag{11}$$

usually gives a good estimate of the standard deviation. It is used to calculate an initial value for all diagonals in 4. It should be noted that in several cases this may lead to a strong underestimation of the variance for some states. Experience shows that this does not harm the optimization step afterwards.

In addition sample points $\mathbf{x}_i$ are chosen according to a probability

$$P_{i,\mathrm{rm}} = 1 - \frac{\frac{\epsilon(\mathbf{x}_i)}{\epsilon_\infty} + \sigma_{\text{glob}}}{1 + 2\sigma_{\text{glob}}} \tag{12}$$

to be added to a pool of sample points, from which a given number are removed after each step. The probablility $P_{i,\mathrm{rm}}$ depends on the local residuals $\epsilon(\mathbf{x}_i)$ relative to the maximum residual $\epsilon_\infty = \max_i \epsilon(\mathbf{x}_i)$. This process reminds of the evaporation of sample points. The evaporation constant $0 \leq \sigma_{\text{glob}} \leq \infty$ defines the strength of the evaporation process and its dependency on the local residuals.
Figure 1 shows the improvement of this combined approach for the adaption of a sum of up to 50 Gaussians to a cosine shaped target density distribution. Since the algorithm adapts the sample points to areas with large residuals, they mostly cluster at the sides where the topological differences to a Gaussian shape is largest. The overall residual achievable with 50 Gaussians is lowered by an order.

2.2 Single Directional Optimization

The gradient of $\epsilon_{\text{ini,sampled}}$ with respect to $\mathbf{p}$ shows some remarkable structure, depending on the the point of linearization. Figure 2 shows the influence of different components in $\mathbf{p}$ on the residual. In general the influence of the amplitude and the variance show one single and distinct minimum of the residual, even though with entirely different shape. The influence of the location reflects the shape of the local residual with respect to the position $\mathbf{x}$. For a gradient based optimization these differences may lead to a large condition number of the Jacobian.
In general it turns out that the components of $\mathbf{p}$ behave locally independent. Examples are given in Figure 3. This means that the components of $\mathbf{p}$ can be optimized independently, which has another advantage: If the situation becomes unsuitable for optimization, it can be treated independently for every component in $\mathbf{p}$. In doubt a single component is not optimized at all. For example an optimization of the variance or the location does not make sense, if the amplitude is close to zero. An adaption of the position is in general not meaningful, if the diagonals λ_{ij} in the shape matrix G_i become close to zero. These observations lead to

Algorithm 1 *Single Directional Optimization (SDO)*

1. *Treat every Gaussian $i = [1, ..., N]$ separately and in each step chose one. In case a new Gaussian was added with only roughly estimated initial values, it is optimized first.*

2. *For each Gaussian i chose first the amplitude, then the inverted variances, then rotational angles or other properties if present and then the positions.*

3. *Minimize the residual with respect to each of them in that order and modify the overall parameter vector*

$$\mathbf{p}(i)_{new} = \mathbf{p}(i) : \min_{\mathbf{p}(i)} \left(\epsilon_{ini,sampled}(\mathbf{p})\right) \tag{13}$$

   with the obtained result. If the result is not within expected limits, do not take the result of the optimization into account.

4. *Restart from the beginning at item 2. If the influence vector does not change significantly any more, go back to step 1 and chose the next Gaussian. If the last Gaussian is reached, restart with the first one for another loop. If $\mathbf{p}$ does not change significantly any more, the optimization is finished.*

The algorithm is inspired by human behaviour to keep the overall status quo and to optimize locally only. With many Gaussians interacting no global changes are achieved. Usually it needs less than five loops to reach convergence.

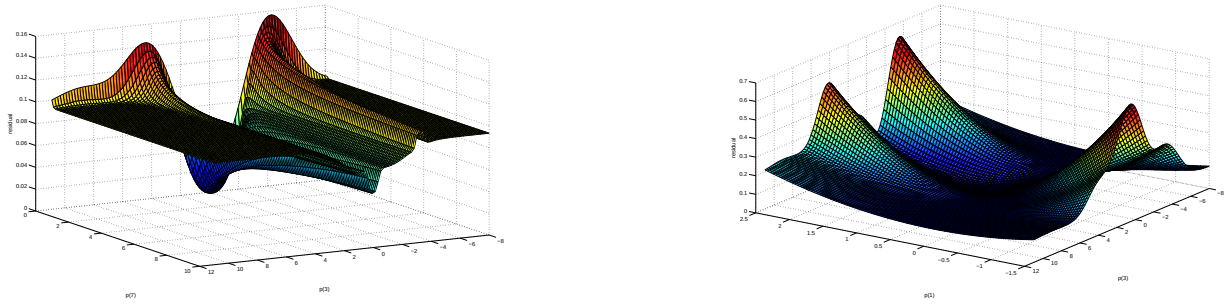
2.3 Initialization Process

It should be noted that it is enough for the algorithm to find a sufficiently good $\mathbf{p}$, such that the density is represented well by a low number of Gaussians. $\mathbf{p}$ does not need to be globally optimal, which usually would be too expensive to achieve. This leads to the following

Algorithm 2 *Initialization process for GBP-DEC*

1. *Use DIRECT or PSO to find the global maximum of the residual $\epsilon^2_{ini,max}(\mathbf{x}_{max}) = \max_{\mathbf{x}}(\rho_0(\mathbf{x}) - \rho_{approx}(\mathbf{x}))^2$.*

2. *If the residual is sufficiently small, e.g. $\epsilon_{ini,max} \leq \hat{\epsilon}_{spatial}$, the decomposition is finished, otherwise proceed.*



(a) residual due to x_1-position and inverted variance

(b) residual due to amplitude and x_1-position

Fig. 3 Combined influence of different positions in the parameter vector $\mathbf{p}$ on the residual. Components of a different type in $\mathbf{p}$ are decoupled over a wide parameter range.

3. *Use the argmax $\mathbf{x}_{max}$ as new center point for an additional Gaussian. The amplitude is $y_i = \rho_0(\mathbf{x}_{max}) - \rho_{approx}(\mathbf{x}_{max})$.*
4. *Remove a number of sample points according to equation (12). Add new sample points in the vicinity of the new Gaussian function.*
5. *Optimize the additional Gaussian and then the other basis functions using single direction optimization, i.e. Algorithm 1, and proceed with step 1.*

2.4 Time Evolution

Elements of Algorithm 2 are directly used for time integration. In [15] the trapezoidal rule

$$\rho(t+\tau) - \frac{\tau}{2}\dot{\rho}(t+\tau) = \rho(t) + \frac{\tau}{2}\dot{\rho}(t) \tag{14}$$

with a time step τ is proposed for integration and with $\dot{\rho}$ calculated numerically from the Fokker-Planck equation (2). It gives the residual

$$\epsilon_{\text{spatial}} = \|\rho(\mathbf{x}, t+\tau) - \frac{\tau}{2}\dot{\rho}(\mathbf{x}, t+\tau) - \rho(\mathbf{x}, t) - \frac{\tau}{2}\dot{\rho}(\mathbf{x}, t)\| \tag{15}$$

for equation (6), which should be smaller than a prescribed spatial tolerance $\hat{\epsilon}_{\text{spatial}}$. According to the initialization process the norm in (15) can be described by an approximated 2-norm

$$\epsilon_{\text{spatial,sampled}} = \sqrt{\sum_{i=1}^{N} \left| \rho(\mathbf{x}_i, t+\tau) - \frac{\tau}{2}\dot{\rho}(\mathbf{x}_i, t+\tau) - \rho(\mathbf{x}_i, t) - \frac{\tau}{2}\dot{\rho}(\mathbf{x}_i, t) \right|^2} \tag{16}$$

or the maximum norm

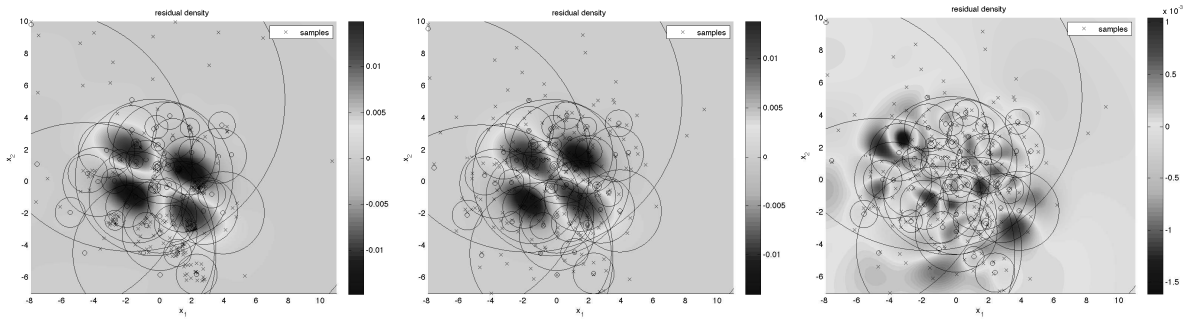
$$\epsilon_{\text{spatial,max}} = \max_{\mathbf{x}} \left| \rho(\mathbf{x}, t+\tau) - \frac{\tau}{2}\dot{\rho}(\mathbf{x}, t+\tau) - \rho(\mathbf{x}, t) - \frac{\tau}{2}\dot{\rho}(\mathbf{x}, t) \right| \tag{17}$$

determined by global optimization. Prior to use (16) for the first time after the initialization, the sample points have to be redistributed, since their location is optimized for the residual with respect to the initial density and not to the residual due to (1).
In [16] a time step adaption is proposed, which uses a comparison step of a different order, e.g.

$$\hat{\rho}(t+\tau) = \rho(t) + \tau\dot{\rho}(t) \tag{18}$$

and leads to

$$\tau_{\text{opt}} = \sqrt{\frac{\sigma\hat{\epsilon}_{\text{t}}}{\tilde{\epsilon}_{\text{t}}}}\tau_{\text{old}} \tag{19}$$



(a) Original situation (b) Linearly propagated (c) Optimization

Fig. 4 Starting the Time Evolution Optimization Process: The sample points from the initialization do not fit to the residual defined by the trapezoidal rule (a). They are redistributed and a linear step is performed (b). Since the degrees of freedom in the shape matrix are limited to its diagonal elements, nonlinearities cannot be captured by the linear analysis. The optimization step (c) improves the residual by an order of magnitude.

for a target residual $\hat{\epsilon}_{\mathrm{t}}$ and a factor of safety $0 < \sigma < 1$. The residual in between the two steps of different order is calculated by a global optimization step

$$\tilde{\epsilon}_{\mathrm{t}} = \sqrt{\max_{\mathbf{x}} \left(\rho(t+\tau) - \hat{\rho}(\mathbf{x}, t+\tau)\right)^2}. \tag{20}$$

If the time step becomes smaller, the spatial accuracy needs to be adapted, i.e.

$$\hat{\epsilon}_{\text{spatial}} = \hat{\epsilon}_{\mathrm{t}} \cdot \tau. \tag{21}$$

Prior to the adaption a linear step

$$y_i(t+\tau) = y_i(t) - \tau y_i(t)\mathrm{trace}(J_f) \tag{22}$$

$$\mathbf{x}_i(t+\tau) = \mathbf{x}_i(t) + \tau \mathbf{f}(\mathbf{x}_i(t), t) \tag{23}$$

$$G_i(t+\tau) = G_i(t) - \tau(J_f G_i(t) + G_i(t) J_f^T). \tag{24}$$

is performed in [15]. It is derived by performing a first order Taylor expansion for equation (1), applying it to (2) and separating terms of the same order. If the density is only represented by very few Gaussians and there is only a limited number of degrees of freedom in the shape matrices, the effect of the linear step on the shape matrices and hence on the residual is usually small. But the adaption of amplitude and center points leads a better starting value for the optimization. It should be noted that while the linear step linearizes equation (1), the trapezoidal rule uses the unmodified equation (1) but approximates the evolution of $\rho(\mathbf{x}, t)$. Thus the drift term of (2) is linearized in time and large movements of the density may lead to a residual. This effect, however, vanishes quickly for small movements as controlled by equation (19).

2.5 The GBP-DEC Algorithm

If the residual cannot be improved by using (16) for optimization, more sample points and eventually more Gaussians are added, cf. [15]. It can be checked, whether the evaluation of (17) can be improved by optimizing (16) with more sample points only. If this is the case, the number of sample points was not sufficient in the previous step. Otherwise (1) shows nonlinearities, which require more Gaussians basis functions. This procedure is called spawning. The number of sample points is reduced automatically in the following time steps by the application of (12).
The suggestion in [27] is used to prune the number of Gaussians if the amplitude is too low and if any two Gaussians with the same variance have almost the same location. Pruning is not essential for convergence as it is for the methodology described in [15], since SDO requires no evaluation of a Jacobian matrix. Still it is useful to eliminate useless Gaussians for performance reasons. Pruning may also be applied for the initialization phase, where dependencies can occur due to insufficient sampling and lead to dependent Gaussians.
The above considerations lead to the new

Algorithm 3 *Gaussian Based Particle method with Deterministic Error Control (GBP-DEC)*

1. *Start the algorithm with a decomposition achieved using Algorithm 2.*
2. *Redistribute sample points.*
3. *Start the algorithm at t = 0.*
4. *Perform a linear step (22), (23), (24) for the parameter vector* $\mathbf{p}$ *to obtain an initial guess* $\mathbf{p}_{new}(t+\tau)$.
5. *Compute a new parameter vector* $\mathbf{p}_{new}(t+\tau)$ *by applying the residual (16) to the optimization (6), i.e.*
$$\mathbf{p}_{new}(t+\tau) : \min_{\mathbf{p}} \epsilon_{spatial,sampled}(\mathbf{p}). \tag{25}$$
6. *Perform a pruning step.*
7. *Compute the residual* $\epsilon_{spatial,max}$ *by using DIRECT or PSO methods for (17) and also obtain the argmax* $\mathbf{x}_{max}$*. If the residual obtained previously could not be lowered for the optimized* $\mathbf{p}_{new}(t+\tau)$ *in step (5), add sample points close to the location of the identified maximum and repeat step (5). Allow a slightly larger residual (e.g. 5%) in the next step as a relaxation factor for* $\epsilon_{spatial,max}$ *in case the correct value was not found in the previous step.*
8. *If the residual is smaller than* $\hat{\epsilon}_{spatial}$*, proceed to step (13), otherwise go on for the spawning procedure.*
9. *Calculate the variance of the additional Gaussian by applying equation (11).*
10. *Add new sample points within the variance around the new sample point at* $\mathbf{x}_{max}$*. Add the Gaussian there as well with the estimated variance and the residual as amplitude.*
11. *Optimize the new Gaussian only and proceed with step 5.*
12. *Calculate* τ_{opt} *according to equation (19). Check whether time step* $\tau_{old} = \tau$ *has been correct and adapt* TOL_x *according to (21).*
13. *If the last time step* τ_{old} *was too large, revert the changes from the last time step and proceed with step (5) again using* $\tau = \sigma\tau_{opt}$*. Otherwise, if the final time of the simulation is not yet reached, set* $\mathbf{p} = \mathbf{p}_{new}(t+\tau)$ *and* $t = t+\tau$ *and proceed with step (4).*

A typical course of the algorithm can be seen in Figure 4, where in the first steps are visualized. The linear step has seemingly little influence but improves the structure of the Gaussians for the optimization e.g. by adjusting their center positions. The crucial step is the optimization according to equation 25 in Algorithm 3, where the improvement becomes visible.

3 Example Applications

Two examples are given to show advantages and disadvantages of the GBP-DEC approach. The first one describes a relatively well behaving nonlinear oscillatory motion with large spatial movements. The second shows a strong nonlinearity and is intended to demonstrate the algorithms behavior close to its limitations.

3.1 Oscillating Motion

The phugoid motion of an uncontrolled glider can be described by

$$\dot{v} = -g\sin(\theta) - R\frac{v^2}{m} \tag{26}$$
$$\dot{\theta} = A\frac{v}{m} - \frac{g}{v}\cos(\theta) \tag{27}$$

using the assumption of small angles. The states of the ordinary differential equation are the velocity v and the pitch angle θ. The other variables, i.e. the aerodynamic lift A and friction constant R, are assumed to be deterministic and chosen such that a low frequency of about 0.05 Hz is obtained. The initial states are assumed to be uncertain. In this example a typical bound derived from engineering judgement or previous analysis is assumed to be $\theta \in [-0.025, 0.025]$ and $v \in [103.6, 106.4]$. Assuming further, that this bound comprises more than 95% of all occurrences, e.g. larger or smaller parameter values are possible but very rare, the variances σ are $\sigma_v \approx 0.5\frac{m}{s}$ and $\sigma_\theta \approx 0.009$rad, respectively. Using these values, a Gaussian distribution in two dimensions is used as initial value for problem (26)-(27).
Using this initial stochastic distribution the GBP-DEC is performed and compared to other methods, i.e.

Calculation		Result				$\overline{t}_{\text{eval}}$	$\overline{m}_{\text{eval}}$
Method	t	Min	Avg	Max	$\bar{\sigma}_{\text{rel}}$		
Linear Sensitivity Analysis	0.0	0.4052	0.4052	0.4052	0.0000	-	-
	1.0	10.9982	10.9982	10.9982	0.0000	-	-
	10.0	0.0000	0.0000	0.0000	0.0000	-	-
Linear Gaussian Particle Method	0.0	0.4052	0.4052	0.4052	0.0000	-	-
	1.0	11.9764	11.9764	11.9764	0.0000	-	-
	10.0	0.0000	0.0000	0.0000	0.0000	-	-
Monte Carlo (10000 samples)	0.0	0.3580	0.7161	1.0742	0.1802	0h:28m	<1MB
	1.0	4.5836	12.3758	19.4806	0.2205	0h:28m	<1MB
	10.0	3.7242	4.8128	6.0160	0.0996	0h:28m	<1MB
Monte Carlo (100000 samples)	0.0	0.3724	0.4469	0.5443	0.0737	4h:20m	2.4MB
	1.0	9.7689	10.7601	11.8316	0.0352	4h:20m	2.4MB
	10.0	4.5550	4.9045	5.3285	0.0368	4h:20m	2.4MB
Finite Volume (11644 cells)	0.0	0.2716	0.2716	0.2716	0.0000	0h:14m	3.1MB
	1.0	8.7270	8.7270	8.7270	0.0000	0h:16m	3.1MB
	10.0	8.2690	8.2690	8.2690	0.0000	0h:23m	3.1MB
Finite Volume (282204 cells)	0.0	0.4184	0.4184	0.4184	0.0000	7h:27	77MB
	1.0	10.9982	10.9982	10.9982	0.0000	8h:09	77MB
	10.0	7.9685	7.9685	7.9685	0.0000	12h:33	77MB
GBP-DEC (initial $\text{TOL}_x = 10^{-3}$)	0.0	0.4061	0.4205	0.4386	0.0394	0h:22m	<1MB
	1.0	10.3914	10.4663	10.5594	0.0081	17h:33m	<1MB
	10.0	5.7729	5.7975	5.8200	0.0040	29h:52	<1MB

Table 1 Scenario 1: For $t = 0, 1$ the probability density at $[104, \frac{\pi}{180}0.5\frac{1}{\sqrt{2}}]^T$ is determined. For $t = 10$ the probability density at $[98, \frac{\pi}{180}2.5]^T$ is calculated.

- solving the linearized problem, by either computing gradients or by using (22)-(24).
- the Monte Carlo method: With its low level of intrusiveness, this method is used frequently. It works well for computing large probabilities but shows slow convergence.
- the finite volume method using a first order upwind scheme: This method solves the Fokker-Planck equation (2) directly.

The problem is relatively difficult to solve for all methods, since it involves large spatial derivatives at locally high densities with steep gradients.
Solving the linearized problem leads to a deterministic solution. Analogously the finite volume method works in a deterministic way. Contrary, the Monte Carlo method leads to a stochastic solution. GBP-DEC does as well not behave deterministic in the sense of an exactly repeatable result. But the results stay within the error bounds given by $\hat{\epsilon}_{\text{t}}$ and $\hat{\epsilon}_{\text{spatial}}$.
It should be noted, that the axes for this problem are of strongly different length scales which requires a pre-scaling of the shape matrices for GBP-DEC.
Table 1 summarizes typical results. Since the initial distribution is given, an exact result is available for comparison at $t = 0$ and matched by the linear methods. Since the gradient is relatively steep, the finite volume method needs a very fine resolution to match the value. Monte Carlo shows a significant variance in the results. GBP-DEC tends to overestimate the density but matches the result within the prescribed accuracy.
After a second of simulation time the linear methods still give a reasonable result. The other methods confirm this result as well. The scatter in the Monte Carlo method is unchanged.
Severe differences become visible after 10 seconds of the simulation time. The linear methods are not suitable any more. Numerical diffusion leads to a significant overestimation of the density for the finite volume methods. Since the gradient is steep at the location, where the density is calculated, the Monte Carlo method is expected to underestimate the density slightly. GBP-DEC shows a low scatter in the results. However, it should be noted that a bias within the prescribed accuracy is possible.
The calculation time for the GBP-DEC is about seven times larger but with only about a tenth of the scatter.

3.2 Ordinary Differential Equations with a Large Nonlinear Contribution

If nonlinearities become too large, the resulting distribution cannot be approximated accurately within the error bounds by Gaussian basis functions any more. In this case particle methods are no more suitable. For example the equations

$$\dot{x}_1 = 0.1x_1 \tag{28}$$

$$\dot{x}_2 = e^{-\frac{(x_2+5)^2}{25}} - 0.1x_1^2 \tag{29}$$

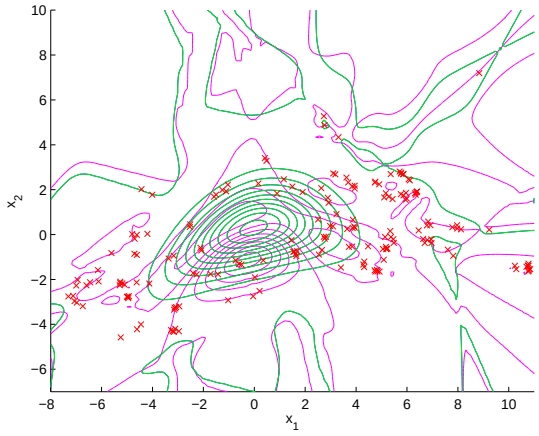


Fig. 5 Distribution at $t = 0.85$. For the GBP-DEC algorithm the green contour lines show the solution. The pink lines give the residual according to the implicit trapezoidal rule. This residual is largest where the density is largest. The nonlinearities are strongest on the left side and on the right side of the distribution. The red crosses denote the locations of the sample points. They are aggregated in the areas with the strongest nonlinearities.

describe a flow field, which is highly nonlinear in the second state x_2 with increasing $|x_1|$. Equation (28), however, ensures a linearly increasing $|x_1|$. The probability density evolves well-behaving at the beginning but then the nonlinearities become stronger on the left and on the right side of the density.
GBP-DEC is applied to a probability density centred at the origin and behaves as expected. Figure 5 shows the distribution evaluated at $t = 0.85$. At this time the number of Gaussians needed to proceed quickly increases towards infinity and the algorithms slows down. The time step gets smaller leading to a lower spatial tolerance according to (21). GBP-DEC comes to a stop as it is supposed to.
If the error bound were only approximated with sample points using (16), the algorithm would proceed without observing large local residuals. Hence, the global search (17) allows to detect situation, where a particle method is not applicable.

4 Conclusion and Way Ahead

Particle methods with Gaussian basis functions provide an efficient way of solving stochastic ordinary differential equations with the Fokker-Planck equation directly. Advances in global optimization are used here to determine residuals during the adaption of the particles to a new time step. The information gathered from the global optimization is used to adapt an efficient estimator for the residual using sample points. This adaption allows to break the exponential behaviour, the sampling approach otherwise has.
Even though the course of the GBP-DEC is not deterministic, the error control is. This is a novel feature of the GBP-DEC method which was not available in previous approaches, notably the TRAIL method [16]. If there is no solution to (1) with a low number of Gaussians, GBP-DEC can observe this, while previous implementations would have progressed without any control.
A significant drawback of GBP-DEC is the strong computational effort. Parallelization is tested for the evaluation of the sample points and works well for later stages of the algorithm and was tested up to 16 cores.
For the optimization of the Gaussian basis functions represented by the parameter vector $\mathbf{p}$ single directional optimization is applied and keeps the optimization problem well conditioned even for a large number of basis functions. It further gives the possibility to optimize different components in $\mathbf{p}$ on different CPUs. The memory consumption is extremely low for this method.
The most difficult part of the algorithm is a sufficiently good and efficient approximation of the norm (16). Especially at the beginning the single directional optimization behaves rather inflexible to correct inefficient placements of Gaussians. Hence a two phase approach, where first all Gaussians are optimized at once before single directional optimization is activated, may further improve computation time. Since in later stages most changes have local effect only, the adaption may be considered spatially local only and parallelized further. Another approach noted in [9] might be to predecompose Gaussians before the linear step is performed such that its accuracy is increased and fewer optimization steps are necessary. This will be part of future work.

References

[1] R.E. Bellmann,“Dynamic Programming”,Dover Publications, ISBN 9780486428093, p. ix, (2003)
[2] M. Ben-Nun, T. J. Martinez, “Nonadiabatic molecular dynamics: Validation of the multiple spawning method for a multidimensional problem”, Journal of Chemical Physics, Vol. 108, Iss. 17, 7244-7257 (1998)
[3] P. Bhattacharyya and B.K. Chakrabarti, “The mean distance to the nth neighbour in a uniform distribution of random points: an application of probability theory”, Eur. J. Phys. 29, 639-645 (2008)
[4] M. Clerc, “Particle Swarm Optimization”, ISBN-10 1905209045, ISBN-13 9781905209040, ISTE (2006)
[5] M. Bjorkman and K. Holmstrom, “Global optimization using the DIRECT algorithm in Matlab”, Advanced Modeling and Optimization, Vol. 1, No. 2 (1999)
[6] A.J. Chorin, O.H. Hald, “Stochastic Tools in Mathematics and Science”, ISBN-10 0-387-28080-4, ISBN-13 978-0387-28080-6, Springer (2006)
[7] B. DeVolder, J. Glimm, J.W. Grove, Y. Kang, Y. Lee, K. Pao, D.H. Sharp and K. Ye, “Uncertainty quantification for multiscale simulations” J. Fluids Engineering 124: 29-41 (2002)
[8] G. Drolshagen, E. J. Heller, “A wavepacket approach to gas-surface scattering: Application to surfaces with imperfections”, Surface Science, Vol. 139, Iss. 1, 260-280, ISSN 0039-6028 (1984)
[9] K. Ellermann, “Zur Analyse zufallserregter Schwingungen mechanischer Systeme”, ISBN 978-3-8322-8663-7, ISSN 1616-0126, Shaker Verlag, Aachen (2009)
[10] M. Feistauer, “Mathematical Methods in Fluid Dynamics”, ISBN 0-582-20988-9, Longman Scientific & Technical, Essex (1993)
[11] M. Feistauer, J. Felcman and I. Straškraba, “Mathematical and Computational Methods for Compressible Flow”, ISBN 0-19-850588-4, Oxford University Press (2003)
[12] M. Griebel, M. A. Schweitzer, “Meshfree Methods for Partial Differential Equations”, Lecture Notes in Computational Science and Engineering, Vol. 26, ISBN: 978-3-540-43891-5, Springer (2003)
[13] I. Horenko, S. Lorenz, C. Schuette and W. Huisinga, “Adaptive approach for non-linear sensitivity analysis of reaction kinetics”, J. of Comp. Chem. , Vol 26, No 9, 941-948, Wiley (2005)
[14] I. Horenko, B. Schmidt and C. Schuette, “Multidimensional classical Liouville dynamics with quantum initial conditions”, J. of Chem. Phys., Vol 117, No 10, 4643-4650, American Institute of Physics (2002)
[15] I. Horenko and M. Weiser, “Adaptive Integration of Molecular Dynamics”, J. Comput. Chem. 24, 1921-1929 (2003)
[16] I. Horenko, M. Weiser, B. Schmidt and C. Schuette, “Fully adaptive propagation of the quantum-classical Liouville equation”, J. of Chem. Phys., Vol 120, No 19, 8913-8923, American Institute of Physics (2004)
[17] D.R. Jones, C.D. Perttunen and B.E. Stuckman, “Lipschitzian Optimization Without Using the Lipschitz Constant”, Journal of Optimization Theory and Application, Vol. 79, No. 1, (1993)
[18] J. Kennedy, R. Eberhart, “Particle swarm optimization”, IEEE International Conference on Neural Networks Proceedings, Vol.4, 1942-1948, IEEE (1995)
[19] J. Kennedy, R. Eberhart, “A New Optimizer Using Particle Swarm Theory”, Sixth International Symposium on Micro Machine and Human Science Proceedings, 0-7803-2676-8/95, IEEE (1995)
[20] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions” SIAM Journal of Optimization, Vol. 9, No. 1, 112-147, Siam (1998)
[21] H. Liu, A. Abraham and M. Clerc,”Chaotic dynamic characteristics in swarm intelligence”, Applied Soft Computing, Vol. 7, Iss. 3, 1019-1026, ISSN 1568-4946, (2007)
[22] H. Neunzert , A. Klar , J. Struckmeier, “ Particle Methods: Theory and Applications”, ICIAM 95: proceedings of the Third International Congress on Industrial and Applied Mathematics (1995)
[23] H. Risken, “The Fokker-Planck Equation, Methods of Solution and Applications”, Second Edition, ISBN 3-540-50498-2, Springer (1989)
[24] C.P. Robert, G. Casella, “Monte Carlo Statistical Methods”, 2nd ed., ISBN-10: 1441919392, Springer, (2004)
[25] M. Scharpenberg and M. Lukáčová-Medviďová, “Stochastic Considerations for Dynamic Systems”, 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference Proceedings, AIAA 2008-6054, Victoria, BC (2008)
[26] B.O. Shubert, “A Sequential Method Seeking the Global Maximum of a Function”, Journal on Numerical Analysis, Vol. 9, 379-388, Siam (1972)
[27] S. Sawada, R. Heather, B. Jackson and H. Metiu, “A strategy for time dependent quantum mechanical calculations using a Gaussian wave packet representation of the wave function” , J. Chem. Phys. 83, 3009 (1985)
[28] S. M. Stigler, “Stochastic Simulation in the Nineteenth Century”, Statistical Science, Vol. 6, No. 1,89-97, Institute of Mathematical Statistics (1991)
[29] A.Y. Weiße, I. Horenko and W. Huisinga, “Adaptive Approach for Modelling Variability In Pharmacokinetics”, CompLife, LNBI 4216, 194-204, Springer-Verlag Berlin Heidelberg (2006)
[30] A. Younis, Z. Dong, and J. Gu, “Trends, Features, and Tests of Common and Recently Introduced Global Optimization Methods”, 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference Proceedings, AIAA 2008-5853, Victoria, BC (2008)