Nada är en gemensam institution mellan
Kungliga Tekniska högskolan och Stockholms universitet.

Adaptivity for Stochastic and Partial Differential Equations with Applications to Phase Transformations

ERIK VON SCHWERIN

Doctoral Thesis
Stockholm, Sweden 2007

TRITA-CSC-A 2007:12
ISSN-1653-5723
ISRN KTH/CSC/A--07/12--SE
ISBN 978-91-7178-744-6

Skolan för Datavetenskap och Kommunikation
KTH
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen måndagen den 17 september 2007 kl 13.00 i sal F3, Lindstedtsvägen 26, Kungl Tekniska högskolan, Stockholm.

Tryck: Universitetsservice US AB

Abstract

This work is concentrated on efforts to efficiently compute properties of systems, modelled by differential equations, involving multiple scales. Goal oriented adaptivity is the common approach to all the treated problems. Here the goal of a numerical computation is to approximate a functional of the solution to the differential equation and the numerical method is adapted to this task.

The thesis consists of four papers. The first three papers concern the convergence of adaptive algorithms for numerical solution of differential equations; based on a posteriori expansions of global errors in the sought functional, the discretisations used in a numerical solution of the differential equiation are adaptively refined. The fourth paper uses expansion of the adaptive modelling error to compute a stochastic differential equation for a phase-field by coarse-graining molecular dynamics.

An adaptive algorithm aims to minimise the number of degrees of freedom to make the error in the functional less than a given tolerance. The number of degrees of freedom provides the convergence rate of the adaptive algorithm as the tolerance tends to zero. Provided that the computational work is proportional to the degrees of freedom this gives an estimate of the efficiency of the algorithm.

The first paper treats approximation of functionals of solutions to second order elliptic partial differential equations in bounded domains of $\mathbb{R}^d$, using isoparametric d-linear quadrilateral finite elements. For an adaptive algorithm, an error expansion with computable leading order term is derived and used in a computable error density, which is proved to converge uniformly as the mesh size tends to zero. For each element an error indicator is defined by the computed error density multiplying the local mesh size to the power of $2 + d$. The adaptive algorithm is based on successive subdivisions of elements, where it uses the error indicators. It is proved, using the uniform convergence of the error density, that the algorithm either reduces the maximal error indicator with a factor or stops; if it stops, then the error is asymptotically bounded by the tolerance using the optimal number of elements for an adaptive isotropic mesh, up to a problem independent factor. Here the optimal number of elements is proportional to the $d/2$ power of the $L^{\frac{d}{d+2}}$ quasi-norm of the error density, whereas a uniform mesh requires a number of elements proportional to the $d/2$ power of the larger L^1 norm of the same error density to obtain the same accuracy. For problems with multiple scales, in particular, these convergence rates may differ much, even though the convergence order may be the same.

The second paper presents an adaptive algorithm for Monte Carlo Euler approximation of the expected value $E[g(X(\tau), \tau)]$ of a given function g depending on the solution X of an Itô stochastic differential equation and on the first exit time τ from a given domain. An error expansion with computable leading order term for the approximation of $E[g(X(T))]$ with a fixed final time $T > 0$ was given in [Szepessy, Tempone, and Zouraris, Comm. Pure and Appl. Math., 54, 1169-1214, 2001]. This error expansion is now extended to the case with stopped diffusion. In the extension conditional probabilities are used to estimate the first exit time error, and differ-

ence quotients are used to approximate the initial data of the dual solutions. For the stopped diffusion problem the time discretisation error is of order $N^{-1/2}$ for a method with N uniform time steps. Numerical results show that the adaptive algorithm improves the time discretisation error to the order N^{-1}, with N adaptive time steps.

The third paper gives an overview of the application of the adaptive algorithm in the first two papers to ordinary, stochastic, and partial differential equation.

The fourth paper investigates the possibility of computing some of the model functions in an Allen–Cahn type phase-field equation from a microscale model, where the material is described by stochastic, Smoluchowski, molecular dynamics. A local average of contributions to the potential energy in the micro model is used to determine the local phase, and a stochastic phase-field model is computed by coarse-graining the molecular dynamics. Molecular dynamics simulations on a two phase system at the melting point are used to compute a double-well reaction term in the Allen–Cahn equation and a diffusion matrix describing the noise in the coarse-grained phase-field.

ISBN 978-91-7178-744-6 • TRITA-CSC-A 2007:12 • ISSN-1653-5723 • ISRN KTH/CSC/A--07/12--SE

Acknowledgments

First of all I wish to thank my always enthusiastic advisor Anders Szepessy for his academic guidance and his encouragement and support.

I am grateful to all my colleagues in the numerical analysis group in the Computational Phase Transformation project for creating a friendly and stimulating atmosphere. A special “thank you” goes to my fellow Ph.D. students of the weekly meetings: Kyoung-Sook Moon, Raúl Tempone, Mattias Sandberg, and Jesper Carlsson.

The stochastic molecular dynamics code is based on a code for Hamiltonian systems written by Måns Elenius who would also always take the time to answer questions. Mikhail Dzugutov and Anatoly Belonoshko introduced me to molecular dynamics simulations. Thank you!

I also thank Daniel Appelö, Jesper Oppelstrup, Tomas Oppelstrup, and Raúl Tempone for reading various parts of the manuscript and providing useful suggestions.

Financial support has been provided by the Swedish Foundation for Strategic Research (SSF), research grant A3 02:123, "Mathematical theory and simulation tools for phase transformations is materials", and is gratefully acknowledged.

Contents

List of Papers

The thesis consists of four papers and an introduction.

Paper I: *"Convergence Rates for an Adaptive Dual Weighted Residual Finite Element Algorithm"*,

K-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone
BIT Numerical Mathematics (2006) 46: 367–407.

The author of this thesis contributed to the proof of convergence of the error density, performed the numerical computations, which influenced the proof, and contributed to the writing of parts of the paper, mainly the numerical section.
This paper is also part of the licentiate thesis [31].

Paper II: *"Adaptive Monte Carlo Algorithms for Stopped Diffusion"*,
A. Dzougoutov, K-S. Moon, E. von Schwerin, and A. Szepessy, R. Tempone,
Lecture Notes in Computational Science and Engineering **44**,
"Multiscale Methods in Science and Engineering", 59–88,
Springer–Verlag, Berlin Heidelberg, 2005.

The author of this thesis derived the error expansion for stopped diffusion in multi dimensional domains and wrote the paper, extending the one dimensional setting in [17], Paper V.
This paper is also part of the licentiate thesis [31].

Paper III: *"An Adaptive Algorithm for Ordinary, Stochastic and Partial Differential Equations"*,
K-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone,
Contemporary Mathematics, **383**,
"Recent Advances in Adaptive Computation", 325–343,
American Mathematical Society, Providence, 2005.

This paper presents an overview of the work of the group on an adaptive algorithm, presented in a series of papers including Paper I and Paper II. The author of this thesis participated in the writing of the paper.

Paper IV: *"A Stochastic Phase-Field Model Computed From Coarse-Grained Molecular Dynamics"*,
E. von Schwerin.

Chapter 1

Introduction

Differential equations are important in the formulation of mathematical models in many areas of science and engineering. Such models may be used to get an understanding of global properties of the system being modelled, from analytical solutions to the differential equations, from qualitative analysis of the dependence on model parameters, or from approximate numerical solutions for particular parameter values. However, mathematical models are also commonly used, not primarily to study global behaviour, but to predict the values of one or several scalar quantities of particular importance for the application at hand. Mathematically, such quantities correspond to functionals of the solutions to the differential equations. When the underlying differential equations are solved numerically, with finite computational resources, it is desirable to minimise the computational work for a given accuracy in the functional values. In goal oriented adaptivity, for a fixed numerical method of approximation, the degrees of freedom are adapted to both the differential equation and the functional in an attempt to minimise the work needed to meet the error tolerance in the goal functional. The first three articles in this thesis aim at increased understanding of optimal convergence rates for goal oriented adaptive algorithms; one adaptive algorithm is studied in different settings, in particular those of deterministic elliptic partial differential equations in bounded d-dimensional domains using isoparametric d-linear quadrilateral finite element approximations, and of Itô stochastic differential equations using the Euler Monte Carlo method.

A common characteristic among many problems where adaptivity is used is that two or more scales are involved and that poor accuracy in the small details may propagate to the large scale properties of the solution. For example, if a

certain partial differential equation, combined with the goal functional, have the property that the solution must be resolved well in parts of the domain to obtain the desired accuracy, then a uniform discretisation may waste much computational effort by over-resolving the solution in the rest of the domain. The situation in the fourth article concerns instead the problem of computing an approximate model on a macroscopic scale from an underlying microscopic model, which is assumed to be more fundamental. This can be seen as a form of adaptivity in the model, although the computations on the microscale and the macroscale are separated. In this case the macroscopic model is a stochastic phase-field equation describing the time evolution of a system with a solid and a liquid phase. The microscopic model of the material is that of interacting particles with positions given by the stochastic Smoluchowski molecular dynamics.

Chapter 2

Background on Adaptive Algorithms

Adaptive and Non-Adaptive Algorithms Consider the problem of computing an approximate value of $g(f)$ of a functional $g\colon X \to \mathbb{R}$ for $f \in F$, where F is a subset of the normed linear space X. Often a numerical method for this problem is on the form

$$g^n(f) = \phi^n(L_1(f), \ldots, L_n(f)), \tag{2.1}$$

where $L_i\colon X \to \mathbb{R}$ are linear functionals and $\phi^n\colon X \to \mathbb{R}$ is linear or nonlinear. The functionals can for example be function evaluations, $L_i(f) = f(x_i)$. The method g^n is called *non-adaptive* if the functionals L_i are the same for all $f \in F$. It is called *adaptive* if the choice of functionals L_i depends on f through the previously computed values $L_1(f), \ldots, L_{i-1}(f)$.

In information based complexity theory there is a general result by Bakhvalov and Smolyak comparing adaptive and non-adaptive methods for approximation of linear functionals, $g : X \to \mathbb{R}$, on a normed linear function space, X. The result states that for any adaptive method (2.1) using a fixed number of linear functionals L_i to approximate the linear g, defined on a symmetric convex subset F of X, there is a linear non-adaptive method whose maximal error, on F with the same number of linear functionals, is as small as that of the adaptive method. A more detailed formulation can be found the overview article [24] by Novak.

How does the adaptive algorithms for computation of linear functionals of solutions to differential equations which are considered here relate to the result of Bakhvalov and Smolyak? The point of view is different in that a fixed

method, for example a finite element method of given order, is considered with the aim to construct an adaptive mesh refinement algorithm for that method. Also, in contrast to keeping the number of steps in the algorithm fixed, the aim here is to create an algorithm where the number of steps, as a function of the the error tolerance is close to optimal as the tolerance tends to zero.

Consider a numerical method based on uniform discretisation of a d-dimensional domain with element size h and with approximation error $\Theta(h^p)$, as $h \to 0$, using the notation that $f = \Theta(g)$ if and only if $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$. Making the error less than a tolerance TOL requires $\Theta(\text{TOL}^{-d/p})$ elements. Assuming that the work is proportional to the number of elements the performance of the method can be expressed in terms of the tolerance, as $\text{TOL} \to 0$. This measure of the efficiency is natural to extend to adaptive algorithms as illustrated in a simple setting in the next example.

Example: Numerical Integration The assumption of a convex domain of definition, F, for the adaptive and non-adaptive methods in the result of Bakhvalov and Smolyak mentioned above is important. To illustrate this and to show how convergence rates for adaptive algorithms are measured here, consider the linear functional given by an integral of a known function, $g(f) = \int_0^T f(t)\,dt$, and let the method of numerical integration be the left point rule (forward Euler). Discretise the time interval $[0, T]$ into N subintervals $0 = t_0 < t_1 < \cdots < t_N = T$ with steps $\Delta t_n := t_{n+1} - t_n$. With $\bar{g}(f)$ denoting the numerical approximation of $g(f)$ the global discretisation error becomes

$$g(f) - \bar{g}(f) = \sum_{n=0}^{N-1} \rho_n (\Delta t_n)^2 + \text{ higher order terms}, \tag{2.2}$$

where the error density function ρ is given by $\rho_n := \frac{df}{dt}(t_n)/2$. As an example of a non-adaptive method consider uniform Δt. Using that the number of time steps is

$$N(\Delta t) = \int_0^T \frac{1}{\Delta t(\tau)}\, d\tau, \tag{2.3}$$

the number N_u of uniform steps to reach a given level of accuracy TOL is asymptotically proportional to TOL^{-1} with the L^1-norm of the function ρ in the proportionality constant,

$$N_u \simeq \frac{T}{\text{TOL}} \|\rho\|_{L^1(0,T)}, \tag{2.4}$$

provided that ρ has constant sign. When the number of steps in (2.3) is minimised with the accuracy constraint that the leading order of (2.2) is TOL, the optimal distribution of time steps is

$$\rho_n \Delta t_n^2 = \text{ constant for all } n.$$

With this choice the number N_a of adaptive steps becomes proportional to TOL^{-1} with the smaller $L^{\frac{1}{2}}$-quasi-norm of the error density as the proportionality constant,

$$N_a \simeq \frac{1}{\mathrm{TOL}} \|\rho\|_{L^{\frac{1}{2}}(0,T)}. \tag{2.5}$$

Since the Euler method uses one function evaluation per step the asymptotic number of steps (2.4) and (2.5) give the convergence rates of the Euler method using uniform and optimal adaptive time steps respectively.

Take for example the integrand $f(t) = 1/\sqrt{t+\epsilon}$ for a small positive parameter $\epsilon \ll T$. Since $\rho(t) = \frac{-1}{4(t+\epsilon)^{3/4}}$ the number of uniform steps becomes

$$N_u \simeq \frac{T/4}{\mathrm{TOL}} \int_0^T \frac{dt}{(t+\epsilon)^{3/2}} \approx \frac{T/4}{\mathrm{TOL}} \frac{1}{\epsilon^{1/2}},$$

while the number of adaptive time steps is smaller,

$$N_a \simeq \frac{1/4}{\mathrm{TOL}} \left(\int_0^T \frac{dt}{(t+\epsilon)^{3/4}} \right)^2 \approx \frac{4\sqrt{T}}{\mathrm{TOL}}.$$

The smaller multiple of 1/TOL with an adaptive approach captures the multiple scales introduced by $\epsilon \ll T$. In this example, the integrand can also be viewed as an approximation of the singular $1/\sqrt{t}$, in which case the parameter must be taken $\epsilon^{1/2} = o(\mathrm{TOL})$, so that $N_a/N_u \to 0$ as $\mathrm{TOL} \to 0$.

If $F = \{f \ : \ ||f'||_{L^{\frac{1}{2}}} < M\}$ for a constant M, then the integrand in the example above is in F for some ϵ depending on M. Note that F is non-convex so that the result by Bakhvalov and Smolyak does not apply to the problem of computing $g(f)$ for $f \in F$. In this class of integrands the choice of uniform steps in the non-adaptive method is motivated by considering integrands $f_s(t) = 1/\sqrt{|t-s|+\epsilon}$ with ϵ just large enough for f_s to be in F for all $s \in [0,T]$. However, it is not always the case that optimal non-adaptive discretisations for a fixed method are uniform, as is illustrated in the next example.

Example: Corner Singularity for an Elliptic Partial Differential Equation Let u, in a domain Ω with a crack as in Figure 2.1, be the solution of the Laplace equation

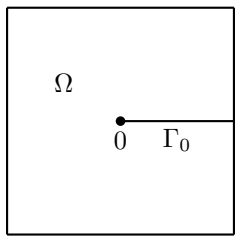


Figure 2.1: Domain with a crack

$$\begin{aligned} -\Delta u &= 0, && \text{in } \Omega, \\ u &= 0, && \text{on } \Gamma_0, \\ u &= f, && \text{on } \partial\Omega \setminus \Gamma_0, \end{aligned}$$

and let $g(f)$, viewed as a linear functional of the Dirichlet boundary values f, be given by the integral of u,

$$(u, 1) = \int_\Omega u \, dx.$$

Even if the boundary conditions are taken from a class of smooth functions the solution u will in general have a form like $u(r, \theta) = \sqrt{r}\alpha(\theta) + \beta(r, \theta)$ in polar coordinates with smooth α and β close to the tip of the crack; see for example the textbook [15] by Johnson. For a given solution method of the boundary value problem, for example bilinear finite elements on a grid with square elements and hanging nodes, the a priori information of the singularity of the derivative of the exact solution can be used to construct non-uniform non-adaptive meshes for this particular geometry. On the other hand, in applications where the domain Ω varies adaptive methods allow the mesh to automatically adapt to the geometry without the detailed a priori knowledge of the solution. This is the situation considered in Paper I.

Chapter 3

An Adaptive Algorithm

This chapter describes an adaptive algorithm for computing approximate solutions to problems which can abstractly be stated as:

$$\begin{aligned} &\text{compute the functional } g(u) \\ &\text{where } u \text{ solves an initial or boundary value problem} \\ &\text{for a differential equation in a } d\text{-dimensional domain } \Omega. \end{aligned} \tag{3.1}$$

For a given method of numerical approximation of u, based on discretisation of the domain Ω, the algorithm constructs the final discretisation by iterative refinements of an initial mesh; the algorithm presupposes an expansion of the error in the scalar quantity $g(u)$ of the form

$$\text{Global error} = \sum \text{local error} \cdot \text{weight} + \text{higher order error}, \tag{3.2}$$

depending on the approximation method and on the problem; compare (2.2) in the numerical integration example. The leading order terms must be computable using information on the current mesh. The weight describes the influence of changes in the differential equation on the functional of its solution. The goal of the adaptive algorithm is to, for the given approximation method, approximate $g(u)$ using an adapted mesh with a minimal number of intervals (elements) for error less than a given tolerance.

Concrete formulations of the abstract (3.1) are for example:

- the computation of $g(u) = (u(T))^2$ where u solves an ordinary differen-

tial equation

$$\begin{aligned} \frac{du}{dt}(t) &= a(t, u(t)), \qquad 0 < t < T, \\ u(0) &= u_0, \end{aligned} \tag{3.3}$$

with flux $a\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ and where an approximate solution u_h is obtained by any p:th order numerical method using $u_h(0) = u_0$ and $\Omega = [0,T]$ is discretised into $0 = t_0 < t_1 < \cdots < t_N = T$.

- the computation of $g(u)$ where u solves an elliptic partial differential equation in a bounded open domain $\Omega \subset \mathbb{R}^d$ and where an approximate solution u_h is obtained using a given finite element method; see the example on page 6.

Equidistribution of Errors Consider (3.1) in a domain of dimension d with a given approximation method of order p. Assume that an asymptotic error expansion (3.2) on the form

$$\text{error} \simeq \sum_n \rho_n h_n^{p+d}$$

is known, where h is the local mesh size, of the non-stretched element, and ρ is independent of h. The number of elements that corresponds to a mesh with size h can be determined by

$$N(h) := \int_\Omega \frac{dx}{h^d(x)}. \tag{3.4}$$

If the sign of the error density varies a very small set of elements may give an error in the functional that is close to zero due to cancellation of error contributions of opposite sign. Thus the optimal mesh may consist of very few elements, but it seems difficult to exploit the cancellation of errors when constructing the mesh. Disregarding the possible cancellation by minimising the number of elements N in (3.4) under the constraint

$$\sum_{n=1}^{N} |\rho_n| h_n^{d+p} = \int_\Omega |\rho(x)| h^p(x) dx = \text{TOL},$$

gives the optimum

$$|\rho|(h^*)^{d+p} = \text{constant} \tag{3.5}$$

with corresponding mesh size function

$$h^*(x) := \frac{\mathrm{TOL}^{\frac{1}{p}}}{|\rho(x)|^{\frac{1}{d+p}}} \left(\int_\Omega |\rho(x)|^{\frac{d}{d+p}} dx \right)^{-\frac{1}{p}}. \tag{3.6}$$

This condition is optimal only for density functions ρ with one sign. Moreover, in higher dimension, $d > 1$, it is optimal only for meshes with non-stretched elements, that is, elements such that each element is described by one element size h.

The adaptive refinement algorithm, described in a generic deterministic form in Algorithm 1 below, is designed to approximate the optimal equidistribution of error contributions (3.5). With $[k]$ denoting quantities on the k:th mesh in the refinement sequence, the accepted mesh k_{stop} ideally fulfils

$$\hat{\rho}_n[k_{\mathrm{stop}}](h_n[k_{\mathrm{stop}}])^{d+p} \approx \frac{\mathrm{TOL}}{N[k_{\mathrm{stop}}]}, \qquad n = 1, 2, \ldots, N[k_{\mathrm{stop}}],$$

where $\hat{\rho}_n$ is a computable approximation of the unsigned error density, $|\rho|$. Thus, after calculating $\hat{\rho}[k]$ from computed approximate primal and dual solutions on level k, the algorithm refines all elements with error indicators $\bar{r}_n[k] := \hat{\rho}_n[k](h_n[k])^{d+p} > s_1 \mathrm{TOL}/N[k]$, where $s_1 \approx 1$ is a constant. The maximal error indicator may reduce slowly when most $\bar{r}_n$ are small, $\bar{r}_n[k] \leq s_1 \mathrm{TOL}/N[k]$, leading to many refinements; to avoid this the refinements stop when all $\bar{r}_n[k] \leq S_1 \mathrm{TOL}/N[k]$ for a constant $S_1 > s_1$. In summary, the new element sizes $h[k+1]$ are obtained from $h[k]$ by:

Algorithm 1: Refinement and stopping

```
forall intervals (elements) n = 1, 2, ..., N[k] do
    r̄_n[k] = ρ̂_n[k](h_n[k])^{d+p}
    if r̄_n[k] > s_1 TOL/N[k] then
        mark interval (element) n for division
    end
end

if max_{1≤n≤N[k]} r̄_n[k] ≤ S_1 TOL/N[k] then
    stop the refinements
else
    divide every marked interval (element) into 2^d sub intervals
    (elements)
end
```

The optimality condition (3.5) was obtained from the assumption of a limit

error density, ρ, and the adaptive algorithm was constructed to approximate (3.5) using a computed approximate error density $\hat{\rho}$. This is meaningful if $\hat{\rho}$ converges to $|\rho|$ as TOL $\to 0$. Thus for any particular application the proof of this convergence is crucial for the theoretical analysis of the algorithm. It is possible to analyse the important properties of stopping, accuracy and efficiency of the algorithm in terms of convergence of $\hat{\rho}$.

Stopping of Algorithm 1 Assume the convergence of $\hat{\rho}$ where this positive approximate error density is bounded away from zero by a lower bound δ which tends to 0 with TOL as

$$\delta = \text{TOL}^{\gamma}, \tag{3.7}$$

for a positive parameter γ which depends on the application. Then the change in the density $\hat{\rho}(K)[k]$ in an element K on refinement level k from its value on the parent element on a previous level, $p(K,k)$ can be bounded; it follows from the convergence assumption and (3.7) and an additional assumption (3.12) on the initial mesh size that there exist functions $\hat{c}$ and $\hat{C}$, close to 1 for sufficiently refined meshes, such that

$$\hat{c}(K) \leq \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]} \leq \hat{C}(K), \tag{3.8a}$$

$$\hat{c}(K) \leq \frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]} \leq \hat{C}(K). \tag{3.8b}$$

The lower bound on the quotients here can be used, together with the refinement and stopping criteria in Algorithm 1, to prove the following theorem, which shows that the slow reduction of the maximal error indicator is avoided for S_1 chosen suitably larger than s_1.

Theorem (Stopping)**.** *With the adaptive refinement and stopping strategy in Algorithm 1, assume that $\hat{c}$ satisfies* (3.8a)–(3.8b)*, for the elements or time steps corresponding to the maximal error indicator on each refinement level, and that*

$$S_1 \geq \frac{2^d}{\hat{c}} s_1, \quad 1 > \frac{\hat{c}^{-1}}{2^{d+p}}.$$

Then each refinement level either decreases the maximal error indicator with the factor $\frac{\hat{c}^{-1}}{2^{d+p}}$, that is

$$\max_{1\leq n\leq N[k+1]} \bar{r}_n[k+1] \leq \frac{\hat{c}^{-1}}{2^{d+p}} \max_{1\leq n\leq N[k]} \bar{r}_n[k], \tag{3.9}$$

or it stops the algorithm.

Accuracy of Algorithm 1 By construction the adaptive algorithm guarantees that the estimate of the global error is bounded by a given error tolerance, TOL. Is also the true global error bounded by TOL asymptotically? The stopping criterion in Algorithm 1 gives an upper bound of the error indicators, which together with the assumed convergence of $\hat{\rho}$ leads to an asymptotic bound of the global error of the kind

$$\limsup_{\mathrm{TOL}\to 0+} \left(\mathrm{TOL}^{-1} \big| g(u) - g(u_h) \big| \right) \leq S_1,$$

where u is the exact solution and u_h the computed approximation. See Theorem 3.3 in Paper I for a precise formulation for the dual weighted residual finite element algorithm considered there for second order elliptic partial differential equations.

Efficiency of Algorithm 1 The goal of the adaptive algorithm is to determine a mesh with a minimal number of elements or time steps, N, for the specified accuracy. The optimality condition (3.6) in the equation (3.4) for N gives the optimal number of adaptive elements

$$N^{\mathrm{opt}} = \int_\Omega \frac{dx}{(h^*(x))^d} = \frac{1}{\mathrm{TOL}^{\frac{d}{p}}} \left(\int_\Omega |\rho[k](x)|^{\frac{d}{d+p}} dx \right)^{\frac{d+p}{p}} = \frac{1}{\mathrm{TOL}^{\frac{d}{p}}} \|\rho\|^{\frac{d}{p}}_{L^{\frac{d}{d+p}}}. \tag{3.10}$$

With a uniform mesh, constant mesh size h, the number of elements, N^{uni}, to achieve $\sum_{i=1}^N |\rho_i| h^{d+p} = \mathrm{TOL}$ becomes instead

$$N^{\mathrm{uni}} = \int_\Omega \frac{dx}{h^d(x)} = \frac{\int_\Omega dx}{\mathrm{TOL}^{\frac{d}{p}}} \left(\int_\Omega |\rho[k](x)| dx \right)^{\frac{d}{p}} = \frac{\int_\Omega dx}{\mathrm{TOL}^{\frac{d}{p}}} \|\rho\|^{\frac{d}{p}}_{L^1}. \tag{3.11}$$

Since, by Jensen's inequality, $\|f\|_{L^{\frac{d}{d+p}}} \leq (\int_\Omega dx)^{\frac{p}{d}} \|f\|_{L^1}$, the asymptotic constant multiplying $1/\mathrm{TOL}^{d/p}$ in the convergence order is smaller for the adaptive method than the uniform element size method. For problems with multiple scale solutions the difference may be significant; compare the integration example in Chapter 1.

From the refinement criterion in Algorithm 1, a lower bound of the error indicators follows for the refined parent error indicator. This, together with the assumption that upper bound of the ratios of the error density (3.8a)–(3.8b) holds for all elements on the final mesh, and an assumption

$$h_K[1] = \Theta(\mathrm{TOL}^s), \tag{3.12}$$

on the initial mesh size to guarantee that, for sufficiently small TOL, all elements on the initial mesh are refined, can be used to show that Algorithm 1 generates a mesh which is optimal, (3.10), up to a multiplicative constant independent of the data,

$$(\mathrm{TOL}^{\frac{d}{p}} N) \leq C \, \|\hat{C}\hat{\rho}\|^{\frac{d}{p}}_{L^{\frac{d}{d+p}}} \leq C \left(\max_{x \in D} \hat{C}(x)^{\frac{d}{p}} \right) \|\hat{\rho}\|^{\frac{d}{p}}_{L^{\frac{d}{d+p}}}, \tag{3.13}$$

with $C \leq (\frac{2^{d+p}}{s_1})^{\frac{d}{p}}$. See Theorem 3.4 in Paper I for a precise formulation in a specific case.

Earlier Applications of Algorithm 1 to Stochastic Differential Equations The work [29, 20] treat the weak approximation of an Itô Stochastic differential equation of the form

$$dX_k(t) = a_k(t, X(t))dt + \sum_{\ell=1}^{\ell_0} b_k^\ell(t, X(t))dW^\ell(t), \quad t > 0, \tag{3.14}$$

where $k = 1, \ldots, d$, and $X(t; \omega)$ is a stochastic process in $\mathbb{R}^d$, with independent one dimensional Wiener processes $W^\ell(t; \omega)$, $\ell = 1, \ldots, \ell_0$. The functions $a(t, x) \in \mathbb{R}^d$ and $b^\ell(t, x) \in \mathbb{R}^d$, $\ell = 1, \ldots, \ell_0$, are given drift and diffusion fluxes.

Weak approximation of the stochastic differential equation by the Euler Monte Carlo method approximates the expected value $E[g(X(T))]$ of a functional of the solution with a sample average of $g(\overline{X}(T))$, where $\overline{X}(t_n)$ are identically distributed samples of a discrete time approximation of $X(t_n)$ in the times $0 = t_0 < t_1 < \cdots < t_N = T$ using the Euler method,

$$\overline{X}(t_{n+1}) - \overline{X}(t_n) = a(t_n, \overline{X}(t_n))\Delta t_n + \sum_{\ell=1}^{\ell_0} b^\ell(t_n, \overline{X}(t_n))\Delta W_n^\ell, \tag{3.15}$$

for $\Delta t_n = t_{n+1} - t_n$, $\Delta W_n^\ell = W^\ell(t_{n+1}) - W^\ell(t_n)$, $n = 0, 1, 2, \ldots, N-1$. The aim of the adaptive algorithm is to, for a given error tolerance, obtain

$$\left| E[g(X(T))] - \frac{1}{M} \sum_{j=1}^{M} g(\overline{X}(T; \omega_j)) \right| \leq \mathrm{TOL} \tag{3.16}$$

with a probability close to one, and doing this with minimal computational work, proportional to the total number of stochastic time steps N_{ω_j} for the M realisations.

The error in (3.16) splits naturally into two parts,

$$E[g(X(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(T;\omega_j))$$
$$= \big(E[g(X(T)) - g(\overline{X}(T))]\big) + \left(E[g(\overline{X}(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(T))\right) \quad (3.17)$$

corresponding to time discretisation error and statistical error.

Talay and Tubaro derived a priori estimates of the error (3.16) in [30]. This is modified to an error expansion with a posteriori computable leading order term in [29] using computable stochastic flows and discrete dual backward problems. In [20] convergence of algorithms based on the error expansion is analysed in terms of stopping, accuracy, and efficiency using both stochastic and deterministic time steps in the control of the time discretisation error. With stochastic adaptive time steps Algorithm 1 controls the refinements and the stopping in the computation of each sample path. In those time steps that are marked for refinement the sample value of the Wiener processes in the midpoints are simulated using Brownian bridges

$$W^l\left(\frac{t_n + t_{n+1}}{2}\right) = \frac{1}{2}\left(W^l(t_n) + W^l(t_{n+1})\right) + z_n^l, \quad (3.18)$$

where z_n^l are independent normally distributed random variables with mean 0 and variance $(t_{n+1} - t_n)/4$, independent also of previous $W^l(t_j)$.

The statistical error, governed by the Central Limit Theorem, is asymptotically bounded by $c_0\overline{\sigma}/\sqrt{M}$ where $\overline{\sigma}$ is the sample average of the standard deviation of $g(\overline{X}(T))$ and c_0 is a positive constant for a confidence interval.

Paper II extends the earlier work on stochastic differential equations to stopped diffusion problems; see Section 4.4.

Chapter 4

Summary of Papers

4.1 An Adaptive Dual Weighted Residual Finite Element Algorithm

Consider an adaptive finite element algorithm to approximate linear functionals

$$g(u) = (u, F) := \int_\Omega uF \, dx$$

of multiscale solutions, $u : \Omega \to \mathbb{R}$, of the second order elliptic partial differential equation

$$-\mathrm{div}(a\nabla u) = f \tag{4.1}$$

in a given open bounded domain $\Omega \subset \mathbb{R}^d$ with Dirichlet boundary condition $u|_{\partial\Omega} = 0$. The weak form of (4.1) is

$$(a\nabla u, \nabla v) = (f, v), \qquad \forall v \in H_0^1(\Omega),$$

where the Sobolev space $H_0^1(\Omega)$ is the Hilbert space of functions on Ω, vanishing on $\partial\Omega$, such that the first derivatives are in $L^2(\Omega)$. The finite element approximate solution, u_h, solves the corresponding discrete variational form,

$$(a\nabla u_h, \nabla v) = (f, v), \qquad \forall v \in V_h, \tag{4.2}$$

where V_h is a finite dimensional subspace of $H_0^1(\Omega)$; see for example [5] by Brenner and Scott. For the purpose of Paper I, V_h is the set of continuous piecewise isoparametric bilinear quadrilateral functions in $H_0^1(\Omega)$, using an adaptive quadrilateral mesh with hanging nodes. In the dual weighted residual

method, see [2] by Becker and Rannacher, a dual function $\varphi \in H_0^1(\Omega)$ defined by

$$(a\nabla v, \nabla\varphi) = (v, F), \quad \forall v \in H_0^1(\Omega),$$

is introduced to describe the sensitivity of the functional value on the fluctuations in the solution to the partial differential equation. From the definition of the dual, the error in the functional value is

$$(u - u_h, F) = (a\nabla(u - u_h), \nabla\varphi) = (\mathcal{R}(u_h), -\varphi),$$

with the residual $\mathcal{R}(v) = -\mathrm{div}(a\nabla v) - f$, defined as a distribution in $H^{-1}(\Omega)$ for $v \in H_0^1(\Omega)$. By this and the orthogonality (4.2) applied to $\pi\varphi \in V_h$, where $\pi\varphi$ is the nodal interpolant on V_h, the error in the functional has the dual weighted residual representation

$$(u - u_h, F) = (\mathcal{R}(u_h), \pi\varphi - \varphi)\,. \tag{4.3}$$

Taking inspiration from [10], by Eriksson et.al., and [2] Paper I contains a derivation of a computable approximation $\sum_K \bar{\rho}_K h_K^{d+2}$ of (4.3) for adaptive meshes with at most one hanging node per edge where the refinements of the initial elements are obtained by successive division of elements into 2^d, so that the transformation of each initial element to the reference tensor element maps the corresponding sub mesh to a tensor hanging node mesh.

The new difficulty when elliptic partial differential equations are considered instead of ordinary differential equations is the analysis of the convergence of the error density.

In contrast to the common approach to derive an a posteriori error estimate, the aim here is to derive a uniformly convergent error density with computable leading order term and formulate an adaptive algorithm with proved convergence rates. The works [1] by Babuška and Vogelius, [9] by Dörfler, and [23] by Morin, Nochetto, and Siebert study the convergence of adaptive algorithms for finite element approximations of partial differential equations.

There are also recent work on the convergence rates of adaptive algorithms for numerical solution of elliptic partial differential equations, in terms of the computational work. DeVore [7] shows the efficiency of adaptive approximation of functions, including wavelet expansions. In [6] Cohen, Dahmen, and DeVore use an adaptive N-term wavelet-based approximation algorithm and proves that it produces a solution which is asymptotically optimal in the energy norm error for linear coercive elliptic problems. In [3] by Binev, Dahmen, and DeVore and [26, 27] by Stevenson, the ideas in [23] are extended to prove optimal energy norm error estimates using piecewise linear elements for the Poisson equation.

4.2 Paper I: Convergence Rates for an Adaptive Dual Weighted Residual Finite Element Algorithm

This paper establishes basic convergence rates for a dual weighted residual finite element algorithm using isoparametric d-linear quadrilateral finite element approximation to functionals of solutions second order elliptic partial differential equations in open bounded domains of $\mathbb{R}^d$.

Section 2 describes an expansion of the error in the functional, based on (4.3), which is shown in Theorem 2.1 to be uniformly convergent as the mesh size tends to zero for smooth primal and dual solutions. The computable error density using localised averages of second order difference quotients of the primal and dual solutions, gives the leading order term of the error expansion; see Corollary 2.2. While the analysis is carried out for d-linear elements, a way to extend it to k:th order isoparametric quadrilateral finite elements is suggested.

The hanging node constraint implies that the refinement step in Algorithm 1 on page 9 must be modified to include a recursive marking of all neighbours that would otherwise violate the constraint. With that modification, the algorithm is analysed Section 3 following the outline in Chapter 3.

Section 4 presents numerical results for a simplified elasticity problem related to a problem with round corner of small radius introducing a small scale in the solution. The results show that the adaptive algorithm is more efficient for this problem than uniform refinements.

Paper I has entry [22] in the bibliography.

4.3 An Adaptive Algorithm for the Stopped Diffusion Problem

Here the objective is to compute adaptive approximations of an expected value

$$E[g(X(\tau),\tau)] \tag{4.4}$$

of a given function, $g : D \times [0,T] \to \mathbb{R}$, where the stochastic process X solves a stochastic differential equation (3.14) and τ is the first exit time

$$\tau := \inf\{0 < t : (X(t),t) \notin D \times (0,T)\}$$

from a given open domain $D \times (0,T) \subset \mathbb{R}^d \times (0,T)$. These so called barrier problems have applications in physics and finance, for example when pricing barrier options.

The expected value (4.4) is approximated by a sample average of $g(\overline{X}(\overline{\tau}),\overline{\tau})$, where $(\overline{X},\overline{\tau})$ is an Euler approximation (3.15) of (X,τ) using stochastic adaptive time steps. Like in (3.17) the global error using M realizations, splits into two parts

$$E[g(X(\tau),\tau)] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(\overline{\tau};\omega_j),\overline{\tau})$$
$$= \left(E[g(X(\tau),\tau) - g(\overline{X}(\overline{\tau}),\overline{\tau})]\right) + \left(E[g(\overline{X}(\overline{\tau}),\overline{\tau})] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(\overline{\tau};\omega_j),\overline{\tau})\right),$$

corresponding to time discretisation error and statistical error.

The main difficulty introduced by the barrier is that the continuous path may exit D even though a discrete approximate solution does not cross the boundary of D. The hitting of the boundary causes the time discretisation error for the Monte Carlo Euler method with N uniform time steps to be of order $N^{-1/2}$ instead of N^{-1} without stopping boundary in $\mathbb{R}^d \times [0,T)$; see [11] by Gobet.

In Mannella [16] and Jansons and Lythe [14] the order N^{-1}, using N uniform time steps is recovered by deciding in each time step whether the continuous path exits a half space domain by simulating a stochastic outcome. In [12] Gobet proves the convergence rate N^{-1} for a similar method, under suitable assumptions including smooth boundary. These methods are efficient when the exit probabilities can be computed accurately, for example when the domain is a half space or has a smooth boundary which can be approximated by tangent planes, but not for a boundary with corners.

4.4 Paper II: Adaptive Monte Carlo Algorithms for Stopped Diffusion

This paper, inspired by Petersen and Buchmann [25], uses an alternative approach to the uniform time step methods of [16], [14] and [12]. The time steps are chosen adaptively for each sample path, decreasing close to the barrier. The advantage of this method is that the exit probability need not be computed accurately, which is difficult for complicated domains D. Section 2 contains a derivation of an expansion of the error with computable leading order term, which is an extension of the corresponding error expansion in [29] for the approximation of $E[g(X(T))]$ for fixed T and $D = \mathbb{R}^d$. The extension uses a conditional probability to estimate the first exit time and it initialises

the dual solutions on the barrier with difference approximations of partial derivatives. Section 3 presents an adaptive algorithm based on the estimates in Section 2. Numerical results presented in Section 4 show that the algorithm recovers the time discretisation error of order N^{-1}, for N adaptive time steps.

Paper II has entry [8] in the bibliography.

4.5 Paper III: An Adaptive Algorithm for Ordinary, Stochastic and Partial Differential Equations

The results presented in the first two papers in this thesis follow previous work on the same adaptive algorithm in other precise settings of (3.1). For an ordinary differential equation (3.3), an error expansion (3.2) is derived by a variational principle in [19] and the convergence properties of the adaptive algorithm are studied in [18]. Weak approximation of an Itô stochastic differential equation is treated in [29, 20]. The application to the barrier problem in Paper II is an extension of the results in the latter two papers. Paper III provides an overview of the applications of the algorithm to both to ordinary, stochastic, and partial differential equations.

Paper III has entry [21] in the bibliography.

4.6 The computation of a stochastic phase-field model by coarse-graining Smoluchowski molecular dynamics

The modelling of nucleation and growth of crystal grains in a sub-cooled liquid involves both macroscopic and microscopic length scales. Diffusion and convection of heat occur on the macroscopic level but the process also depends on interface effects, where the width of the solid–liquid interface can extend over just a few inter-atomic distances. For the study of the continuum level time evolution of the phase transformations phase-field methods are widely used. A stochastic phase-field model for solidification

$$\frac{\partial}{\partial t}\left(c_V T + Lg(\phi)\right) = \nabla \cdot (\lambda \nabla T)\,, \tag{4.5a}$$

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (k_1 \nabla \phi) - k_2\Big(f'(\phi) + g'(\phi) k_3 (T_{\mathrm{M}} - T)\Big) + \text{noise}, \tag{4.5b}$$

gives a macroscopic description of the time evolution of two phases, here liquid and solid, co-existing in a material. Here T denotes the temperature,

c_V the specific heat at constant volume, L the latent heat of solidification, and T_{M} the melting point. The function ϕ is an order parameter that is used to distinguish between the two phases. The model functions $f(\phi)$ and $g(\phi)$ are constructed such that the system of partial differential equations, in the absence of noise, has two stable stationary solutions corresponding to pure single phase systems. An overview of the phase-field method applied to the modelling of solidification can be found in [4] by Boettinger, Warren, Beckermann, and Karma.

For a specific phase transition, the choice of model functions in the phase-field model, while motivated by thermodynamics, has to be made by the researcher based on his or her knowledge of the problem at hand. To avoid or complement this modelling on the macroscopic level, a research goal is to compute, if possible, the model functions and parameters, $f(\phi)$, $g(\phi)$, k_1, k_2, k_3, and the characteristics of the noise from computations on a microscale model; in [28] Szepessy suggests a method for doing this. To achieve this the phase-field, ϕ, must be defined in terms of quantities computable on the microscale. The underlying microscopic model is a stochastic molecular dynamics model, where the positions, X_i^t, of individual atoms follow the Smoluchowski dynamics

$$dX_i^t = -\nabla_{X_i} U(X^t)\, dt + \sqrt{2k_{\mathrm{B}}T}\, dW_i^t, \qquad i = 1, 2, \ldots, N,$$

for a given potential U which is a modelling choice on the microscopic level. Assuming that the total potential energy splits naturally as a sum of contributions from the individual atoms

$$U(X) = \sum_{i=1}^{N} m_i(X),$$

for example as in the case when the potential is defined by pairwise interactions between atoms, a microscopic phase-field variable is introduced as a smooth spatial average of the contributions through

$$m(x; X) = \sum_{i=1}^{N} m_i(X)\eta(x - X_i).$$

The smoothness depends on the choice of the mollifier η. A stochastic differential equation for a coarse-grained approximation ${m_{\mathrm{cg}}}^t$ of the microlevel phase-field is obtained by an optimal control argument.

Extensive research is being performed on methods to combine computations on different scales in material physics and other sciences; for the specific

case of modelling of dendritic solidification see for example the overview [13] by Hoyt, Asta, and Karma.

4.7 Paper IV: A Stochastic Phase-Field Model Computed From Coarse-Grained Molecular Dynamics

This paper presents results from numerical experiments on the method for computing stochastic phase-field models for phase transformations by coarse-graining molecular dynamics suggested in [28], as outlined above. On the microscopic level the simulated material is modelled by Smoluchowski molecular dynamics where the potential is defined by the so called exponential-6 pair potential with parameter values used for the simulation of Argon at high pressures. At such high pressures solid and liquid phases of the noble gas element can co-exist even above room temperature and the numerical experiments are made on the phase transformations between a solid crystal and a liquid.

The numerical experiments, consisting of molecular dynamics simulations on a two phase system at the melting point, are used to compute a double-well reaction term in the Allen–Cahn equation (4.5b) and a diffusion matrix describing the noise in the coarse-grained phase-field.

Paper IV has entry [32] in the bibliography.

Bibliography

[1] I. Babuška and M. Vogelius. Feedback and adaptive finite element solution of one-dimensional boundary value problems. *Numer. Math.*, 44(1): 75–102, 1984.

[2] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, pages 1–102, 2001.

[3] P. Binev, W. Dahmen, and R. DeVore. Adaptive finite element methods with convergence rates. *Numer. Math.*, 97(2):219–268, 2004.

[4] W. J. Boettinger, J. A. Warren, C. Beckermann, and A. Karma. Phase-field simulation of solidification. *Annu. Rev. Mater. Res.*, 32:163–194, 2002.

[5] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods*. Number 15 in Texts in Applied Mathematics. Springer–Verlag, New York, 1994.

[6] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations: convergence rates. *Math. Comp.*, 70(233): 25–75, 2001.

[7] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, pages 51–150, 1998.

[8] A. Dzougoutov, K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone. Adaptive monte carlo algorithms for stopped diffusion. In B. Engquist, P. Lötstedt, and O. Runborg, editors, *Multiscale Methods in Science and Engineering*, volume 44 of *Lecture Notes in Computational Science and Engineering*, pages 59–88. Springer–Verlag, Berlin Heidelberg, 2005. Appended as Paper II.

[9] W. Dörfler. A convergent adaptive algorithm for poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[10] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, pages 105–158, 1995.

[11] E. Gobet. Weak approximation of killed diffusion using euler schemes. *Stochastic Process. Appl.*, 87(2):167–197, 2000.

[12] E. Gobet. Euler schemes and half-space approximation for the simulation of diffusion in a domain. *ESAIM Probab. Stat.*, 5:261–297, 2001.

[13] J. J. Hoyt, M. Asta, and A. Kharma. Atomistic and continuum modeling of dendritic solidification. *Materials Science and Engineering R*, 41:121–163, 2003.

[14] K. M. Jansons and G. D. Lythe. Efficient numerical solution of stochastic differential equations using exponential timestepping. *J. Stat. Phys.*, 100 (5/6):1097–1109, 2000.

[15] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Studentlitteratur, Lund, 1987.

[16] R. Mannella. Absorbing boundaries and optimal stopping in a stochastic differential equation. *Phys. Lett. A*, 254(5):257–262, 1999.

[17] K.-S. Moon. *Adaptive Algorithms for Deterministic and Stochastic Differential Equations*. PhD thesis, KTH (Royal Institute of Technology, Stockholm), 2003. ISBN 91-7283-553-2.

[18] K.-S. Moon, A. Szepessy, R. Tempone, and G. E. Zouraris. Convergence rates for adaptive approximation of ordinary differential equations. *Numer. Math.*, 96:99–129, 2003.

[19] K.-S. Moon, A. Szepessy, R. Tempone, and G. E. Zouraris. A variational principle for adaptive approximation of ordinary differential equations. *Numer. Math.*, 96:131–152, 2003.

[20] K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris. Convergence rates for adaptive weak approximation of stochastic differential equations. *Stoch. Anal. Appl.*, 23(3):511–558, 2005.

[21] K-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone. An adaptive algorithm for ordinary, stochastic and partial differential equations. In Z.-C. Shi, Z. Chen, T. Tang, and D. Yu, editors, *Recent Advances in Adaptive Computation*, volume 383 of *Contemp. Math.*, pages 325–343. American Mathematical Society, Providence, 2005. Appended as Paper III.

[22] K-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone. Convergence rates for an adaptive dual weighted residual finite element algorithm. *BIT*, 46:367–407, 2006. Appended as Paper I.

[23] P. Morin, R. H. Nochetto, and K. G. Siebert. Convergence of adaptive finite element methods. *SIAM Rev.*, 44(4):631–658, 2002. ISSN 0036-1445.

[24] E. Novak. On the power of adaption. *J. Complexity*, 12:199–237, 1996.

[25] W. P. Petersen and F. M. Buchmann. Solving dirichlet problems numerically using the feynman-kac representation. *BIT*, 43(3):519–540, 2003.

[26] R. P. Stevenson. An optimal adaptive finite element method. *SIAM J. Numer. Anal.*, 42(5):2188–2217, 2004. ISSN 0036-1429.

[27] R. P. Stevenson. Optimality of a standard adaptive finite element method. *Found. Comput. Math.*, 7(2):245–269, 2007.

[28] A. Szepessy. Atomistic and continuum models for phase change dynamics. In M. Sanz-Solé, J. Soria, J. L. Varona, and J. Verdera, editors, *Proceedings of the International Congress of Mathematicians, Madrid, August 22–30, 2006, Volume III*, volume III, pages 1563–1582. EMS Ph, 2007.

[29] A. Szepessy, R. Tempone, and G. E. Zouraris. Adaptive weak approximation of stochastic differential equations. *Comm. Pure Appl. Math.*, 54: 1169–1214, 2001.

[30] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8:483–509, 1990.

[31] E. von Schwerin. *Convergence Rates of Adaptive Algorithms for Stochastic and Partial Differential Equations*. Licentiate thesis, KTH (Royal Institute of Technology, Stockholm), 2005. ISBN 91-7283-959-7.

[32] E. von Schwerin. A stochastic phase-field model computed from coarse-grained molecular dynamics. Technical report, KTH (Royal Institute of Technology, Stockholm), 2007. Appended as Paper IV.

Paper I

CONVERGENCE RATES FOR AN ADAPTIVE DUAL WEIGHTED RESIDUAL FINITE ELEMENT ALGORITHM

KYOUNG-SOOK MOON[1] and ERIK VON SCHWERIN[2] and
ANDERS SZEPESSY[3] and RAÚL TEMPONE[4] *

[1] *Department of Mathematics, University of Maryland, Collage Park, MD 20742, USA.*
email: moon@math.umd.edu

[2] *School of Computer Science and Communication, KTH, S-100 44 Stockholm, Sweden.*
email: schwerin@nada.kth.se

[3] *School of Computational Science, Florida State University,*
Dirac Science Library, Tallahassee, FL 32306-4120, USA,
Department of Mathematics, 208 James J. Love Building, Florida State University,
Tallahassee, FL 32306-4510, USA
email: rtempone@scs.fsu.edu

[4] *Department of Mathematics, KTH, S-100 44 Stockholm, Sweden.*
email: szepessy@nada.kth.se

Abstract.

Basic convergence rates are established for an adaptive algorithm based on the dual weighted residual error representation,

$$\text{error} = \sum_{\text{elements}} \text{error density} \times \text{mesh size}^{2+d},$$

applied to isoparametric d-linear quadrilateral finite element approximation of functionals of multi scale solutions to second order elliptic partial differential equations in bounded domains of $\mathbb{R}^d$. In contrast to the usual aim to derive an a posteriori error estimate, this work derives, as the mesh size tends to zero, a uniformly convergent error expansion for the error density, with computable leading order term. It is shown that the optimal adaptive isotropic mesh uses a number of elements proportional to the $d/2$ power of the $L^{\frac{d}{d+2}}$ quasi-norm of the error density; the same error for approximation with a uniform mesh requires a number of elements proportional to the $d/2$ power of the larger L^1 norm of the same error density. A point is that this measure recognizes different convergence rates for multi scale problems, although the convergence order may be the same. The main result is a proof that the adaptive algorithm based on successive subdivisions of elements reduces the maximal error indicator with a factor or stops with the error asymptotically bounded by the tolerance using the optimal number of elements, up to a problem independent factor. An important step is to prove uniform convergence of the expansion for the error density, which is based on localized averages of second order difference quotients of the primal and dual finite element solutions. The averages are used since the difference quotients themselves do not

*This work was partially supported by the Swedish Research Council grants 2002-6285 and 2002-4961, by a Swedish Foundation for Strategic Research grant and by the European network HYKE, funded by the EC as contract HPRN-CT-2002-00282.

converge pointwise for adapted meshes. The proof uses weak convergence techniques with a symmetrizer for the second order difference quotients and a splitting of the error into a dominating contribution, from elements with no hanging nodes or edges on the initial mesh, and a remaining asymptotically negligible part. Numerical experiments for an elasticity problem with a crack and different variants of the averages show that the algorithm is useful in practice also for relatively large tolerances, much larger than the small tolerances needed to theoretically guarantee that the algorithm works well.

AMS subject classification (2000): 65N12, 65N30, 65N50.

Key words: adaptive methods, mesh refinement algorithm, a posteriori error estimate, computational complexity, finite elements.

Contents

1 Introduction to Adaptive Finite Element Algorithms

This work analyzes the convergence rate of an adaptive finite element algorithm to approximate functionals of multi scale solutions, $u : \Omega \to \mathbb{R}$, of the second order elliptic partial differential equation

$$-\mathrm{div}(a\nabla u) = f \tag{1.1}$$

in a given open bounded domain $\Omega \subset \mathbb{R}^d$ with Dirichlet boundary condition $u|_{\partial\Omega} = 0$. The paper presents the linear two dimensional case, $d = 2$, with data a and f, where a is a symmetric positive definite matrix $a : \Omega \to \mathbb{R}^{d\times d}$ and $f : \Omega \to \mathbb{R}$. The results directly generalize to higher dimensions, $d > 2$. The adaptivity is based only on the Galerkin approximation error, neglecting for instance quadrature and data error. It is easy to extend the study to some nonlinear problems, see Remark 2.4. This work uses some simplifying properties of

linear quadrilateral finite element approximation; an extension to approximation by piecewise polynomials of degree $k > 1$ is discussed in Section 2.4.

There are numerous studies on error estimates for adaptive finite element methods applied to partial differential equations, e.g., [1], [4, 5], [7], [9, 10], [18, 19], [20, 21], [25], and some work on the convergence of adaptive algorithms [6], [16], [26]. However, important numerical complexity theory, on how convergence rates for adaptive finite element algorithms depend on the computational work, is not as well developed, but there are recent contributions. The work [15] shows the efficiency of adaptive approximation of functions, including wavelet expansions, based on smoothness conditions in Besov spaces. Inspired by this approximation result, first the work [13] proves that a wavelet-based adaptive N-term approximation algorithm produces a solution with asymptotically optimal error in the energy norm for linear coercive elliptic problems. Then [11, 27] extend the ideas of [26] to prove similar optimal error estimates in the energy norm for piecewise linear elements applied to the Poisson equation. The modification includes a somewhat complicated coarsening step in the adaptive algorithm to obtain bounds on the work.

Our work focuses on isoparametric d-linear quadrilateral finite element approximation of functionals of the solution to (1.1), inspired by [10], [18] and [19], using the residual and dual weight functions to estimate the error. Section 2 uses the dual weighted residual method to derive, as the mesh size tends to zero, a uniformly convergent expansion of this error, with computable leading order term $\sum_K \bar{\rho}_K h_K^{2+d}$ where h_K is the mesh size and $\bar{\rho}_K$ is the error density for element K. This is in contrast to the usual aim to derive an a posteriori error estimate for adaptive refinements. Section 3 applies this expansion to prove convergence rates, depending on the number of degrees of freedom, for an adaptive finite element algorithm. The simpler problem to approximate a given function is studied in [2] where adaptive refinements are based on an error expansion for approximation, in function spaces of rectangular bi-p elements.

What is the right measure of convergence rates for adaptive finite element algorithms applied to (1.1)? For a constant mesh size h, approximations with error $\mathcal{O}(h^p)$ require computational work with $\mathcal{O}(1/h^d)$ operations, using optimal multigrid solvers. The accuracy $\epsilon \equiv \mathcal{O}(h^p)$ is hence asymptotically determined by the number of elements $N = \mathcal{O}(1/h^d) = \mathcal{O}(\epsilon^{-d/p})$. This simple asymptotic complexity estimate, $\mathcal{O}(\epsilon^{-d/p})$, is one of the most basic and well used numerical analysis measures of the performance of approximations. Analogously, for adaptive methods, it seems natural to study the approximation error and the associated work, proportional to the number of elements, as the tolerance parameter tends to zero. For the second order accurate piecewise linear finite elements on a uniform mesh, the number of elements needed to reach a given approximation error turns out to be proportional to the $d/2$ power of the L^1-norm of the error density; this work shows that the smallest number of isotropic elements in an adaptive mesh is proportional to the $d/2$ power of the smaller $L^{\frac{d}{d+2}}$ quasi-norm of the same error density. These norms of the error density are therefore good measures of the convergence rates and define our optimal number

of elements, explained in Section 3. A simple elasticity problem with a round corner of radius ϵ shows in Section 4 that the error density is $\rho = \frac{\mathcal{O}(1)}{(r+\epsilon)^3}$ with the polar coordinate $r > 0$ in $\mathbb{R}^2$, so that $\|\rho\|_{L^{1/2}} = \mathcal{O}(1)$ and $\|\rho\|_{L^1} = \frac{\mathcal{O}(1)}{\epsilon}$. Therefore the optimal mesh of adaptive elements becomes $\frac{\mathcal{O}(1)}{\mathrm{TOL}}$ for error TOL while the number of uniform elements is $\frac{\mathcal{O}(1)}{\mathrm{TOL}}\frac{1}{\epsilon}$ for the same error. A point is that this measure distinguishes between different convergence rates for multi scale problems, although the convergence order may be the same.

Section 3 constructs a simple algorithm which, given an error tolerance, TOL, subdivides the elements with error indicators, $|\bar{\rho}_K|h_K^{2+d}$, greater than TOL/N and stops if all N elements have sufficiently small error indicators. In particular the algorithm has no coarsening step. Theorems 3.1, 3.3 and 3.4 in Section 3 prove that each refinement level of this adaptive algorithm decreases the maximal error indicator with a factor, less than 1, or stops with an error asymptotically bounded by TOL and with asymptotically optimal number of elements, N, in the finest mesh, up to a problem independent factor. The total number of elements, including all refinement levels, can be bounded by $\mathcal{O}(N)$, provided the tolerance in each refinement level decreases by a constant factor, see Theorem 3.5. Varying tolerance has the drawback that the final stopping tolerance is not a priori known; on the other hand, with constant tolerance, the total number of elements including all levels is bounded by the larger $\mathcal{O}(N \log N)$.

Chapter one in [23] describes the relation of the adaptive convergence rate result in Section 3 to Bakhvalov's and Smolyak's complexity result [8], which shows that, using a fixed number of function evaluations, there is for each adaptive method a non-adaptive method which has as small maximal error as the adaptive method, for approximation of linear functionals in a convex symmetric subset of a normed linear function space. The difference in the assumptions, which favors adaptive approximations, is that in Section 3 the discretization method is fixed and only the mesh is varying and that the performance of the algorithm is characterized by non-convex function sets, e.g. functions with the $L^{\frac{d}{d+2}}$ quasi-norm bounded by a constant.

The reports [23] and [22, 28] introduced adaptive algorithms for weak approximation of ordinary and stochastic differential equations, respectively. Their extension to partial differential equations here is partly straightforward except for the pointwise convergence of the error density and a hanging node constraint; to prove convergence of the error density for approximation of ordinary differential equations is simple, while the corresponding convergence result for partial differential equations is hard, requiring unions of structured adapted meshes and detailed analysis special to bilinear finite elements. In fact the work [10] writes "The strategies for mesh adaption is largely based on heuristic grounds. One hard open problem is the rigorous proof of the convergence of local residual terms and weights to certain 'limits' ". Note that such pointwise convergence of the error density, based on second order difference quotients, is well known for structured uniform meshes; however Remark 2.1 below shows by an example that second order difference quotients of smooth functions do not in general

converge pointwise for adapted meshes. To prove convergence of the second order difference quotients, in the error density, our proof instead uses localized averages with a symmetrizer and a splitting of the error into a dominating contribution, from elements with no hanging nodes or edges on the initial mesh, and a remaining asymptotically negligible part.

2 Convergence of the Error Density

2.1 An Error Representation

The finite element approximation u_h, of u in (1.1), is based on the standard variational formulation in the function space V_h of continuous piecewise isoparametric bilinear quadrilateral functions in $H_0^1(\Omega)$, using an adaptive quadrilateral mesh with hanging nodes cf. [10]. The Sobolev space $H_0^1(\Omega)$ is the usual Hilbert space of functions on Ω, vanishing on $\partial\Omega$, with bounded first derivatives in $L^2(\Omega)$. Let $\mathcal{T}$ denote the set of convex quadrilaterals K and let h_K be the local mesh size, i.e. the length of the longest edge of K. Let $\mathcal{T}_e$ denote all interior edges in $\mathcal{T}$. The aim is to compute a linear functional value $(u, F) := \int_\Omega uF dx$ for a given function $F \in L^2(\Omega)$. Let $(\cdot,\cdot)$ denote the duality pairing on $H^{-1} \times H_0^1$, which reduces to the usual inner product in $L^2(\Omega)$ on $L^2 \times L^2$. Define the residual $\mathcal{R}(v) = -\mathrm{div}(a\nabla v) - f$ as a distribution in $H^{-1}(\Omega)$ for $v \in H_0^1(\Omega)$. Then the variational problems for $u \in H_0^1(\Omega)$ and $u_h \in V_h$ are

$$\begin{aligned} (\mathcal{R}(u), v) &= 0, \quad \forall v \in H_0^1(\Omega), \\ (\mathcal{R}(u_h), v) &= 0, \quad \forall v \in V_h. \end{aligned} \tag{2.1}$$

Define the dual function $\varphi \in H_0^1(\Omega)$ by

$$(a\nabla v, \nabla\varphi) = (F, v), \quad \forall v \in H_0^1(\Omega), \tag{2.2}$$

to obtain

$$(u - u_h, F) = (a\nabla(u - u_h), \nabla\varphi) = (\mathcal{R}(u_h), -\varphi).$$

The orthogonality (2.1) implies $(u - u_h, F) = (\mathcal{R}(u_h), v - \varphi)$ for all $v \in V_h$, and the choice $v = \pi\varphi \in V_h$, where π is the nodal interpolant on V_h, yields the dual weighted error representation

$$(u - u_h, F) = (\mathcal{R}(u_h), \pi\varphi - \varphi)\,. \tag{2.3}$$

The global error (2.3) can be split into residual parts supported in the interior

of quadrilaterals and on their edges

$$
\begin{aligned}
(u-u_h,F) &= \sum_{K\in\mathcal{T}}\int_K(-\mathrm{div}(a\nabla u_h)-f)(\pi\varphi-\varphi)dx\\
&\quad+\sum_{K\in\mathcal{T}}\int_{\partial K} n\cdot a\nabla u_h\,(\pi\varphi-\varphi)\,ds\\
&= \sum_{K\in\mathcal{T}}\int_K(-\mathrm{div}(a\nabla u_h)-f)(\pi\varphi-\varphi)dx\\
&\quad-\sum_{e\in\mathcal{T}_e}\int_e n\cdot a[\nabla u_h]\,(\pi\varphi-\varphi)\,ds,
\end{aligned}
\tag{2.4}
$$

where n is the outward normal to the element K, on ∂K, and on the edge e the symbol n denotes one of the normals (it does not matter which) with $[w](x) := \lim_{s\to 0+}\big(w(x+sn)-w(x-sn)\big)$. The continuity of u_h on Ω implies

$$
[\nabla u_h] = n[n\cdot\nabla u_h] =: \; n[\frac{\partial u_h}{\partial n}].
$$

2.2 Approximation of the Error Density

The goal in this section is to derive a computable approximation of the error representation (2.4). An adaptive algorithm providing a reliable error bound and efficient use of the degrees of freedom can use an error expansion

$$
(u-u_h,F)\simeq\sum_K\bar\rho_K h_K^{2+d} \tag{2.5}
$$

where the error density $\bar\rho$ is essentially independent of the mesh size and the asymptotic error density is used to find the optimal mesh.

Precise analysis of the adaptive algorithms for ordinary [23] and stochastic differential equations [22, 24] was obtained by proving convergence of an error density. This work generalizes those adaptive algorithms to partial differential equations. The main new ingredient is to prove convergence of the error density. For general meshes this convergence of the error density $\bar\rho$ does not hold, since the orientation of the elements varies. The purpose here is to analyze the asymptotic behavior of the error density $\bar\rho$ for adaptive refinements, with general quadrilateral initial meshes: successive division of reference square elements into four similar squares generates hanging node meshes consisting of unions of structured adapted sub meshes, where the domain of each structured sub mesh is an initial element; viewed in the initial reference element the structured adaptive mesh is an adaptive hanging node mesh with square elements. We restrict the study to such unions of structured adaptive hanging node meshes. The use of quadrilaterals can directly be extended to higher space dimension using tensor reference elements. Other refinements using e.g. subdivision of a simplex, in three and higher dimensions, cf. [17], generate new edges which are not parallel to the old and would require additional analysis.

To define the error density $\bar{\rho}$ we use the isoparametric bilinear quadrilateral finite element approximation $\varphi_h \in V_h$, of the dual function φ in (2.2), defined by

$$(a\nabla v, \nabla\varphi_h) = (F, v), \quad \forall v \in V_h. \tag{2.6}$$

Then one would like to use second order difference quotients of u_h and φ_h to approximate the error density. On uniform meshes the second difference quotients of u_h and φ_h converge and the proof uses the translation invariance of the mesh. However, non uniform adapted meshes are not translation invariant and Remark 2.1 shows that a second order difference quotient of the discrete functions u_h or φ_h does not in general converge pointwise to the corresponding second order derivatives of u or φ, respectively. We solve this problem by using instead localized averages of second order difference quotients and a splitting of the error into a dominating contribution from elements with no hanging nodes or edges on the initial mesh and a remaining asymptotically negligible part.

Consider a multiscale problem and an adaptive algorithm seeking to equidistribute the error indicators $\bar{\rho}_h h_K^4$ in an error expansion (2.5), by successively dividing elements into four, using general regular quadrilateral initial meshes. The notion of multiscale means here that the error density $\bar{\rho}$ will be uniformly bounded, although it may be very large, and essentially independent of the mesh size, as shown in Theorem 2.1. There is a smooth mapping of each initial element to a square, so that the refined initial element is mapped to a square hanging node mesh. Let $\mathcal{T}_I$ denote the subset of elements with an edge on the initial mesh. Asymptotically as the tolerance tends to zero, the total area, $\int_{\mathcal{T}_I} dx$, of the elements with edges in common with the initial mesh tends to zero as the maximal edge length, h_{max}, tends to zero. Assume that all second order difference quotients of u_h are uniformly bounded. The approximation error has then by (2.4) the bound

$$|(u - u_h, F)| = \sum_{K\in\mathcal{T}} \mathcal{O}(h_K^4).$$

Therefore the contribution to this sum from the elements with an edge on the initial mesh is bounded by

$$\sum_{K\in\mathcal{T}_I} \mathcal{O}(h_K^4) = o(h_{max}^2)$$

and hence these elements will give an asymptotically negligible contribution to the total error, as $h_{max} \to 0+$. We show in Theorem 2.1 that the error density has a precise expansion using that the isoparametric bilinear coordinate transformation $X^{-1} : [0,1]^2 \to K_I$ maps the square and the square hanging node mesh to the initial element K_I and its refined hanging node quadrilateral mesh.

Let us now study the transformation of the variational formulation under such a mapping $X : K_I \to [0,1]^2$

$$\sum_{ij} \int_{K_I} (a_{ij} \frac{\partial u_h}{\partial x_j} \frac{\partial v}{\partial x_i} - fv) dx = \int_{[0,1]^2} (aX' u_h' \cdot X' v' - fv) J dx'$$

where X' is the Jacobian of X and J is the Jacobian determinant. Here we abuse the notation by writing v instead of $(v \circ X^{-1})$ and similarly for a, u_h, and f, for $x \in K_I$. Moreover, we write $v' = \frac{\partial v}{\partial x_i'}$ instead of $\frac{\partial (v \circ X^{-1})}{\partial x_i'}$.

Therefore the variational equation in the transformed coordinates, x', takes the same form with a and f replaced by $a^* := J(X')^t a X'$ and $f^* := Jf$, respectively. Note that a^* and f^* are as smooth on $X(K_I)$ as the functions a and f are on K_I. To avoid messy notation, we will not always use the prime notation for coordinates obviously in the reference elements; we will also avoid notation for the dependence of X on the initial element K_I and assume that we for a point $x \in \Omega$ choose the mapping X that corresponds to the initial element K_I which contains x. We will use the set of transformed elements

$$\mathcal{T}' \coloneqq \{X(K) : \ K \in \mathcal{T}\}.$$

To define the approximate error density, $\bar{\rho}$, we will use averages of second difference as follows. Consider a function w which is defined on a discretization of an interval $[0, \hat{a}]$ with nodes $\{x_j : \ j = 0, \ldots, \bar{N} + 1\} =: \bar{\mathcal{N}}$, where $x_0 = 0$ and $x_{\bar{N}+1} = \hat{a}$. Let $h_+ := x_{i+1} - x_i$ and $h_- := x_i - x_{i-1}$ denote two consecutive edge sizes. Then define the average mesh size $\bar{h}$ and the difference quotients

$$\begin{aligned} \bar{h}_i &:= \frac{h_+ + h_-}{2} = \frac{x_{i+1} - x_{i-1}}{2}, \\ Dw(x_i) &:= \frac{w(x_i + h_+) - w(x_i)}{h_+}, \\ D^2 w(x_i) &:= \frac{1}{\bar{h}_i} \left(\frac{w(x_i + h_+) - w(x_i)}{h_+} - \frac{w(x_i) - w(x_i - h_-)}{h_-} \right). \end{aligned} \tag{2.7}$$

The localized average is based on a non negative function, $\psi^{x_j} : \mathcal{N} \to \mathbb{R}$, $j = 1, \ldots, \bar{N}$, with $\psi_i^j := \psi^{x_j}(x_i)$ and a positive parameter α, measuring the width of the average, where the averaging function satisfies

$$\psi_\cdot^j \geq 0, \tag{2.8}$$

$$\|\bar{h}_\cdot D^2 \psi_\cdot^j\|_{\ell^1} = \mathcal{O}(\alpha^{-2}), \tag{2.9}$$

$$\psi_0^j + \psi_{\bar{N}+1}^j = \mathcal{O}(\alpha^{-1}), \tag{2.10}$$

$$\psi_0^j = \psi_1^j, \quad \psi_{\bar{N}}^j = \psi_{\bar{N}+1}^j, \tag{2.11}$$

and the weak convergence for any $v \in \mathcal{C}^1([0, \hat{a}])$

$$\Big| \sum_{i=1}^{\bar{N}} \psi_i^j \bar{h}_i v(x_i) - v(x_j) \Big| = \mathcal{O}(\alpha). \tag{2.12}$$

Define the average difference

$$\overline{D^2 w}(x_j) := \sum_{i=1}^{\bar{N}} \psi_i^j \bar{h}_i D^2 w(x_i). \tag{2.13}$$

Section 2.3 presents an efficient method to compute such averages $Y \in \mathbb{R}^{\bar{N}+2}$ of $D^2 w \in \mathbb{R}^{\bar{N}}$ based on the equation

$$Y_i - \alpha^2 D^2 Y_i = D^2 w_i, \quad i = 1, \ldots, \bar{N}$$

with homogeneous Neumann boundary conditions, $Y_0 = Y_1,\ Y_{\bar{N}} = Y_{\bar{N}+1}$. Section 4 reports numerical results with different alternative averages, including the fast nearest neighbor variant. The convergence proof requires α to be sufficiently large, slightly larger than the pointwise errors $|u - u_h|,\ |\nabla(u - u_h)|,\ |\varphi - \varphi_h|$ and $|\nabla(\varphi - \varphi_h)|$.

Let us define $\bar{h} D_i^2 w$ as the difference quotients $\bar{h} D^2 w$, in (2.7), with respect to the x_i' reference directions $i = 1, 2$, respectively, and analogously for $D_i w$. The the weight function ψ^j centered at a nodal point $x_j' \in [0,1]^2$ then yields the averaged values $\overline{D_i^2 w}$ by (2.13). The approximate error density, $\bar{\rho}$, in the transformed coordinates is now defined by

$$\overline{\rho}_K \coloneqq \frac{1}{48} \sum_{j=1}^{4} \left(a_{11}^* \overline{D_1^2 u_h}\ \overline{D_1^2 \varphi_h} + a_{22}^* \overline{D_2^2 u_h}\ \overline{D_2^2 \varphi_h} \right)(x_j^K) \tag{2.14}$$

where $x_1^K, x_2^K, x_3^K, x_4^K$ are the four corners of the square $K \in \mathcal{T}'$ illustrated in Figure 2.1. To derive this error density and make the ideas as transparent as possible we focus on a simple but general case based on solutions u and φ with multiple scales and $\mathcal{C}^3$ regularity. Let $\mathcal{T}_H$ denote the subset of elements with hanging nodes in neighbors and let

$$\bar{\mathcal{T}}_H \coloneqq \bigcup_{K \in \mathcal{T}_H} K.$$

Our proof uses the assumption that the area $\int_{\bar{\mathcal{T}}_H} dx$ tends to zero asymptotically. This can be motivated by the fact that the error density is uniformly bounded and by (2.8-2.12) changes very little in a distance of order α, where $1 \gg \alpha \gg h$; therefore a mesh size change from h to $h/2$ requires the error density to approximately change by a factor of 16, for equidistributed error indicators $\bar{\rho}_K h_K^4$. This change of $\bar{\rho}$ can not happen within a distance α and consequently the quotient of the number of hanging node elements and the total number of elements is at most $\mathcal{O}(h/\alpha)$ and the total area of hanging node elements becomes at most $\int_{\bar{\mathcal{T}}_H} dx = \mathcal{O}(h_{max}/\alpha)$.

We will use an adaptive algorithm where the error indicators become approximately equidistributed: we prove in Section 3 that

$$c \frac{\mathrm{TOL}}{N} \leq \hat{\rho}_K h_K^4 \leq S_1 \frac{\mathrm{TOL}}{N}, \tag{2.15}$$

for problem independent constants c and S_1, the number N of elements in the final mesh and where

$$\hat{\rho} \coloneqq \max(|\bar{\rho}|, \delta) \tag{2.16}$$

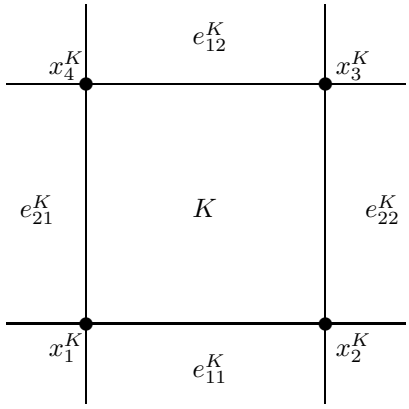


Figure 2.1: Corners x_j^K and edges e_{ij}^K of a square $K \in \mathcal{T}'$

is a positive approximation of the error density depending on a positive parameter δ tending to zero as the error tolerance, TOL, tends to zero. A consequence of u and φ in $\mathcal{C}^3$ and the lower bound of the error indicator $\hat{\rho}$ is that the quotient of the maximal, h_{max}, and minimal, h_{min}, mesh sizes becomes

$$\frac{h_{max}}{h_{min}} \leq C\Big(\frac{\hat{\rho}_{max}}{\hat{\rho}_{min}}\Big)^{1/4} \leq \frac{C_u}{\delta^{1/4}} =: \sqrt{C_\delta}, \tag{2.17}$$

where $C = (S_1/c)^{1/4}$ is independent of TOL, u and φ; while the constant C_u is independent of TOL but depends on u and φ. Section 4 shows an example where $C_u = \mathcal{O}(\epsilon^{-3/4})$ for an elasticity problem with a crack of radius ϵ. The upper bound in (2.15) follows directly from the stopping rule of the algorithm (3.11)-(3.12) and the lower bound is proved from the refinement criterion of a parent error indicator; with the notation of Section 3 we have $c = s_1/(\hat{C}2^{d+2})$ from (3.33).

Let $W^{1,\infty}(\Omega)$ denote the usual Sobolev space of functions with bounded first order derivatives in $L^\infty(\Omega)$ and let h_{max} be the maximal edge length in the mesh of V_h. Sometimes we drop the set and write $W^{1,\infty}$ and L^∞ also for functions in the reference set $[0,1]^2$. To prove convergence of the error density, $\bar{\rho}$, we will use the assumption that for some $\gamma \in (0,1]$

$$\begin{aligned} \|u - u_h\|_{W^{1,\infty}(\Omega)} + \|\varphi - \varphi_h\|_{W^{1,\infty}(\Omega)} &= \mathcal{O}(C_\delta h_{max}), \\ \|u - u_h\|_{L^\infty(\Omega)} + \|\varphi - \varphi_h\|_{L^\infty(\Omega)} &= \mathcal{O}(h_{max}^{2\gamma}). \end{aligned} \tag{2.18}$$

The work [12] proves such estimates for finite element approximations of the coercive linear problems (1.1) and (2.2), with piecewise isoparametric bilinear

quadrilateral elements and quasi uniform meshes, provided $u, \varphi \in \mathcal{C}^2(\bar{\Omega})$, see [14] for nonlinear problems. Our meshes are quasi uniform with a possible large constant C_u if $|\bar{\rho}|$ is bounded away from zero. Including the case with $|\bar{\rho}|$ close to zero needs the use of the modified error density, $\hat{\rho}$ defined in equation (2.16), and yields the bound (2.17) of the quotient of the maximal and minimal mesh size in the final mesh and such a bound changes an $\mathcal{O}(h_{max})$ estimate of the right hand side in (2.18) to $\mathcal{O}(C_\delta h_{max})$ with $C_\delta = C_u^2/\delta^{1/2}$.

To have solutions u and φ in $\mathcal{C}^3(\bar{\Omega})$, with general data f and F, is not compatible with Ω being a domain with corners. Therefore the constraint of the mesh, required by V_h, yields restrictions on the data. One solution is to treat only the approximation in an interior domain $\Omega_h \subset \Omega$ and let u_h and φ_h have the boundary values $u|_{\partial\Omega_h}$ and $\varphi|_{\partial\Omega_h}$, respectively. Another solution is to let Ω be a polygonal domain and use constraints on f and F; i.e. in a square $[0,1]^2$, let f and F satisfy $\sum_{n,m=1}^{\infty}(n^2+m^2)^4 f_{nm}^2 < \infty$, with the Fourier coefficients $f_{nm} = \int_0^1 f(x,y)\sin(\pi x)\sin(\pi y)dxdy$, which by Sobolev's inequality implies $u,\ \varphi \in \mathcal{C}^3([0,1]^2)$.

Our main result in this section is

THEOREM 2.1. *Assume that $a \in \mathcal{C}^1(\bar{\Omega})$ and that the solutions $u \in \mathcal{C}^3(\bar{\Omega})$, $\varphi \in \mathcal{C}^3(\bar{\Omega})$ of* (1.1) *and* (2.2)*, respectively, are for some $\gamma \in (0,1]$ approximated uniformly with error satisfying* (2.18) *using piecewise isoparametric bilinear quadrilateral elements and a refined mesh, with at most one hanging node per edge, obtained by successively dividing the reference square elements into four similar squares. Assume also that the total area of the elements with a hanging node on a neighbor or with an edge on the initial mesh is asymptotically zero:*

$$\int_{\bar{\mathcal{T}}_H \cup \bar{\mathcal{T}}_I} dx = o(1), \qquad \text{as } h_{max} \to 0^+. \tag{2.19}$$

Then the global error (2.3) *has the error expansion*

$$(u-u_h, F) = \sum_{K\in\mathcal{T}'} \left(\overline{\rho}_K + \mathcal{O}(h_{max}^\gamma/\alpha + \alpha)\right) h_K^4 + \mathcal{O}(C_\delta h_{max}) \int_{\bar{\mathcal{T}}_H \cup \bar{\mathcal{T}}_I} h_K dx \tag{2.20}$$

with uniformly convergent computable error density $\bar{\rho}$, defined by (2.14) *and* (2.7)-(2.13) *with*

$$\frac{h_{max}^{-\gamma}}{\alpha} = o(1), \tag{2.21}$$

as $h_{max} \to 0^+$, satisfying

$$\bar{\rho} = \widetilde{\rho} + \mathcal{O}(h_{max}^\gamma/\alpha + \alpha), \tag{2.22}$$

where

$$\widetilde{\rho} := \frac{1}{12}\left(a_{11}^* \frac{\partial^2 u}{\partial x_1^2}\frac{\partial^2 \varphi}{\partial x_1^2} + a_{22}^* \frac{\partial^2 u}{\partial x_2^2}\frac{\partial^2 \varphi}{\partial x_2^2}\right)$$

is evaluated in the transformed coordinates on $[0,1]^2$. The computable error density $\bar{\rho}$ yields the leading order error contribution in the following sense

COROLLARY 2.2. *Suppose that the assumptions in Theorem 2.1 hold and in addition that the error indicators satisfy* (2.15)*, which is verified for the adaptive algorithm in Section 3. Then the lower bound for the error density,* $\delta \to 0+$*, as* $\mathrm{TOL} \to 0+$*, can be chosen such that* $\sum_K \bar{\rho}_K h_K^4$ *is the leading order error contribution in the sense*

$$(u - u_h, F) = \sum_{K \in \mathcal{T}'} \bar{\rho}_K h_K^4 + o(\mathrm{TOL}) \tag{2.23}$$

where

$$\sum_{K \in \mathcal{T}'} \bar{\rho}_K h_K^4 \leq S_1 \mathrm{TOL}, \tag{2.24}$$

and, provided $\|\tilde{\rho}\|_{L^{1/2}(\Omega)} > 0$*,*

$$\liminf_{\mathrm{TOL} \to 0+} \mathrm{TOL}^{-1} \sum_{K \in \mathcal{T}'} |\bar{\rho}_K| h_K^4 \geq c. \tag{2.25}$$

Observe that the regularity condition $u \in C^3(\bar{\Omega})$ and the assumption that $\|\tilde{\rho}\|_{L^{1/2}(\Omega)} > 0$ exclude the very special case where the exact solution u or φ is in V_h. Note also that the convergence of $\bar{\rho}$ is uniform while the convergence of $\check{\rho}$, defined by $(u - u_h, F) = \sum_{K \in \mathcal{T}'} \check{\rho} h_K^4$, is in $L^1(\Omega)$ by assumption (2.19). It is important to notice that our restriction of the data, required by $u,\ \varphi \in \mathcal{C}^3(\bar{\Omega})$, includes examples with substantial adaptive gain. Section 3 shows that the optimal number of adaptive elements is $N^{opt} = \mathrm{TOL}^{-1} \|\bar{\rho}\|_{L^{\frac{d}{d+2}}}^{d/2}$, while the number of uniform elements becomes $N^{uni} = \mathrm{TOL}^{-1} \|\bar{\rho}\|_{L^1}^{d/2}$ to achieve the same error TOL. Although $u,\ \varphi \in \mathcal{C}^3(\bar{\Omega})$ their norms in these spaces may be large so that $\|\bar{\rho}\|_{L^{\frac{d}{d+2}}} \ll \|\bar{\rho}\|_{L^1}$. Section 4 shows such an example with $\frac{\|\rho\|_{L^1}}{\|\rho\|_{L^{1/2}}} = \mathcal{O}(\epsilon^{-1})$ for an elasticity problem, with ϵ related to the radius of a round corner. Note also that the error term $\mathcal{O}(h_{max}^{\gamma}/\alpha + \alpha)$ can be expressed in terms of the tolerance, by Lemma 3.2, so that the error term becomes negligible for sufficiently small tolerances. In the example in Section 4 this relative error is of the order $\mathrm{TOL}^{1/4}/\epsilon$.

The proof of the theorem is based on the uniform convergence of the averaged second differences $\overline{D^2 u_h}$ and $\overline{D^2 \varphi_h}$, derived in Lemma 2.4, and the convergence of $h^{-2}(\pi\varphi - \varphi)$ established in Lemma 2.5. The pointwise convergence of the averaged differences is essentially a consequence of the observation that the difference operator $\bar{h} D^2$ is symmetric, which is proved in Lemma 2.3, and weak convergence. We first prove the lemmas and then the theorem.

Theorem 2.1 can be generalized to some cases where the optimal adaptive isotropic mesh uses a number of elements proportional to $(\|\widetilde{\rho}\|_{L^{\frac{d}{d+2}}}/\mathrm{TOL})^{d/2} < \infty$ while the same error for approximation with a uniform mesh requires a much larger number of elements proportional to a higher power of TOL^{-1}, because $\|\widetilde{\rho}\|_{L^1} = \infty$. The assumption $u,\ \varphi \in \mathcal{C}^3(\bar{\Omega})$ is then violated. But for some such

problems it is possible to determine a spatially varying averaging and extend the analysis in Theorem 2.1; see Example 4.1 for an example.

REMARK 2.1 (AVERAGES ARE NEEDED). *The uniform convergence in Lemma 2.4 applies to the averaged second differences* $\overline{D^2 u_h}$ *and* $\overline{D^2 \varphi_h}$. *On the other hand, with uniform meshes the difference quotients without averaging converge uniformly to the corresponding second derivatives. The following example explains why the second order difference quotients of the interpolant on meshes with hanging nodes do not converge uniformly on* Ω*; numerical tests show that the corresponding finite element solution of Poisson's equation behaves similarly to the interpolant, i.e., second order difference quotients of the finite element solution with hanging nodes do not converge uniformly. Let* $u(x_1, x_2) = x_2^2$*, so that* $\partial_{x_1 x_1} u = 0$ *everywhere, and compute the second difference* $D_1^2 \pi u$ *in the node* $(-h, 0)$ *neighboring the hanging node* $(0, 0)$ *as in Figure 2.2. Using bilinear finite elements the nodal interpolant* πu *is equal to* u *in all proper nodes, but in the hanging node* $\pi u(0,0) = h^2 \neq 0 = u(0,0)$. *Then* $D_1^2 \pi u(-h, 0) = h^2/h^2 = 1$ *but* $\partial_{x_1 x_1} u(-h, 0) = 0$. *In spite of this, Lemma 2.4 shows that the averaged difference quotients converge uniformly under the condition that* $\alpha^{-1} = o(h_{max}^{-\gamma})$ *as* $h_{max} \to 0$.

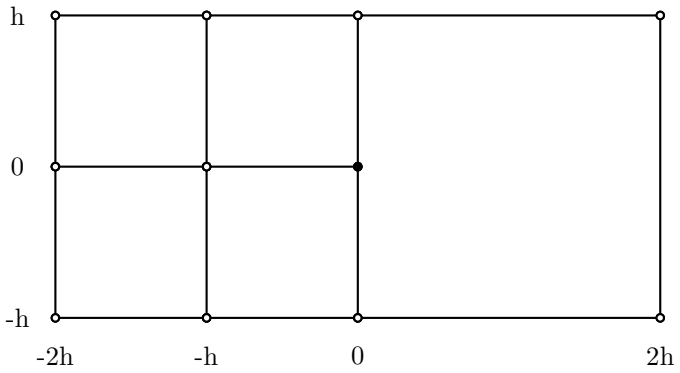


Figure 2.2: Difference quotients without averaging do not converge uniformly to second derivatives in the presence of hanging nodes.

REMARK 2.2 (LOCALIZATION). *One would like to take* $\alpha = \alpha(x_j) \simeq h(x_j)$*, for instance with an average based only on a few neighboring elements. Our convergence proof determines* α *by* (2.18) *and* (2.21)*, where in particular* α *depends on the global pointwise error* $\varphi - \varphi_h$. *Example 4.1 shows different variants of local averages. On the other hand, in practice the algorithm seems to work reasonably well even without the localized averages, see Section 4 and similar algorithms in [10].*

On a uniform mesh, the difference operator D^2 is symmetric so that summation by parts behaves like integration by parts. On an adapted non uniform mesh, the difference operator D^2 is not symmetric, however we have

LEMMA 2.3. *The difference operator $\bar{h}D^2$ is symmetric, i.e., the diagonal matrix $\bar{h}$ is a symmetrizer for D^2, and hence, for all $v, w \in \mathbb{R}^{\bar{N}+2}$, the summation by parts formula*

$$\sum_{i=1}^{\bar{N}} \bar{h}_i w_i D^2 v_i = \sum_{i=1}^{\bar{N}} v_i \bar{h}_i D^2 w_i + \frac{v_{\bar{N}+1} w_{\bar{N}}}{h_{\bar{N}}} + \frac{v_0 w_1}{h_0} - \frac{v_{\bar{N}} w_{\bar{N}+1}}{h_{\bar{N}}} - \frac{v_1 w_0}{h_0}$$

holds.

PROOF. Summation by parts proves the lemma:

$$\begin{aligned}
\sum_{i=1}^{\bar{N}} w_i \bar{h}_i D^2 v_i &= \sum_{i=1}^{\bar{N}} (\frac{v_{i+1} - v_i}{h_i} - \frac{v_i - v_{i-1}}{h_{i-1}}) w_i \\
&= \sum_{i=1}^{\bar{N}} v_i (\frac{w_{i+1} - w_i}{h_i} - \frac{w_i - w_{i-1}}{h_{i-1}}) \\
&\quad + \frac{v_{\bar{N}+1} w_{\bar{N}}}{h_{\bar{N}}} + \frac{v_0 w_1}{h_0} - \frac{v_{\bar{N}} w_{\bar{N}+1}}{h_{\bar{N}}} - \frac{v_1 w_0}{h_0} \\
&= \sum_{i=1}^{\bar{N}} v_i \bar{h}_i D^2 w_i \\
&\quad + \frac{v_{\bar{N}+1} w_{\bar{N}}}{h_{\bar{N}}} + \frac{v_0 w_1}{h_0} - \frac{v_{\bar{N}} w_{\bar{N}+1}}{h_{\bar{N}}} - \frac{v_1 w_0}{h_0}.
\end{aligned}$$

□

This symmetry of $\bar{h}D^2$ is the essential ingredient to obtain convergence of the averages $\overline{D^2 u_h}$ and $\overline{D^2 \varphi_h}$.

LEMMA 2.4. *Assume that $u, \varphi \in \mathcal{C}^3(\bar{\Omega})$ and*

$$\begin{aligned}
h_{max}^{\gamma} \|u - u_h\|_{W^{1,\infty}} + h_{max}^{\gamma} \|\varphi - \varphi_h\|_{W^{1,\infty}} \\
+ \|u - u_h\|_{L^\infty} + \|\varphi - \varphi_h\|_{L^\infty} = \mathcal{O}(h_{max}^{2\gamma})
\end{aligned}$$

holds; then the averaged second difference quotients and second derivatives of u_h and φ_h, in the initial reference element coordinates, satisfy

$$\begin{aligned}
\|\overline{D_i^2 u_h} - \partial_{x_i x_i} u\|_{L^\infty} + \|\overline{D_i^2 \varphi_h} - \partial_{x_i x_i} \varphi\|_{L^\infty} \\
= \mathcal{O}(h_{max}^{\gamma}/\alpha + \alpha), \quad \text{as } h_{max} \to 0.
\end{aligned} \tag{2.26}$$

PROOF. Let $(w, w_h) = (u, u_h)$ or $(w, w_h) = (\varphi, \varphi_h)$ and consider the average error $\overline{D_i^2 (w_h - w)}$. Lemma 2.3 shows that the operator $\bar{h} D_i^2$ is symmetric. This symmetry and the assumptions (2.9)-(2.10) yield, with $\cdot$ denoting the standard

scalar product in $\mathbb{R}^{\bar{N}}$,

$$\begin{aligned}
&|\overline{D_i^2(w_h-w)}(x_j)| = |\bar{h}D_i^2(w_h-w)\cdot\psi^j| \\
&= \Big|(w_h-w)\cdot\bar{h}D_i^2\psi^j \\
&\quad + \frac{(w_h-w)(x_{\bar{N}+1})-(w_h-w)(x_{\bar{N}})}{h_{\bar{N}}}\psi^j(x_{\bar{N}+1}) \\
&\quad - \frac{(w_h-w)(x_1)-(w_h-w)(x_0)}{h_0}\psi^j(x_0)\Big| \\
&\le \|w_h-w\|_{\ell^\infty}\|\bar{h}D_i^2\psi^j\|_{\ell^1} + 2\|w_h'-w'\|_{L^\infty}\max_{x\in\{0,a\}}|\psi^j(x)| \\
&= \mathcal{O}(\frac{h_{max}^\gamma}{\alpha}) \to 0,
\end{aligned}$$

provided $\alpha^{-1} = o(h_{max}^{-\gamma})$ as $h_{max}\to 0$.

The function ψ^j also satisfies the weak convergence (2.12) which together with the uniform convergence $D_i^2 w - \partial_{x_ix_i}w = \mathcal{O}(h_{max})$ and $w\in\mathcal{C}^3(\bar{\Omega})$ imply $\overline{D_i^2 w} - \partial_{x_ix_i}w = \mathcal{O}(\alpha)$. Therefore we conclude that $\overline{D_i^2 w_h} - \partial_{x_ix_i}w = \mathcal{O}(h_{max}^\gamma/\alpha+\alpha)$, in the nodal points, uniformly as $h_{max}\to 0$; the $C^3(\bar{\Omega})$-bound on w implies the L^∞-estimate. □

Let

$$\begin{aligned}
\mathcal{T}_H' &:= \{X(K):\ K\in\mathcal{T}_H\}, \\
\mathcal{T}_I' &:= \{X(K):\ K\in\mathcal{T}_I\}.
\end{aligned}$$

LEMMA 2.5. *Assume that $w\in\mathcal{C}^3([0,1]^2)$. Consider a point $x'\in[0,1]^2$ in an initial reference element and a sequence of squares $K\in\mathcal{T}'\setminus\mathcal{T}_H'$ containing x'. Then*

$$h_K^{-4}\int_K(\pi w(x)-w(x))\,dx - \frac{1}{12}\Delta w(x') = \mathcal{O}(h_K). \tag{2.27}$$

PROOF. Apply the tensor property $\pi=\pi_1\pi_2$, where π_i is the piecewise linear nodal interpolant in the x_i direction, to split

$$\pi w - w = \pi_1(\pi_2 w - w) + \pi_1 w - w. \tag{2.28}$$

Translate the square K to the reference square $K_0 := [0,h]\times[0,h]$, for $h=h_K$. Approximation of w by a quadratic function on K_0 shows that

$$\pi_i w - w = \frac{1}{2}\partial_{x_ix_i}w(x')x_i(h-x_i) + \mathcal{O}(h^3),\ \text{on } K_0. \tag{2.29}$$

The integral

$$\int_0^h x_i(h-x_i)dx_i = \frac{h^3}{6}$$

combined with (2.28)-(2.29) proves the lemma. □

Proof of Theorem 2.1. The error representation (2.4) divides the residual into parts supported in the interior of squares and on their edges.

The residual in the interior of the elements becomes

$$-\mathrm{div}(a^*\nabla u_h) - f^* = -2a^*_{12}\frac{\partial^2 u_h}{\partial x_1 \partial x_2} - \sum_{ij}\frac{\partial a^*_{ij}}{\partial x_i}\frac{\partial u_h}{\partial x_j} - f^*$$

which has the mixed derivative term $2a^*_{12}\frac{\partial u_h}{\partial x_1 \partial x_2} = 2a^*_{12}D_1D_2u_h$. We seek a converging error density and since second order differences of u_h may not converge uniformly, we study the error in replacing $D_1D_2u_h$ by $\partial^2 u/\partial x_1\partial x_2$ using the summation by parts formula

$$\sum_{i=1}^{n}(\alpha_i - \alpha_{i-1})\beta_i = \sum_{i=1}^{n-1}\alpha_i(\beta_i - \beta_{i+1}) + \alpha_n\beta_n - \alpha_0\beta_1 \tag{2.30}$$

applied to $\alpha_K = D_2u_h - D_2u$ evaluated at the east edge e^K_{22} of element K in Figure 2.1 and $\beta_K = \int_K a^*_{12}(\pi\varphi - \varphi)dx'$. Let us denote the set of boundary elements in the x'_1-direction, corresponding to n and 1 in (2.30), by $\partial_1(\mathcal{T}'_H \cup \mathcal{T}'_I)$. This summation by parts yields

$$\begin{aligned}
&\sum_{K\in\mathcal{T}'\setminus(\mathcal{T}'_H\cup\mathcal{T}'_I)}\int_K a^*_{12}D_1D_2u_h(\pi\varphi-\varphi)dx'\\
&= \sum_{K\in\mathcal{T}'\setminus(\mathcal{T}'_H\cup\mathcal{T}'_I)} D_1D_2u\int_K a^*_{12}(\pi\varphi-\varphi)dx'\\
&\quad+\sum_{K\in\partial_1(\mathcal{T}'_H\cup\mathcal{T}'_I)}(D_2u_h - D_2u)\int_K a^*_{12}(\pi\varphi-\varphi)dx'/h_K\\
&\quad+\sum_{K\in\mathcal{T}'\setminus\left(\mathcal{T}'_H\cup\mathcal{T}'_I\cup\partial_1(\mathcal{T}'_H\cup\mathcal{T}'_I)\right)} D_2(u-u_h)D_1\int_K a^*_{12}(\pi\varphi-\varphi)dx'.
\end{aligned} \tag{2.31}$$

The last term has the bound

$$D_1\int_K a^*_{12}(\pi\varphi-\varphi)dx' = \mathcal{O}(h_K^4)$$

since K and its neighbors have no hanging nodes or edges on the initial mesh. Observe that across edges in the initial mesh the matrix a^* changes discontinuously and on hanging node elements $\pi\varphi$ has discontinuous derivatives; therefore summation by parts on all elements would give larger difference quotients of order h_K^3. The assumption (2.18) implies $D_2(u-u_h) = \mathcal{O}(C_\delta h_{max})$ and yields

$$\begin{aligned}
&\sum_{K\in\mathcal{T}'\setminus(\mathcal{T}'_H\cup\mathcal{T}'_I\cup\partial_1(\mathcal{T}'_H\cup\mathcal{T}'_I))} D_2(u-u_h)D_1\int_K a^*_{12}(\pi\varphi-\varphi)dx'\\
&\qquad = \mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}'\setminus(\mathcal{T}'_H\cup\mathcal{T}'_I\cup\partial_1(\mathcal{T}'_H\cup\mathcal{T}'_I))} h_K^4,
\end{aligned}$$

which is an asymptotically negligible contribution to the total error $\sum_K \mathcal{O}(h_K^4)$. The boundary term has by (2.18) the bound

$$\sum_{K\in\partial_1(\mathcal{T}_H'\cup\mathcal{T}_I')}\int_K (D_2u_h - D_2u)a_{12}^*(\pi\varphi-\varphi)dx'/h_k = \mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}_H'\cup\mathcal{T}_I'} h_K^3.$$

Similarly, replacing in the remaining interior error the u_h increments by corresponding derivatives of the solution u and using Lemmas 2.4 and 2.5 give

(2.32)

$$\begin{aligned}
&\sum_{K\in\mathcal{T}'}\int_K \big(-2a_{12}^*\frac{\partial u_h}{\partial x_1'\partial x_2'} - \sum_{ij}\frac{\partial a_{ij}^*}{\partial x_i'}\frac{\partial u_h}{\partial x_j'} - f^*\big)(\pi\varphi-\varphi)dx'\\
&= \sum_{K\in\mathcal{T}'}\int_K \left(a_{11}^*\frac{\partial^2 u}{\partial x_1'\partial x_1'} + a_{22}^*\frac{\partial^2 u}{\partial x_2'\partial x_2'}\right)(\pi\varphi-\varphi)\,dx'\\
&\quad + \mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}_H'\cup\mathcal{T}_I'} h_K^3 \;+\; \mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}'\setminus(\mathcal{T}_H'\cup\mathcal{T}_I')} h_K^4\\
&= \sum_{K\in\mathcal{T}'}\left(\sum_{j=1}^4 (a_{11}^*\overline{D_1^2u_h} + a_{22}^*\overline{D_2^2u_h})(\overline{D_1^2\varphi_h} + \overline{D_2^2\varphi_h})(x_j^K)\frac{h_K^4}{48} + h_K^4\mathcal{O}(\frac{h_{max}^\gamma}{\alpha}+\alpha)\right)\\
&\quad + \mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}_H'\cup\mathcal{T}_I'} h_K^3.
\end{aligned}$$

Consider the error associated with the edges in the error representation (2.4). The normal derivative $\partial u_h/\partial x_i$ is the same on the opposite edges of a square in $\mathcal{T}'$. Therefore the regularity $\varphi\in\mathcal{C}^3(\bar\Omega)$ and the uniform convergence $\nabla u_h\to\nabla u$ of (2.18) yield for $K\in\mathcal{T}'\setminus(\mathcal{T}_H'\cup\mathcal{T}_I')$

$$\begin{aligned}
&\int_{\partial K} n\cdot a^*\, n\frac{\partial u_h}{\partial n}(\pi\varphi-\varphi)ds'\\
&= \int_{e_{12}^K}\left.\frac{\partial u_h}{\partial x_2}\right|_{e_{12}}\left(a_{22}^*\left(\pi\varphi-\varphi\right)|_{e_{12}} - a_{22}^*\left(\pi\varphi-\varphi\right)|_{e_{11}}\right)dx_1'\\
&\quad + \int_{e_{22}^K}\left.\frac{\partial u_h}{\partial x_1}\right|_{e_{22}}\left(a_{11}^*\left(\pi\varphi-\varphi\right)|_{e_{22}} - a_{11}^*\left(\pi\varphi-\varphi\right)|_{e_{21}}\right)dx_2'\\
&= \int_{e_{12}^K}\left.\frac{\partial u}{\partial x_2}\right|_{e_{12}}\left(a_{22}^*\left(\pi\varphi-\varphi\right)|_{e_{12}} - a_{22}^*\left(\pi\varphi-\varphi\right)|_{e_{11}}\right)dx_1'\\
&\quad + \int_{e_{22}^K}\left.\frac{\partial u}{\partial x_1}\right|_{e_{22}}\left(a_{11}^*\left(\pi\varphi-\varphi\right)|_{e_{22}} - a_{11}^*\left(\pi\varphi-\varphi\right)|_{e_{21}}\right)dx_2' + h_K^4\mathcal{O}(C_\delta h_{max}),
\end{aligned}$$

where $e_{ij}\coloneqq e_{ij}^K$ in Figure 2.1. Use $[\partial u_h/\partial n]=\mathcal{O}(C_\delta h_{max})$ in $\mathcal{T}_H'\cup\mathcal{T}_I'$, obtained from (2.18) and $[\partial u/\partial n]=0$, and summation by parts over the other edges, as

in (2.31), to obtain

$$\sum_{e\in\mathcal{T}_e}\int_e n\cdot an[\frac{\partial u_h}{\partial n}](\pi\varphi-\varphi)ds'$$

$$\begin{aligned}(2.33)\quad &-\sum_{K\in\mathcal{T}'\setminus(\mathcal{T}'_H\cup\mathcal{T}'_I)}\left(\int_{e_{11}^K}a^*_{22}\Big(\left.\frac{\partial u}{\partial x_2}\right|_{e_{12}}-\left.\frac{\partial u}{\partial x_2}\right|_{e_{11}}\Big)(\pi\varphi-\varphi)\,dx'_1\right.\\ &\quad+\left.\int_{e_{21}^K}a^*_{11}\Big(\left.\frac{\partial u}{\partial x_1}\right|_{e_{22}}-\left.\frac{\partial u}{\partial x_1}\right|_{e_{21}}\Big)(\pi\varphi-\varphi)\,dx'_2+h_K^4\mathcal{O}(C_\delta h_{max})\right)\\ &\quad+\sum_{K\in\partial_i(\mathcal{T}'_H\cup\mathcal{T}'_I)}\int_{e_{ij}^K}n\cdot a^*\,n(\frac{\partial u_h}{\partial n}-\frac{\partial u}{\partial n})(\pi\varphi-\varphi)ds'\\ &\quad+\mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}'_H\cup\mathcal{T}'_I}\mathcal{O}(h_K^3).\end{aligned}$$

The assumptions (2.18) and (2.19) show that the boundary term becomes asymptotically negligible

$$\sum_{K\in\partial_i(\mathcal{T}'_H\cup\mathcal{T}'_I)}\int_{e_{ij}^K}n\cdot a^*\,n(\frac{\partial u_h}{\partial n}-\frac{\partial u}{\partial n})(\pi\varphi-\varphi)ds'=\mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}'_H\cup\mathcal{T}'_I}h_K^3.$$

We have

$$\left(\left.\frac{\partial u}{\partial x_2}\right|_{e_{12}}-\left.\frac{\partial u}{\partial x_2}\right|_{e_{11}}\right)/h_K=\partial_{x_2x_2}u+\mathcal{O}(h_K)$$

therefore the regularity assumptions $a\in\mathcal{C}^1(\bar\Omega)$ and $u\in\mathcal{C}^3(\bar\Omega)$, together with Lemmas 2.4 and 2.5 applied to u_h and φ_h imply that right hand side of (2.33) has an expansion with leading order terms

(2.34)

$$\begin{aligned}\int_{e_{11}^K}a^*_{22}\left(\left.\frac{\partial u}{\partial x_2}\right|_{e_{12}}-\left.\frac{\partial u}{\partial x_2}\right|_{e_{11}}\right)(\pi\varphi-\varphi)\,dx'_1=&\frac{h_K^4}{48}\sum_{j=1}^4a^*_{22}\overline{D_2^2u_h}\;\overline{D_1^2\varphi_h}(x_j^K)\\ &+h_K^4\mathcal{O}(h_{max}^\gamma/\alpha+\alpha),\end{aligned}$$

and similarly for the edges in the other direction. Note that the convergence $\mathcal{O}(h_{max}^\gamma/\alpha+\alpha)\to 0$ in (2.34) is uniform on $\bar\Omega$. Finally, the mixed second differences in the sum over all edges in (2.33) cancel by the same terms from the interior residual to complete the proof of (2.20). The error estimate (2.22) then follows from Lemma 2.5. □

Proof of the Corollary. The steps in this proof are done in more detail in Section 3. Here is a short preview. Summation of all indicators with the upper bound in (2.15) proves (2.24)

$$\sum_K\bar\rho_Kh_K^4\le\sum_K\hat\rho_Kh_K^4\le S_1\mathrm{TOL}.$$

The upper bound in (2.15) also gives

$$N = \int_\Omega \frac{dx}{h^2} \geq \sqrt{\frac{N}{S_1 \mathrm{TOL}}} \int_\Omega \sqrt{\hat{\rho}} dx$$

and consequently

$$\frac{1}{N} \leq \frac{S_1 \mathrm{TOL}}{\|\hat{\rho}\|_{L^{1/2}}}$$

and

$$h_{max}^2 \leq \frac{S_1 \mathrm{TOL}}{\sqrt{\delta} \int_\Omega \sqrt{\hat{\rho}} dx}.$$

Let $h_{max}^\gamma/\alpha + \alpha = \mathcal{O}(h_{max}^{\hat{\gamma}})$ be the definition of $\hat{\gamma} \in (0,1]$. Then a choice of δ such that $(h_{max}^\gamma/\alpha + \alpha)\hat{\rho} \leq (\mathrm{TOL}/\sqrt{\delta})^{\hat{\gamma}}/\delta = o(1)$ and $C_\delta \int_{\bar{T}_H \cup \bar{T}_I} dx/\sqrt{\delta} = o(1)$ together with the expansion (2.20) proves (2.24).

It remains to establish the lower bound. The lower bound (2.15) on the error indicators imply as above

$$\sum_K |\bar{\rho}_K| h_K^4 = \int_\Omega |\bar{\rho}| h^2 dx \geq \frac{c\mathrm{TOL} \int_\Omega |\bar{\rho}|/\sqrt{\hat{\rho}}\, dx}{\int_\Omega \sqrt{\hat{\rho}} dx}$$

which combined with the uniform convergence of $\hat{\rho}$ and $\bar{\rho}$ proves the lower bound (2.25) in the limit $\mathrm{TOL} \to 0+$. □

2.3 Efficient Computation of the Averages

Let

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

We will use $\psi^j : \mathcal{N} \to \mathbb{R}$ determined, for $j = 1, \ldots, \bar{N}$, by

$$\bar{h}_i \psi_i^j - \alpha^2 \bar{h}_i D^2 \psi_i^j = \delta_{ij}, \quad i = 1, \ldots, \bar{N} \tag{2.35}$$

with homogeneous Neumann boundary conditions $\psi_0^j = \psi_1^j$ and $\psi_{\bar{N}}^j = \psi_{\bar{N}+1}^j$. The averages $Y_j := \overline{D^2 v}(x_j) = \sum_{i=1}^N \bar{h}_i D^2 v_i \psi_i^j$ then solve the dual equation to (2.35) with the right hand side $\bar{h}_i D^2 v_i$. The symmetry of $\bar{h} D^2$ and the summation by parts formula in Lemma 2.3 show that Y can be efficiently computed by

$$Y_j - \alpha^2 D^2 Y_j = D^2 v_j, \quad j = 1, \ldots, \bar{N} \tag{2.36}$$

with homogeneous Neumann boundary conditions, $Y_0 = Y_1,\ Y_{\bar{N}} = Y_{\bar{N}+1}$.

It remains to verify the conditions (2.8)-(2.12) for ψ. The following discrete minimum principle argument first observes that $D^2 \psi_{i^*}^j \geq 0$ at a minimum point i^* and hence (2.35) yields non negative ψ

$$\psi_i^j \geq \psi_{i^*}^j = \frac{\delta_{i^* j}}{\bar{h}_{i^*}} + \alpha^2 D^2 \psi_{i^*}^j \geq 0.$$

The remaining conditions can be derived from the following estimate of the weighted ℓ^1 norm $\sum_i \bar{h}_i \psi_i^j w_i$, with the weights $w_i = \cosh(\frac{x_i - x_p}{\bar{\alpha}})$ for $\bar{\alpha} \geq 2\alpha$ and $x_p \in \mathcal{N}$: multiplication by w_i in (2.35) and summation by parts shows

$$\begin{aligned} &\sum_{i=1}^{\bar{N}} \psi_i^j \bar{h}_i (w_i - \alpha^2 D^2 w_i) \\ &+ \alpha^2 \psi_{\bar{N}+1}^j \frac{w_{\bar{N}+1} - w_N}{h_{\bar{N}}} - \alpha^2 \psi_0^j \frac{w_1 - w_0}{h_0} = \cosh(\frac{x_j - x_p}{\bar{\alpha}}). \end{aligned} \tag{2.37}$$

We have

$$\begin{aligned} D^2 w_i &= \bar{\alpha}^{-2} w_i + \mathcal{O}(h_{max}), \\ \frac{w_{\bar{N}+1} - w_{\bar{N}}}{h_{\bar{N}}} &= \bar{\alpha}^{-1} \sinh(\frac{x_{\bar{N}} - x_p}{\bar{\alpha}}) + \mathcal{O}(h_{max}), \\ \frac{w_1 - w_0}{h_0} &= \bar{\alpha}^{-1} \sinh(\frac{x_0 - x_p}{\bar{\alpha}}) + \mathcal{O}(h_{max}). \end{aligned} \tag{2.38}$$

Therefore all terms in the left hand side of (2.37) are non negative for $\bar{\alpha} \geq 2\alpha$, with sufficiently small h_{max}, and provide estimates to verify the conditions (2.9)-(2.12).

Note first that the choice $w = 1$, corresponding to $\bar{\alpha} = \infty$ in (2.37), implies $\sum_i \bar{h}_i \psi_i^j = 1$ and hence by (2.35) we have $\|\alpha^2 \bar{h} D^2 \psi^j\|_{\ell^1} \leq 2$.

The estimates (2.37) and (2.38) show first that the boundary values satisfy

$$\begin{aligned} \psi_0^j &= \mathcal{O}(1/\min(x_j, \alpha)), \\ \psi_{\bar{N}+1}^j &= \mathcal{O}(1/\min(\hat{a} - x_j, \alpha)), \end{aligned}$$

and with a second choice of weight function $w_i = e^{-x_i/\bar{\alpha}}$ and $w_i = e^{(x_i - \hat{a})/\bar{\alpha}}$, respectively, we have similarly

$$\begin{aligned} \psi_0^j &= \mathcal{O}(1/\alpha), \\ \psi_{\bar{N}+1}^j &= \mathcal{O}(1/\alpha). \end{aligned}$$

Finally, to verify the weak convergence we estimate

$$|\sum_i \psi_i^j (v_i - v_j) \bar{h}_i| \leq \|\psi^j w \bar{h}\|_{\ell^1} \|(v_\cdot - v_j) w^{-1}\|_{\ell^\infty}$$

and use (2.37) together with a uniform bound on the difference quotients of v to obtain

$$\|(v_\cdot - v_j) w^{-1}\|_{\ell^\infty} \leq \mathcal{O}(\max_i |x_i - x_j| e^{-|x_i - x_j|/\bar{\alpha}}) = \mathcal{O}(\bar{\alpha}),$$

so that $|\sum_i \psi_i^j v_i \bar{h}_i - v_j| \leq \mathcal{O}(\bar{\alpha})$.

2.4 Higher Order Polynomials

The analysis in Section 2 uses isoparametric d-linear quadrilateral finite elements and splits the error to a dominating part, from domains where the mesh is locally uniform, and a negligible part, from a small domain with hanging nodes. This kind of analysis seems possible to extend to higher order elements. In the case of approximation with isoparametric order k quadrilateral finite elements, the leading order error term becomes $\sum_K \bar{\rho} h_K^{2k+d}$ with the error density in the transformed coordinates defined by

$$\overline{\rho}_K := c \sum_{j=1}^{4} \big(a_{11}^* \overline{D_1^{k+1} u_h}\ \overline{D_1^{k+1} \varphi_h} + a_{22}^* \overline{D_2^{k+1} u_h}\ \overline{D_2^{k+1} \varphi_h} \big)(x_j^K), \tag{2.39}$$

where D_i^{k+1} is a $k+1$ order difference quotient approximating the $k+1$ order partial derivative $\partial^{k+1}/\partial x_i^{k+1}$ and c is a constant determined below.

To somehow explain this error density consider a uniformly refined square, in the transformed coordinates, as the computational domain and assume that all difference quotients of order $k+2$ of u_h are bounded and that all derivatives of order $k+2$ of φ are bounded. Instead of using the interpolant of φ in the orthogonality (2.1), take the L^2 projection $\hat{\pi}\varphi$ as test function in the error representation (2.3) to obtain the error representation

$$(u - u_h, F) = (\mathcal{R}(u_h), \hat{\pi}\varphi - \varphi)\,.$$

By the tensor property $\hat{\pi} = \hat{\pi}_1 \hat{\pi}_2$, this representation splits into two one dimensional problems

$$\begin{aligned}(\mathcal{R}(u_h), \hat{\pi}\varphi - \varphi) &= (\mathcal{R}(u_h), \hat{\pi}_1\varphi - \varphi) + (\mathcal{R}(u_h), \hat{\pi}_1(\hat{\pi}_2\varphi - \varphi)) \\ &= (\mathcal{R}(u_h), \hat{\pi}_1\varphi - \varphi) + (\hat{\pi}_1\mathcal{R}(u_h), \hat{\pi}_2\varphi - \varphi)\,,\end{aligned}$$

where $\hat{\pi}_i$ is the one dimensional L^2 projection in the x_i-direction. It is then sufficient to consider the error density for one dimensional problems; the use of the L^2 projection is motivated by the following step

$$(\mathcal{R}(u_h), \hat{\pi}_1\varphi - \varphi) = (\mathcal{R}(u_h) - \pi_1\mathcal{R}(u_h), \hat{\pi}_1\varphi - \varphi)\,.$$

Here the function $\pi_1\mathcal{R}(u_h)$ is the nodal interpolant of the discrete but discontinuous function $\mathcal{R}(u_h) = f + a_0 u_h' + a_h''$ which takes the same value as $\mathcal{R}(u_h)$ in the interior nodes and mean value of the left and right limits at the edges. The residual parts have the estimates

$$\begin{aligned}\|f - \pi_1 f\|_{L^\infty} &= \mathcal{O}(h^{k+1}), \\ \|a_0 u_h' - \pi_1(a_0 u_h')\|_{L^\infty} &= \mathcal{O}(h^k), \\ \|a u_h'' - \pi_1(a u_h'')\|_{L^\infty} &= \mathcal{O}(h^{k-1});\end{aligned}$$

therefore the last part with the jump at the second derivative dominates the error; in the case $k = 2$ we see directly that $u_h'' - \pi_1 u_h''$ is to leading order proportional to the jump $[u_h''] = h D_1^3 u_h$, which is a third order difference quotient.

To estimate the error $\hat{\pi}_1\varphi - \varphi$ pointwise make a Taylor expansion to order $k+2$ centered at a nodal point x_p and note that $I - \hat{\pi}_1$ has a null space of polynomials of order k, so that

$$(\hat{\pi}_1\varphi - \varphi)(x) = \varphi^{(k+1)}(x_p)\frac{(x-x_p)^{k+1}}{(k+1)!} + \mathcal{O}(h^{k+2}),\ |x - x_p| \le h.$$

Assume that u_h is the nodal interpolant of a smooth function w. Expand w into a Taylor series of order $k+2$ as above and let w_1 be its Taylor polynomial of order k. The jump of u_h'' is proportional to the difference of $w - w_1$ in two neighboring elements which to leading order is proportional to $w^{(k+1)}h^{k-1}$, so that $[u_h''] \sim h^{k-1}D^{k+1}u_h$.

We have now motivated that the leading order error term is given by (2.39) for some constant c. To determine the right constant c, consider $\varphi = w = x_1^{k+1}$ and compute by hand

$$c = \int_0^1 \Big((\pi_1 w)'' - \pi_1\big((\pi_1 w)''\big)\Big)(\varphi - \hat{\pi}_1\varphi)dx_1/(h^k(k+1)!)^2.$$

Remark 2.3 (One dimension). *Note that in one dimension, $d = 1$, the edge part of the residual vanishes since $\pi\varphi = \varphi$ in the nodes.*

Remark 2.4 (Nonlinear problems). *A nonlinear problem $a = a(u, x)$ and $f = f(\nabla u, u, x)$ for a nonlinear functional, $\int_\Omega g(u(x), x)dx$, gives a different dual problem for φ, but the same approximation property*

$$\|u - u_h\|_{W^{1,\infty}} + \|\varphi - \varphi_h\|_{W^{1,\infty}} = \mathcal{O}(C_\delta h_{max})$$

and the regularity $u \in \mathcal{C}^3(\bar{\Omega})$, $\varphi \in \mathcal{C}^3(\bar{\Omega})$ also yield estimates of the linearization error and imply the conclusion in the theorem.

Remark 2.5 (Alternative error densities). *Let $s \in [0, 1]$ and*

$$\mathcal{R}^*(\varphi_h) = -div(a\nabla\varphi_h) - F.$$

Then

$$\begin{aligned}(u - u_h, F) = s\,&(\mathcal{R}(u_h), \pi\varphi - \varphi)\\ &+(1-s)\,(\pi u - u, \mathcal{R}^*(\varphi_h))\end{aligned} \tag{2.40}$$

are alternative global error estimates for $s \in [0, 1]$, cf. [10]. The isoparametric bilinear quadrilateral approximation with hanging nodes in this work shows that in fact also the local error densities are asymptotically the same for all $s \in [0, 1]$

$$\bar{\rho} \to \frac{1}{12}\left(a_{11}^*\frac{\partial^2 u}{\partial x_1^2}\frac{\partial^2 \varphi}{\partial x_1^2} + a_{22}^*\frac{\partial^2 u}{\partial x_2^2}\frac{\partial^2 \varphi}{\partial x_2^2}\right), \tag{2.41}$$

as $h_{max} \to 0$: Theorem 2.1 is formulated for the case $s = 1$, but the symmetry of the problem makes Theorem 2.1 applicable to $s \in [0, 1]$. For instance, $s = 0$ is the case where the role of u_h and φ_h are interchanged, with respect to $s = 1$,

and since also the error density is symmetric with respect to this interchange of u_h *and* φ_h *we conclude that the error representation* (2.40) *yields the same error density* (2.41) *for all* $s \in [0,1]$ *asymptotically. Here the approximation errors* $\overline{D_i^2 u_h} \to \frac{\partial^2 u}{\partial x_i^2}$ *and* $\overline{D_i^2 \varphi_h} \to \frac{\partial^2 \varphi}{\partial x_i^2}$ *are the dominating errors in the convergence. The same conclusion of the error density asymptotically independent of* s *holds also for nonlinear problems with* $a = a(u,x)$, $f = f(\nabla u, u, x)$ *and* $g = g(u,x)$.

3 Convergence Rates for the Adaptive Mesh Algorithm

This section constructs an adaptive algorithm and analyzes its stopping, accuracy and efficiency, using the convergence of the error density in Theorem 2.1 to motivate the bound (3.14). The analysis is largely based on the similar work on ordinary differential equations in [23]. The main difference is the hanging node constraint present here.

3.1 Adaptive Refinements and Stopping

Theorem 2.1 proves that the error expansion

(3.1)
$$\begin{aligned}(u-u_h, F) &= \sum_{K\in\mathcal{T}'} (\overline{\rho}_K + \mathcal{O}(\frac{h_{max}^\gamma}{\alpha} + \alpha))h_K^{2+d} + \mathcal{O}(C_\delta h_{max}) \sum_{K\in\mathcal{T}_H'\cup\mathcal{T}_I'} h_K^{1+d} \\ &=: \sum_K \check{\rho}_K h_K^{2+d},\end{aligned}$$

with C_δ defined in (2.17), has a well defined leading order error density $\overline{\rho}$ which converges uniformly as $h_{max} \to 0+$. The two error terms

$$\sum_{K\in\mathcal{T}'} \mathcal{O}(\frac{h_{max}^\gamma}{\alpha} + \alpha)h_K^{2+d} \quad \text{and} \quad \mathcal{O}(C_\delta h_{max}) \sum_{K\in\mathcal{T}_H'\cup\mathcal{T}_I'} h_K^{1+d}$$

are fundamental in the analysis of the algorithm. Assume that α has been chosen such that

$$\frac{h_{max}^\gamma}{\alpha} + \alpha = \mathcal{O}(h_{\max}^{\widehat{\gamma}}), \tag{3.2}$$

where $\widehat{\gamma} > 0$. In the adaptive algorithm below we will use the positive approximate error density $\hat{\rho}_K$ defined by

$$\hat{\rho}|_K = \hat{\rho}_K := \max\left(|\overline{\rho}_K|,\ \delta\right) \tag{3.3}$$

where

$$\delta := \mathrm{TOL}^{\bar{\gamma}} \tag{3.4}$$

satisfies the two lower bounds

$$\bar{\gamma} < \frac{\widehat{\gamma}}{\widehat{\gamma}+2} \ \text{ and } C_\delta \int_{\bar{\mathcal{T}}_H\cup\bar{\mathcal{T}}_I} dx/\delta = o(1) \ \text{ as TOL} \to 0+, \tag{3.5}$$

and the upper bound

$$\delta = o(1) \text{ as } \mathrm{TOL} \to 0+ .$$

The lower bounds on $\delta > 0$ are motivated by the requirements that $h_{max} \to 0$ as $\mathrm{TOL} \to 0$, that the bounds for the error density in (3.14) hold, see Lemma 3.2, and that the error from hanging node elements become asymptotically negligible, see Theorem 3.3. The convergence $\hat{\rho} \to |\tilde{\rho}|$ requires the upper bound $\delta \to 0$.

Let us now motivate the optimal choice of element sizes

$$\big|\rho\big|h^{d+2} = \text{constant},$$

for hypothetical linear tensor reference finite elements with no other constraint than tensor cube reference elements and a mesh independent error density ρ. Define first, for a mesh with elements $\{K_1, K_2, K_3, \ldots, K_N\}$, the piecewise constant error density and mesh functions $\rho|_{K_i} \equiv \rho_i \equiv \rho_{K_i}$, $\hat{\rho}|_{K_i} \equiv \hat{\rho}_i \equiv \hat{\rho}_{K_i}$ and $h|_{K_i} \equiv h_i \equiv h_{K_i}$. The number of elements that corresponds to a mesh with size h can be determined by

$$N(h) \equiv \int_\Omega \frac{dx}{h^d(x)}. \tag{3.6}$$

It seems hard to use the sign of the error indicator for constructing the mesh, since with only two elements the error can be zero just by chance: let $\int_0^1 f(s)ds = 0$ be the integral of a continuous function where also $f(0) = f(1) = 0$. This integral can be computed by the Euler method without error for a very particular choice of just the two elements $(0, \bar{s}), (\bar{s}, 1)$, with an interior point $\bar{s}$ satisfying $f(\bar{s}) = 0$, but any other choice of two elements gives in general very large error. Instead we choose to minimize the number of elements N in (3.6) under the more stringent constraint

$$\sum_{i=1}^{\bar{N}} |\rho_i| h_i^{d+2} = \int_\Omega |\rho(x)| h^2 dx = \mathrm{TOL}. \tag{3.7}$$

This yields, with a standard application of a Lagrange multiplier, the optimal element sizes h^* satisfying

$$|\rho|(h^*)^{d+2} = \ \textit{constant} \tag{3.8}$$

and

$$h^* := \frac{\mathrm{TOL}^{\frac{1}{2}}}{|\rho|^{\frac{1}{d+2}}} \left(\int_\Omega |\rho(x)|^{\frac{d}{d+2}} dx \right)^{-\frac{1}{2}}. \tag{3.9}$$

This condition is optimal only for density functions ρ with one sign and for meshes with shape regular elements, i.e., non stretched elements. To use the sign of the density or orientation of stretched elements in an optimal way is not considered in this work.

The goal of the adaptive algorithm described below is to construct a mesh of Ω such that

$$\hat{\rho}_i h_i^{d+2} \approx \frac{\mathrm{TOL}}{N}, \quad i = 1, \ldots, N, \tag{3.10}$$

which is an approximation of the optimal (3.8). Let the index $[k]$ refer to the refinement level in the sequence of adaptively refined meshes. To achieve (3.10) let $s_1 \approx 1$ be a given constant, start with an initial mesh of size $h[1]$ and then specify iteratively a new mesh $h[k+1]$, from $h[k]$, using the following dividing strategy:

$$\begin{aligned}
&\textbf{for all elements } i = 1, 2, \ldots, N[k] \\
&\quad \bar{r}_i[k] := \hat{\rho}_i[k](h_i[k])^{d+2} \\
&\quad \textbf{if} \quad \bar{r}_i[k] > s_1 \frac{\mathrm{TOL}}{N[k]} \quad \textbf{then} \\
&\qquad \text{mark element } i \text{ for division and recursively mark all neighbors} \\
&\qquad \text{that need division due to the hanging node constraint:} \\
&\qquad \text{at most one hanging node per edge} \\
&\quad \textbf{endif} \\
&\textbf{endfor} \\
&\textbf{divide every marked element in } \mathcal{T}' \text{ into } 2^d \text{ uniform sub elements.}
\end{aligned} \tag{3.11}$$

With this dividing strategy, it is natural to use the stopping criterion:

$$\textbf{if} \quad \Big(\max_{1 \le i \le N[k]} \bar{r}_i[k] \le S_1 \frac{\mathrm{TOL}}{N[k]} \Big) \quad \textbf{then} \quad \text{stop.} \tag{3.12}$$

Here S_1 is a given constant, with $S_1 > s_1 \approx 1$, determined more precisely as follows: we want that the maximal error indicator decays quickly to the stopping level $S_1 \mathrm{TOL}/N$, but when almost all error indicators $\bar{r}_i$ satisfy $\bar{r}_i < s_1 \frac{\mathrm{TOL}}{N}$ the reduction of the error may be slow. Theorem 3.1 shows that a slow reduction is avoided if S_1 satisfies (3.15).

The remainder of this section analyzes in three theorems the adaptive algorithm based on (3.11) and (3.12) with respect to stopping, accuracy and efficiency. To analyze the decay of the maximal error indicator, it is useful to understand the variation of the density $\hat{\rho}$ at different refinement levels, in particular we will consider an element $K[k]$ and its parent on a previous refinement level, $p(K,k)$, with the corresponding error density $\hat{\rho}(K)[p(K,k)]$. With the assumption that $h_{max} \to 0$ as $\mathrm{TOL} \to 0+$ Theorem 2.1 shows that there is a limit error density $\widetilde{\rho}$ such that

$$\lim_{L^1} \check{\rho} = \widetilde{\rho}, \ \overline{\rho} \to \widetilde{\rho} \text{ and } \hat{\rho} \to |\widetilde{\rho}|, \text{ as } \mathrm{TOL} \to 0+. \tag{3.13}$$

A consequence of the uniform convergence $\hat{\rho} \to |\widetilde{\rho}|$, as $\mathrm{TOL} \to 0+$, and (3.3) is that for all elements K and all refinement levels k there exists positive functions

$\hat{c}$ and $\hat{C}$ close to 1 for sufficiently fine meshes, such that the error density satisfies

$$\begin{aligned} \hat{c}(K) &\leq \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]} \leq \hat{C}(K), \\ \hat{c}(K) &\leq \frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]} \leq \hat{C}(K), \end{aligned} \tag{3.14}$$

provided $\max_{K,k} h_K[k]$ is sufficiently small. In other words, (3.14) holds with e.g. $\hat{c} = 2^{-1} = \hat{C}^{-1}$ for sufficiently small $\max_{K,k} h_K[k]$. Note that the condition (3.14) also implies a related constraint on the optimal mesh, see Remark 3.1.

THEOREM 3.1 (STOPPING). *Suppose the assumptions of Theorem 2.1 hold and the adaptive algorithm uses the strategy* (3.11)-(3.12). *Assume that $\hat{c}$ satisfies* (3.14)*, for the elements corresponding to the maximal error indicator on each refinement level, and that*

$$S_1 \geq \frac{2^d}{\hat{c}} s_1, \quad 1 > \frac{\hat{c}^{-1}}{2^{d+2}}. \tag{3.15}$$

Then each refinement level either decreases the maximal error indicator with the factor

$$\max_{1\leq i\leq N[k+1]} \bar{r}_i[k+1] \leq \frac{\hat{c}^{-1}}{2^{d+2}} \max_{1\leq i\leq N[k]} \bar{r}_i[k], \tag{3.16}$$

or stops the algorithm.

We have in [23] tested several alternative stopping rules, such as the usual $|\sum_i \bar{\rho}_i h_i^{2+d}| \leq \mathrm{TOL}$. Our stopping rule (3.12) has the advantage that it implies bounds on h_{max}, see Lemma 3.2, which may explain its more accurate error estimates.

Proof. Define the piecewise constant error indicator function $\bar{r}|_K \equiv \bar{r}_K$, for all elements K. There is a point $x^* \in \Omega$ giving the maximal error indicator value

$$\bar{r}(x^*)[k+1] = \max_{1\leq i\leq N[k+1]} \bar{r}_i[k+1]$$

on refinement level $k+1$. The corresponding indicator $\bar{r}(x^*)[k]$, on the previous level, satisfies precisely one of the following three statements

$$\bar{r}(x^*)[k] \leq \frac{s_1 \mathrm{TOL}}{N[k]}, \tag{3.17}$$

$$\frac{s_1 \mathrm{TOL}}{N[k]} < \bar{r}(x^*)[k] \leq 2^{d+2} \frac{s_1 \mathrm{TOL}}{N[k]}, \tag{3.18}$$

$$\bar{r}(x^*)[k] > 2^{d+2} \frac{s_1 \mathrm{TOL}}{N[k]}. \tag{3.19}$$

If (3.17) holds either the element containing x^* is not divided on level $k+1$ or it is divided on level $k+1$ by the hanging node condition. In any case, (3.14) implies

$$\bar{r}(x^*)[k+1] \leq \frac{\hat{c}^{-1} s_1 \mathrm{TOL}}{N[k]}. \tag{3.20}$$

Condition (3.15) and the bound $N[k+1] \le 2^d N[k]$ imply

$$\frac{S_1 \mathrm{TOL}}{N[k+1]} \ge \frac{\hat{c}^{-1} s_1 \mathrm{TOL}}{N[k]},$$

which together with (3.20) show that the algorithm stops at level $k+1$ if (3.17) holds.

Similarly, if (3.18) holds, the element containing x^* is divided on level $k+1$, so that $\bar{r}(x^*)[k+1] \le \frac{\hat{c}^{-1} s_1 \mathrm{TOL}}{N[k]}$ again and consequently the algorithm stops at level $k+1$.

Finally if (3.19) holds, the element containing x^* is divided and by (3.14)

$$\bar{r}(x^*)[k+1] \le \frac{\hat{c}^{-1}}{2^{d+2}} \bar{r}(x^*)[k] \le \frac{\hat{c}^{-1}}{2^{d+2}} \max_{1 \le i \le N[k]} \bar{r}_i[k],$$

which proves the theorem. □

Let us verify that the choice (3.4) of δ implies that $h_{max} \to 0$ and that the functions $\hat{c}$ and $\hat{C}$ in (3.14) are close to 1 for sufficiently small tolerances.

LEMMA 3.2. *Suppose $\hat{\rho}$ is given by* (3.3) *and δ by* (3.4). *If the algorithm stops by stopping criterion* (3.12), *then*

$$h_{max}[J] = \mathcal{O}\left(\mathrm{TOL}^{(1-\bar{\gamma})/2}\right), \tag{3.21}$$

for the final mesh J. Suppose in addition that the convergence (2.22) *of the error density holds with α in* (3.2) *and that $h_{max}[1] = \mathcal{O}(\mathrm{TOL}^s)$ for some s with $\bar{\gamma}/\widehat{\gamma} < s < (1-\bar{\gamma})/2$; then*

$$\left| \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]} - 1 \right| = \mathcal{O}\left(\mathrm{TOL}^{s\widehat{\gamma}-\bar{\gamma}}\right) \to 0, \quad \text{as TOL} \to 0,$$
$$\left| \frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]} - 1 \right| = \mathcal{O}\left(\mathrm{TOL}^{s\widehat{\gamma}-\bar{\gamma}}\right) \to 0, \quad \text{as TOL} \to 0.$$

Proof. When the algorithm stops, on level J, the error indicators satisfy the bound

$$(\hat{\rho} h^{d+2})(K)[J] \le \frac{S_1 \mathrm{TOL}}{N[J]}, \quad \text{for all } K. \tag{3.22}$$

Consequently we have by (3.3)

$$\delta h_{max}^{d+2}[J] \le \frac{S_1 \mathrm{TOL}}{N[J]},$$

which using (3.4) proves (3.21):

$$h_{max}^2[J] \le \frac{S_1 \mathrm{TOL}}{\delta N[J] h_{max}^d[J]} \le \frac{S_1 \mathrm{TOL}^{1-\bar{\gamma}}}{\int_\Omega dx}.$$

The convergence (2.22) and the definition (3.3) imply

$$\hat{\rho} = \max(|\widetilde{\rho}| + \mathcal{O}(h_{max}^{\gamma}/\alpha + \alpha), \delta)$$

where $|\widetilde{\rho}|$ is the limit of $\hat{\rho}$. Therefore, by (3.2) and (3.4) we have

$$\left| \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]} - 1 \right| \leq \max_{k \leq J} \frac{\mathcal{O}(h_{max}^{\gamma}/\alpha + \alpha)[k]}{\delta} \leq \frac{\mathcal{O}(h_{\max}^{\widehat{\gamma}}[1])}{\delta} = \mathcal{O}\left(\mathrm{TOL}^{s\widehat{\gamma}-\bar{\gamma}}\right).$$

The same estimate for $\frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]}$ finishes the proof. □

Remark 3.1 (Mesh constraints). *The error density condition* (3.14) *also implies constraints on the optimal mesh; for instance the assumption* $\frac{1}{2}(\bar{\rho}(x)[k] + \bar{\rho}(x+h)[k]) = \bar{\rho}(x)[k-1]$ *shows that*

$$2\hat{c} - 1 \leq \left| \frac{\bar{\rho}(x+h)[k]}{\bar{\rho}(x)[k]} \right| \leq 2\hat{C}^{-1} - 1. \tag{3.23}$$

□

3.2 Accuracy of the Adaptive Algorithm

The adaptive algorithm guarantees that the estimate of the global error is bounded by a given error tolerance, TOL. An important question is whether the true global error is bounded by TOL asymptotically. Using the upper bound (3.12) of the error indicators and the convergence of ρ and $\bar{\rho}$ in Theorem 2.1, the global error has the following estimate.

Theorem 3.3 (Accuracy). *Suppose that the assumptions of Lemma 3.2 hold. Then the adaptive algorithm* (3.11)-(3.12) *satisfies*

$$\limsup_{\mathrm{TOL}\to 0+} \left(\mathrm{TOL}^{-1} \big| (u - u_h, F) \big| \right) \leq S_1. \tag{3.24}$$

Proof. When the adaptive algorithm stops, (2.20), (3.3), (3.12) and (3.21) imply

(3.25)

$$\begin{aligned}
\frac{|(u-u_h,F)|}{\mathrm{TOL}} &= \frac{|\sum_{i=1}^{N}(\bar{\rho}_i h_i^{d+2} + \mathcal{O}(h_{max}^{\widehat{\gamma}})h_i^{d+2}) + \mathcal{O}(C_\delta h_{max})\sum_{K\in\mathcal{T}_H\cup\mathcal{T}_I} h_K^{1+d}|}{\mathrm{TOL}} \\
&\leq \mathrm{TOL}^{-1}\Big(\sum_{i=1}^{N}(|\bar{\rho}_i| + \mathcal{O}(h_{\max}^{\widehat{\gamma}}))h_i^{d+2} + \mathcal{O}(C_\delta h_{max}) \sum_{K\in\mathcal{T}_H\cup\mathcal{T}_I} h_K^{1+d}\Big) \\
&\leq \sum_{i=1}^{N}(1 + \frac{\mathcal{O}(h_{\max}^{\widehat{\gamma}})}{\hat{\rho}_i})\frac{S_1}{N} + \mathcal{O}(C_\delta \frac{h_{max}^2}{\mathrm{TOL}}) \int_{\bar{\mathcal{T}}_H\cup\bar{\mathcal{T}}_I} dx \\
&\leq S_1 + \mathcal{O}\left(\mathrm{TOL}^{(\widehat{\gamma}-\bar{\gamma}(\widehat{\gamma}+2))/2}\right) + \mathcal{O}(C_\delta \delta^{-1}) \int_{\bar{\mathcal{T}}_H\cup\bar{\mathcal{T}}_I} dx,
\end{aligned}$$

which together with the assumptions (3.5) and (2.19) prove (3.24) in the limit as TOL → 0+. □

3.3 Efficiency of the Adaptive Algorithm

An important issue for the adaptive method is its efficiency; we want to determine a mesh with as few elements as possible providing the desired accuracy. From the definition (3.6) and the optimality condition (3.9), the number of optimal adaptive elements, $N^{\rm opt}$, satisfies

$$N^{\rm opt} = \int_\Omega \frac{dx}{(h^*(x))^d} = \frac{1}{{\rm TOL}^{\frac{d}{2}}} \left(\int_\Omega |\rho[k](x)|^{\frac{d}{d+2}} dx \right)^{\frac{d+2}{2}},$$

i.e.

$$N^{\rm opt} = \frac{1}{{\rm TOL}^{\frac{d}{2}}} \|\rho\|^{\frac{d}{2}}_{L^{\frac{d}{d+2}}}. \tag{3.26}$$

Here $\|\cdot\|_{L^{\frac{d}{d+2}}}$ is the quasi-norm defined by

$$\|f\|_{L^{\frac{d}{d+2}}} := \Big(\int_\Omega |f(x)|^{\frac{d}{d+2}} dx \Big)^{\frac{d+2}{d}}.$$

On the other hand, for the uniform mesh with elements $h = \textit{constant}$, the number of elements, $N^{\rm uni}$, to achieve $\sum_{i=1}^N |\rho_i| h_i^{d+2} = {\rm TOL}$ becomes

$$N^{\rm uni} = \int_\Omega \frac{dx}{h^d(x)} = \frac{\int_\Omega dx}{{\rm TOL}^{\frac{d}{2}}} \left(\int_\Omega |\rho[k](x)| dx \right)^{\frac{d}{2}},$$

i.e.

$$N^{\rm uni} = \frac{\int_\Omega dx}{{\rm TOL}^{\frac{d}{2}}} \|\rho\|^{\frac{d}{2}}_{L^1}. \tag{3.27}$$

Hence, the number of uniform elements is measured in the L^1-norm while the optimal number of elements is measured in the $L^{\frac{d}{d+2}}$ quasi-norm. Jensen's inequality implies $\|f\|_{L^{\frac{d}{d+2}}} \leq (\int_\Omega dx)^{\frac{2}{d}} \|f\|_{L^1}$, therefore an adaptive method may use fewer elements than the uniform element size method. Example 4.1 shows a case where adaptive refinements give a better order of convergence than uniform refinements.

The following theorem uses a lower bound of the error indicators, obtained from the refinement criterion (3.11) for the refined parent error indicator and the ratio of the error density (3.14), to show that the algorithm (3.11)-(3.12) generates a mesh which is optimal, up to a multiplicative constant.

THEOREM 3.4 (EFFICIENCY). *Assume that $\hat{C} = \hat{C}(x)$ satisfies* (3.14) *for all elements at the final refinement level, that the assumptions of Lemma 3.2 hold and that the initial mesh satisfies $h_K[1] \geq {\rm TOL}^s$ for all elements K and some s with $\bar{\gamma}/\widehat{\gamma} < s < (1-\bar{\gamma})/2$. Then there exists a constant $C > 0$, bounded by $(\frac{2^{d+2}}{s_1})^{\frac{d}{2}}$, such that, for sufficiently small* TOL, *the final number of adaptive elements N, of the algorithm* (3.11)-(3.12), *satisfies*

$$({\rm TOL}^{\frac{d}{2}} N) \leq C \ \|\hat{C}\hat{\rho}\|^{\frac{d}{2}}_{L^{\frac{d}{d+2}}} \leq C \left(\max_{x\in\Omega} \hat{C}(x)^{\frac{d}{2}} \right) \|\hat{\rho}\|^{\frac{d}{2}}_{L^{\frac{d}{d+2}}}, \tag{3.28}$$

and

$$\begin{aligned}\lim_{\mathrm{TOL}\to 0+} \|\hat{\rho}\|_{L^{\frac{d}{d+2}}} &= \|\widetilde{\rho}\|_{L^{\frac{d}{d+2}}},\\ \lim_{\mathrm{TOL}\to 0+} \max_{x\in\Omega} \hat{C}(x)^{\frac{d}{2}} &= 1,\end{aligned} \tag{3.29}$$

i.e. the number of elements is asymptotically optimal up to the problem independent factor $C \leq (\frac{2^{d+2}}{s_1})^{\frac{d}{2}}$.

Proof. Let us first verify that the initial mesh is coarse enough so that all elements are divided, i.e. that

$$\bar{r}(K)[1] \geq s_1 \mathrm{TOL}/N[1]. \tag{3.30}$$

We have

$$\bar{r}(K)[1] \geq \delta h_K^{2+d}[1] \geq \mathrm{TOL}^{\bar{\gamma}+2s} h_K^d[1] \geq \mathrm{TOL}^{\bar{\gamma}+2s} \frac{\int_\Omega dx}{N[1]} > s_1 \frac{\mathrm{TOL}}{N[1]}$$

provided $\mathrm{TOL}^{\bar{\gamma}+2s-1} > 1/\int_\Omega dx$. By assumption $\bar{\gamma}+2s-1<0$ and we see that (3.30) indeed holds for the initial mesh with sufficiently small TOL.

When the adaptive algorithm stops, on level J, the error indicators satisfy the upper bound

$$\bar{r}_K[J] = (\hat{\rho}(K) h_K^{d+2})[J] \leq \frac{S_1 \mathrm{TOL}}{N[J]}, \quad \forall K \in \mathcal{T}.$$

Each element $K[J]$ has a parent on a previous level, $p(K,J)$ (not necessary the previous level $J-1$), which was divided. We shall show that this parent indicator $\bar{r}(K)[p(K,J)]$ satisfies the lower bound

$$\bar{r}(K)[p(K,J)] > \frac{s_1 \mathrm{TOL}}{N[p(K,J)]}, \tag{3.31}$$

and this lower bound is the essential step to obtain the estimate (3.28). If this parent was not refined by hanging node constraints, the lower bound holds. In fact, it also holds if the refinement was made by hanging node constraints: then the parent has a refined neighbor element which has half the mesh size while the error densities $\hat{\rho}_i$ and $\hat{\rho}_j$ of two neighboring elements satisfy by Theorem 2.1

(3.32)

$$\begin{aligned}\big|1-\frac{\hat{\rho}_j}{\hat{\rho}_i}\big| &= \frac{|\hat{\rho}_i - |\widetilde{\rho}_i| + |\widetilde{\rho}_i| - |\widetilde{\rho}_j| + |\widetilde{\rho}_j| - \hat{\rho}_j|}{\hat{\rho}_i}\\ &= \mathcal{O}(\frac{h_{max}^{\widehat{\gamma}}[1]}{\delta} + \frac{h_{max}[1]}{\delta}) = \mathcal{O}\left(\mathrm{TOL}^{s\widehat{\gamma}-\bar{\gamma}}\right) \to 0 \quad \text{as TOL} \to 0+.\end{aligned}$$

Therefore the error indicator of the parent is a factor 2^{d+1} larger than for the neighbor. Hence, starting from source elements, where the indicator is marked

for refinement not by the hanging node constraint, the error indicators for successive connected hanging node neighbors increase and consequently also the hanging node refinements satisfy the lower bound (3.31).

The indicators of the parent elements therefore satisfy the lower bound

$$\begin{aligned}\hat{\rho}(K)[p(K,J)]2^{d+2}h_K^{d+2}[J] &= (\hat{\rho}(K)h^{d+2}(K))[p(K,J)]\\ &> \frac{s_1\mathrm{TOL}}{N[p(K,J)]}\\ &\geq \frac{s_1\mathrm{TOL}}{N[J]}.\end{aligned}$$

The estimate on the number of elements now follows by relating the error indicators to the lower bounds of their parents:

$$\begin{aligned}h^{d+2}(K)[J] &> \frac{s_1\mathrm{TOL}}{N[J]}\frac{1}{2^{d+2}}\frac{1}{\hat{\rho}(K)[p(K,J)]}\\ &\geq \frac{s_1\mathrm{TOL}}{N[J]2^{d+2}}\frac{1}{\hat{C}\hat{\rho}(K)[J]}.\end{aligned} \tag{3.33}$$

This and (3.6) imply

$$N[J] = \int_\Omega \frac{dx}{h^d(x)[J]} < \frac{(N[J])^{\frac{d}{d+2}}2^d}{(s_1\mathrm{TOL})^{\frac{d}{d+2}}}\int_\Omega \left|\hat{C}\hat{\rho}\right|^{\frac{d}{d+2}}dx$$

which together with Hölder's inequality proves (3.28):

$$\begin{aligned}N[J] &\leq \left(\frac{2^{d+2}}{s_1}\right)^{\frac{d}{2}}\left(\frac{1}{\mathrm{TOL}}\right)^{\frac{d}{2}}\|\hat{C}\hat{\rho}\|_{L^{\frac{d}{d+2}}}^{\frac{d}{2}}\\ &\leq \left(\frac{2^{d+2}}{s_1}\right)^{\frac{d}{2}}\left(\frac{1}{\mathrm{TOL}}\right)^{\frac{d}{2}}(\|\hat{C}\|_{L^\infty}\|\hat{\rho}\|_{L^{\frac{d}{d+2}}})^{\frac{d}{2}}.\end{aligned}$$

Combining the last estimate with the uniform convergence $\hat{\rho} \to |\widetilde{\rho}|$ and Lemma 3.2 establish the asymptotic result (3.29). □

Remark 3.2 (Efficiency without Theorem 2.1). *The conclusion* (3.28) *in Theorem 3.4 can be obtained for all* TOL *also without use of Theorem 2.1 by replacing the estimate* (3.32)*, requiring Theorem 2.1, by the assumption that the quotient of the error densities for two neighboring elements is bounded below by* $2^{-(d+2)}$.

3.4 Implementation of the Adaptive Algorithm

This subsection presents a detailed implementation of the adaptive algorithm (3.11)-(3.12). The dividing strategy (3.11) is applied iteratively until the approximate solution is sufficiently resolved, in other words, until the approximate error density $\hat{\rho}$ and the elements satisfy the stopping criterion (3.12):

Initialization

The user chooses:

1. an initial error tolerance, TOL,
2. an initial coarse (uniform) mesh, and
3. a number, s_1, in (3.11) and a rough estimate of $\hat{c}$ in (3.14) to compute S_1 using (3.15),
4. numbers $\bar{\gamma}$ and α in (2.36) and (3.4).

Set the mesh level k to 0.

Step I

Increase the mesh level k by 1. Compute the second order accurate approximation $u_h[k] \in V_h[k]$ in (2.1) and compute the approximate weight $\varphi_h[k] \in V_h[k]$, using the method (2.6). Compute the density $\hat{\rho}_i[k]$ using (2.14),(3.3),(3.4) and an average, i.e., (2.36).

Step II

If $\left(\max_{1\leq i\leq N[k]} \bar{r}_i[k] \leq \frac{S_1 \mathrm{TOL}}{N[k]} \right)$ **then** stop the program

else

do (3.11)

go to Step I

endif.

3.5 Decreasing Tolerance

This subsection studies an adaptive algorithm allowing the tolerance to decrease slightly as the mesh is refined. The decreasing tolerance is motivated by efficiency – the efficiency of the algorithm depends on the total work including all refinement levels. If the number of elements in each refinement iteration increases only very slowly, the total work becomes proportional to the product of the number of elements in the finest mesh times the number of refinement levels. The condition (3.9) shows that the number of refined levels, J, satisfies

$$\min h = 2^{-J} h[1] = \mathcal{O}(\mathrm{TOL}^{1/2}). \tag{3.34}$$

A relation $\min h = \mathcal{O}(\mathrm{TOL}^{\alpha}),\ \alpha > 0$ still holds for many singular densities, as in Example 4.1. Therefore, $J = \mathcal{O}(\log(\mathrm{TOL}^{-1})) \simeq \log N$, so that the total number of elements, including all mesh levels, of the algorithm (3.11)-(3.12) would be essentially bounded by

$$\mathcal{O}(N \log N). \tag{3.35}$$

A more efficient refinement algorithm is obtained by successively decreasing the tolerance, $\mathrm{TOL}[k+1] < \mathrm{TOL}[k]$, in each refinement so that

$$\frac{N[k]}{N[k+1]} \leq \bar{c} < 1 \tag{3.36}$$

always holds. The condition (3.36) would imply that the total number of elements satisfy

$$\sum_{k=1}^{J} N[k] \leq \frac{N[J]}{1-\bar{c}}. \tag{3.37}$$

Therefore, a slightly decreasing tolerance may be more efficient than a constant tolerance. Including the assumption

$$c' \leq \frac{\mathrm{TOL}[k+1]}{\mathrm{TOL}[k]} \leq 1 \tag{3.38}$$

and replacing $\hat{c}$ by $c'\hat{c}$ in (3.15) directly generalizes Theorems 3.1, 3.3 and 3.4 to slightly varying tolerance, where TOL in (3.24) and (3.28) then denotes the final stopping tolerance. However, an unattractive consequence of varying tolerance is that the stopping tolerance becomes a priori uncertain, see Remark 3.3 and Theorem 3.5.

REMARK 3.3. *A decreasing tolerance is useful if there are few elements with their error indicators, $\bar{r}_i$, in the set $(s_1\mathrm{TOL}/N, \infty)$. To include a decreasing tolerance, modify the algorithm by adding the command "Set* $\mathtt{V} = 0$*" in the end of* **Step I** *and replace* **Step II** *by:*

Step II

If $\big(\max_{1\leq i\leq N[k]} \bar{r}_i[k] \leq \frac{S_1\mathrm{TOL}}{N[k]}\big)$ **then** *stop the program*

else

do (3.11)

if $(N[k]/N[k+1] > \bar{c}$ & $\mathtt{V} = 0)$, **then**

$\mathrm{TOL} = \mathrm{TOL}[k](1 - \frac{\bar{c}^{-1}-1}{2^d-1})$, $\mathtt{V} = 1$ *and* **go to Step II**,

else

go to Step I.

endif

endif

Include in the initialization also a choice of the factor $\bar{c}$ to increase the number of elements in (3.36).

Assume that the set $(c's_1\mathrm{TOL}/N, s_1\mathrm{TOL}/N]$ contains a fraction $c''N$ of the elements, where $2^{-d} < c' < 1$; for instance, if the error indicators, $\bar{r}_i$, are uniformly distributed in $[0, s_1\mathrm{TOL}/N]$, with a negligible part outside of this set,

there holds $c'' = 1 - c'$*, which yields* $\bar{c} = \frac{1}{1+c''(2^d-1)} = \frac{1}{1+(1-c')(2^d-1)}$ *and motivates* $c' = 1 - \frac{\bar{c}^{-1}-1}{2^d-1}$ *in the algorithm. A refinement approximately maps the error indicator set*

$$(c' s_1 \mathrm{TOL}/N, s_1 \mathrm{TOL}/N]$$

to

$$(c' s_1 \mathrm{TOL}/(N 2^{d+2}), s_1 \mathrm{TOL}/(N 2^{d+2})].$$

Then the next refinement continues with essentially a similar distribution of the error indicators, provided c' *is not too small. When the algorithm stops, the final tolerance satisfies*

$$\mathrm{TOL}[0] \geq \mathrm{TOL}[J] \geq \mathrm{TOL}[0](c')^J = \mathrm{TOL}^{1+\mathcal{O}(|\log c'|)},$$

which for c' *close to* 1 *is only a slight change.* □

Let us now show that the total number of elements, for all mesh levels, can be bounded by a constant times the number of elements in the finest mesh with decreasing tolerance. Its proof uses that the tolerance decreases sufficiently, which simplifies the analysis. A more refined study, with less demanding assumptions on the tolerance, following the idea in Remark 3.3 would need deeper understanding of the distribution of the error indicators $\bar{r}_i$. In contrast to the basic Theorems 3.1, 3.3 and 3.4, the following result has the drawback that it uses a uniform bound in (3.14) which yields a condition, on c', that in practice can be too restrictive although it seems reasonable for very small tolerances. The proof is also more complicated and less natural than the previous proofs.

THEOREM 3.5. *The total number of elements satisfies the bound*

$$\sum_{k=1}^{J} N[k] = \mathcal{O}(N[J]),$$

for a variant of the adaptive algorithm in Section 3.4 where all levels have decreasing tolerance

$$\mathrm{TOL}[k+1] = \mathrm{TOL}[k] c'$$

satisfying $0 < c' < 1/\hat{C}$*, provided all initial elements are divided,* $S_1 \geq s_1 2^d/(\hat{c}c')$ *and* (3.14) *holds uniformly for all elements. Proof.* Let $s_2 := s_1/(\hat{C}c'2^{d+2})$ and $\mathcal{N}_0[k] := \{i : s_2 \mathrm{TOL}[k]/N[k] \leq \bar{r}_i[k] \leq s_1 \mathrm{TOL}[k]/N[k]\}$. We shall first verify that

$$\min_{K,k} (\bar{r}_K N/\mathrm{TOL})[k] \geq s_2. \tag{3.39}$$

Assume first that

$$\min_K \bar{r}_K[k] > s_1 \mathrm{TOL}/N[k],$$

then all elements are divided on level $k+1$ and by (3.14)

$$\begin{aligned}\bar{r}(K)[k+1] &= \big(\hat{\rho}(K)h(K)^{d+2}\big)[k+1] \\ &\geq \frac{1}{\hat{C}}\hat{\rho}(K)[k]\frac{h^{d+2}(K)[k]}{2^{d+2}} \\ &= \frac{1}{\hat{C}2^{d+2}}\bar{r}(K)[k] \\ &> \frac{s_1\mathrm{TOL}[k]}{\hat{C}2^{d+2}N[k]}\end{aligned}$$

therefore

$$\begin{aligned}\min_K \bar{r}(K)[k+1] &> \frac{s_1\mathrm{TOL}[k+1]}{\hat{C}c'2^{d+2}N[k+1]} \\ &= \frac{s_2\mathrm{TOL}[k+1]}{N[k+1]}.\end{aligned}$$

Then if $K \in \mathcal{N}_0[k]$ the element K is not divided on level $k+1$, unless the hanging node constraint required division but then the error indicator is bigger than its source of the hanging node constraint, see the proof of Theorem 3.4. For K which is not divided on level $k+1$ it holds that

$$\begin{aligned}\bar{r}(K)[k+1] &\geq \frac{1}{\hat{C}}\,\bar{r}(K)[k] \\ &\geq \frac{1}{\hat{C}}\,s_2\,\mathrm{TOL}[k]/N[k] \\ &\geq \frac{1}{\hat{C}c'}\,s_2\,\mathrm{TOL}[k+1]/N[k+1] \\ &> s_2\,\mathrm{TOL}[k+1]/N[k+1].\end{aligned}$$

Therefore we conclude, by induction, that the error indicators satisfy

$$\min_{K,k}(\bar{r}(K)N/\mathrm{TOL})[k] \geq s_2.$$

The next step is show that at most m consecutive levels can have the slow increase $N[k]/N[k+1] > \bar{c}$. This will imply that the total number of elements is bounded by a constant times the final number of elements. Assume the contrary that

$$\frac{N[k]}{N[k+1]} > \bar{c}, \quad k = \kappa, \ldots, \kappa+m, \tag{3.40}$$

where m and $\bar{c}$ are chosen to satisfy

$$\frac{(c')^m}{\hat{c}} < \frac{s_2}{s_1}, \tag{3.41}$$

$$1 < \bar{c}^{-1} < 2^{d/(m+1)}, \tag{3.42}$$

and let $N_0[k] := \#\mathcal{N}_0[k]$ and $N_+ := N - N_0$. The bound (3.39) shows that error indicators are either in $\mathcal{N}_0$ or in the refinement region where $\bar{r}_i > s_1 \mathrm{TOL}/N$. Therefore we have

$$N[k+1] = N_0[k] + 2^d N_+[k]$$

which combined with (3.40) show that the number of divided elements, $N_+[k]$, satisfies

$$N_+[k] < \frac{\bar{c}^{-1} - 1}{2^d - 1} N[k]. \tag{3.43}$$

The tolerance decreases, so that after m levels the dividing barrier is

$$s_1 \mathrm{TOL}[\kappa + m]/N[\kappa + m] < (c')^m s_1 \mathrm{TOL}[\kappa]/N[\kappa].$$

All elements in $\mathcal{N}_0[\kappa]$ must have been divided after m levels, since if they have not all been divided some error indicator is larger than $\hat{c} s_2 \mathrm{TOL}[\kappa]/N[\kappa]$ and condition (3.41) gives the contradiction that an element with error indicator, $\bar{r}_K[\kappa + m]$, in the dividing region will not be divided

$$s_1 \frac{\mathrm{TOL}[\kappa + m]}{N[\kappa + m]} < (c')^m s_1 \frac{\mathrm{TOL}[\kappa]}{N[\kappa]} < \hat{c} s_2 \frac{\mathrm{TOL}[\kappa]}{N[\kappa]} < \bar{r}_K[\kappa + m].$$

Dividing of all elements in $\mathcal{N}_0[\kappa]$ shows that $N_0[\kappa]$ must be smaller than the sum of divided elements

$$N_0[\kappa] \leq \sum_{j=1}^{m} N_+[\kappa + j] \tag{3.44}$$

which also leads to a contradiction, since by (3.43)

$$N_0[\kappa] = N[\kappa] - N_+[\kappa] > \frac{2^d - \bar{c}^{-1}}{2^d - 1} N[\kappa]$$

and by combining (3.43) and (3.40)

$$\begin{aligned} N_+[\kappa + j] &< \frac{\bar{c}^{-1} - 1}{2^d - 1} N[\kappa + j] \\ &< \frac{\bar{c}^{-1} - 1}{2^d - 1} \bar{c}^{-1} N[\kappa + j - 1] \\ &< \frac{\bar{c}^{-1} - 1}{2^d - 1} \bar{c}^{-j} N[\kappa], \end{aligned}$$

so that by the assumption (3.42)

$$N_0[\kappa] - \sum_{j=1}^{m} N_+[\kappa + j] > \frac{2^d - \bar{c}^{-m-1}}{2^d - 1} N[\kappa] > 0,$$

which contradicts (3.44). Hence, the number of consecutive levels, where $N[k]/N[k+1] > \bar{c}$, must be smaller than $m+1$ and therefore

$$\sum_{k=1}^{J} N[k] \leq \frac{mN[J]}{1-\bar{c}} = \mathcal{O}(N[J]).$$

□

4 An Application

In this section we study a simplified elasticity problem related to a round corner, which yields a solution with multiple scales, and show that the adaptive algorithm solves this problem more efficiently than with a uniform mesh; the number of adaptive elements divided by the number of uniform elements becomes $\mathcal{O}(\epsilon)$, for the same error, where ϵ is related to the radius of the corner.

The numerical results in this section were obtained with an implementation of the adaptive algorithm in Section 3.4 using the error expansion (2.20) and the approximate error density (2.14). Table 4.1 compares different choices of the averaging of the second differences and Figure 4.4 shows that the requirements on $\hat{C}$ and $\hat{c}$ in (3.14) are fulfilled.

EXAMPLE 4.1. *An example with adaptive gain is computation of the functional*

$$(u,1) = \int_\Omega u \, dx,$$

where the function u solves the Laplace equation in a domain which generates different scales

$$\begin{aligned} -\Delta u &= 0, && \text{in } \Omega = \{x \in \mathbb{R}^2 : \ \epsilon < |x| < 1\} \setminus \Gamma_0, \\ u &= g, && \text{on } \Gamma_C = \{x \in \mathbb{R}^2 : \ |x| = \epsilon\}, \\ u &= 0, && \text{on } \Gamma_0 = (\epsilon, 1] \times \{0\}, \\ u &= g, && \text{on } \partial\Omega \setminus (\Gamma_0 \cup \Gamma_C). \end{aligned}$$

Here g is the function $(r,\theta) \mapsto r^{1/2}\sin(\theta/2)$, where (r,θ) are the polar coordinates, so that the exact solution, u, is $r^{1/2}\sin(\theta/2)$. Note that the Dirichlet boundary condition on Γ_C is equivalent to the Robin boundary condition

$$\frac{\partial u}{\partial n} + \frac{u}{2\epsilon} = 0, \qquad \text{on } \Gamma_C, \tag{4.1}$$

which yields the dual problem

$$-\Delta\varphi = 1 \quad \text{in } \Omega, \quad \varphi|_{\partial\Omega\setminus\Gamma_C} = 0, \quad \left.\frac{\partial\varphi}{\partial n} + \frac{\varphi}{2\epsilon}\right|_{\Gamma_C} = 0.$$

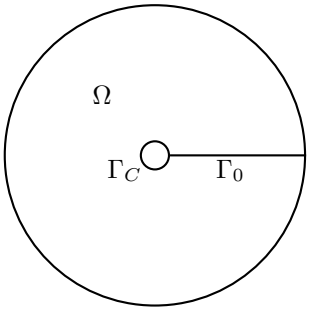


Figure 4.1: Domain with a crack with rounded tip

An expansion of φ, by separation of variables, implies that the singular mode $r^{1/2}\sin(\theta/2)$ is present also in φ. Following (2.22)*, this mode $r^{1/2}\sin(\theta/2)$ in u and φ therefore yields*

$$\hat{\rho}(x) = \frac{\mathcal{O}(1)}{r^3}, \quad \text{for } r > \epsilon, \tag{4.2}$$

where the leading order term in $\mathcal{O}(1)$ is positive. This implies by (3.28) *that the optimal number of adaptive elements N, for error* TOL*, satisfies*

$$\text{TOL } N = \mathcal{O}(1), \quad \text{as TOL} \to 0^+,$$

while (3.27) *shows that the number of uniform elements for the same error grows much faster for small ϵ,*

$$\text{TOL } N^{\text{uni}} = \mathcal{O}(1)/\epsilon, \quad \text{as TOL} \to 0^+.$$

Our numerical tests are for simplicity in computer implementation on a qualitatively related problem: A crude approximation of a round corner at Γ_C on a square grid is to replace the boundary condition on $\Gamma_C \cup \Gamma_0$ with the penalty formulation: $\min(\|\nabla u\|^2_{L^2(\Omega)} + \beta\|u\|^2_{L^2(\Gamma_0)})$ for $\beta \gg 1$ with the computational domain $\Omega = [-1,1]^2 \setminus \Gamma_0$. This gives the Robin boundary condition

$$\frac{\partial u}{\partial n} + \beta u = 0, \quad \text{on } \Gamma_0, \tag{4.3}$$

the Dirichlet condition $u = g$ on $\partial\Omega \setminus \Gamma_0$, and $\Delta u = 0$ in Ω. We see from Theorem 2.1 and Lemma 3.2 that the error density converges with a relative error $\frac{\text{TOL}^{1/4}\mathcal{O}(1)}{\epsilon}$ in the case of a rounded tip and similarly for the Robin boundary condition (4.3) *with $\beta \simeq \epsilon^{-1}$, so to have theoretical estimates on the error density very small tolerances are necessary. Numerical results show that the bound* (3.14) *holds with $\hat{c}$ independent of ϵ and* TOL*, which makes the algorithm useful also for larger tolerances.*

4.1 Numerical results for $\epsilon = 0$

In Table 4.1 and Figures 4.2–4.4 below we present numerical results for the singular case $\epsilon = 0$.

In practice the adaptive algorithm works also when the error density is computed without averaging. The proofs of Theorems 3.1, 3.3, and 3.4 use uniform convergence of $\hat{\rho}$, since the algorithm uses local properties of $\hat{\rho}$; on the other hand, the error is an integral over Ω and therefore weaker convergence of $\hat{\rho}$ may suffice to estimate the error. Figure 4.4 shows that the minimal requirements on $\hat{c}$ and $\hat{C}$, defined in (3.14)*, to prove Theorems 3.1, 3.3, and 3.4 behave well.*

refinements		N	error	error estimate
uniform		32768	$4.8 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$
adaptive	(a)	20288	$4.4 \cdot 10^{-6}$	$6.9 \cdot 10^{-6}$
	(b)	18203	$5.1 \cdot 10^{-6}$	$7.7 \cdot 10^{-6}$
	(c)	18929	$4.8 \cdot 10^{-6}$	$7.1 \cdot 10^{-6}$
	(d)	16634	$5.5 \cdot 10^{-6}$	$8.0 \cdot 10^{-6}$
	(e)	18884	$5.1 \cdot 10^{-6}$	$7.5 \cdot 10^{-6}$

Table 4.1: The adaptive algorithm (3.11)–(3.12) uses far less elements than the number of uniform elements needed to get comparable accuracy in Example 4.1. The error $(u - u_h, 1)$ is estimated by (2.5) using signed error density, spatially varying averaging $\alpha(x) = \sqrt{r(x)h(x)}$ or $\alpha = 0$, and the following combinations of parameters:
(a) $\alpha = \sqrt{rh}$, $\delta = \sqrt{\mathrm{TOL}}$,
(b) $\alpha = 0$, $\delta = \sqrt{\mathrm{TOL}}$,
(c) $\alpha = \sqrt{rh}$, $\delta = 0$,
(d) $\alpha = 0$, $\delta = 0$,
(e) local averaging over 5 nodes, $\delta = \sqrt{\mathrm{TOL}}$.
They all give similar error and number of elements for the tolerance $\mathrm{TOL} = 2^{-15}$. The computations used $s_1 = 1$ and $S_1 = 10$.

The numerical computations in the previous example give an optimal refinement near the corner singularity, $h \approx \sqrt{\mathrm{TOL}} r^{3/4}$, which is the same needed to control the error in u and φ, measured with the energy norm. This fact is a direct consequence of u and φ having the same behavior in the vicinity of the corner singularity.

However, if we consider instead an example where the solution u is smooth, $D^2 u = \mathcal{O}(1)$, and φ is related to the computation of a point value, then we may have $\rho \sim r^{-3}$ yielding a more stringent refinement than the one needed to control the error in u with the energy norm.

Acknowledgment.

The authors thank Professor Ivo Babuška and Professor Stig Larsson for useful comments.

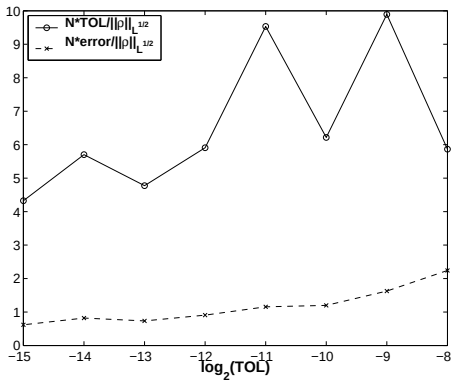


Figure 4.2: Efficiency estimate of the accepted mesh in the case (a) in Table 4.1 for a sequence of tolerances. The number of elements, N, on the accepted mesh is compared to the estimated optimal $N^{\text{opt}} = ||\hat{\rho}||_{L^{1/2}}/\text{TOL}$, where the quasi-norm of the error density is computed on the finest mesh, corresponding to $\text{TOL} = 2^{-15}$.

REFERENCES

1. M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Comput. Methods Appl. Mech. Engrg., 142 (1997), pp. 1-88.
2. I. Babuška, J. Hugger, T. Strouboulis, K. Copps, and S. K. Gangaraj, *The asymptotically optimal meshsize function for bi-p degree interpolation over rectangular elements* J. Comput. Appl. Math., 90 (1998), no. 2, pp. 185–221.
3. I. Babuška, A. Miller, and M. Vogelius, *Adaptive methods and error estimation for elliptic problems of structural mechanics*, in Adaptive computational methods for partial differential equations, SIAM, Philadelphia, Pa., 1983, pp. 57–73.
4. I. Babuška and W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., 15 (1978), pp. 736–754.
5. I. Babuška, T. Strouboulis, and S. K. Gangaraj, *Guaranteed computable bounds for the exact error in the finite element solution. I. One dimensional model problem*, Comput. Methods Appl. Mech. Engrg., 176 (1999), pp. 51–79.
6. I. Babuška and M. Vogelius, *Feedback and adaptive finite element solution of one-dimensional boundary value problems*, Numer. Math., 44 (1984), no. 1, pp. 75–102.
7. R. E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp., 44 (1985), pp. 283–301.
8. N. S. Bakhvalov, *On the optimality of linear methods for operator approximation in convex classes of functions*, USSR Comp. Math. and Math. Phys., 11 (1971), pp. 244–249.
9. R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: basic analysis and examples*, East-West J. Numer. Math., 4 (1996), no. 4, pp. 237–264.
10. R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica, vol. 10, (2001), pp. 1–102.

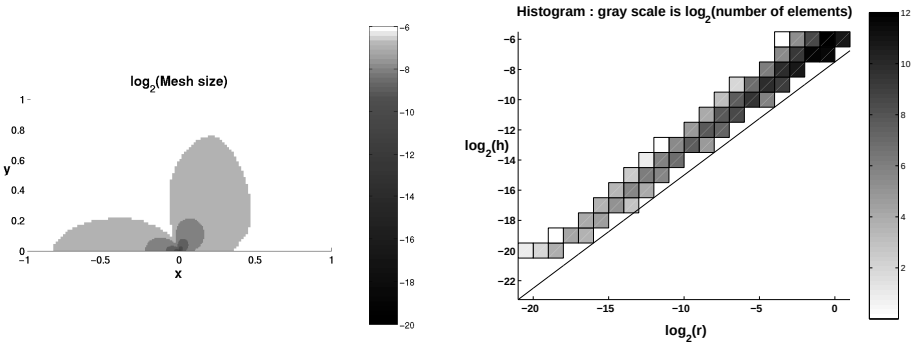


Figure 4.3: This is the mesh of the accepted solution for case (a) in Table 4.1. The other cases produce similar meshes. The element sizes h vary with distance from the origin approximately like $r^{3/4}$. The reference line is the function $\sqrt{\text{TOL}}\, r^{3/4}$.

11. P. Binev, W. Dahmen, and R. DeVore, *Adaptive finite element methods with convergence rates*, Numer. Math., 97 (2004), no. 2, pp. 219–268.
12. S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Texts in Applied Mathematics 15, Springer–Verlag, New York, 1994.
13. A. Cohen, W. Dahmen and R. DeVore, *Adaptive wavelet methods for elliptic operator equations: convergence rates*, Math. Comp., 70 (2001), no. 233, pp. 25–75.
14. F. Christian and G. Santos, *A posteriori estimators for nonlinear elliptic partial differential equations*, J. Comput. Appl. Math., 103 (1999), pp. 99–114.
15. R. A. DeVore, *Nonlinear approximation*, Acta Numerica, vol. 7, (1998), pp. 51–150.
16. W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33 (1996), no. 3, pp. 1106–1124.
17. H. Edelsbrunner, D. R. Grayson, *Edgewise subdivision of a simplex.*, ACM Symposium on Computational Geometry, (Miami, FL, 1999), Discrete Comput. Geom., 24 (2000), pp. 707–719.
18. K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numerica, vol. 4, (1995), pp. 105–158.
19. C. Johnson and A. Szepessy, *Adaptive finite element methods for conservation laws based on a posteriori error estimates*, Comm. Pure Appl. Math., 48 (1995), pp. 199–234.
20. L. Machiels, A. T. Patera, J. Peraire, and Y. Maday, *A general framework for finite element a posteriori error control: application to linear and nonlinear convection-dominated problems*, in "Proceedings of ICFD Conference on Numerical Methods for Fluid Dynamics", Numerical Methods for Fluid Dynamics VI, Oxford, England, 1998.
21. Y. Maday, A. T. Patera and, J. Peraire, *A general formulation for a posteriori bounds for output functionals of partial differential equations; applications to the eigenvalue problem*, C. R. Acad. Sci. Paris Sér. I Math., 328 (1999), pp. 823–828.
22. K.-S. Moon, *Adaptive algorithms for deterministic and stochastic differential equations*, Doctoral thesis, ISBN 91-7283-553-2, Royal Institute of Technology, Stockholm, 2003, `http://www.nada.kth.se/~moon`

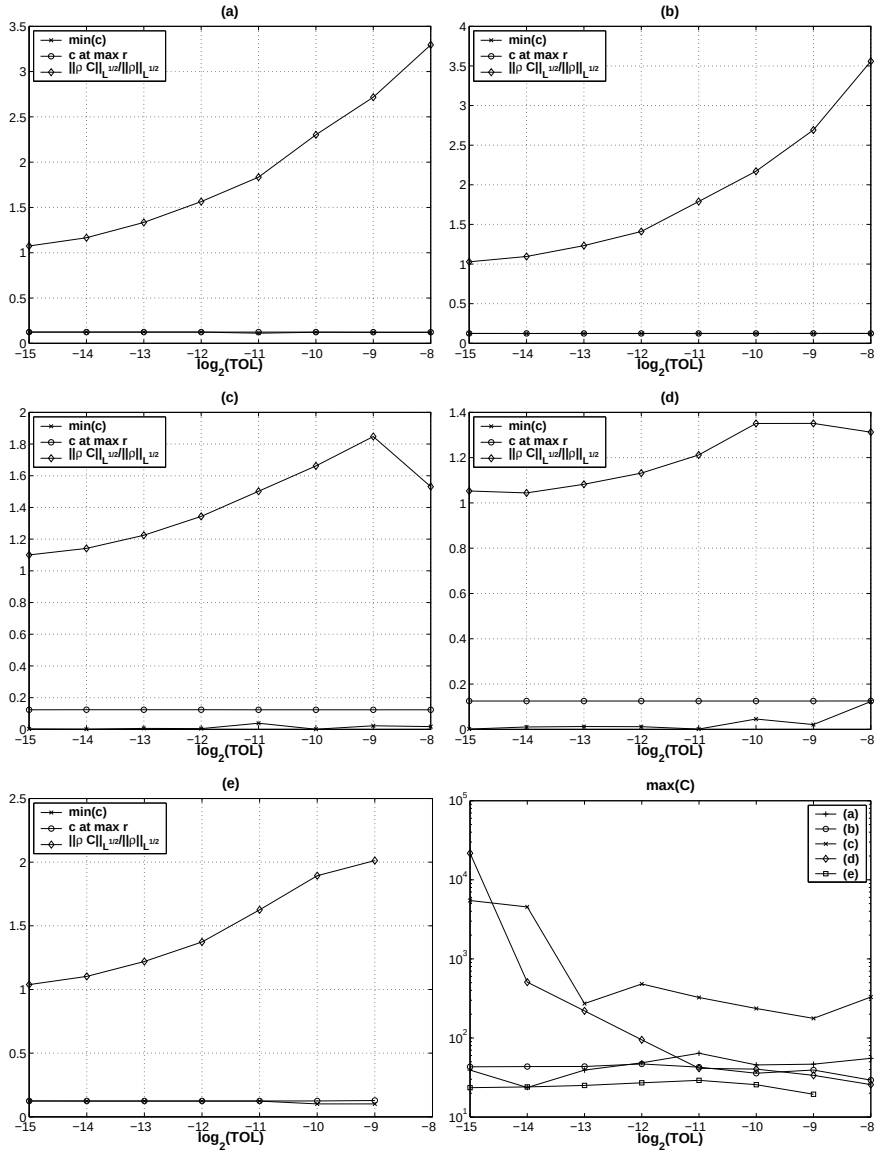


Figure 4.4: For the five cases (a)–(e) in Table 4.1, the quotients $\hat{C}(K) = \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]}$ and $\hat{c}(K) = \min\{\frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]}, \frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]}\}$ have been computed for a sequence of tolerances. The values shown for $\hat{c}$ are the minimal over all levels, k, while for $\hat{C}(K)$, K is an element on the accepted mesh. The minimal requirements for Theorems 3.1 and 3.3 use bounds on $\hat{c}^{-1}$ for the maximal error indicator, which is here approximately 8, because of (4.2) and $\epsilon = 0$. For Theorem 3.4 the minimal requirements use bounds on $||\hat{\rho}\hat{C}||_{L^{1/2}}$, which is close to $||\hat{\rho}||_{L^{1/2}}$ computed on the accepted mesh using TOL $= 2^{-15}$; the maximal $\hat{C}$ may, especially with $\delta = 0$, be significantly larger.

23. K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *Convergence rates for adaptive approximation of ordinary differential equations*, Numer. Math., 96 (2003), pp. 99–129.
24. K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *Convergence rates for adaptive weak approximation of stochastic differential equations*, Stoch. Anal. Appl., 23 (2005), no. 3, pp. 511–558.
25. K.-S. Moon, A. Szepessy, R. Tempone, and G.E. Zouraris, *Hyperbolic differential equations and adaptive numerics*, in Theory and numerics of differential equations, Durham 2000, J. F. Blowey, J.P. Coleman, and A.W. Craig, eds., Springer Verlag, Berlin, 2001.
26. P. Morin, R.H. Nochetto, and K.G. Siebert, *Convergence of adaptive finite element methods*, SIAM Rev. 44 (2002), no. 4, pp. 631–658.
Revised reprint of *Data oscillation and convergence of adaptive FEM*, SIAM J. Numer. Anal. 38 (2000), no. 2, pp. 466–488.
27. R. Stevenson, *An optimal adaptive finite element method*, SIAM J. Numer. Anal., 42 (2005), pp. 2188–2217.
28. A. Szepessy, R. Tempone, and G. E. Zouraris, *Adaptive weak approximation of stochastic differential equations*, Comm. Pure Appl. Math., 54 (2001), pp. 1169–1214.

Paper II

Adaptive Monte Carlo Algorithms for Stopped Diffusion

Anna Dzougoutov[1], Kyoung-Sook Moon[2], Erik von Schwerin[1], Anders Szepessy[1,3], and Raúl Tempone[4]

[1] Department of Numerical Analysis and Computer Science, KTH, S–100 44 Stockholm, Sweden
annadz@kth.se, schwerin@nada.kth.se
[2] Department of Mathematics, University of Maryland, College Park, MD 20742, USA,
moon@math.umd.edu
[3] Department of Mathematics, KTH, S–100 44 Stockholm, Sweden,
szepessy@nada.kth.se
[4] ICES, The University of Texas at Austin, 1 Texas Longhorns, Austin, Texas 78712, and
IMERL, Facultad de Ingeniería, Julio Herrera y Reissig 565, 11200 Montevideo Uruguay,
rtempone@ices.utexas.edu

Summary. We present adaptive algorithms for weak approximation of stopped diffusion using the Monte Carlo Euler method. The goal is to compute an expected value $E[g(X(\tau),\tau)]$ of a given function g depending on the solution X of an Itô stochastic differential equation and on the first exit time τ from a given domain. The adaptive algorithms are based on an extension of an error expansion with computable leading order term, for the approximation of $E[g(X(T))]$ with a fixed final time $T > 0$ and diffusion processes X in $\mathbb{R}^d$, introduced in [17] using stochastic flows and dual backward solutions. The main steps in the extension to stopped diffusion processes are to use a conditional probability to estimate the first exit time error and introduce difference quotients to approximate the initial data of the dual solutions. Numerical results show that the adaptive algorithms achieve the time discretization error of order N^{-1} with N adaptive time steps, while the error is of order $N^{-1/2}$ for a method with N uniform time steps.

Key words: adaptive mesh refinement algorithm, diffusion with boundary, barrier option, Monte Carlo method, weak approximation

1 Introduction

In this paper, we compute adaptive approximations of an expected value

$$E[g(X(\tau),\tau)] \tag{1}$$

of a given function, $g : D \times [0,T] \to \mathbb{R}$, where the stochastic process X solves an Itô stochastic differential equation (SDE)

$$\mathrm{d}X_i(t) = a_i(X(t))\,\mathrm{d}t + \sum_{l=1}^{l_0} b_i^l(X(t))\,\mathrm{d}W^l(t)\ ,\ i = 1,2,\ldots,d,\ t > 0 \tag{2}$$

and τ is the first exit time

$$\tau := \inf\{0 < t : (X(t),t) \notin D \times (0,T)\} \tag{3}$$

from a given open domain $D \times (0,T) \subset \mathbb{R}^d \times (0,T)$. The functions $a : \mathbb{R}^d \to \mathbb{R}^d$ and $b^l : \mathbb{R}^d \to \mathbb{R}^d$ for $l = 1,2,\ldots,l_0$, are given drift and diffusion fluxes and $W^l(t;\omega)$ for $l = 1,2,\ldots,l_0$, are independent Wiener processes. Such problems arise in physics and finance, for instance when computing the value of barrier options.

In the case when the dimension of the problem is large or when the related partial differential equation is difficult to formulate or to solve, the Monte Carlo Euler method is used to compute the expected value. The main difficulty in the approximation of the stopped (or killed) diffusion on the boundary ∂D is that a continuous sample path may exit the given domain D even though a discrete approximate solution does not cross the boundary of D. This hitting of the boundary makes the time discretization error $N^{-1/2}$ for the Monte Carlo Euler method with N uniform time steps, see [7], while the discretization error is of order N^{-1} without stopping boundary in $\mathbb{R}^d \times [0,T)$. The work [13] and [9] reduce the large $N^{-1/2}$ first exit error to N^{-1}. The idea is to generate a uniformly distributed random variable in $(0,1)$ for each time step and compare it with a known exit probability to decide if the continuous path exits the domain during this time interval. A similar method with N uniform time steps in a domain with smooth boundary is proven to converge with the rate N^{-1} under some appropriate assumptions in [8]. Different Monte Carlo methods for stopped diffusions are compared computationally in [5]. To use these methods, the exit probability needs to be computed accurately.

Inspired by Petersen and Buchmann [16], this work uses the alternative to reduce the computational error by choosing adaptively the size of the time steps near the boundary, which has the advantage that the exit probability does not need to be computed accurately. Section 2 derives an expansion of the error with computable leading order term. Section 3 presents an adaptive algorithm based on the error estimate where the time discretization error is of order N^{-1} with N adaptive time steps.

Using the Monte Carlo Euler method, the expected value (1) can be approximated by a sample average of $g(\overline{X}(\overline{\tau}),\overline{\tau})$, where $(\overline{X},\overline{\tau})$ is an Euler approximation of (X,τ). The global error can then be split into time discretization error and statistical error,

$$
\begin{aligned}
&E[g(X(\tau),\tau)] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(\overline{\tau};\omega_j),\overline{\tau}) \\
&= \big(E[g(X(\tau),\tau) - g(\overline{X}(\overline{\tau}),\overline{\tau})]\big) + \left(E[g(\overline{X}(\overline{\tau}),\overline{\tau})] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(\overline{\tau};\omega_j),\overline{\tau})\right) \\
&=: \mathcal{E}_T + \mathcal{E}_S \qquad (4)
\end{aligned}
$$

where M is the number of realizations. The statistical error, $\mathcal{E}_S$ in (4), is asymptotically bounded by $c_0\overline{\sigma}/\sqrt{M}$ using the Central Limit Theorem, where $\overline{\sigma}$ is the sample average of the standard deviation of $g(\overline{X}(\overline{\tau}),\overline{\tau})$ and c_0 is a positive constant for a confidence interval, see Sect. 3.1.

Talay and Tubaro [18] and Bally and Talay [4] prove an a priori error expansion of $E[g(X(T)) - g(\overline{X}(T))]$ for the case without stopping boundary, i.e. for diffusion processes in $\mathbb{R}^d \times [0,T]$. In the same setting without a stopping boundary, the work [17] proves an expansion of the error with computable leading order term, error $\simeq E\left[\sum_{n=1}^{N} r_n\right]$, using an error density, $\rho = r_n/\Delta t_n^2$, which depends on computable discrete primal and dual solutions. Given this error estimate, consider an algorithm which for each realization refines the solution, $\overline{X}$, by the adaptive time stepping:

for all time steps $n = 1, \ldots, N$
  if $\left(r_n \geq \dfrac{\mathrm{TOL}_T}{E[N]}\right)$ **then**
    divide Δt_n into 2 equal substeps, and generate
    the intermediate value of W by the Brownian bridge (5),
  else let the new step be the same as the old
  endif
endfor,

with the stopping criterion:

$$\textbf{if } \left(\max_{1\leq n\leq N} r_n < S\frac{\mathrm{TOL}_T}{E[N]}\right) \textbf{ then stop.}$$

The intermediate sample points from W are constructed by the Brownian bridge, cf. [10],

$$W^l\left(\frac{t_n + t_{n+1}}{2}\right) = \frac{1}{2}\left(W^l(t_n) + W^l(t_{n+1})\right) + z_n^l \qquad (5)$$

where z_n^l are independent random variables in $N(0,(t_{n+1}-t_n)/4)$, i.e. they are normally distributed with mean 0 and variance $(t_{n+1}-t_n)/4$, independent also of previous $W^l(t_j)$. Letting c_0 be the confidence interval parameter, related to the statistical error $c_0\overline{\sigma}/\sqrt{M} \simeq \mathrm{TOL}_S$, in (4), with $\mathrm{TOL} = 3\mathrm{TOL}_T =$

$3\mathrm{TOL}_S/2$, and assuming $S > C$ are constants such that $C^{-1} \leq \frac{\rho_{parent}}{\rho_{child}} \leq C$, the work [15] proves that the algorithm stops with asymptotically optimal expected number of time steps and the error asymptotically bounded by TOL with large probability (up to problem independent factors):

$$E[N] \lesssim 4CE[N_{optimal}] \text{ and } P(\frac{\text{error}}{\text{TOL}} \leq \frac{S}{3} + \frac{2}{3}) \gtrsim (2\pi)^{-1/2} \int_{-c_0}^{c_0} \mathrm{e}^{-x^2/2} \mathrm{d}x \ .$$

In Sect. 2, we approximate the time discretization error, $\mathcal{E}_T$ in (4), in computable form by extending the error estimate in [17] to weak approximation of stopped diffusion. As in [18] and [17], the first step to derive an error estimate is to introduce a continuous Euler path. Then the error between the exact and continuous Euler path is approximated using stochastic flows and dual backward solutions in Sect. 2.3. The main idea in this extension is to use difference quotients to replace the stochastic flows that do not exist at the boundary. The approximate error between the continuous and the discrete Euler path is derived by a conditional probability using Brownian bridges in Sect. 2.2. Note that the exit probability is used here only to decide the time steps, not to approximate the expected values directly. Therefore the accuracy of the approximation of the exit probability is not crucial.

The computation of the dual solutions may be costly in high dimension. A simplified variant of the algorithm based on the local error is obtained by replacing the dual solutions by 1.

The paper is organized as follows. The computable error estimate for stopped diffusions is derived in the next section and based on this error estimate we develop adaptive algorithms in Sect. 3. Finally some numerical results of adaptive refinements in one and two space dimension are given in Sect.4. This paper is an extension of the preprint paper 5 in [14] where stopped diffusion in one dimension is studied.

2 Error Expansion

Consider a domain $D \subset \mathbb{R}^d$ and assume that the initial position $X(0) = X_0$ lies in D. The goal is to compute the expected value $E[g(X(\tau), \tau)]$ of a given function g which depends on the stochastic process X and the first exit time τ defined in (3).

First discretize the time interval $[0, T]$ into N subintervals $0 = t_0 < t_1 < \ldots < t_N = T$ and let $\overline{X}$ denote the Euler approximation of the process X; start with $\overline{X}(0) = X_0$ and compute $\overline{X}(t_{n+1})$ for $n = 0, 1, \ldots, N-1$ by

$$\overline{X}_i(t_{n+1}) = \overline{X}_i(t_n) + a_i(\overline{X}(t_n))\, \Delta t_n + \sum_{l=1}^{l_0} b_i^l(\overline{X}(t_n))\, \Delta W_n^l, \quad i = 1, 2, \ldots, d, \tag{6}$$

where $\Delta t_n := t_{n+1} - t_n$ denote time increments and $\Delta W_n^l := W^l(t_{n+1}) - W^l(t_n)$ denote Wiener increments. Approximate the first exit time τ with

$$\overline{\tau} := \min_{1 \le n \le N} \{t_n : (\overline{X}(t_n), t_n) \notin D \times [0,T)\} \tag{7}$$

using the Euler approximation path $\overline{X}$ instead of the exact path X.

Introduce, for theoretical purposes only, a continuous Euler path $\overline{X}(t)$ by

$$\overline{X}_i(t) = \overline{X}_i(t_n) + \int_{t_n}^{t} a_i(\overline{X}(t_n))\,\mathrm{d}t + \sum_{l=1}^{l_0} \int_{t_n}^{t} b_i^l(\overline{X}(t_n))\,\mathrm{d}W_t^l, \quad i = 1, 2, \ldots, d, \tag{8}$$

for $t \in [t_n, t_{n+1})$ and denote by

$$\widetilde{\tau} := \inf\{0 < t : (\overline{X}(t), t) \notin D \times [0,T)\} \tag{9}$$

the exit time of the continuous Euler path. Then the time discretization error of the Euler approximation can be split in two parts:

$$\begin{aligned} &E[g(X(\tau),\tau) - g(\overline{X}(\overline{\tau}),\overline{\tau})] \\ &\qquad = E[g(X(\tau),\tau) - g(\overline{X}(\widetilde{\tau}),\widetilde{\tau})] + E[g(\overline{X}(\widetilde{\tau}),\widetilde{\tau}) - g(\overline{X}(\overline{\tau}),\overline{\tau})] \\ &\qquad =: \mathcal{E}_C + \mathcal{E}_D. \end{aligned} \tag{10}$$

In [7], Gobet proves the following a priori error estimate with N uniform time steps:

$$E[f(X(\tau),\tau) - f(\overline{X}(\overline{\tau}),\overline{\tau})] = \mathcal{O}(N^{-1/2}). \tag{11}$$

In order to improve the convergence rate in (11), we adaptively refine the mesh according to computable error estimates. Error estimates for $\mathcal{E}_D$ and $\mathcal{E}_C$ are derived in Theorem 1 and Theorem 2 respectively.

2.1 Notation

In this paper, ∂_i denotes the derivative with respect to x_i, i.e. $\partial_i := \partial/\partial x_i$, and similarly for ∂_{ij} and ∂_{ijk}. If same subscript appears twice in a term, the term denotes the sum over the range of this subscript, e.g., $c_{ik}\partial_k b_j := \sum_k c_{ik}\partial_k b_j$. We use $X_t := X(t)$ and $\overline{X}_t := \overline{X}(t)$ for the continuous cases and $\overline{X}^n := \overline{X}(t_n)$ for the discrete case. The piecewise constant mesh function Δt is defined by

$$\Delta t(s) := \Delta t_n \quad \text{for } s \in [t_n, t_{n+1}) \text{ and } n = 0, 1, \ldots, N-1 \tag{12}$$

and

$$\Delta t_{\max} := \max_{n,\omega} \Delta t_n(\omega).$$

We let $\mathbf{1}_A$ denote the indicator function, i.e. $\mathbf{1}_A(y) = 1$ if $y \in A$, otherwise $\mathbf{1}_A(y) = 0$.

2.2 Expansion of Exiting Error using Probability

Consider the time discretization error between the continuous and the discrete Euler path, denoted by $\mathcal{E}_D$ in (10). In the case when the continuous Euler path ends at time $t = T$, i.e. $\widetilde{\tau} = T = \overline{\tau}$, there is no time discretization error between two Euler paths since $E[g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau})\mathbf{1}_{\widetilde{\tau}=T}] = E[g(\overline{X}_{\overline{\tau}}, \overline{\tau})\mathbf{1}_{\overline{\tau}=T}]$. On the other hand, if the continuous Euler path is stopped at $\widetilde{\tau} < T$ then it is possible that $\widetilde{\tau} < \overline{\tau}$. Figure 1 shows an illustrative Monte Carlo trajectory where the continuous Euler path $\overline{X}(t) \in \mathbb{R}$ exits the domain $D = (-\infty, \lambda)$ at $t = \widetilde{\tau} < T$, but the discrete Euler process $\overline{X}^n$ does not stop until much later, $\overline{\tau} = T$.

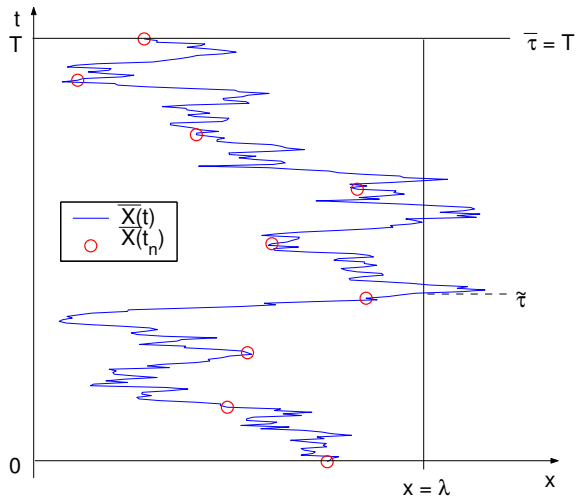


Fig. 1. An illustrative Euler Monte Carlo trajectory when $\widetilde{\tau} < \overline{\tau}$

Taking the above effect into account, $\mathcal{E}_D$ can be estimated using the probability of the continuous Euler path exiting in a time interval (t_n, t_{n+1}) conditioned on the values of $\overline{X}^n$ and $\overline{X}^{n+1}$ in the discrete Euler process. Consider the particular case of a half space $D = \{x \in \mathbb{R}^d : \langle v, x\rangle_{\mathbb{R}^d} < \lambda\}$ for a constant λ and a constant unit vector v. The probability $P_{\overline{X},n}$ of $\overline{X}(t)$ exiting at some $t \in (t_n, t_{n+1})$ has an explicit expression, see e.g. [12], [1],

$$\begin{aligned} P_{\overline{X},n} &:= \mathbb{P}\left[\max_{t\in[t_n,t_{n+1}]} \langle v, \overline{X}_t\rangle_{\mathbb{R}^d} \geq \lambda \middle| \overline{X}^n = z^1, \overline{X}^{n+1} = z^2\right] \\ &= \exp\left(-2\frac{(\lambda - \langle v, z^1\rangle_{\mathbb{R}^d})(\lambda - \langle v, z^2\rangle_{\mathbb{R}^d})}{\sigma^2 \Delta t_n}\right) \end{aligned} \tag{13}$$

where $\langle v, z^1\rangle_{\mathbb{R}^d} < \lambda$ and $\langle v, z^2\rangle_{\mathbb{R}^d} < \lambda$ and $\sigma^2 = v_i b(\overline{X}^n)^\ell_i b(\overline{X}^n)^\ell_j v_j$. The work [3] studies estimates for the exit probability of the Brownian bridge in general cases of one dimension, e.g. with time dependent lower and upper

boundaries. For a family of non degenerate SDEs in high dimension, including the half space case, the exit probabilities are expressed as asymptotic series in [6], [2]. In the more general case, we can approximate D locally near the boundary by its tangent half space and use the approximation of the exit probability for the half space case, see [7], [8].

We have the following error representation for $\mathcal{E}_D$, formulated for the case $D = \{x \in \mathbb{R}^d : x_1 < \lambda\}$:

Theorem 1. *Let $\overline{X}(t)$ and $\overline{X}(t_n)$ be the continuous and discrete Euler approximations defined in* (8) *and* (6) *respectively. Let χ be the σ-algebra generated by $\{\overline{X}^n : n = 0, 1, \ldots, N\}$. Then the error $\mathcal{E}_D$ has the representation*

$$E[g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau})] = E\left[\sum_{n=0}^{N-1} \left(g(\overline{X}_{\xi_n}, \xi_n) - g(\overline{X}_{\overline{\tau}}, \overline{\tau})\right) \widehat{P}_{\overline{X},n}\right] \tag{14}$$

for $\xi_n \in (t_n, t_{n+1})$, $\overline{X}_{\xi_n} = (\lambda, \overline{X}_{2,\xi_n}, \ldots, \overline{X}_{d,\xi_n})$ satisfying

$$g(\overline{X}_{\xi_n}, \xi_n) = E\big[g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) \;\big|\; \chi, \tilde{\tau} \in [t_n, t_{n+1})\big]$$

and where $\widehat{P}_{\overline{X},n}$ are conditional first exit probabilities defined by

$$\begin{aligned} \widehat{P}_{\overline{X},n} &= P_{\overline{X},n} \prod_{k=0}^{n-1} (1 - P_{\overline{X},k}), \qquad\qquad n = 1, 2, \ldots, N-1, \\ \widehat{P}_{\overline{X},0} &= P_{\overline{X},0}, \end{aligned} \tag{15}$$

using the conditional exit probabilities from (13).

Proof. Since $E[(g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau}))\mathbf{1}_{\widetilde{\tau}=T}] = 0$ and $\mathbf{1}_{\widetilde{\tau}<T} = \sum_{n=0}^{N-1} \mathbf{1}_{\widetilde{\tau}\in[t_n, t_{n+1})}$ we obtain

$$E[g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau})] = E\left[\sum_{n=0}^{N-1} (g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau}))\mathbf{1}_{\widetilde{\tau}\in[t_n, t_{n+1})}\right],$$

and after smoothing with the σ-algebra χ generated by $\{\overline{X}^n : n = 0, 1, \ldots, N\}$

$$E[g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau})] = E\left[\sum_{n=0}^{N-1} E\left[\left(g(\overline{X}_{\widetilde{\tau}}, \widetilde{\tau}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau})\right) \mathbf{1}_{\widetilde{\tau}\in[t_n, t_{n+1})} \big| \chi\right]\right]. \tag{16}$$

In the right hand side we have

$$E[g(\overline{X}_{\overline{\tau}}, \overline{\tau})\mathbf{1}_{\widetilde{\tau}\in[t_n, t_{n+1})} \;\big|\; \chi] = g(\overline{X}_{\overline{\tau}}, \overline{\tau})\, \mathbb{P}[\widetilde{\tau} \in [t_n, t_{n+1})|\chi] \tag{17}$$

since $g(\overline{X}_{\overline{\tau}}, \overline{\tau}) \in \chi$, and, using the independence of the different coordinate directions in the Brownian bridge and the mean value theorem for integration

$$E[g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})\mathbf{1}_{\widetilde{\tau}\in[t_n,t_{n+1})}\,|\,\chi] = E\big[g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})\,|\chi,\widetilde{\tau}\in[t_n,t_{n+1})\big]\mathbb{P}[\widetilde{\tau}\in[t_n,t_{n+1})\,|\chi]$$
$$= g(\overline{X}_{\xi_n},\xi_n)\mathbb{P}[\widetilde{\tau}\in[t_n,t_{n+1})\mid\chi], \tag{18}$$

for some $\xi_n\in(t_n,t_{n+1})$, $\overline{X}_{\xi_n}=(\lambda,\overline{X}_{2,\xi_n},\ldots,\overline{X}_{d,\xi_n})$. Inserting (17) and (18) into (16) we get

$$E[g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})-g(\overline{X}_{\overline{\tau}},\overline{\tau})]=E\left[\sum_{n=0}^{N-1}\big(g(\overline{X}_{\xi_n},\xi_n)-g(\overline{X}_{\overline{\tau}},\overline{\tau})\big)\,\mathbb{P}[\widetilde{\tau}\in[t_n,t_{n+1})\,|\,\chi]\right]. \tag{19}$$

To compute the probability in (19), we observe that the event $\{\widetilde{\tau}\in[t_n,t_{n+1})\}$ is equivalent to

$$\big\{\overline{X}_{t\in[t_0,t_1)}\in D,\ldots,\overline{X}_{t\in[t_{n-1},t_n)}\in D,\ \text{and}\ \overline{X}_{t\in[t_n,t_{n+1})}\notin D\big\}$$

and that the events $\big\{\overline{X}_{t\in[t_n,t_{n+1})}\in D\big\}$, for $n=0,1,\ldots,N-1$, are independent with respect to χ. Thus, using the conditional exit probabilities $P_{\overline{X},k}$, we obtain

$$\widehat{P}_{\overline{X},n}:=\mathbb{P}[\widetilde{\tau}\in[t_n,t_{n+1})\mid\chi]=P_{\overline{X},n}\prod_{k=0}^{n-1}(1-P_{\overline{X},k})$$

and

$$\widehat{P}_{\overline{X},0}:=\mathbb{P}[\widetilde{\tau}\in[t_0,t_1)\mid\chi]=P_{\overline{X},0},$$

which together with (19) proves (14).

Remark 1. For uniform time steps we know from [7] that $E[g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})-g(\overline{X}_{\overline{\tau}},\overline{\tau})]=\mathcal{O}(\sqrt{\Delta t})$. To obtain a computable approximation of $E\big[g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})\ \big|\ \chi,\widetilde{\tau}\in[t_n,t_{n+1})\big]$ approximate by a linear function

$$g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})=g(\overline{X}(t_n),t_n)+B(\overline{X}_{\widetilde{\tau}}-\overline{X}(t_n))+\mathcal{O}(|\overline{X}_{\widetilde{\tau}}-\overline{X}(t_n)|^2+|\widetilde{\tau}-t_n|).$$

The last two terms have expected value $E[\ldots|\chi,\widetilde{\tau}\in[t_n,t_{n+1})]=\mathcal{O}(\Delta t)$ and $\overline{X}_{\xi_n}=(\lambda,\overline{X}_{2,\xi_n},\ldots,\overline{X}_{d,\xi_n})$ is based on pinned Brownian motions $Y:=(\overline{X}_2,\ldots,\overline{X}_d)$ independent of $\widetilde{\tau}$. Hence the expected value $E[Y|\chi,\widetilde{\tau}\in[t_n,t_{n+1})]$ is

$$Y(t_n)+(Y(t_{n+1}-Y(t_n))\frac{E[\widetilde{\tau}|\chi,\widetilde{\tau}\in[t_n,t_{n+1})]-t_n}{t_{n+1}-t_n}.$$

The expected value $E[\widetilde{\tau}|\chi,\widetilde{\tau}\in[t_n,t_{n+1})]$ can be calculated from the explicit probability distribution of the exit time for Brownian bridges in [1].

2.3 Error Expansion Using Dual Solutions

In this subsection, we derive a computable error estimate between the exact and the continuous Euler path, i.e. $\mathcal{E}_C$ in (10). The main result is stated in Theorem 2 and the proof is presented afterwards.

The error estimate uses the discrete dual functions $\varphi(t_n)$, $\varphi'(t_n)$ and $\varphi''(t_n)$, taking values in $\mathbb{R}^d$, $\mathbb{R}^{d^2}$ and $\mathbb{R}^{d^3}$ respectively, defined as follows. For simplicity we describe the case when D is the half space $\{x : x_1 < \lambda\}$; see Remark 2. Introduce the notation

$$\begin{aligned} c_i(t_n, x) &= x_i + \Delta t_n a_i(x) + b_i^l(x)\Delta W_n^l, && i = 1, 2, \ldots, d, \\ \beta_{ij}(x) &= \frac{1}{2} b_i^l(x) b_j^l(x), && i, j = 1, 2, \ldots, d. \end{aligned}$$

Then the function φ is defined by the dual backward problem

$$\varphi_i(t_n) = \partial_i c_j(t_n, \overline{X}^n)\varphi_j(t_{n+1}), \qquad t_n < \overline{\tau}, \quad i = 1, 2, \ldots, d, \tag{20}$$

$$\varphi_i(\overline{\tau}) = \begin{cases} \partial_i g(\overline{X}_{\overline{\tau}}, \overline{\tau}), & \text{if } \overline{\tau} = T, \quad i = 1, 2, \ldots, d, \text{ or} \\ & \text{if } \overline{\tau} < T \text{ and } i = 2, \ldots, d, \\ -(g(\hat{\overline{X}}_{\hat{\overline{\tau}}}, \hat{\overline{\tau}}) - g(\overline{X}_{\overline{\tau}}, \overline{\tau}))/\Delta x, & \text{if } \overline{\tau} < T \text{ and } i = 1; \end{cases} \tag{21}$$

since $\partial_1 g(\overline{X}_{\overline{\tau}}, \overline{\tau})$ does not exist if $\overline{\tau} < T$ we have introduced the restarted Euler approximation $\hat{\overline{X}}(t_n)$ for $t_n \in [\overline{\tau}, \hat{\overline{\tau}}]$ with initial value $\hat{\overline{X}}(\overline{\tau}) = \overline{X}(\overline{\tau}) + \gamma\Delta x$, where γ is an inward unit normal vector, Δx is a small positive number and $\hat{\overline{\tau}}$ denotes the first exit time of $\hat{\overline{X}}$, i.e. $\hat{\overline{\tau}} := \min\{t_n : \overline{\tau} < t_n \text{ and } \hat{\overline{X}}^n \notin D\}$. The first variation φ' satisfies, cf. [17],

$$\begin{aligned} \varphi'_{ik}(t_n) &= \partial_i c_j(t_n, \overline{X}^n)\partial_k c_m(t_n, \overline{X}^n)\varphi'_{jm}(t_{n+1}) \\ &\quad + \partial^2_{ik} c_j(t_n, \overline{X}^n)\varphi_j(t_{n+1}), && t_n < \overline{\tau}, \end{aligned} \tag{22}$$

$$\varphi'_{ik}(\overline{\tau}) = \delta^2_{ik} g(\overline{X}_{\overline{\tau}}, \overline{\tau}), \tag{23}$$

where we interpret $\delta^2_{ik} g(\overline{X}_{\overline{\tau}}, \overline{\tau})$ as the corresponding second derivatives when possible and make use of difference quotients otherwise. If no simplifying property of the domain D and the drift b_i^l is present we may use additional restarted processes, similar to $\hat{\overline{X}}$, and difference quotients to define the initial values of φ' and φ''. Interpreting $\delta^3_{ikp} g(\overline{X}_{\overline{\tau}}, \overline{\tau})$ analogously to $\delta^2_{ik} g(\overline{X}_{\overline{\tau}}, \overline{\tau})$, the second variation φ'' satisfies

$$\begin{aligned} \varphi''_{ikp}(t_n) &= \partial_i c_j(t_n, \overline{X}^n)\partial_k c_m(t_n, \overline{X}^n)\partial_p c_r(t_n, \overline{X}^n)\varphi''_{jmr}(t_{n+1}) \\ &\quad + \partial^2_{ip} c_j(t_n, \overline{X}^n)\partial_k c_m(t_n, \overline{X}^n)\varphi'_{jm}(t_{n+1}) \\ &\quad + \partial_i c_j(t_n, \overline{X}^n)\partial^2_{kp} c_m(t_n, \overline{X}^n)\varphi'_{jm}(t_{n+1}) \\ &\quad + \partial^2_{ik} c_j(t_n, \overline{X}^n)\partial_p c_m(t_n, \overline{X}^n)\varphi'_{jm}(t_{n+1}) \\ &\quad + \partial^3_{ikp} c_j(t_n, \overline{X}^n)\varphi_j(t_{n+1}), && t_n < \overline{\tau}, \end{aligned} \tag{24}$$

$$\varphi''_{ikr}(\overline{\tau}) = \delta^3_{ikp} g(\overline{X}_{\overline{\tau}}, \overline{\tau}). \tag{25}$$

Remark 2. For more general domains we may approximate ∂D with the tangent plane at the stopping point $(\overline{X}_{\overline{\tau}}, \overline{\tau})$, compute derivatives of g in the directions of the tangent plane and use difference quotients in the normal direction and then transform back to the original coordinate directions.

The time discretization error $\mathcal{E}_C$ in (10) has the following error expansion:

Theorem 2. *Let $X(t)$, $\overline{X}(t)$ and $\overline{X}(t_n)$ be the exact, the continuous Euler and the discrete Euler path defined in* (2), (8) *and* (6) *respectively. Assume that the functions a, b and g are bounded in $\mathcal{C}^6(D)$ and $\mathcal{C}^6(D \times [0,T])$ respectively. Then the time discretization error $\mathcal{E}_C$ has the error expansion*

$$E[g(X_\tau,\tau) - g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})] = E\left[\sum_{n=0}^{N-1} \mathbf{1}_{t_{n+1}\leq\overline{\tau}} \rho_n \Delta t_n^2\right] + \mathcal{O}\left(\Delta x + \sqrt{\Delta t_{\max}} + \frac{\sqrt{\Delta t_{\max}}}{\Delta x^k}\right) E\left[\sum_{n=0}^{N-1} \Delta t_n^2\right] \tag{26}$$

where Δx is a small positive constant and $k \in \{1,2,3\}$ is the highest order of difference quotient used in (20)–(25) *to define φ, φ', φ'', and*

$$\begin{aligned}\rho_n = {} & \frac{1}{2}(\partial_t a_k + a_j\partial_j a_k + \beta_{ij}\partial^2_{ij} a_k)(\overline{X}^n)\varphi_k(t_{n+1}) \\ & + \frac{1}{2}\left(\partial_t\beta_{km} + 2\beta_{jm}\partial_j a_k + a_j\partial_j\beta_{km} + \beta_{ij}\partial^2_{ij}\beta_{km}\right)(\overline{X}^n)\varphi'_{km}(t_{n+1}) \\ & + (\beta_{jr}\partial_j\beta_{km})(\overline{X}^n)\varphi''_{kmr}(t_{n+1}).\end{aligned} \tag{27}$$

Remark 3. If we do not solve the backward dual problems (20)–(25), but instead set $\varphi \equiv \varphi' \equiv \varphi'' \equiv 1$ we obtain adaptivity based on the local error.

The proof of Theorem 2 has several steps and we present them by following three lemmas. Let us first introduce a solution u of the Kolmogorov backward equation

$$\begin{aligned} \partial_t u + a_i\partial_i u + \beta_{ij}\partial^2_{ij} u &= 0, && (x,t) \in D \times [0,T), \\ u(x,T) &= g(x,T), && x \in D, \\ u(x,t) &= g(x,t), && (x,t) \in \partial D \times [0,T]. \end{aligned} \tag{28}$$

Then by the Feynman-Kac formula u can be represented by the expectation

$$u(x,t) = E[g(X_\tau,\tau) \mid X(t) = x]. \tag{29}$$

Let $\overline{a_i}$ and $\overline{b_i}$ be the piecewise constant functions defined by $\overline{a_i}(t) = a_i(\overline{X}^n)$ and $\overline{b_i}(t) = b_i(\overline{X}^n)$ for $t \in [t_n, t_{n+1})$. Similarly define $\overline{\beta_{ij}} = \frac{1}{2}\overline{b_i^l b_j^l}$. Then the time discretization error $\mathcal{E}_C$ has the following representation :

Lemma 1. *Let $X(t)$ and $\overline{X}(t)$ be the exact and the continuous Euler path defined by* (2) *and* (8) *respectively and let the function u be defined by* (29). *Suppose that the assumptions in Theorem 2 hold. Then the time discretization error between these two paths has the representation*

$$E[g(X_\tau,\tau)-g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})] = E\left[\int_0^{\widetilde{\tau}} \left((a_i-\overline{a_i})\partial_i u + (\beta_{ij}-\overline{\beta_{ij}})\partial^2_{ij}u\right)(\overline{X}_t,t)\,\mathrm{d}t\right]. \tag{30}$$

Proof. Apply the Itô formula to the function u in (29) to get

$$\mathrm{d}u(\overline{X}_t,t) = \left(\partial_t u + \overline{a_i}\partial_i u + \overline{\beta_{ij}}\partial^2_{ij}u\right)(\overline{X}_t,t)\,\mathrm{d}t + \overline{b_i^l}\partial_i u(\overline{X}_t,t)\,\mathrm{d}W_t^l.$$

Here the definition of the continuous Euler scheme in (8) is used, i.e. $d\overline{X}_i(t) = \overline{a_i}\,\mathrm{d}t + \overline{b_i^l}\,\mathrm{d}W_t^l$ for $t\in[t_n,t_{n+1})$. Integrate both sides from 0 to $\widetilde{\tau}$ and take the expectation to obtain

$$\begin{aligned} E[u(\overline{X}_{\widetilde{\tau}},\widetilde{\tau}) - u(\overline{X}_0,0)] &= E\left[\int_0^{\widetilde{\tau}} (\partial_t u + \overline{a_i}\partial_i u + \overline{\beta_{ij}}\partial^2_{ij}u)(\overline{X}_t,t)\,\mathrm{d}t\right] \\ &\quad + E\left[\int_0^{\widetilde{\tau}} \overline{b_i^l}\partial_i u(\overline{X}_t,t)\,\mathrm{d}W_t^l\right]. \end{aligned} \tag{31}$$

Note that the Itô integral in (31) is not adapted to the standard filtration generated by W alone. Instead consider the filtration $\mathcal{G}_t$, the σ-algebra generated by $\{W^l(s), \Delta t(s) : s\le t,\, l=1,2,\ldots,l_0\}$. Then from Lemma 4.2 in [15] the Itô integral in (31) is a martingale with respect to $\mathcal{G}_t$ and since $\widetilde{\tau}$ is a stopping time, we therefore have

$$E\left[\int_0^{\widetilde{\tau}} \overline{b_i^l}\partial_i u(\overline{X}_t,t)\mathrm{d}W_t^l\right] = 0.$$

In the left hand side of (31) we use the boundary conditions in (28)

$$E[u(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})] = E[g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})],$$

and the Feynman-Kac formula (29)

$$u(\overline{X}_0,0) \;=\; E[g(X_\tau,\tau)\mid X_0=\overline{X}_0] = \;E[g(X_\tau,\tau)].$$

Finally we use the Kolmogorov backward equation (28) to eliminate $\partial_t u$ in the first expectation of the right hand side in (31) and conclude (30).

Using the discrete time steps, the error representation (30) can be written

$$\begin{aligned} &E[g(X_\tau,\tau)-g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})] \\ &\quad = E\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}} \mathbf{1}_{t\le\widetilde{\tau}}\left((a_i-\overline{a_i})\partial_i u + (\beta_{ij}-\overline{\beta_{ij}})\partial^2_{ij}u\right)(\overline{X}_t,t)\,\mathrm{d}t\right]. \end{aligned} \tag{32}$$

Lemma 2. *Let $X(t)$ and $\overline{X}(t)$ be the exact path and the continuous Euler path defined in* (2) *and* (8) *respectively and assume that the assumptions in Theorem 2 hold. Then the time discretization error between these two paths has the following expansion*

$$E[g(X_\tau,\tau)-g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})]=E\left[\sum_{n=0}^{N-1}\mathbf{1}_{t_{n+1}\leq\overline{\tau}}\widetilde{\rho}_n\Delta t_n^2\right]+\mathcal{O}(\sqrt{\Delta t_{\max}})E\left[\sum_{n=0}^{N-1}\mathcal{O}(\Delta t_n^2)\right] \tag{33}$$

where

$$\begin{aligned}\widetilde{\rho}_n &= \frac{1}{2}(\partial_t a_k + a_j\partial_j a_k + \beta_{ij}\partial^2_{ij}a_k)(\overline{X}^n)\partial_k u(\overline{X}^{n+1},t_{n+1})\\ &+\frac{1}{2}\left(\partial_t\beta_{km}+2\beta_{jm}\partial_j a_k + a_j\partial_j\beta_{km}+\beta_{ij}\partial^2_{ij}\beta_{km}\right)(\overline{X}^n)\partial^2_{km}u(\overline{X}^{n+1},t_{n+1})\\ &+(\beta_{jr}\partial_j\beta_{km})(\overline{X}^n)\partial^3_{kmr}u(\overline{X}^{n+1},t_{n+1}).\end{aligned} \tag{34}$$

Proof. Apply the Itô formula to each term in (32) to get

$$\begin{aligned}a_i(\overline{X}_t)-\overline{a_i}(\overline{X}_t) &= a_i(\overline{X}_t)-a_i(\overline{X}_n)\\ &=\int_{t_n}^t\left(\partial_s a_i+\overline{a_k}\partial_k a_i+\overline{\beta_{jk}}\partial^2_{jk}a_i\right)(\overline{X}_s)\,\mathrm{d}s\\ &+\int_{t_n}^t\overline{b^l_j}\partial_j a_i(\overline{X}_s)\,\mathrm{d}W^l_s,\end{aligned}$$

and similarly

$$\begin{aligned}&\beta_{ij}(\overline{X}_t)-\overline{\beta_{ij}}(\overline{X}_t)\\ &\quad=\int_{t_n}^t\left(\partial_s\beta_{ij}+\overline{a_k}\partial_k\beta_{ij}+\overline{\beta_{km}}\partial^2_{km}\beta_{ij}\right)(\overline{X}_s)\,\mathrm{d}s+\int_{t_n}^t\overline{b^l_k}\partial_k\beta_{ij}(\overline{X}_s)\,\mathrm{d}W^l_s.\end{aligned}$$

Substitute the above integrals in (32) and use Malliavin derivatives, see [17], for example

$$\begin{aligned}&E\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}}\mathbf{1}_{t\leq\widetilde{\tau}}\int_{t_n}^t\overline{b^l_j}\partial_j a_i(\overline{X}_s)\partial_i u(\overline{X}_t,t)\,\mathrm{d}W^l_s\,\mathrm{d}t\right]\\ &\qquad=E\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}}\mathbf{1}_{t\leq\widetilde{\tau}}\int_{t_n}^t 2\overline{\beta_{jm}}\partial_j a_i(\overline{X}_s)\partial^2_{im}u(\overline{X}_t,t)\;\mathrm{d}s\;\;\mathrm{d}t\right]\end{aligned}$$

to get

$$
\begin{aligned}
&E[g(X_\tau,\tau) - g(\overline{X}_{\widetilde{\tau}},\widetilde{\tau})] \\
&= E\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}} \mathbf{1}_{t\le\widetilde{\tau}}\left(\int_{t_n}^{t}\left(\partial_s a_i + \overline{a_k}\partial_k a_i + \overline{\beta_{jk}}\partial^2_{jk}a_i\right)(\overline{X}_s)\,\mathrm{d}s\;\partial_i u(\overline{X}_t,t)\right.\right. \\
&\quad + \int_{t_n}^{t}\left(\partial_s\beta_{km} + 2\overline{\beta_{jm}}\partial_j a_k + \overline{a_j}\partial_j\beta_{km} + \overline{\beta_{ij}}\partial^2_{ij}\beta_{km}\right)(\overline{X}_s)\,\mathrm{d}s\,\partial^2_{km}u(\overline{X}_t,t) \\
&\qquad\qquad \left.\left. + \int_{t_n}^{t} 2\overline{\beta_{jr}}\partial_j\beta_{km}(\overline{X}_s)\,\mathrm{d}s\;\partial^3_{kmr}u(\overline{X}_t,t)\right)\mathrm{d}t\right]. \quad (35)
\end{aligned}
$$

Each term in (35) has the form

$$
E\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}}\int_{t_n}^{t}\mathbf{1}_{t\le\widetilde{\tau}} f(\overline{X}_s)\;h(\overline{X}_t,t)\,\mathrm{d}s\,\mathrm{d}t\right] \quad (36)
$$

where f is a function of a_i, β_{ij} and their derivatives representing the local error and h is a function of the derivatives of u. Finally apply the a priori error estimate (11) to the expected value (36) to conclude

$$
\begin{aligned}
&E\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}}\int_{t_n}^{t}\mathbf{1}_{t\le\widetilde{\tau}} f(\overline{X}_s)\;h(\overline{X}_t,t)\,\mathrm{d}s\,\mathrm{d}t\right] \\
&= E\left[\sum_{n=0}^{N-1}\frac{1}{2}\mathbf{1}_{t_{n+1}\le\overline{\tau}} f(\overline{X}^n)h(\overline{X}^{n+1},t_{n+1})\Delta t_n^2\right] + \mathcal{O}(\sqrt{\Delta t_{\max}})E\left[\sum_{n=0}^{N-1}\Delta t_n^2\right]
\end{aligned}
$$

which proves (33).

Note that the quantities $\partial_i u, \partial^2_{ij}u$ and $\partial^3_{ijk}u$ in (34) not are computable. The adaptive algorithms will use the computable approximations (20)-(25) for these functions. From the construction of u we have

$$
\partial_k u(x,t) = E[\partial_i u(X_\tau,\tau)X'_{ik}(\tau;t) \mid X'_{ij}(t) = \delta_{ij}, X(t) = x], \quad (37)
$$

where δ_{ij} denotes the Kronecker δ–function and $X'_{ij}(s;t) := \partial X_i(s; X(t) = x)/\partial x_j$ is the first variation of $X(s)$ with respect to a perturbation in the initial location at time t, i.e. it satisfies

$$
\begin{aligned}
&\mathrm{d}X'_{ij}(s) = \partial_k a_i(X(s))X'_{kj}(s)\,\mathrm{d}s + \partial_k b^l_i(X(s))X'_{kj}(s)\,\mathrm{d}W^l(s), \quad t < s < \tau, \\
&X'_{ij}(t) = \delta_{ij}.
\end{aligned} \quad (38)
$$

The goal is to approximate $\partial_k u(\overline{X}^n, t_n)$ in (34) by conditional expected values of the computable quantities φ_k defined in (20)-(21) and similarly to approximate $\partial^2_{ij}u$ and $\partial^3_{ijk}u$ by expected values of φ'_{ij} and φ''_{ijk} in (22)-(23) and (24)-(25) respectively.

Note that if the continuous exact path finishes at $\tau = T$ then by the definition of u, we have $\partial_k u(X_T, T) = \partial_k g(X_T, T)$ so that

$$\begin{aligned} &E[\partial_i u(X_\tau, \tau) X'_{ik}(\tau; t) \mathbf{1}_{\tau=T} \mid X'_{ij}(t) = \delta_{ij}, X(t) = x] \\ &\qquad = E[\partial_i g(X_T, T) X'_{ik}(T; t) \mathbf{1}_{\tau=T} \mid X'_{ij}(t) = \delta_{ij},\ X(t) = x]. \quad (39) \end{aligned}$$

However, for $\tau < T$ the first variation $\partial_i g(X_\tau, \tau)$ exists only in the directions tangent to the boundary ∂D, $i = 2, \ldots, d$. In the direction normal to ∂D we approximate $\partial_1 u(X_\tau, \tau)$ in (37) by the expected value of a difference quotient of g and remove this second expected value. To do this we introduce a small positive constant Δx. Once the continuous exact path crosses the boundary, we start a new realization $\hat{X}$ with the initial value

$$\hat{X}(\tau) = X(\tau) + \gamma \Delta x \in D,$$

where γ denotes an inward unit normal vector. The new realization $\hat{X}_t$ evolves by (2) for $\tau < t < \hat{\tau}$ until it stops with the first exit time $\hat{\tau} \in (\tau, T]$. Then by the Taylor expansion we have

$$\partial_1 u(X_\tau, \tau) = -\frac{u(\hat{X}_\tau, \tau) - u(X_\tau, \tau)}{\Delta x} + \mathcal{O}(\Delta x)$$

and the Feynman-Kac formula (29) gives

$$\partial_1 u(X_\tau, \tau) = -\frac{E[g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau) | \mathcal{G}_\tau]}{\Delta x} + \mathcal{O}(\Delta x)$$

where $\mathcal{G}_t$ is the σ-algebra generated by $\{W^l(s), \Delta t(s) : s \leq t, l = 1, 2, \ldots, l_0\}$. Use the measurability of $X'_{ik}(\tau; t) \mathbf{1}_{\tau<T} \in \mathcal{G}_\tau$ to get

$$\begin{aligned} &E\left[\partial_1 u\left(X_\tau, \tau\right) X'_{1k}(\tau; t) \mathbf{1}_{\tau<T} \mid X'_{ij}(t) = \delta_{ij},\ X(t) = x\right] \\ &= E\left[E\left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau)}{-\Delta x} \;\middle|\; \mathcal{G}_\tau \right] X'_{1k}(\tau; t) \mathbf{1}_{\tau<T} \;\middle|\; \begin{array}{l} X'_{ij}(t) = \delta_{ij}, X(t) = x, \\ \hat{X}_\tau = X_\tau + \gamma \Delta x \end{array} \right] \\ &\quad + \mathcal{O}(\Delta x) \\ &= E\left[E\left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau)}{-\Delta x} X'_{1k}(\tau; t) \mathbf{1}_{\tau<T} \;\middle|\; \mathcal{G}_\tau \right] \;\middle|\; \begin{array}{l} X'_{ij}(t) = \delta_{ij}, X(t) = x, \\ \hat{X}_\tau = X_\tau + \gamma \Delta x \end{array} \right] \\ &\quad + \mathcal{O}(\Delta x) \\ &= E\left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau)}{-\Delta x} X'_{1k}(\tau; t) \mathbf{1}_{\tau<T} \;\middle|\; \begin{array}{l} X'_{ij}(t) = \delta_{ij}, X(t) = x, \\ \hat{X}_\tau = X_\tau + \gamma \Delta x \end{array} \right] + \mathcal{O}(\Delta x), \end{aligned}$$

and thus

$$E[\partial_i u(X_\tau,\tau)X'_{ik}(\tau;t)\mathbf{1}_{\tau<T} \mid X'_{ij}(t)=\delta_{ij}, X(t)=x]$$
$$= E\left[-\frac{g(\hat{X}_{\hat{\tau}},\hat{\tau})-g(X_\tau,\tau)}{\Delta x}X'_{1k}(\tau;t)\mathbf{1}_{\tau<T}\;\middle|\;\begin{array}{l}X'_{ij}(t)=\delta_{ij}, X(t)=x,\\ \hat{X}_\tau = X_\tau+\gamma\Delta x\end{array}\right]$$
$$+ E\left[\sum_{i=2}^{d}\partial_i g(X_\tau,\tau)X'_{ik}(\tau;t)\mathbf{1}_{\tau<T}\;\middle|\;X'_{ij}(t)=\delta_{ij}, X(t)=x\right]+\mathcal{O}(\Delta x). \quad (40)$$

The expected values in the right hand sides of (39) and (40) can be approximated using Euler approximations and the error in doing so is estimated by repeated use of the a priori error estimate (11). Let thus $(\hat{\overline{X}},\hat{\overline{\tau}})$ be the Euler approximation of $(\hat{X},\hat{\tau})$ and gather all X_i, X_{ij} in a stochastic process Y_t, taking values in $\mathbb{R}^{d+d^2}$. Then Y_t satisfies the system of SDEs, (2) and (38), which we write

$$\mathrm{d}Y(t) = A(Y(t))\,\mathrm{d}t + B^l(Y(t))\,\mathrm{d}W^l(t), \qquad t>t_0, \quad Y(t_0)=Y_0. \quad (41)$$

Similarly define the corresponding Euler approximation $\overline{Y}$ of Y as the solution of

$$\overline{Y}(t_{n+1}) = \overline{Y}(t_n) + A(\overline{Y}(t_n))\Delta t_n + B^l(\overline{Y}(t_n))\Delta W_n^l, \quad n\geq 0, \quad Y(t_0)=Y_0. \quad (42)$$

Consider first the case $\tau = T$; apply the a priori error estimate (11) to the functions $\check{f}(Y_\tau,\tau) = \partial_i g(X_\tau,\tau)X'_{ik}(\tau;t)\mathbf{1}_{\tau=T}$, for $k=1,2,\ldots,d$, to get

$$E[\check{f}(Y_\tau,\tau) - \check{f}(\overline{Y}_{\overline{\tau}},\overline{\tau})] = \mathcal{O}(\sqrt{\Delta t_{\max}}).$$

When $\tau < T$, the second expected value in the right hand side of (40) is treated similarly as when $\tau = T$. For the first expected value in the right hand side in (40), extend Y_t to $\mathcal{Y}_t$ containing also the d-dimensional process $\hat{X}_t$, which solves (2) for $\tau < t < \hat{\tau}$. Then $\mathcal{Y}_t$ has two exit times $\theta = (\tau,\hat{\tau})^{\mathsf{T}}$. Denote by $\overline{\mathcal{Y}}$ and $\overline{\theta}$ the corresponding Euler approximations and apply (11) to the functions $\check{f}(\mathcal{Y}_\theta,\theta) = -\frac{g(\hat{X}_{\hat{\tau}},\hat{\tau})-g(X_\tau,\tau)}{\Delta x}X'_{1k}(\tau;t)\mathbf{1}_{\tau<T}$, for $k=1,2,\ldots,d$, to obtain

$$E[\check{f}(\mathcal{Y}_\theta,\theta) - \check{f}(\overline{\mathcal{Y}}_{\overline{\theta}},\overline{\theta})] = \mathcal{O}\left(\frac{\sqrt{\Delta t_{\max}}}{\Delta x}\right)$$

and consequently

$$\begin{aligned}\partial_k u(x,t) = {} & E[\partial_i g(\overline{X}_{\overline{\tau}},\overline{\tau})X'_{ik}(\overline{\tau};t)\mathbf{1}_{\overline{\tau}=T} \mid \overline{X}'_{ij}(t)=\delta_{ij},\ \overline{X}(t)=x] \\ & + E\left[\frac{g(\hat{\overline{X}}_{\hat{\overline{\tau}}},\hat{\overline{\tau}}) - g(\overline{X}_{\overline{\tau}},\overline{\tau})}{-\Delta x}\overline{X}'_{1k}(\tau;t)\mathbf{1}_{\overline{\tau}<T} \;\middle|\; \begin{matrix}\overline{X}'_{ij}(t)=\delta_{ij},\overline{X}(t)=x,\\ \hat{\overline{X}}_{\overline{\tau}}=\overline{X}_{\overline{\tau}}+\gamma\Delta x\end{matrix}\right] \\ & + E\left[\sum_{i=2}^{d}\partial_i g(\overline{X}_{\overline{\tau}},\overline{\tau})\overline{X}'_{ik}(\tau;t)\mathbf{1}_{\overline{\tau}<T}\middle|\overline{X}'_{ij}(t)=\delta_{ij},\overline{X}(t)=x\right] \\ & + \mathcal{O}\left(\Delta x + \sqrt{\Delta t_{\max}} + \frac{\sqrt{\Delta t_{\max}}}{\Delta x}\right). \end{aligned} \tag{43}$$

This is an expansion of the expected value of φ_k defined in (20)-(21). The higher derivatives $\partial^2_{ij}u$ and $\partial^3_{ijk}u$ can be computed in a similar way and we have the error expansion:

Lemma 3. *Suppose the assumptions in Theorem 2 hold. Then the function u defined by (29) and the dual functions φ, φ' and φ'' defined by (20)-(25) satisfy, for $\alpha = 1, 2, 3$,*

$$\partial^\alpha u(\overline{X}(t_n),t_n) - E[\varphi^\alpha(t_n)\,|\,\mathcal{F}_n] = \mathcal{O}\left(\Delta x + \sqrt{\Delta t_{\max}} + \frac{\sqrt{\Delta t_{\max}}}{\Delta x^\alpha}\right) \tag{44}$$

where $\mathcal{F}_n$ denotes the σ-algebras generated by $\{W^l(s), \Delta t(s) : s \le t_n,\ l = 1,2,\ldots,l_0\}$, $\varphi^1 = \varphi_i$, $\varphi^2 = \varphi'_{ij}$ and $\varphi^3 = \varphi''_{ijk}$ for some i, j, k and $\partial^\alpha u$ is the corresponding α:th order derivative of u.

Proof. For $\alpha = 1$, the approximation (43) and the definition (20)-(21) yield (44).

Following [17] extend Y to be $(X, X', X'', X''')^{\mathrm{T}}$ satisfying the SDE similar to (41) with $Y(t_0) = (x, I, 0, 0)^{\mathrm{T}}$ where I is the $d \times d$-identity matrix. Here the first variation X' of X is defined in (38) and the other higher variations are defined similarly by taking the derivatives to the right hand side of (38). Introduce the corresponding Euler approximate $\overline{Y} = (\overline{X}, \overline{X}', \overline{X}'', \overline{X}''')^{\mathrm{T}}$ satisfying the SDE similar to (42) and let $(\overline{X}', \overline{X}'', \overline{X}''')$ denote the Euler approximations of (X', X'', X'''). For the case when $\tau = T$ and $\alpha = 2$ or 3, we use the a priori error estimate (11) for the extended systems Y and $\overline{Y}$ with

$$\begin{aligned}
\check{f}(Y_\tau,\tau) &= \left(\partial_i g X''_{ikn} + \partial^2_{ir} g X'_{ik} X'_{rn}\right)\mathbf{1}_{\tau=T} && \text{if } \alpha = 2,\\
\check{f}(Y_\tau,\tau) &= (\partial_i g X'''_{iknm} + \partial^2_{ir} g X'_{ik} X''_{rnm} \\
&\quad + \partial^2_{ir} g X'_{in} X''_{rkm} + \partial^2_{ir} g X'_{im} X''_{rkn} \\
&\quad + \partial^3_{irv} g X'_{ik} X'_{rn} X'_{vm})\mathbf{1}_{\tau=T} && \text{if } \alpha = 3,
\end{aligned}$$

where $\check{f}(Y_\tau,\tau)$ in the case $\alpha = 2$ derives from

$$\begin{aligned}\partial_{kn} u(x,t) = E\big[&\partial_i u(X_\tau,\tau)X''_{ikn}(\tau) + \partial_{ir}u(X_\tau,\tau)X'_{ik}(\tau)X'_{rn}(\tau) \mid \\ &X''_{ikn}(t) = 0,\ X'_{ij}(t) = \delta_{ij},\ X(t) = x\big]\end{aligned}$$

with $u(X_\tau, \tau) = g(X_\tau, \tau)$ if $\tau = T$ and similarly for $\alpha = 3$. The extension to the case $\tau < T$ is similar to the first order derivative treated above; this time second and third order difference quotients appear leading to terms $\mathcal{O}(\sqrt{\Delta t_{max}}/\Delta x^2)$ and $\mathcal{O}(\sqrt{\Delta t_{max}}/\Delta x^3)$ respectively.

Proof of Theorem 2. The measurability of the function f_n depending on the derivatives of a and β, e.g. $f_n = \mathbf{1}_{t_{n+1} \le \overline{\tau}}(\partial_t a_k + a_j \partial_j a_k + \beta_{ij} \partial^2_{ij} a_k)(\overline{X}^n)\Delta t_n^2 \in \mathcal{F}_{n+1}$, proves

$$E\left[\sum_{n=0}^{N-1} f_n E[\varphi_k(t_{n+1})|\mathcal{F}_{n+1}]\right] = E\left[E\left[\sum_{n=0}^{N-1} f_n \varphi_k(t_{n+1})\middle|\mathcal{F}_{n+1}\right]\right]$$
$$= E\left[\sum_{n=0}^{N-1} f_n \varphi_k(t_{n+1})\right]. \tag{45}$$

Similar representations hold for the other terms in (34). Consequently, the combination of Lemma 2-3 and the removal of the second expectation (45) prove (26). □

Remark 4. In the case of only first order difference quotients, the optimal size of the constant Δx for the difference quotient in (44) is $\mathcal{O}((\Delta t_{\max})^{1/4})$ and $\Delta x = \mathrm{TOL}_T^{1/4}$ is used for the adaptive algorithm in Sect. 3 where TOL_T is a given time discretization error tolerance. In the one dimensional example in Sect. 4 we use the Kolmogorov equation to replace higher order derivatives on the boundary with lower order terms. For instance $\varphi'(\overline{\tau}) = -\beta^{-1}(\partial_t g(\overline{X}_{\overline{\tau}}, \overline{\tau}) + a(\overline{X}_{\overline{\tau}})\varphi(\overline{\tau}))$ and $\varphi'''(\overline{\tau}) = \beta^{-1}((\partial_t g(\hat{\overline{X}}_{\hat{\overline{\tau}}}, \hat{\overline{\tau}}) - \partial_t g(\overline{X}_{\overline{\tau}}, \overline{\tau}))/\Delta x + \partial_x a(\overline{X}_{\overline{\tau}})\varphi(\overline{\tau}) + (a + \partial_x \beta)(\overline{X}_{\overline{\tau}})\varphi'(\overline{\tau}))$.

3 Adaptive Algorithms for Stopped Diffusion

This section presents adaptive algorithms for the stopped diffusion problems. As described in Sect. 2, the computational error is separated into the following three terms : the time discretization error between the exact and the continuous Euler path $\mathcal{E}_C$, the time discretization error between the continuous and discrete Euler approximation $\mathcal{E}_D$, and the statistical error $\mathcal{E}_S$, i.e.

$$\begin{aligned}
&E[g(X(\tau), \tau)] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(\overline{\tau}; \omega_j), \overline{\tau}) \\
&= E[g(X(\tau), \tau) - g(\overline{X}(\widetilde{\tau}), \widetilde{\tau})] + E[g(\overline{X}(\widetilde{\tau}), \widetilde{\tau}) - g(\overline{X}(\overline{\tau}), \overline{\tau})] \\
&+ \left(E[g(\overline{X}(\overline{\tau}), \overline{\tau})] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(\overline{\tau}; \omega_j), \overline{\tau})\right) \\
&=: \mathcal{E}_C + \mathcal{E}_D + \mathcal{E}_S.
\end{aligned} \tag{46}$$

For a given error tolerance TOL, the goal is to minimize the computational work, which is roughly $\mathcal{O}(M{\cdot}N) = \mathcal{O}(\mathrm{TOL}_S^{-2}\mathrm{TOL}_T^{-1})$ where TOL_S and TOL_T denote a statistical tolerance and a time discretization tolerance respectively. Thus we obtain

$$\mathrm{TOL}_S = \frac{2}{3}\mathrm{TOL} \quad \text{and} \quad \mathrm{TOL}_T = \frac{1}{3}\mathrm{TOL} \tag{47}$$

by solving

$$\min \mathrm{TOL}_S^{-2}\mathrm{TOL}_T^{-1} \quad \text{subject to} \quad \mathrm{TOL}_S + \mathrm{TOL}_T = \mathrm{TOL}.$$

3.1 Control of the Statistical Error

Let us first introduce some notation. Define the sample average $\mathcal{A}(Y;M)$ and the sample standard deviation $\overline{\sigma}(Y;M)$ of Y by

$$\mathcal{A}(Y;M) := \frac{1}{M}\sum_{j=1}^{M} Y(\omega_j), \quad \overline{\sigma}(Y;M) := \left(\mathcal{A}(Y^2;M) - (\mathcal{A}(Y;M))^2\right)^{\frac{1}{2}}.$$

Then from the Central Limit Theorem, the statistical error $\mathcal{E}_S$ in (46) satisfies

$$|\mathcal{E}_S| \leq \mathtt{E}_S(Y;M) := c_0 \ \frac{\overline{\sigma}(Y;M)}{\sqrt{M}} \tag{48}$$

with probability close to one asymptotically, where $Y = g(\overline{X}_{\overline{\tau}}, \overline{\tau})$ and c_0 is a constant corresponding to a confidence interval. For example, $c_0 \geq 1.65$ gives asymptotically the probability greater than 0.90.

3.2 Control of the Time Discretization Error

In this subsection, we present two refinement strategies to control the time discretization error. For a given partition $0 = t_0 < t_1 < \ldots < t_N = T$, the piecewise constant mesh function Δt is defined by (12) and the corresponding number $N(\Delta t)$ of steps is

$$N(\Delta t) := \int_0^T \frac{1}{\Delta t(s)}\mathrm{d}s\,.$$

Then the optimal choice of the time steps is formulated by minimizing the computational work $E[N(\Delta t)]$ such that $\Delta t \in \mathcal{K}$ subject to given accuracy constraints. The feasible set $\mathcal{K}$ for the mesh function Δt is defined by

$$\mathcal{K} := \{\Delta t : \Delta t \text{ is stochastic, positive and piecewise constant on } [0,T] \text{ for each realization } \}.$$

Total Time Discretization Error

The goal is to make the total time discretization error, $\mathcal{E}_T = \mathcal{E}_C + \mathcal{E}_D$ defined in (4), bounded by a given time discretization error tolerance TOL_T in (47). Therefore the accuracy constraint is

$$E\left[\sum_{n=0}^{N-1} r_n\right] \leq \mathrm{TOL}_T \tag{49}$$

where the error indicators r_n are defined for $n = 0, 1, \ldots, N-1$, by

$$\begin{aligned} r_n := \Big| & \mathbf{1}_{t_{n+1} \leq \overline{\tau}} \rho_n \Delta t_n^2 \\ & + \Big(g\big(\mathrm{proj}_{\partial D} \frac{1}{2}(\overline{X}(t_n) + \overline{X}(t_{n+1})), \frac{1}{2}(t_n + t_{n+1})\big) - g(\overline{X}_{\overline{\tau}}, \overline{\tau}) \Big) \hat{P}_{\overline{X},n} \Big| \end{aligned} \tag{50}$$

with ρ_n in (27) and $\hat{P}_{\overline{X},n}$ in (15) and $\mathrm{proj}_{\partial D}$ the orthogonal projection to ∂D.

To have as few time steps as possible, we try to make

$$r_n(\omega) = \text{constant}, \quad \forall n \text{ and } \forall \omega$$

and by (49) the natural choice of the constant is then

$$r_n(\omega) = \frac{\mathrm{TOL}_T}{E[N]}, \quad \forall n \text{ and } \forall \omega. \tag{51}$$

The choice (51) is optimal in the case without stopping boundary, see [15], [17], i.e. without the second term in (50). Numerical tests on one dimensional processes show that the error $\mathcal{E}_D$ in (46), corresponding the second term in (50), converges exponentially fast as the number of adaptive steps is increased. Therefore an over-refinement in this part of the error does not seem to cost much. Note that in practice the quantity $E[N]$ is not known and we can only estimate it by the sample average $\overline{N}[j] := \mathcal{A}(N; M[j])$ of the final number of time steps in the jth batch of $M[j]$ numbers of realizations. Then the statistical error, $|E[N] - \overline{N}[j]|$, is bounded by $\mathtt{E}_S(N; M[j])$, with probability close to one, by the same argument as in (48).

To achieve (51), start with an initial mesh $\Delta t[1]$ and then specify iteratively a new partition $\Delta t[k+1]$ from $\Delta t[k]$, using the following refinement strategy: for each realization in the mth batch and for all time steps $n = 0, 1, \ldots, N[k]-1$,

$$\begin{aligned} & \textbf{if } \left(r_n[k] \geq \frac{\mathrm{TOL}_T}{\overline{N}[m-1]} \right) \textbf{ then} \\ & \qquad \text{divide } \Delta t_n[k] \text{ into 2 equal substeps, and} \\ & \qquad \text{generate the intermediate value of } W \text{ by Brownian bridges (5)} \\ & \textbf{else} \text{ let the new step be the same as the old} \\ & \textbf{endif}, \end{aligned} \tag{52}$$

with the stopping criterion: for each realization of the mth batch

$$\textbf{if } \left(\max_{1\le n\le N[k]} r_n[k] < S\frac{\mathrm{TOL}_T}{\overline{N}[m-1]} \right) \textbf{ then } \text{stop.} \tag{53}$$

Here S is a given constant, motivated as follows: we want the maximal error indicator to decay quickly to the stopping level $S\mathrm{TOL}_T/\overline{N}$, but when almost all r_n satisfy $r_n \le \mathrm{TOL}_T/\overline{N}$, the reduction of the error may be slow. The constant S is introduced to cure this slow reduction.

Splitting of the Time Discretization Error

Let us compare the adaptive algorithm (52)-(53) with the following *ad hoc* refinement algorithm. First we split the time discretization tolerance $\mathrm{TOL}_T = \mathrm{TOL}_C + \mathrm{TOL}_D$ by $\mathrm{TOL}_C = \mathrm{TOL}_D = \mathrm{TOL}_T/2$ and define the error indicators r_n^C and r_n^D by

$$\begin{aligned} r_n^C &:= \mathbf{1}_{t_{n+1}\le\overline{\tau}} |\rho_n| \Delta t_n^2 \\ r_n^D &:= \left| g\big(\mathrm{proj}_{\partial D} \frac{1}{2}(\overline{X}(t_n) + \overline{X}(t_{n+1})), \frac{1}{2}(t_n + t_{n+1})\big) - g(\overline{X}_{\overline{\tau}}, \overline{\tau}) \right| \hat{P}_{\overline{X},n} \end{aligned} \tag{54}$$

with ρ_n in (27) and $\hat{P}_{\overline{X},n}$ in (15). This alternative refinement strategy is to take into account the computational observation that only a few time intervals for each realization have large error indicators r_n^D compared to the others, see Fig. 2, an illustrative Monte Carlo realization of r_n^D for Example 1 in Sect. 4.

Start the algorithm with an initial mesh $\Delta t[1]$ and then specify iteratively a new partition $\Delta t[k+1]$ from $\Delta t[k]$ using following refinement strategy: for each realization in the mth batch and for all time step $n = 0, 1, \ldots, N[k]-1$,

$$\begin{aligned} &\textbf{if } \left(r_n^C[k] \ge \frac{\mathrm{TOL}_C}{\overline{N}[m-1]} \textbf{ or } r_n^D[k] \ge \mathrm{TOL}_D \right) \textbf{ then}, \\ &\qquad \text{divide } \Delta t_n[k] \text{ into 2 equal substeps} \\ &\textbf{else let the new step be the same as the old one} \\ &\textbf{endif}. \end{aligned} \tag{55}$$

until the following stopping criteria is fulfilled: for each realization of the mth batch

$$\textbf{if } \left(\max_{1\le n\le N[k]} r_n^C[k] < S_C\frac{\mathrm{TOL}_C}{\overline{N}[m-1]} \textbf{ and } \max_{1\le n\le N[k]} r_n^D[k] < S_D\mathrm{TOL}_D \right) \textbf{ then } \text{stop.} \tag{56}$$

Here S_C and S_D are given constants to cure the slow reduction when almost all r_n^C or r_n^D satisfy $r_n^C \le \mathrm{TOL}_C/\overline{N}$ or $r_n^D \le \mathrm{TOL}_D$.

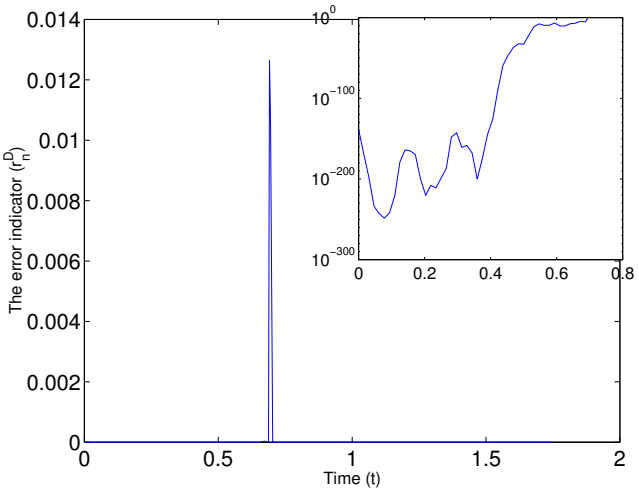


Fig. 2. Example 1: An illustrative Monte Carlo realization of r_n^D with TOL $= 0.1$

3.3 The Adaptive Algorithms

The adaptive stochastic time stepping algorithms have structures similar to a basic Monte Carlo algorithm, with an additional inner loop for individual mesh refinement for each realization of a Brownian motion. First we split the specified error tolerance by (47): the outer loop computes the batches of realizations of $\overline{X}$, until an estimate for the statistical error (48) is below the tolerance, TOL_S; then in the inner loop, for each realization, we apply our refinement strategy (52) or (55) to a given initial mesh iteratively until the error indicators satisfy the stopping criteria (53) or (56) with a given time discretization tolerance TOL_T. This procedure, in the inner loop, needs to sample the Wiener process W on finer partitions, given its values on coarser, which is accomplished by Brownian bridge refinements (5).

The adaptive algorithm based on the refinement (52) and the stopping (53) is called `Algorithm A` and the algorithm based on the refinement (55) and the stopping (56) is called `Algorithm B`. We first describe `Algorithm A` in detail and define the additional changes for `Algorithm B` afterwards.

`Algorithm A`

Initialization
Choose:

1. an error tolerance, $\text{TOL} \equiv \text{TOL}_S + \text{TOL}_T$,
2. a number $N[1]$ of initial uniform steps $\Delta t[1]$ for $[0, T]$, with $\text{TOL}\, N[1]$ bounded from above and below by positive constants, and set $\overline{N}[0] = N[1]$,

3. a number $M[1]$ of initial realizations, with $\mathrm{TOL}^2\, M[1]$ bounded from above and below by positive constants,
4. the stopping constant S in (53),
5. a positive constant c_0 for a confidence interval and an integer MCH≥ 2 to determine the number of realizations in (58),
6. a constant Δx for the difference quotient in (43), see Remark 4.

Set the iteration counter for realization batches $m = 1$ and the stochastic error to $\mathtt{E}_S[m] = +\infty$.

Do while ($\mathtt{E}_S[m] > \mathrm{TOL}_S$)
For realizations $j = 1, \ldots, M[m]$
Set the number of time levels for realization j to $k = 1$ and set the error indicator to $r[k] = +\infty$.
Start with the initial partition $\Delta t[k]$ and generate $\Delta W[k]$.
Compute for realization j, $g(\overline{X}(T))[J]$ and $N[J]$ by calling `routine Control--Time--Error` where $k = J$ is the number of final time levels for an accurate mesh of this realization.
end-for
Compute the sample average $Eg \equiv \mathcal{A}\left(g(\overline{X}(T)); M[m]\right)$, the sample standard deviation $\mathcal{S}[m] \equiv \mathcal{S}(g(\overline{X}(T)); M[m])$ and the a posteriori bound for the statistical error $\mathtt{E}_S[m] \equiv \mathtt{E}_S(g(\overline{X}(T)); M[m])$ in (48).
if ($\mathtt{E}_S[m] > \mathrm{TOL}_S$)
Discard all old $M[m]$ realizations and determine a larger $M[m+1]$ by `change_`M ($M[m]$, $\mathcal{S}[m]$, TOL_S; $M[m+1]$), in (58), and update $\overline{N} = \mathcal{A}\left(N[J]; M[m]\right)$, where the random variable $N[J]$ is the final number of time steps on each realization.
end-if
Increase m by 1.
end-do

Accept Eg as an approximation of $E[g(X(T))]$, since the estimate of the computational error is bounded by TOL.

`routine Control--Time--Error`$\big(\Delta t[k], \Delta W[k], r[k], \overline{N}[m-1];$
$g(\overline{X}(T))[J], N[J]\big)$

Do while ($r[k]$ violates the stopping (53))
Compute the Euler approximation $\overline{X}[k]$ in (6) and the error indicator $r[k]$ in (50) on $\Delta t[k]$ with the known Wiener increments $\Delta W[k]$.
if ($r[k]$ violates the stopping (53))
Do the refinement process (52) to compute $\Delta t[k+1]$ from $\Delta t[k]$ and compute $\Delta W[k+1]$ from $\Delta W[k]$ using Brownian bridges (57).
end-if
Increase k by 1.
end-do
Set the number of the final level $J = k - 1$.

`end of Control--Time--Error`

At the new time steps $t_i' \equiv (t_i[k] + t_{i+1}[k])/2$, on level $k+1$, the new sample points from W are constructed by the Brownian bridge, cf. [10],

$$W^\ell(t_i') = \frac{1}{2}\big(W^\ell(t_i[k]) + W^\ell(t_{i+1}[k])\big) + z_i^\ell \tag{57}$$

where z_i^ℓ are independent random variables, also independent of $W(t_j[k])$ for all i, j and ℓ, and each component z_i^ℓ is normal distributed with mean zero and variance $(t_{i+1}[k] - t_i[k])/4$.

`routine change_`M $(M_{in}, \mathcal{S}_{in}, \mathrm{TOL}_S; M_{out})$

$$\begin{aligned} M^* &= \min\Big\{\text{integer part } \Big(\tfrac{c_0\,\mathcal{S}_{in}}{\mathrm{TOL}_S}\Big)^2,\ \mathrm{MCH} \times M_{in}\Big\} \\ n &= \text{integer part } (\log_2 M^*) + 1 \\ M_{out} &= 2^n. \end{aligned} \tag{58}$$

`end of change_`M

Here $\mathrm{MCH} \geq 2$ is a positive integer parameter introduced to avoid a large new number of realizations in the next batch due to a possibly inaccurate sample standard deviation $\overline{\sigma}[m]$. Indeed, $M[m+1]$ cannot be greater than $\mathrm{MCH} \times M[m]$.

`Algorithm B`

In addition to the **Initialization** of `Algorithm A`, choose the error tolerances $\mathrm{TOL}_T = \mathrm{TOL}_C + \mathrm{TOL}_D$ and the stopping constants S_C and S_D in (56). Inside the **Do while** loop of `Algorithm A`, use $(r^C[k], r^D[k])$ in (54) instead of $r[k]$ and the refinement (55) and stopping (56).

4 Numerical Experiments

This section presents numerical results from a one dimensional problem with a `C++` implementation of `Algorithm A` and `Algorithm B` described in Sect. 3 and for a two dimensional problem with a corner singularity with `Matlab` implementation. The numerical results in 1D are obtained using the pseudo-random number generator, `drand48()`, in standard C library functions. The Box-Muller method is used to generate standard Gaussian random variable from the uniformly distributed pseudo-random numbers, see for example [11].

4.1 A One Dimensional Domain

In all computations, the following constants are chosen for the initialization of both `Algorithm A` and `Algorithm B`: the number of time steps in the initial partition, $\Delta t[1]$, of $[0,T]$ is $N[1] = 4$; the initial number of realizations is $M[1] = 128$; the stopping constant $S = 4$ is used in (53) and $S_C = 4, S_D = 1$ in (56); the constants to determine the number of realizations in (58) are $c_0 = 1.65$ and $MCH = 16$, and the constant $\Delta x = \mathrm{TOL}_T^{1/4}$ is used for the difference quotient in (43).

To describe the behavior of the adaptive algorithm, let us first define some notation. The index Q, which is the ratio between the approximate error and the exact error, is defined by

$$Q := \frac{E_{approx}}{E_{exact}} := \frac{\mathtt{E}_S + |\mathtt{E}_T|}{\left|E[g(X_\tau,\tau)] - \mathcal{A}(g(\overline{X}_{\overline{\tau}},\overline{\tau}); M)\right|}. \tag{59}$$

Here the statistical error $\mathtt{E}_S$ is defined by (48) and the time discretization error $\mathtt{E}_T$ is defined by

$$\mathtt{E}_T := \mathcal{A}\left(\sum_{n=0}^{N-1} \mathbf{1}_{t_{n+1}\leq\overline{\tau}}\rho_n \Delta t_n^2 + \left(g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\overline{X}_{\overline{\tau}},\overline{\tau})\right)\hat{P}_{\overline{X},n}; M\right),$$

where λ defines the domain $D = (-\infty, \lambda)$.

Example 1. Consider (2) with $d = 1$,

$$a(t,x) = \frac{11}{36}x, \quad b(t,x) = \frac{1}{6}x, \quad t \in [0,T], \quad x \in (-\infty, 2)$$

and the initial condition $X(0) = 1.6$ and $T = 2$. For $g(x,t) = x^3\mathrm{e}^{-t}$ with $x \in \mathbb{R}$, this problem has the exact solution $E[g(X_\tau,\tau)] = u(X(0),0) = X(0)^3$, where the solution u of the Kolmogorov backward equation (28) is $u(x,t) = x^3\mathrm{e}^{-t}$.

To check the behavior of the error expansion described in Sect. 2, Example 1 is constructed such that most of the realizations exit at $\overline{\tau} < T$, for instance, with $\mathrm{TOL} = 0.01$, 99% of the paths exit at $\overline{\tau} < T$ and $\mathcal{A}(\overline{\tau}; M) \simeq 0.77$.

Table 1 shows the comparisons between `Algorithm A` and `Algorithm B` for the computational results of Example 1. As the error tolerance TOL decreases, E_{exact} decreases and is bounded by a given TOL. The sample standard deviation of the number of time steps is around 35% of the average of the number of time steps. The histogram in Fig. 5 indeed shows that highly varying step sizes are used for individual realizations.

To check the accuracy of the error estimate in Sect. 2, choose the number of realizations M sufficiently large so that the total statistical error is small compared to the time discretization error. Here we use $M = 2^{22} = 4,194,304$,

Table 1. Example 1: Comparisons of the final number of the realizations, M, the sample average of the final number of steps, $\mathcal{A}(N;M)$, the sample standard deviation of the final number of steps, $\overline{\sigma}(N;M)$, and the exact error, E_{exact} for different error tolerances, TOL

		Algorithm A			Algorithm B		
TOL	M	$\mathcal{A}(N;M)$	$\overline{\sigma}(N;M)$	E_{exact}	$\mathcal{A}(N;M)$	$\overline{\sigma}(N;M)$	E_{exact}
0.5	2^7	27	11.7	0.028	24	6.9	0.02
0.1	2^{11}	81	30.6	0.024	84	25.8	0.06
0.05	2^{13}	126	44.0	0.015	158	54.2	0.02
0.01	2^{18}	453	170.7	0.003	700	287.7	0.005

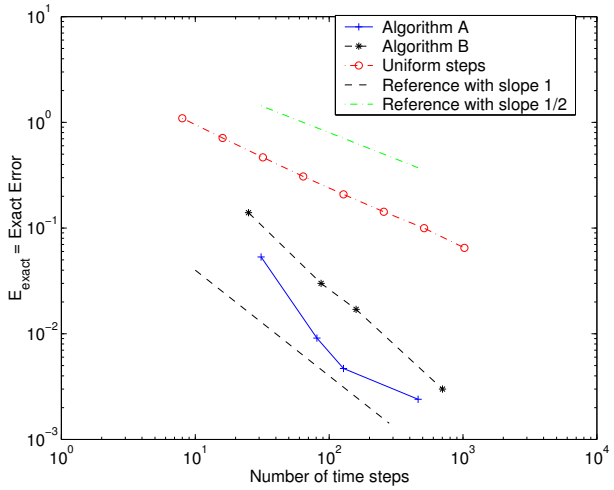


Fig. 3. Example 1: Comparison of the convergence rates with uniform and adaptive meshes. The convergence rate of the adaptive method is of order N^{-1} with N adaptive time steps, while the rate for the uniform method is of order $N^{-1/2}$ with N uniform time steps

which makes the statistical error approximately 0.001. Then the comparison of the convergence between the uniform and the adaptive method is shown in Fig. 3. The x-axis denotes the number of time steps for the uniform method and the sample average of the final number of steps for the adaptive method. The y-axis is the exact error E_{exact} defined by (59). The number of steps $N = 2^k, k = 3, 4, \ldots, 10$ are used for the uniform method and for adaptive method the tolerances TOL $= 0.5, 0.1, 0.05$ and 0.01 are used. Figure 3 shows that the convergence rate of the adaptive method is of order N^{-1} with N adaptive time steps, while the uniform method converges with the rate $N^{-1/2}$ with N uniform time steps.

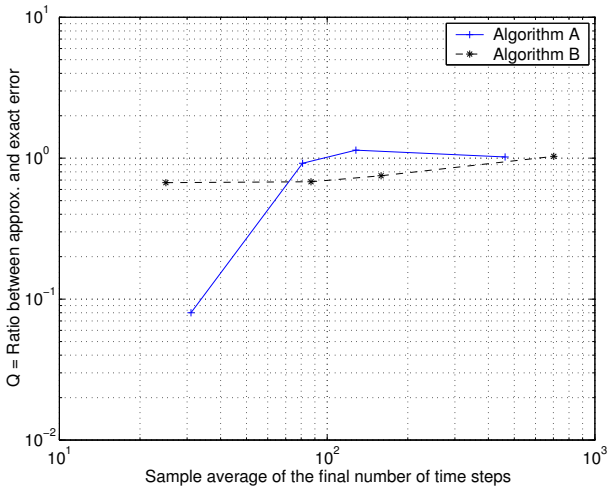


Fig. 4. Example 1: The ratio of the approximate and exact error on adaptive mesh. The ratio tends to 1 as the number of time steps increases

Figure 4 shows the convergence of the ratio Q between the approximate and the exact error in (59), still with $M = 2^{22}$ so that the statistical error is negligible. As predicted by Theorem 1 and 2, Fig. 4 shows that the ratio Q tends to 1 as N increases. From Fig. 3 and 4, `Algorithm B` seems more stable than `Algorithm A` for Example 1, on the other hand `Algorithm A` achieves smaller exact error for the same number of time steps.

Figure 5 shows the histogram of the step sizes depending on the distance from the boundary with TOL $= 0.05$ and $M = 2^{22}$ realizations of `Algorithm A`. The histogram of `Algorithm B` also has a similar appearance. The x-axis denotes base 2 log-scale of the step size, ranging from 2^{-35} to 2^{-5}, the y-axis denotes base 2 log-scale of the distance from the boundary, ranging from 2^{-20} to 1, and the z-axis denotes base 2 log-scale of the number of steps. To compensate the large error near the boundary, relatively small step sizes are used close to the boundary compared to further away from the boundary.

4.2 A Two Dimensional Domain

The methods described in the previous sections are implemented for a two dimensional domain, with a corner, which does not satisfy the conditions of smoothness required in [7]. The idea is to find out if the adaptive method can give some improvements to the standard Euler algorithm even though the approximations of the exit probabilities are somewhat incorrect. The known methods for improving the time discretization error rely on the possibility to locally approximate the boundary by its tangent plane. This is obviously difficult in the case for domains with sharp corners.

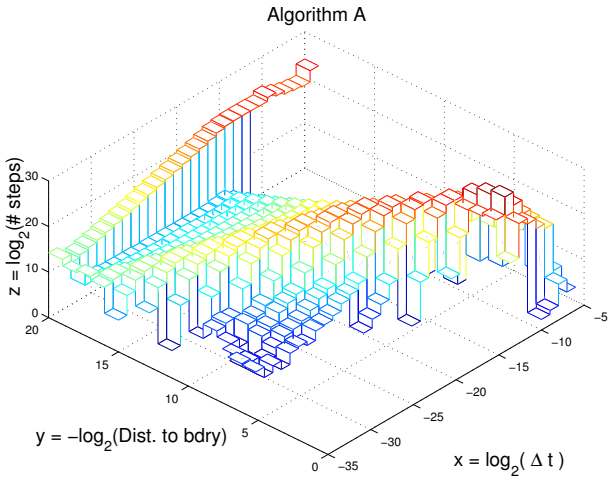


Fig. 5. Example 1: The histogram of the step sizes depending on the distance from the boundary using `Algorithm A`. Relatively small step sizes are used close to the boundary to improve the accuracy

The method used by Gobet [7] is strictly dependent on the value of the exit probability of the continuous Euler process between two time levels. When dealing with domains with non smooth boundary, for example corners, this method may give large errors, since it makes use of the assumption that the boundary can be locally approximated by its tangent plane. A domain with a sharp corner, however, cannot be locally well approximated as a tangent plane.

A prerequisite for any adaptive method is some sort of error estimate to decide which regions need refining and which do not. However, one of the advantages of adaptive methods in general is that they do not require a great deal of exactness in this error estimate in order to function in a satisfactory manner. In fact, it is often enough to check that the behavior of the error estimate is qualitatively similar to the real error, i.e. that the estimate increases and decreases similarly as the actual error.

The domain D for our test problem is chosen to be the one shown in Fig. 6 and in this domain we consider the problem

$$\begin{aligned} &u_t + \frac{1}{2}\Delta u = 0, \quad t < T \\ &u(\cdot, T) = g(\cdot, T) \\ &u(x,t) = g(x,t), \quad x \in \partial D \end{aligned} \tag{60}$$

which is solved by the expectation $u(x,t) = E[F(X_\tau, \tau) \mid X_t = x]$ for a pure Brownian motion $\mathrm{d}X_j(t) = \mathrm{d}W^j(t),\ j = 1, 2$, where $F = g(\cdot, \tau)$ if the process X exited D first at time τ before T, and $F = g(\cdot, T)$ if no exit occurred before T.

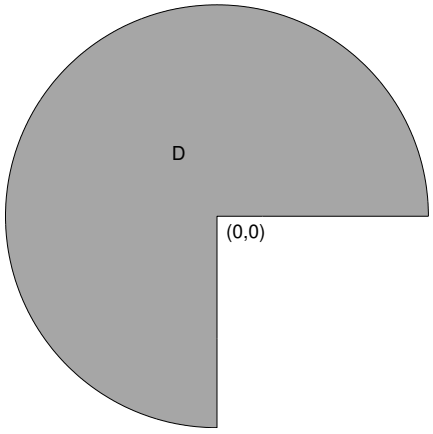


Fig. 6. The computational domain D with a corner at the origin

The boundary condition is chosen so that the behavior of the process W near the corner has a much greater impact than the behavior near the arc. Therefore, the boundary condition is chosen as $g(x,t) = 10\mathrm{e}^{-\sqrt{x^2+y^2}-0.1t}$ and we let the process start within D, near the corner at the origin. We also choose a large enough radius, $R = 10$, of the arc boundary and short enough time interval, $T = 1$, so it becomes highly unlikely for the process to reach the arc. The goal is to approximate $u(-0.209, 0.249, 0) = 0.544$.

The algorithm for this type of domain differs from the one for smooth domains only in the approximation of the exit probabilities P_i. To apply the algorithms formally, it is assumed that the corner is slightly 'rounded'. In the quadrant $x_1 < 0$ and $x_2 > 0$ it can then be imagined that the corner is a circular arc with infinitely small radius, in which case an inward pointing normal vector from the boundary to a point X_t is simply given by X_t itself. The tangent plane must then be orthogonal to X_t and pass through the corner at the origin. By proceeding in this way the tangent plane is quite easy to find, but it is obviously not a good approximation of the boundary near the corner. Using this crude estimate for the tangent plane it becomes easy to calculate distances to the tangent planes of the points in the Euler path, and thereby to calculate rough estimates of the exit probabilities. Near the corner, these estimates of the exit probabilities will, however, be quite far from correct. In all three quadrants the algorithm will over-estimate the exit probabilities.

The adaptive algorithm proceeds as described in the previous sections but now using the exit probabilities P_i as described above and

$$r_i = \Big(g\big(\mathrm{proj}_{\partial D}\big(\frac{1}{2}(\overline{X}_{t_i} + \overline{X}_{t_{i+1}})\big), \frac{t_i + t_{i+1}}{2}\big) - g(\overline{X}_{\overline{\tau}}, \overline{\tau})\Big)\widehat{P}_i,$$

where $\mathrm{proj}_{\partial D}$ is the orthogonal projection to the boundary ∂D. A dilemma arises when trying to calculate the error estimate for the case when the discrete process crosses the 'tangent plane' but does not exit the domain D. An

example of this is shown in Fig. 7. When calculating the exit probability for such a step, Mannella's and Gobet's method would proceed as earlier, and consider that the process indeed has exited the domain. For the adaptive method however, this seems an unnecessarily erroneous way to proceed, and the exit probability is calculated by reflecting the point which has exited back onto the other side of the tangent plane. This procedure results in a completely incorrect exit probability for some steps, but as this does not occur too often, it seems to be an acceptable way of testing the convergence properties of the adaptive algorithm. It is important to note that it is necessary to limit the refinement, for example by limiting the length of the time steps so that the incorrect behavior of the exit probabilities for these few steps will not cause the algorithm to refine indefinitely.

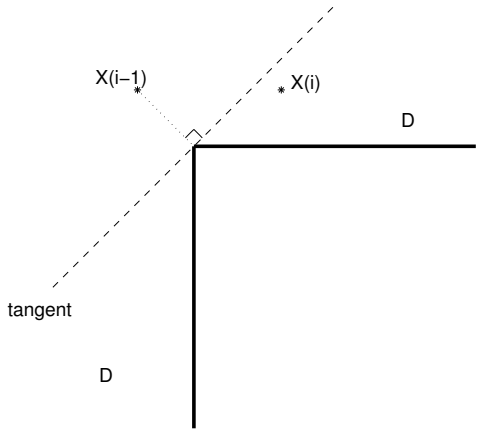


Fig. 7. The discrete process has crossed the 'tangent plane' but is still within D

The solution u of (60) has large derivatives near the corner. This resulted in an even slower convergence rate than $\mathcal{O}(N^{1/2})$ for the standard Euler algorithm, see Fig. 8. Even so, a considerable improvement in the convergence was achieved by using the adaptive algorithm for stochastic differential equations, resulting in a convergence rate which is better than $O(\frac{1}{N})$ and maybe even an exponential rate for this case with $\mathrm{d}X = \mathrm{d}W$, see Fig. 9. As seen in Fig. 9, our implementation of Mannella's and Gobet's method in the corner case gave only a slight improvement to the standard Euler method and was not as effective as the adaptive algorithm. The number of realizations, M, was chosen so that the statistical error was negligible as compared to the time discretization error. For this purpose, $M = 2^{22}$ proved sufficient.

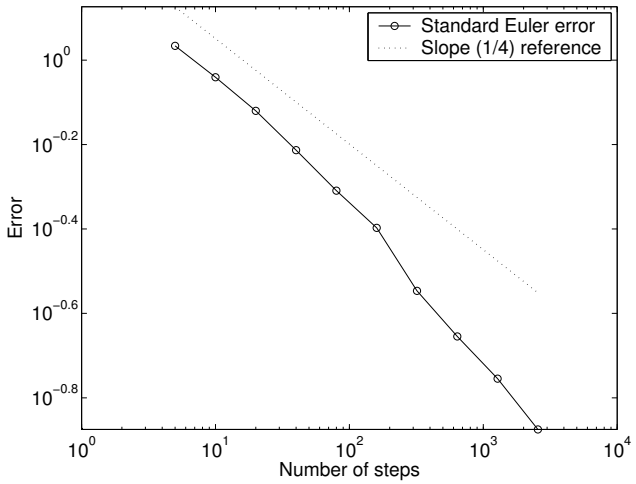


Fig. 8. Error for the Standard Euler method

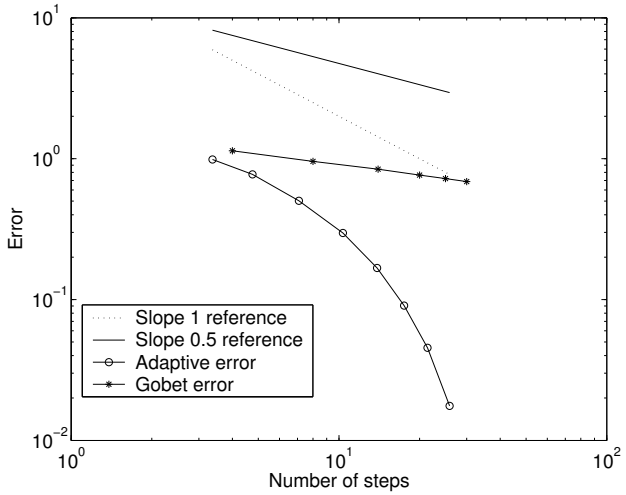


Fig. 9. Error for the adaptive algorithm and Gobet's method

Acknowledgments

The authors thank Michael Tehranchi for the reference [1]. This work is supported by the Swedish Research Council grants 2002-6285 and 2002-4961, the Swedish Foundation for Strategic Research, and the European network HYKE, funded by the EC as contract HPRN-CT-2002-00282.

References

1. Abundo M.: Some conditional crossing results of Brownian motion over a piecewise-linear boundary. Statist. Probab. Lett. **58**, no. 2, 131–145, (2002)
2. Baldi P.: Exact asymptotics for the probability of exit from a domain and applications to simulation. Ann. Probab. **23**, no. 4, 1644–1670, (1995)
3. Baldi P., Caramellino L. and Iovino M.G.: Pricing general barrier options: a numerical approach using sharp large deviations. Math. Finance **9**, no. 4, 293–322, (1999)
4. Bally V. and Talay D.: The law of the Euler scheme for stochastic differential equations, I. Convergence rate of the distribution function. Probab. Theory Related Fields **104**, no. 1, 43–60, (1996)
5. Buchmann F.M.: Computing exit times with the Euler scheme. Research report no. **2003-02**, ETH, (2003).
6. Fleming W.H. and James M.R.: Asymptotic series and exit time probabilities. Ann. Probab. **20**, no. 3, 1369–1384, (1992)
7. Gobet E.: Weak approximation of killed diffusion using Euler schemes. Stochastic Process. Appl. **87**, no. 2, 167–197, (2000)
8. Gobet E.: Euler schemes and half-space approximation for the simulation of diffusion in a domain. ESAIM Probab. Statist. **5**, 261–297, (2001)
9. Jansons K.M. and Lythe G.D.: Efficient numerical solution of stochastic differential equations using exponential timestepping. J. Stat. Phys. **100**, no. 5/6, 1097–1109, (2000)
10. Karatzas I. and Shreve S.E.: Brownian motion and stochastic calculus. Graduate Texts in Mathematics, **113**. Springer-Verlag, New York, (1991)
11. Kloeden P.E. and Platen E.: Numerical solution of stochastic differential equations. Applications of Mathematics, **23**. Springer-Verlag, Berlin, (1992)
12. Lépingle D.: Un schéma d'Euler pour équations différentielles stochastiques réfléchies. C. R. Acad. Sci. Paris Sér. I Math. **316**, no. 6, 601–605, (1993)
13. Mannella R.: Absorbing boundaries and optimal stopping in a stochastic differential equation. Phys. Lett. A **254**, no. 5, 257–262, (1999)
14. Moon K.-S.: Adaptive Algorithms for Deterministic and Stochastic Differential Equations. PhD Thesis, Royal Institute of Technology, Department of Numerical Analysis and Computer Science, Stockholm (2003)
15. Moon K.-S., Szepessy A., Tempone R. and Zouraris G.E.: Convergence rates for adaptive weak approximation of stochastic differential equations. Accepted in Stoch. Anal. Appl. (2005)
16. Petersen W.P. and Buchmann F.M.: Solving Dirichlet problems numerically using the Feynman-Kac representation. BIT **43**, no. 3, 519–540, (2003)
17. Szepessy A., Tempone R. and Zouraris G.E.: Adaptive weak approximation of stochastic differential equations. Comm. Pure Appl. Math. **54**, no. 10, 1169–1214, (2001)
18. Talay D. and Tubaro L.: Expansion of the global error for numerical schemes solving stochastic differential equations. Stochastic Anal. Appl. **8**, no. 4, 483–509, (1990)

Paper III

An Adaptive Algorithm for Ordinary, Stochastic and Partial Differential Equations

Kyoung-Sook Moon, Erik von Schwerin, Anders Szepessy, and Raúl Tempone

Abstract. The theory of a posteriori error estimates suitable for adaptive refinement is well established. This work focuses on the fundamental, but less studied, issue of convergence rates of adaptive algorithms. In particular, this work describes a simple and general adaptive algorithm applied to ordinary, stochastic and partial differential equations with proven convergence rates. The presentation has three parts: The error approximations used to build error indicators for the adaptive algorithm are based on error expansions with computable leading order terms. It is explained how to measure optimal convergence rates for approximation of functionals of the solution, and why convergence of the error density is always useful and subtle in the case of stochastic and partial differential equations. The adaptive algorithm, performing successive mesh refinements, either reduces the maximal error indicator by a factor or stops with the error asymptotically bounded by the prescribed accuracy requirement. Furthermore, the algorithm stops using the optimal number of degrees of freedom, up to a problem independent factor.

1. Introduction to the Adaptive Algorithm

This work presents an overview of the authors work on the convergence rate of an adaptive algorithm to compute functionals of solutions to ordinary, stochastic and partial differential equations. The main ingredient of the adaptive algorithm is an error expansion of the form

$$\text{Global error} = \sum \text{local error} \cdot \text{weight} + \text{higher order error}, \tag{1}$$

with computable leading order terms. The weight is the sensitivity of the functional of the solution with respect to perturbations in the differential equation. For an ordinary differential equation, the error expansion (1) can be derived by the variational principle in [**24**] and for weak approximation of stochastic differential equations the error expansion (1) can be derived based on computable stochastic

1991 *Mathematics Subject Classification.* 65L50,65C30, 65N50.

Key words and phrases. adaptive methods, mesh refinement algorithm, a posteriori error estimate, computational complexity.

Support from the Swedish Research Council grants 2002-6285 and 2002-4961, UdelaR in Uruguay, a Swedish Foundation for Strategic Research grant and the European network HYKE, funded by the EC as contract HPRN-CT-2002-00282, is acknowledged.

flows and discrete dual backward problems in [**29**]. For partial differential equations, [**22**] derives an asymptotic expansion of the error using the dual weighted residual method.

The goal of the adaptive algorithm for differential equations based on these error expansions, is to approximate the desired quantity of interest with an adapted mesh using as few mesh elements as possible, for a given level of accuracy using a given approximation method, for instance the Euler method or piecewise bilinear finite elements with varying mesh size. Based on the a posteriori error expansions of the form (1) the global error can be asymptotically approximated by

$$\text{Global error} \approx \sum_{\mathcal{K}} \text{error indicator}, \tag{2}$$

where $\mathcal{K}$ is a set of time steps or elements. A typical adaptive algorithm does two things iteratively :

- **(i)** if the error indicators satisfy an accuracy condition, then it stops; otherwise
- **(ii)** the algorithm chooses where to refine the mesh, recomputes the error indicators and then makes an iterative step to **(i)**.

Therefore the indicators not only estimate the localization of the global error but also give information on how to refine the mesh in order to achieve optimal efficiency.

Despite the established use of adaptive algorithms and the well developed theory of a posteriori error estimates, e.g. [**1, 2, 3, 4, 5, 7, 8, 9, 17, 18, 19, 20**] the theoretical study of adaptive mesh refinement algorithms is more recent, cf. [**6, 10, 13, 15, 23, 22, 25, 27, 28**]. To introduce the main ingredients, let us consider a simple integration problem, namely for a given function $X : [0, T] \to \mathbb{R}$ approximate the integral $g(X) = \int_0^T X(t)dt$. Let us first discretize the time interval $[0, T]$ into N subintervals $0 = t_0 < t_1 < \cdots < t_N = T$ with corresponding time steps $h_n := t_{n+1} - t_n$. Now, if we approximate $g(X)$ using the left point rule (forward Euler) denoted by $\bar{g}$, our global discretization error becomes

$$\text{Global Error } = g(X) - \bar{g} = \sum_{n=0}^{N-1} (h_n)^2 \rho_n + \text{ higher order terms}, \tag{3}$$

with the error density function ρ defined by $\rho_n := \frac{dX}{dt}(t_n)/2$. From the definition of the number of time steps

$$N(h) := \int_0^T \frac{1}{h(\tau)} d\tau, \tag{4}$$

the number N_u of uniform steps to reach a given level of accuracy TOL turns out to be asymptotically proportional to the L^1-norm of the function ρ, i.e.

$$N_u \simeq \frac{T}{\text{TOL}} \|\rho\|_{L^1(0,T)}.$$

On the other hand, by minimizing the number of steps in (4) subject to an accuracy constraint, i.e. imposing the leading order of (3) to be TOL, a standard Lagrangian relaxation yields the optimal choice

$$h_n^2 \, \rho_n = \text{ constant for all } n$$

and as a consequence the number N_a of adaptive steps becomes proportional to the smaller $L^{\frac{1}{2}}$ quasi-norm of the error density, i.e.

$$N_a \simeq \frac{1}{\mathrm{TOL}} \|\rho\|_{L^{\frac{1}{2}}(0,T)}.$$

For instance, if we have a singular case, $X(t) = 1/\sqrt{t}$, we can instead compute with $X_\epsilon(t) = 1/\sqrt{t+\epsilon}$, choosing the positive parameter ϵ such that

$$\left| \int_0^T (X(t) - X_\epsilon(t))\, dt \right| = o(\mathrm{TOL})$$

i.e. $\epsilon^{1/2} \lesssim o(\mathrm{TOL})$. Therefore, the number of uniform time steps becomes

$$N_u \simeq \frac{T/4}{\mathrm{TOL}} \int_0^T \frac{dt}{(t+\epsilon)^{3/2}} \simeq \frac{T/4}{\mathrm{TOL}} \frac{1}{\epsilon^{1/2}} \gtrsim \mathcal{O}(\mathrm{TOL}^{-2})$$

while the number of adaptive time steps is the smaller

$$N_a \simeq \frac{1/4}{\mathrm{TOL}} \left(\int_0^T \frac{dt}{(t+\epsilon)^{3/4}} \right)^2 \approx \mathcal{O}(\mathrm{TOL}^{-1})$$

which clearly shows the advantage of an adaptive approach. In the sequel we will consider multiscale problems, i.e. problems that have very different scales so that the error density, although being bounded, may be very large, e.g. $\frac{1}{\epsilon^{3/2}}$.

Thus, having motivated the need for adaptivity in the previous example, we now state the main questions to answer, namely
What is the notion of error density for ordinary, stochastic and partial differential equations?
What is a suitable approximation for such an error density?
What can be concluded about the convergence rate of the adaptive algorithm?

In this paper, Section 2 describes an adaptive algorithm based on previously derived a posteriori error expansions for ODEs [**24**], SDEs [**25**], and PDEs [**22**]. Then, Section 3 presents results on the convergence rates of the adaptive algorithm and Section 4 illustrates the behavior of the adaptive algorithm using a numerical example.

2. Convergence of the Error Density and the Adaptive Algorithm

2.1. An Error Expansion for ODEs. Let us consider an ordinary differential equation (ODE) of the form

$$\frac{dX(t)}{dt} = a(t, X(t)),\ 0 < t \leq T, \tag{5}$$

with an initial value $X(0) = X_0 \in \mathbb{R}^d$ and a given flux $a : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$. First discretize the time interval $[0,T]$ into N subintervals $0 = t_0 < t_1 < \ldots < t_N = T$ and let $\overline{X}$ be an approximation of X in (5) by any p-th order accurate numerical method, satisfying the same initial condition $\overline{X}(0) = X(0) = X_0$.

We are interested in computing a function value $g(X(T))$ for a given general function $g : \mathbb{R}^d \to \mathbb{R}$, which may represent the quantities of physical interest. Using the variational principle, we show in [**24**] that the global error has the expansion

$$g(X(T)) - g(\overline{X}(T)) = \sum_{n=1}^{N} \left(\bar{e}(t_n), \bar{\Psi}(t_n) \right) + \int_0^T o(h^p(t)) dt, \tag{6}$$

where $(\cdot,\cdot)$ is the standard scalar product on $\mathbb{R}^d$. Here the approximate local error is defined by $\bar{e}(t_n) := \gamma(\bar{\overline{X}}(t_n) - \overline{X}(t_n))$, where γ is a Richardson extrapolation constant and the approximate local exact solution $\bar{\overline{X}}$ is computed with smaller time steps or a higher order method than $\overline{X}$. The weight $\bar{\Psi}$ is an approximation of Ψ, which solves the dual equation

$$
\begin{aligned}
-\frac{d\Psi(s)}{ds} &= (a')^*(s, X(s))\Psi(s),\ s < T, \\
\Psi(T) &= g'(X(T)),
\end{aligned}
\tag{7}
$$

where $(a')^*(s,x)$ is the transpose of the Jacobian matrix.

Therefore the leading order term in (6) has the approximate error density

$$
\bar{\rho}_n := \frac{\left(\bar{e}(t_n), \bar{\Psi}(t_n)\right)}{h_n^{p+1}},
$$

which is then used in the adaptive algorithm, see Section 2.4.

2.2. An Error Expansion for SDEs. Let us consider an Itô Stochastic differential equation (SDE) of the form

$$
dX_k(t) = a_k(t, X(t))dt + \sum_{\ell=1}^{\ell_0} b_k^\ell(t, X(t))dW^\ell(t),\ t > 0,
\tag{8}
$$

where $k = 1, \ldots, d$ and $(X(t;\omega))$ is a stochastic process in $\mathbb{R}^d$, with randomness generated by the independent one dimensional Wiener processes $W^\ell(t;\omega)$, $\ell = 1, \ldots, \ell_0$, on the probability space $(\Omega, \mathcal{F}, P)$. The functions $a(t,x) \in \mathbb{R}^d$ and $b^\ell(t,x) \in \mathbb{R}^d$, $\ell = 1, \ldots, \ell_0$, are given drift and diffusion fluxes.

The goal is to construct approximations to the expected value $E[g(X(T))]$ by a Monte Carlo method, for a given function $g : \mathbb{R}^d \to \mathbb{R}$. Examples of such an expected value are the computation of option prices in mathematical finance, stochastic climate prediction, wave propagation in random media, etc. The Monte Carlo Euler method approximates the unknown process X by the Euler method $\overline{X}(t_n)$ which is a time discretization based on the nodes $0 = t_0 < t_1 < \cdots < t_N = T$ where

$$
\overline{X}(t_{n+1}) - \overline{X}(t_n) = h_n a(t_n, \overline{X}(t_n)) + \sum_{\ell=1}^{\ell_0} \Delta W_n^\ell b^\ell(t_n, \overline{X}(t_n)),
\tag{9}
$$

and $h_n \equiv t_{n+1} - t_n$, $\Delta W_n^\ell \equiv W^\ell(t_{n+1}) - W^\ell(t_n)$, $n = 0, 1, 2, \ldots, N-1$. The aim of the adaptive algorithm is to choose the size of the time steps, h_n, and the number of independent identically distributed samples $\overline{X}(\cdot, \omega_j)$, $j = 1, 2, \ldots, M$, such that the computational work, $N \cdot M$, is minimal while the approximation error is bounded by a given error tolerance, TOL, i.e. the event

$$
\left| E[g(X(T))] - \frac{1}{M} \sum_{j=1}^{M} g(\overline{X}(T;\omega_j)) \right| \leq \text{TOL}
\tag{10}
$$

has a probability close to one. A priori error estimates of the computational error in (10) was first derived by Talay and Tubaro in [**30**]. The work [**29**] modified Talay's and Tubaro's error expansion to an expansion with computable leading order term in a posteriori form, based on computable stochastic flows and discrete dual

backward problems. Stopped diffusion, including for example the barrier option, is an example where adaptive time steps improve the convergence rate, see [**14**].

Assume that the process X satisfies (8) and its approximation, $\overline{X}$, is given by (9), we have, see [**29, 25**]

THEOREM 2.1 (Error expansion for SDEs). *Suppose there are positive constants k and C and an integer m_0 with the bounds*

$$\begin{array}{l} g \in \mathcal{C}^{m_0}_{loc}(\mathbb{R}^d),\ |\partial_\alpha g(x)| \leq C(1+|x|^k), \quad \text{for all} \quad |\alpha| \leq m_0, \\ E\left[|X(0)|^{2k+d+1} + |\overline{X}(0)|^{2k+d+1}\right] \leq C, \end{array}$$

and

$$a \text{ and } b \text{ are bounded in } \mathcal{C}^{m_0}([0,T]\times\mathbb{R}^d).$$

Assume that $\overline{X}$ is constructed by the forward Euler method with step sizes h_n produced by the stochastic time step version of the adaptive algorithm in Section 2.4 and the corresponding $\Delta W_n \equiv W(t_{n+1}) - W(t_n)$ are generated by Brownian bridges. Assume also that $\overline{X}(0) = X(0)$ and $E[|X(0)|^{k_0}] \leq C$ for some $k_0 \geq 16$. Then the time discretization error has the expansion

$$\begin{aligned} E[g(X(T)) \quad - \quad g(\overline{X}(T))] &= E\left[\sum_{n=1}^{N} \bar{\rho}_n(h_n)^2\right] \\ &+\mathcal{O}\left(\sqrt{\frac{\mathrm{TOL}}{c(\mathrm{TOL})}}\left(\frac{C(\mathrm{TOL})}{c(\mathrm{TOL})}\right)^{8/k_0}\right) E\left[\sum_{n=1}^{N}(h_n)^2\right] \end{aligned} \tag{11}$$

with computable leading order terms, where

$$\begin{aligned} \bar{\rho}_n(t_n,\overline{X}) \quad \equiv \quad & \frac{1}{2}\left(\frac{\partial}{\partial t}a_k + \partial_j a_k a_j + \partial_{ij} a_k d_{ij}\right)\varphi_k(t_{n+1}) \\ & +\frac{1}{2}\left(\frac{\partial}{\partial t}d_{km} + \partial_j d_{km} a_j + \partial_{ij} d_{km} d_{ij} + 2\partial_j a_k d_{jm}\right)\varphi'_{km}(t_{n+1}) \\ & +\partial_j d_{km} d_{jr}\varphi''_{kmr}(t_{n+1}), \end{aligned} \tag{12}$$

with $d_{ij} = \frac{1}{2}\sum_{l=1}^{l_0} b_i^l b_j^l$. The terms in the sum of (12) are evaluated at the a posteriori known points $(t_n, \overline{X}(t_n))$, i.e.

$$\partial_\alpha a \equiv \partial_\alpha a(t_n, \overline{X}(t_n)),\ \partial_\alpha b \equiv \partial_\alpha b(t_n, \overline{X}(t_n)),\ \partial_\alpha d \equiv \partial_\alpha d(t_n, \overline{X}(t_n)).$$

Here $\varphi \in \mathbb{R}^d$ is the solution of the discrete dual backward problem

$$\begin{array}{l} \varphi_i(t_n) = \partial_i c_j(t_n, \overline{X}(t_n))\varphi_j(t_{n+1}),\ t_n < T, \\ \varphi_i(T) = \partial_i g(\overline{X}(T)), \end{array} \tag{13}$$

with

$$c_i(t_n, x) \equiv x_i + h_n a_i(t_n, x) + \Delta W_n^\ell b_i^\ell(t_n, x) \tag{14}$$

and its first and second variation

$$\varphi'_{ij} \quad \equiv \quad \partial_{x_j(t_n)}\varphi_i(t_n) \equiv \frac{\partial \varphi_i(t_n; \overline{X}(t_n) = x)}{\partial x_j}, \tag{15}$$

$$\varphi''_{ikm}(t_n) \quad \equiv \quad \partial_{x_m(t_n)}\varphi'_{ik}(t_n) \equiv \frac{\partial \varphi'_{ik}(t_n; \overline{X}(t_n) = x)}{\partial x_m}, \tag{16}$$

which satisfy

$$
\begin{aligned}
\varphi'_{ik}(t_n) &= \partial_i c_j(t_n, \overline{X}(t_n))\partial_k c_p(t_n, \overline{X}(t_n))\varphi'_{jp}(t_{n+1}) \\
&\quad +\partial_{ik} c_j(t_n, \overline{X}(t_n))\varphi_j(t_{n+1}),\ t_n < T, \\
\varphi'_{ik}(T) &= \partial_{ik} g(\overline{X}(T)),
\end{aligned} \tag{17}
$$

and

$$
\begin{aligned}
\varphi''_{ikm}(t_n) = \ & \partial_i c_j(t_n, \overline{X}(t_n))\partial_k c_p(t_n, \overline{X}(t_n))\partial_m c_r(t_n, \overline{X}(t_n))\varphi''_{jpr}(t_{n+1}) \\
& +\partial_{im} c_j(t_n, \overline{X}(t_n))\partial_k c_p(t_n, \overline{X}(t_n))\varphi'_{jp}(t_{n+1}) \\
& +\partial_i c_j(t_n, \overline{X}(t_n))\partial_{km} c_p(t_n, \overline{X}(t_n))\varphi'_{jp}(t_{n+1}) \\
& +\partial_{ik} c_j(t_n, \overline{X}(t_n))\partial_m c_p(t_n, \overline{X}(t_n))\varphi'_{jp}(t_{n+1}) \\
& +\partial_{ikm} c_j(t_n, \overline{X}(t_n))\varphi_j(t_{n+1}),\ t_n < T, \\
\varphi''_{ikm}(T) = \ & \partial_{ikm} g(\overline{X}(T)),
\end{aligned} \tag{18}
$$

respectively.

The previous result can also be directly applied to the particular case of deterministic time steps. Observe that the error expansion in Theorem 2.1 has the form

$$
E[g(X(T)) - g(\overline{X}(T))] = E\left[\sum_{n=1}^{N} \bar{\rho}_n h_n^2\right] + \text{ higher order terms} \tag{19}
$$

and due to the almost sure convergence of the density $\bar{\rho}_n$ as we refine the discretization, see [**25**], it is suitable for use in the adaptive algorithm.

The computational error in (10) naturally separates into the time discretization error and the statistical error

$$
\begin{aligned}
& E[g(X(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(T;\omega_j)) \\
&= \left(E[g(X(T)) - g(\overline{X}(T))]\right) + \left(E[g(\overline{X}(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\overline{X}(T;\omega_j))\right) \\
&\equiv \mathcal{E}_T + \mathcal{E}_S.
\end{aligned} \tag{20}
$$

The time steps for the realizations of the approximate solution $\overline{X}$ are determined from statistical approximations of the time discretization error, $\mathcal{E}_T$, and the number, M, of realizations of $\overline{X}$ is determined from the statistical error, $\mathcal{E}_S$. The statistical error and the time discretization error are combined in order to bound the computational error (20). Therefore we split a given error tolerance TOL into a statistical tolerance, TOL_S, and a time discretization tolerance, TOL_T. The computational work is roughly $\mathcal{O}(N \cdot M) = \mathcal{O}(\text{TOL}_T^{-1}\ \text{TOL}_S^{-2})$, therefore we use

$$
\text{TOL}_T = \frac{1}{3}\text{TOL} \ \text{ and } \ \text{TOL}_S = \frac{2}{3}\text{TOL}, \tag{21}
$$

by minimizing $\text{TOL}_T^{-1}\ \text{TOL}_S^{-2}$ under the constraint $\text{TOL}_T + \text{TOL}_S = \text{TOL}$.

From the central limit theorem, the statistical error is bounded by the following quantity, i.e. the event

$$
|\mathcal{E}_S(\overline{X}; M)| \leq \mathtt{E}_S(\overline{X}; M) \equiv c_0\ \frac{\mathcal{S}(\overline{X}; M)}{\sqrt{M}} \tag{22}
$$

has probability close to one, where $\mathcal{S}(\overline{X}; M)$ is the sample standard deviation of $\overline{X}$ and c_0 is a constant related to the confidence interval.

The time discretization error is approximated by (19) and its contribution from each of the realizations controlled according to the adaptive algorithm described in Section 2.4.

2.3. An Error Expansion for PDEs. Consider a problem to compute a linear functional

$$g(u) := \int_D u\, G dx$$

for a given function $G \in L^2(D)$ and u is the solution of a second order elliptic partial differential equation of the form

$$-\mathrm{div}(a\nabla u) = f \tag{23}$$

in a given open bounded domain $D \subset \mathbb{R}^d$ with Dirichlet boundary data $u|_{\partial D} = 0$.

The finite element approximation u_h, of u in (23), is based on the standard variational formulation in the function set V_h of continuous piecewise isoparametric bilinear quadrilateral functions in $H_0^1(D)$, using an adaptive quadrilateral mesh with hanging nodes cf. [**9**]. The Sobolev space $H_0^1(D)$ is the usual Hilbert space of functions on D, vanishing on ∂D, with bounded first derivatives in $L^2(D)$. Let $\mathcal{T}$ denote the set of convex quadrilaterals K and let h_K be the local mesh size, i.e. the length of the longest edge of K. Then the variational problems for $u \in H_0^1(D)$ and $u_h \in V_h$ are

$$\begin{aligned} \int_D a\nabla u \cdot \nabla v\, dx &= \int_D f\, v\, dx, \quad \forall v \in H_0^1(D), \\ \int_D a\nabla u_h \cdot \nabla v\, dx &= \int_D f\, v\, dx, \quad \forall v \in V_h. \end{aligned} \tag{24}$$

A central role in the dual weighted error representation for $g(u) - g(u_h)$ is played by the dual function $\varphi \in H_0^1(D)$ which satisfies

$$\int_D a\nabla\varphi \cdot \nabla v\, dx = \int_D G\, v\, dx, \quad \forall v \in H_0^1(D). \tag{25}$$

Besides, its finite element approximation $\varphi_h \in V_h$, defined by

$$\int_D a\nabla\varphi_h \cdot \nabla v\, dx = \int_D G\, v\, dx, \quad \forall v \in V_h \tag{26}$$

is used to construct the error density $\bar{\rho}$ in Theorem 2.2.

For general meshes the convergence of the error density does not hold, since the orientation of the elements varies. Thus, here the analysis considers the asymptotic behavior of the error density $\bar{\rho}$ for adaptive refinements, with general quadrilateral initial meshes: successive division of reference square elements into four similar squares generates hanging node meshes consisting of unions of structured adapted meshes, where each structured mesh has the domain of an initial element; viewed in the initial reference element the structured adaptive mesh is an adaptive hanging node mesh with square elements. We restrict the study to such unions of structured adaptive hanging node meshes. The use of quadrilaterals can directly be extended to higher space dimension using tensor reference elements. Other refinements using e.g. subdivision of a simplex, in three and higher dimensions cf. [**16**], generate new edges which are not parallel to the old and would require additional analysis.

There is a smooth mapping of each initial element to a square, so that the refined initial element is mapped to a square hanging node mesh. Let $\mathcal{T}_I$ denote the subset of elements with an edge on the initial mesh. Theorem 2.2 states that the error density has a precise expansion using that the isoparametric bilinear coordinate transformation $X^{-1} : [0,1]^2 \to K_I$ maps the square and the square hanging node mesh to the initial element K_I and its refined hanging node quadrilateral mesh.

Let us now study the transformation of the variational formulation under such a mapping $X : K_I \to [0,1]^2$

$$\sum_{ij} \int_{K_I} (a_{ij} \frac{\partial u_h}{\partial x_j} \frac{\partial v}{\partial x_i} - fv) dx = \int_{[0,1]^2} (aX' u_h' \cdot X' v' - fv) J dx'$$

where X' is the Jacobian of X and J is the Jacobian determinant. Here we abuse the notation by writing v instead of $(v \circ X^{-1})$ and similarly for a, u_h, and f, for $x \in K_I$. Besides, we write $v' = \frac{\partial v}{\partial x_i'}$ instead of $\frac{\partial (v \circ X^{-1})}{\partial x_i'}$.

Therefore the variational equation in the transformed coordinates, x', takes the same form with a and f replaced by $a^* \equiv J(X')^t a X'$ and $f^* \equiv Jf$, respectively. Note that a^* and f^* are as smooth on $X(K_I)$ as the functions a and f are on K_I. To avoid messy notation, we will not always use the prime notation for coordinates obviously in the reference elements; we will also avoid notation for the dependence of X on the initial element K_I and assume that we for a point $x \in D$ choose the mapping X that corresponds to the initial element K_I which contains x. We will use the set of transformed elements $\mathcal{T}' \equiv \{X(K) : \ K \in \mathcal{T}\}$.

To define the approximate error density, $\bar{\rho}$, we will use averages of second difference quotients as follows. Consider a function w which is defined on a discretization of an interval $[0, L]$ with nodes $\{x_j : \ j = 0, \ldots, \bar{N} + 1\} =: \bar{\mathcal{N}}$, where $x_0 = 0$ and $x_{\bar{N}+1} = L$. Let $h_+ \equiv x_{j+1} - x_j$ and $h_- \equiv x_j - x_{j-1}$ denote two consecutive edge sizes. Then define the average mesh size $\bar{h}$ and the difference quotients

$$\begin{aligned} \bar{h}_j &\equiv \frac{h_+ + h_-}{2} = \frac{x_{j+1} - x_{j-1}}{2}, \\ Dw(x_j) &\equiv \frac{w(x_j + h_+) - w(x_j)}{h_+} \\ D^2 w(x_j) &\equiv \frac{1}{\bar{h}_j} \left(\frac{w(x_j + h_+) - w(x_j)}{h_+} - \frac{w(x_j) - w(x_j - h_-)}{h_-} \right). \end{aligned} \tag{27}$$

Define $\overline{D^2 w} \in \mathbb{R}^{\bar{N}}$, implicitly as the solution $Y \in \mathbb{R}^{\bar{N}+2}$ of an auxiliary equation, i.e.

$$\begin{aligned} \overline{D^2 w}_n \equiv & Y_n, \ n = 1, \ldots, \bar{N}, \text{ where} \\ Y_n - \alpha^2 D^2 Y_n = & D^2 w_n, \quad n = 1, \ldots, \bar{N}, \end{aligned} \tag{28}$$

with homogeneous Neumann boundary conditions, $Y_0 = Y_1$, $Y_{\bar{N}} = Y_{\bar{N}+1}$. The work [**22**] reports numerical results of different alternative averages, including the fast nearest neighbor variant. The convergence proof requires α to be sufficiently large compared to the mesh size.

Let us define $\bar{h} D_i^2 w$ as the difference quotients $\bar{h} D^2 w$, in (27), with respect to the x_i' reference directions $i = 1, 2$, respectively, and analogously for $D_i w$. The

approximate error density, $\bar{\rho}$, in the transformed coordinates is now defined by

$$\overline{\rho}_K \equiv \frac{1}{48}\sum_{j=1}^{4}\left(a_{11}^*\overline{D_1^2 u_h}\ \overline{D_1^2\varphi_h} + a_{22}^*\overline{D_2^2 u_h}\ \overline{D_2^2\varphi_h}\right)(x_j^K) \tag{29}$$

where $x_1^K, x_2^K, x_3^K, x_4^K$ are the four corners of the square $K \in \mathcal{T}'$ illustrated in Figure 1.

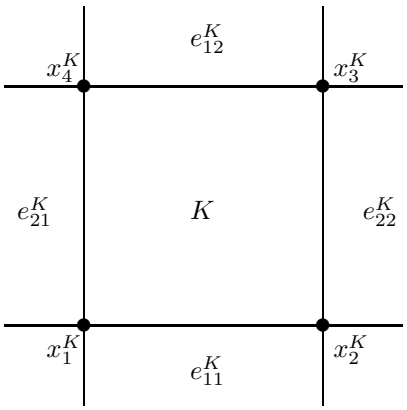


FIGURE 1. Corners x_j^K and edges e_{ij}^K of a square $K \in \mathcal{T}'$

Let $\mathcal{T}_H$ denote the subset of elements with hanging nodes in neighbors and let $\bar{\mathcal{T}}_H \equiv \bigcup_{K\in\mathcal{T}_H} K$. Let $W^{1,\infty}(D)$ denote the usual Sobolev space of functions with bounded first order derivatives in $L^\infty(D)$ and let h_{max} and h_{min} be the maximal and the minimal edge length in the mesh of V_h.

The proof of convergence of the error density, $\bar{\rho}$, uses the assumption that for some $\gamma \in (0, 1]$

$$\begin{aligned}\|u-u_h\|_{W^{1,\infty}(D)} + \|\varphi-\varphi_h\|_{W^{1,\infty}(D)} &= \mathcal{O}(C_\delta h_{max}),\\ \|u-u_h\|_{L^\infty(D)} + \|\varphi-\varphi_h\|_{L^\infty(D)} &= \mathcal{O}(h_{max}^{2\gamma}).\end{aligned} \tag{30}$$

In [**11**] such estimates are proved for finite element approximations of the coercive linear problems (23) and (25), with piecewise isoparametric bilinear quadrilateral elements and quasi uniform meshes, provided $u, \varphi \in \mathcal{C}^2(\bar{D})$; see [**12**] for nonlinear problems. If $|\bar{\rho}|$ is bounded away from zero, then the algorithm described in Section 2.4 produces quasi uniform meshes, h_{max}/h_{min} bounded by a constant that is possibly large depending on u and φ but independent of TOL. To include the case with $|\bar{\rho}|$ close to zero the algorithm uses a positive approximate error density $\hat{\rho} \geq \delta > 0$, defined in (42), with a parameter δ which tends to zero as TOL tends to zero; see Remark 2.3. With N denoting the number of elements in the final mesh, it is proved in [**22**] that

$$c\frac{\mathrm{TOL}}{N} \leq \hat{\rho}_K h_K^4 \leq S_1\frac{\mathrm{TOL}}{N}, \tag{31}$$

for problem independent constants c and S_1. With u and φ in $\mathcal{C}^3$ the quotient of the maximal and minimal mesh sizes becomes

$$\frac{h_{max}}{h_{min}} \leq C\Big(\frac{\hat{\rho}_{max}}{\hat{\rho}_{min}}\Big)^{1/4} \leq \frac{C_u}{\delta^{1/4}} \equiv \sqrt{C_\delta}, \tag{32}$$

where $C = (S_1/c)^{1/4}$ is independent of TOL, u and φ; while the constant C_u is independent of TOL but depends on u and φ. The δ-dependence of the quotient between h_{max} and h_{min} changes an $\mathcal{O}(h_{max})$ estimate of the right hand side in (30) to $\mathcal{O}(C_\delta h_{max})$.

The main result in [**22**] is

THEOREM 2.2. *Assume that $a \in \mathcal{C}^1(\bar{D})$ and that the solutions $u \in \mathcal{C}^3(\bar{D})$, $\varphi \in \mathcal{C}^3(\bar{D})$ of* (23) *and* (25)*, respectively, are for some $\gamma \in (0,1]$ approximated uniformly with error satisfying* (30) *using piecewise isoparametric bilinear quadrilateral elements and a refined mesh, with at most one hanging node per edge, obtained by successively dividing the reference square elements into four similar squares. Assume also that the total area of the elements with a hanging node on a neighbor or with an edge on the initial mesh is asymptotically zero:*

$$\int_{\bar{\mathcal{T}}_H \cup \bar{\mathcal{T}}_I} dx = o(1), \qquad \text{as } h_{max} \to 0+. \tag{33}$$

Then the global error has the expansion

$$g(u) - g(u_h) = \sum_{K\in\mathcal{T}'} \Big(\overline{\rho}_K + \mathcal{O}(h_{max}^\gamma/\alpha + \alpha)\Big)h_K^4 + \mathcal{O}(C_\delta h_{max}) \int_{\bar{\mathcal{T}}_H \cup \bar{\mathcal{T}}_I} h_K dx \tag{34}$$

with uniformly convergent computable error density $\bar{\rho}$, defined by (29) *and* (27)-(28) *for $\alpha^{-1} = o(h_{max}^{-\gamma})$, satisfying*

$$\bar{\rho} = \widetilde{\rho} + \mathcal{O}(h_{max}^\gamma/\alpha + \alpha) \tag{35}$$

where

$$\widetilde{\rho} \equiv \frac{1}{12}\left(a_{11}^* \frac{\partial^2 u}{\partial x_1^2}\frac{\partial^2 \varphi}{\partial x_1^2} + a_{22}^* \frac{\partial^2 u}{\partial x_2^2}\frac{\partial^2 \varphi}{\partial x_2^2}\right)$$

is evaluated in the transformed coordinates on $[0,1]^2$.

Note that the convergence of $\bar{\rho}$ is uniform while the convergence of $\check{\rho}$, defined by $g(u) - g(u_h) \equiv \sum_{K\in\mathcal{T}'} \check{\rho} h_K^4$, is in $L^1(D)$ by assumption (33). It is important to notice that our restriction of the data, required by $u,\ \varphi \in \mathcal{C}^3(\bar{D})$, includes examples with substantial adaptive gain. Section 3 shows that the optimal number of adaptive elements is $N^{opt} \simeq \mathrm{TOL}^{-1}\|\bar{\rho}\|_{L^{\frac{d}{d+2}}}^{d/2}$, while the number of uniform elements becomes $N^{uni} \simeq \mathrm{TOL}^{-1}\|\bar{\rho}\|_{L^1}^{d/2}$ to achieve the same error TOL. Although $u,\ \varphi \in \mathcal{C}^3(\bar{D})$ their norms in these spaces may be large so that $\|\bar{\rho}\|_{L^{\frac{d}{d+2}}} \ll \|\bar{\rho}\|_{L^1}$.

In general, second order difference quotients of the interpolant on meshes with hanging nodes do not converge uniformly on D and this is why the averages are needed [**22**].

Theorem 2.2 proves that the error expansion

$$g(u) - g(u_h) = \sum_{K\in\mathcal{T}'} (\overline{\rho}_K + \mathcal{O}(\frac{h_{max}^\gamma}{\alpha} + \alpha))h_K^{2+d} + \mathcal{O}(C_\delta h_{max}) \sum_{K\in\mathcal{T}_H' \cup \mathcal{T}_I'} h_K^{1+d} \tag{36}$$

has a well defined leading order error density $\overline{\rho}$ which converges uniformly as $h_{max} \to 0+$. We now assume that α has been chosen such that

$$\frac{h_{max}^{\gamma}}{\alpha} + \alpha = \mathcal{O}(h_{\max}^{\widehat{\gamma}}), \tag{37}$$

where $\widehat{\gamma} > 0$.

2.4. The Adaptive Algorithm. To motivate the approximate equidistribution of the error indicators in an adaptive algorithm, consider an asymptotic error expansion

$$\text{error} \simeq \sum_n \rho_n h_n^{p+d},$$

where h is the local isotropic mesh size and ρ is independent of h. The number of elements that corresponds to a mesh with size h can be determined by

$$N(h) \equiv \int_D \frac{dx}{h^d(x)}. \tag{38}$$

It seems hard to use the sign of the error indicators for constructing the mesh. Instead, we minimize the number of elements N in (38) under the more stringent constraint

$$\sum_{n=1}^{\bar{N}} |\rho_n| h_n^{d+p} = \int_D |\rho(x)| h^p dx = \text{TOL}, \tag{39}$$

with $D = [0, T]$ and $d = 1$ for ODEs and SDEs. The global order of convergence satisfies $p = 1$ for the Euler-Maruyama SDE approximation and $p = 2$ for the d-linear finite elements approximations used here. A standard application of a Lagrange multiplier yields the optimum

$$|\rho|(h^*)^{d+p} = \ \textit{constant} \tag{40}$$

and

$$h^* \equiv \frac{\text{TOL}^{\frac{1}{p}}}{|\rho|^{\frac{1}{d+p}}} \left(\int_D |\rho(x)|^{\frac{d}{d+p}} dx \right)^{-\frac{1}{p}}. \tag{41}$$

This condition is optimal only for density functions ρ with one sign, and in the PDE case, for meshes with shape regular elements, i.e. non stretched elements. To use the sign of the density or orientation of stretched elements in an optimal way is not considered here.

In the adaptive algorithm below we will use the positive approximate error density $\hat{\rho}_K$ defined by

$$\hat{\rho}|_K \equiv \hat{\rho}_K \equiv \min\left(\max\left(|\overline{\rho}_K|,\ \delta\right), \text{TOL}^{-r}\right) \tag{42}$$

with $r > 0$ and where the lower bound, $\delta > 0$, is chosen according to

Remark 2.3 (Lower bound for the error density).

$$\delta \equiv \text{TOL}^{\bar{\gamma}} \tag{43}$$

where the parameter $\bar{\gamma}$ is $0 < \bar{\gamma} < 1/(p+1)$ for ODEs, $\bar{\gamma} = 1/9$ for SDEs, and for PDEs it is chosen such that satisfies the two lower bounds

$$\bar{\gamma} < \frac{\widehat{\gamma}}{\widehat{\gamma} + 2} \quad \textit{and} \int_{\bar{\mathcal{T}}_H \cup \bar{\mathcal{T}}_I} dx/\delta = o(1) \quad \textit{as } \text{TOL} \to 0+, \tag{44}$$

and the upper bound $\delta = o(1)$ *as* $\mathrm{TOL} \to 0+$. *The lower bounds on* $\delta > 0$ *are motivated by the requirements that* $h_{max} \to 0$ *as* $\mathrm{TOL} \to 0$, *that the bounds for the error density in* (51) *hold and that the error from hanging node elements becomes asymptotically negligible, see Theorem 3.2. The convergence of* $\hat{\rho}$ *towards the exact density requires the upper bound* $\delta \to 0$.

The goal of the adaptive algorithm described below is to construct a mesh such that

$$\hat{\rho}_n h_n^{d+p} \approx \frac{\mathrm{TOL}}{N},\ n = 1, \ldots, N, \tag{45}$$

which is an approximation of the optimal (40). Let the index $[k]$ refer to the refinement level in the sequence of adaptively refined meshes. For a mesh with elements $\{K_1, K_2, K_3, \ldots, K_N\}$, we consider the piecewise constant error density and mesh functions $\rho|_{K_n} \equiv \rho_n \equiv \rho_{K_n}$, $\hat{\rho}|_{K_n} \equiv \hat{\rho}_n \equiv \hat{\rho}_{K_n}$ and $h|_{K_n} \equiv h_n \equiv h_{K_n}$. To achieve (45) let $s_1 \approx 1$ be a given constant, start with an initial mesh of size $h[1]$ and then specify iteratively a new mesh $h[k+1]$, from $h[k]$, using the following dividing strategy:

(46)

for all intervals (elements) $n = 1, 2, \ldots, N[k]$

$\bar{r}_n[k] \equiv \hat{\rho}_n[k](h_n[k])^{d+p}$

if $\bar{r}_n[k] > s_1 \dfrac{\mathrm{TOL}}{N[k]}$ **then**

mark interval (element) n for division.

(In addition, for the PDE case mark recursively all neighbors

that need division due to the hanging node constraint:

at most one hanging node per edge.)

endif

endfor

divide every marked interval (element) into 2^d uniform sub intervals (elements).

With this dividing strategy, it is natural to use the stopping criterion:

$$\textbf{if} \quad \Big(\max_{1 \le n \le N[k]} \bar{r}_n[k] \le S_1 \frac{\mathrm{TOL}}{N[k]} \Big) \quad \textbf{then} \quad \text{stop.} \tag{47}$$

Here S_1 is a given constant, with $S_1 > s_1 \approx 1$, determined more precisely as follows: we want that the maximal error indicator decays quickly to the stopping level $S_1 \mathrm{TOL}/N$, but when almost all error indicators $\bar{r}_n$ satisfy $\bar{r}_n < s_1 \frac{\mathrm{TOL}}{N}$ the reduction of the error may be slow. Theorem 3.1 shows that a slow reduction is avoided if S_1 satisfies (52).

Remark 2.4 (SDE case: Stochastic Time Steps). *Let* $\overline{g} \equiv \sum_{j=1}^{M} g(\bar{X}(T); \omega_j)$ *be the sample average approximation of the expected value* $E[g(X(T))]$ *and let* $\overline{N}[m]$ *be the sample average of the final number of time steps in the* m*-th batch of* $M[m]$ *realizations. In this case* (46), *is used iteratively for each of the realizations,* $j = 1, \ldots, M[m]$, *with* TOL_T *instead of* TOL *and with* $\overline{N}$ *instead of* N,

see [**25**]. *Replacing the integrals* $\int_D \dots\, dx$ *by* $\int\int \dots\, dt\, dP = E\int \dots\, dt$ *formally motivates the equidistribution of the error indicators for each realization of the Brownian motion.*

3. Convergence Rates for the Adaptive Mesh Algorithm

This section presents results on the stopping, accuracy and efficiency properties of adaptive algorithm introduced in Section 2.

3.1. Adaptive Refinements and Stopping. To analyze the decay of the maximal error indicator, it is useful to understand the variation of the density $\hat{\rho}$ at different refinement levels, in particular we will consider an element or time step $K[k]$ and its parent on a previous refinement level, $p(K,k)$, with the corresponding error density $\hat{\rho}(K)[p(K,k)]$. It is possible to verify that the choice (43) of δ implies that $h_{max} \to 0$ as TOL $\to 0+$, see [**22**], [**23**], [**25**]. Hence Theorem 2.2 shows, for the PDE, that there is a limit error density $\widetilde{\rho}$ such that

$$\check{\rho} \xrightarrow{L^1} \widetilde{\rho},\ \overline{\rho} \to \widetilde{\rho} \text{ and } \hat{\rho} \to |\widetilde{\rho}|,\ \text{as TOL} \to 0+. \tag{48}$$

Similarly, the choice (43) of δ is used to show in [**23**] for ODEs that

$$\hat{\rho} \to |\widetilde{\rho}|,\ \text{as TOL} \to 0+, \tag{49}$$

and in [**25**] that for each realization of the SDE, with $0 < \alpha < 1/2$,

$$\lim_{\text{TOL}\to 0+} h_{max}^{-\alpha}(\hat{\rho} - |\widetilde{\rho}|) = 0, \qquad \text{almost surely.} \tag{50}$$

A consequence of the uniform convergence $\hat{\rho} \to |\widetilde{\rho}|$, as TOL $\to 0+$, and (42) is that for all elements K and all refinement levels k there exists positive functions $\hat{c}$ and $\hat{C}$ close to 1 for sufficiently refined meshes, such that the error density satisfies

$$\begin{aligned} \hat{c}(K) &\leq \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]} \leq \hat{C}(K), \\ \hat{c}(K) &\leq \frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]} \leq \hat{C}(K), \end{aligned} \tag{51}$$

provided $\max_{K,k} h_K[k]$ is sufficiently small. In other words, (51) holds with e.g. $\hat{c} = 2^{-1} = \hat{C}^{-1}$ for sufficiently small $\max_{K,k} h_K[k]$. For SDEs the functions $\hat{c}$ and $\hat{C}$ are close to 1, almost surely.

THEOREM 3.1 (Stopping). *Suppose the adaptive algorithm uses the strategy* (46)-(47). *Assume that* $\hat{c}$ *satisfies* (51)*, for the elements or time steps corresponding to the maximal error indicator on each refinement level, and that*

$$S_1 \geq \frac{2^d}{\hat{c}} s_1, \quad 1 > \frac{\hat{c}^{-1}}{2^{d+p}}. \tag{52}$$

Then each refinement level either decreases the maximal error indicator with the factor

$$\max_{1\leq n\leq N[k+1]} \bar{r}_n[k+1] \leq \frac{\hat{c}^{-1}}{2^{d+p}} \max_{1\leq n\leq N[k]} \bar{r}_n[k], \tag{53}$$

or stops the algorithm.

Here, the global order of convergence is $p = 1$ for the Euler-Maruyama SDE approximation and $p = 2$ for the d-linear finite elements approximations.

3.2. Accuracy of the Adaptive Algorithm. The adaptive algorithm guarantees that the estimate of the global error is bounded by a given error tolerance, TOL. An important question is whether the true global error is bounded by TOL asymptotically. Using the upper bound (47) of the error indicators and the convergence of ρ and $\bar{\rho}$ in Theorem 2.2, or the convergence (49), (50) respectively, the global error has the following estimate.

THEOREM 3.2 (Accuracy). *Suppose* (42)–(43) *hold and that, for PDEs* (37) *and the assumptions of Theorem 2.2 hold, or, for ODEs and SDEs,* (49) *and* (50) *holds, respectively. Then the adaptive algorithm* (46)–(47) *satisfies*

$$\limsup_{\mathrm{TOL}\to 0+} \left(\mathrm{TOL}^{-1} \left| g(X(T)) - g(\overline{X}(T)) \right| \right) \leq S_1, \quad \textit{for the ODE,}$$

$$\limsup_{\mathrm{TOL}\to 0+} \left(\mathrm{TOL}^{-1} \left| g(u) - g(u_h) \right| \right) \leq S_1, \quad \textit{for the PDE,}$$

and, for the SDE, with the number of realizations M and any $c_0 > 0$ determined by (22)*,*

$$\liminf_{\mathrm{TOL}\to 0+} P\left(\frac{1}{\mathrm{TOL}} \left| E[g(X(T))] - \frac{1}{M} \sum_{j=1}^{M} g(\overline{X}(T;\omega_j)) \right| \leq \frac{S_1 + 2}{3} \right) \geq \int_{-c_0}^{c_0} \frac{e^{-x^2/2}}{\sqrt{2\pi}}\, dx.$$

3.3. Efficiency of the Adaptive Algorithm. An important issue for the adaptive method is its efficiency; we want to determine a mesh with as few elements or time steps as possible providing the desired accuracy. From the definition (38) and the optimality condition (41), the number of optimal adaptive elements, N^{opt}, satisfies

$$N^{\mathrm{opt}} = \int_D \frac{dx}{(h^*(x))^d} = \frac{1}{\mathrm{TOL}^{\frac{d}{p}}} \left(\int_D |\rho[k](x)|^{\frac{d}{d+p}} dx \right)^{\frac{d+p}{p}} = \frac{1}{\mathrm{TOL}^{\frac{d}{p}}} \|\rho\|_{L^{\frac{d}{d+p}}}^{\frac{d}{p}}. \tag{54}$$

On the other hand, for the uniform mesh with elements $h = \textit{constant}$, the number of elements, N^{uni}, to achieve $\sum_{i=1}^{N} |\rho_i| h^{d+p} = \mathrm{TOL}$ becomes

$$N^{\mathrm{uni}} = \int_D \frac{dx}{h^d(x)} = \frac{\int_D dx}{\mathrm{TOL}^{\frac{d}{p}}} \left(\int_D |\rho[k](x)| dx \right)^{\frac{d}{p}} = \frac{\int_D dx}{\mathrm{TOL}^{\frac{d}{p}}} \|\rho\|_{L^1}^{\frac{d}{p}}. \tag{55}$$

Hence, the number of uniform elements is measured in the L^1-norm while the optimal number of elements is measured in the $L^{\frac{d}{d+p}}$ quasi-norm. Jensen's inequality implies $\|f\|_{L^{\frac{d}{d+p}}} \leq (\int_D dx)^{\frac{p}{d}} \|f\|_{L^1}$, therefore an adaptive method may use fewer elements than the uniform element size method. For the SDE we get the optimal expected number of adaptive steps $E[N^{\mathrm{opt}}] = \frac{1}{\mathrm{TOL}} \left(E \int_0^T \sqrt{|\rho|}\, dt \right)^2 = \frac{1}{\mathrm{TOL}} \|\rho\|_{L^{\frac{1}{2}}(dtdP)}$ while with uniform time steps $E[N^{\mathrm{uni}}] = \frac{T}{\mathrm{TOL}} \int_0^T E|\rho|\, dt$.

The following theorem uses a lower bound of the error indicators, obtained from the refinement criterion (46) for the refined parent error indicator and the ratio of the error density (51), to show that the algorithm (46)-(47) generates a mesh which is optimal, up to a multiplicative constant independent of the data. In order to guarantee that, for sufficiently small TOL, all elements on the initial mesh are refined, the initial mesh size is assumed to obey

$$h_K[1] \geq \mathrm{TOL}^s, \tag{56}$$

where the parameter s has the upper bound $s < \frac{1-\bar{\gamma}}{p}$ and the lower bounds $0 < s$, for ODEs and SDEs, and $\frac{\bar{\gamma}}{\hat{\gamma}} < s$, for PDEs.

THEOREM 3.3 (Efficiency). *Assume that $\hat{C} = \hat{C}(t)$, $\hat{C}(t,\omega)$ or $\hat{C}(x)$ satisfies* (51) *for all elements at the final refinement level, that the assumptions of Theorem 3.2 hold, and that the initial mesh satisfies* (56) *for all elements K. Then there exists a constant $C > 0$, bounded by $(\frac{2^{d+p}}{s_1})^{\frac{d}{p}}$, such that, for sufficiently small* TOL*, the final number of adaptive time steps or elements N, of the algorithm* (46)-(47) *for ODEs or PDEs, satisfies*

$$(\mathrm{TOL}^{\frac{d}{p}} N) \leq C \, \|\hat{C}\hat{\rho}\|^{\frac{d}{p}}_{L^{\frac{d}{d+p}}} \leq C \left(\max_{x \in D} \hat{C}(x)^{\frac{d}{p}} \right) \|\hat{\rho}\|^{\frac{d}{p}}_{L^{\frac{d}{d+p}}}, \tag{57}$$

and

$$\lim_{\mathrm{TOL} \to 0+} \|\hat{\rho}\|_{L^{\frac{d}{d+p}}} = \|\widetilde{\rho}\|_{L^{\frac{d}{d+p}}},$$

$$\lim_{\mathrm{TOL} \to 0+} \max_{x \in D} \hat{C}(x)^{\frac{d}{p}} = 1,$$

i.e. the number of elements is asymptotically optimal up to the problem independent factor $C \leq (\frac{2^{d+p}}{s_1})^{\frac{d}{p}}$. For the SDE case the final sample average $\bar{N}[m] = \frac{1}{M[m]} \sum_{j=1}^{M[m]} N(\omega_j)$ of the number of adaptive steps of the algorithm (46)-(47) *satisfies*

$$\frac{\mathrm{TOL}_T \bar{N}[m]^2}{\bar{N}[m-1]} < C^2 \left(\int_0^T \frac{1}{M[m]} \sum_{j=1}^{M[m]} \sqrt{\hat{\rho}\hat{C}} \, dt \right)^2$$

and asymptotically

$$\limsup_{\mathrm{TOL}_T \to 0+} \mathrm{TOL}_T E[N] \leq C^2 \|\widetilde{\rho}\|_{L^{\frac{1}{2}}(dtdP)}$$

4. A Numerical Example

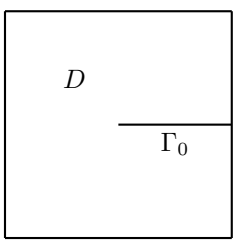
This section presents numerical results obtained with an implementation of the adaptive algorithm in Section 2.4 for a simplified elasticity problem using the error expansion (34) and the approximate error density (29). A more detailed description of the problem and the numerical results is given in [**22**].

EXAMPLE 4.1. The problem is to compute the functional

$$g(u) = \int_D u \, dx,$$

where the function u solves the Laplace equation in a slit domain

$$\begin{aligned} -\Delta u &= 0, && \text{in } D = (-1,1)^2 \setminus \Gamma_0, \\ u &= 0, && \text{on } \Gamma_0 = [0,1) \times \{0\}, \\ u &= u_b, && \text{on } \partial D \setminus \Gamma_0. \end{aligned}$$



With the function u_b given by $(r,\theta) \mapsto r^{1/2}\sin(\theta/2)$, in the polar coordinates (r,θ), the exact solution, u, is $r^{1/2}\sin(\theta/2)$. This singular problem is related to a problem

with a rounded tip, leading to a smooth solution with multiple scales, see [**22**]. Numerical results show that the adaptive algorithm needs far less elements than the number of uniform elements needed to get comparable accuracy for this problem; see Table 1. Figure 2 shows that the number of elements in the accepted adaptively refined mesh is close to the estimated optimal number.

The singular mode $r^{1/2}\sin(\theta/2)$ in u is present also in the solution to the dual problem

$$-\Delta\varphi = 1 \ \text{ in } D, \quad \varphi|_{\partial D} = 0.$$

Thus the conditions $u, \phi \in C^3(\bar{D})$ in Theorem 2.2 are violated and uniform convergence of the error density does not hold. Instead the error density grows like

$$\hat{\rho}(x) = \frac{\mathcal{O}(1)}{r^3}. \tag{58}$$

Still the numerical results in Figure 4 show that the minimal requirement on $\hat{c}$ and $\hat{C}$ in (51) to prove Theorem 3.1, 3.2 and 3.3 behave well.

refinements	N	error	error estimate	TOL
uniform	32768	$4.8 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$	
adaptive	725	$3.2 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$2.0 \cdot 10^{-3}$
	3464	$3.8 \cdot 10^{-5}$	$4.5 \cdot 10^{-5}$	$2.4 \cdot 10^{-4}$
	20288	$4.4 \cdot 10^{-6}$	$6.9 \cdot 10^{-6}$	$3.1 \cdot 10^{-5}$

TABLE 1. The adaptive algorithm (46)–(47) uses far less elements than the number of uniform elements needed to get comparable accuracy in Example 4.1. The error $g(u) - g(u_h)$ is estimated by $\sum_K \bar{\rho}_K h_K^4$ using the signed error density (29), spatially varying averaging $\alpha(x) = \sqrt{r(x)h(x)}$, and lower bound $\delta = \sqrt{\text{TOL}}$. The adaptive refinement and stopping used the parameter values $s_1 = 1$ and $S_1 = 10$.

References

[1] M. Ainsworth and J. T. Oden, A posteriori error estimation in finite element analysis, Comput. Methods Appl. Mech. Engrg., **142** (1997), 1-88.

[2] I. Babuška, J. Hugger, T. Strouboulis, K Copps and S.K. Gangaraj, The asymptotically optimal meshsize function for bi-p degree interpolation over rectangular elements. J. Comput. Appl. Math. 90 (1998), no. 2, 185–221.

[3] I. Babuška, A. Miller and M. Vogelius, Adaptive methods and error estimation for elliptic problems of structural mechanics, in *Adaptive computational methods for partial differential equations* (SIAM, Philadelphia, Pa., 1983) 57-73.

[4] I. Babuska and W. C. Rheinboldt, Error estimates for adaptive finite element computations, SIAM J. Numer. Anal., **15**(1978), 736–754.

[5] I. Babuska, T. Strouboulis and S. K. Gangaraj, Guaranteed computable bounds for the exact error in the finite element solution. I. One dimensional model problem, Comput. Methods Appl. Mech. Engrg., **176** (1999), 51–79.

[6] I. Babuška, and M. Vogelius, Feedback and adaptive finite element solution of one-dimensional boundary value problems, Numer. Math. **44** (1984), no. 1, 75-102.

[7] R. E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp. **44** (1985), 283–301.

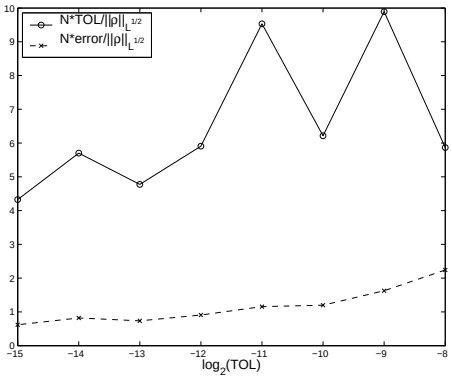


FIGURE 2. Efficiency estimate of the accepted mesh with the parameters in Table 1 for a sequence of tolerances. The number of elements, N, on the accepted mesh is compared to the estimated optimal $N^{\rm opt} = ||\hat{\rho}||_{L^{1/2}}/\mathrm{TOL}$, where the quasi-norm of the error density is computed on the finest mesh, corresponding to $\mathrm{TOL} = 2^{-15}$. Note that by the construction of the algorithm the final stopping error is expected to be only within a multiplicative factor $[1/4, 4]$ from the accuracy requirement TOL (without fine tuning of s_1).

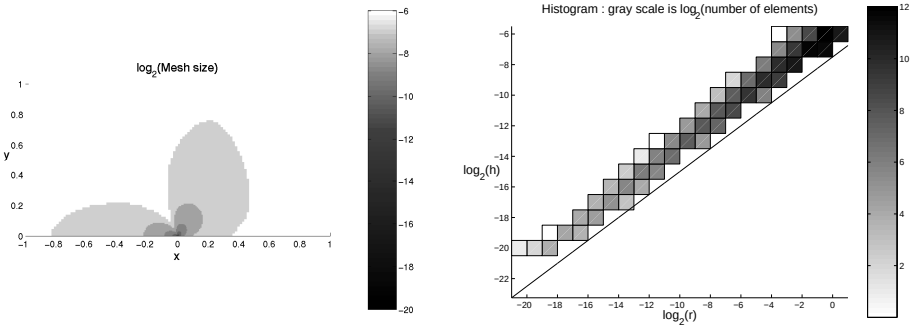


FIGURE 3. This is the mesh of the accepted solution corresponding to the smallest tolerance in Table 1. The element sizes h vary with the distance from the origin approximately like the optimal grading $r^{3/4}$. The reference line is the function $\sqrt{\mathrm{TOL}}\, r^{3/4}$.

[8] R. Becker and R. Rannacher, A feed-back approach to error control in finite element methods: basic analysis and examples, East-West J. Numer. Math., **4** (1996), no. 4, 237-264.

[9] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numerica, (2001), 1-102.

[10] P. Binev, W. Dahmen and R. DeVore, Adaptive finite element methods with convergence rates. Numer. Math. 97 (2004), no. 2, 219–268.

[11] S. C. Brenner and L. R. Scott, The mathematical theory of finite element methods, Texts in Applied Mathematics 15, Springer–Verlag, New York, 1994.

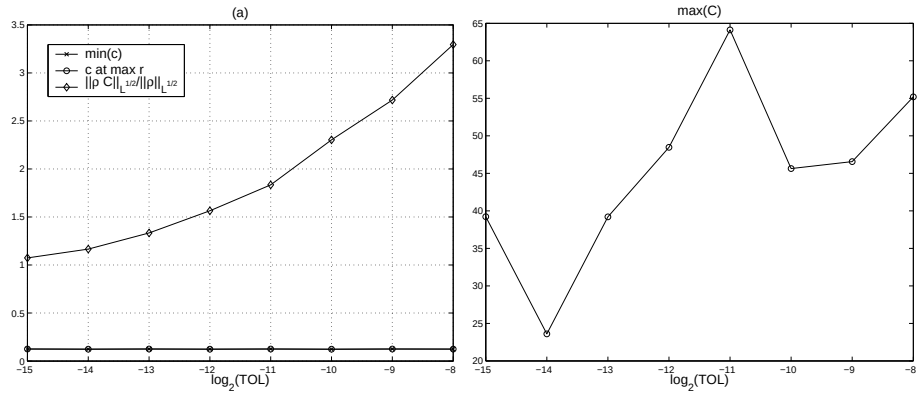


FIGURE 4. The quotients $\hat{c}(K) = \min\{\frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]}, \frac{\hat{\rho}(K)[k-1]}{\hat{\rho}(K)[k]}\}$ and $\hat{C}(K) = \frac{\hat{\rho}(K)[p(K,k)]}{\hat{\rho}(K)[k]}$ have been computed for a sequence of tolerances, where K is an element on the accepted mesh, k. The minimal requirements for Theorems 3.1 and 3.2 use bounds on $\hat{c}^{-1}$ for the maximal error indicator, which is here approximately 8, because of (58). For Theorem 3.3 the minimal requirements use bounds on $||\hat{\rho}\hat{C}||_{L^{1/2}}$, which is close to $||\hat{\rho}||_{L^{1/2}}$ computed on the accepted mesh using $\mathrm{TOL} = 2^{-15}$; the maximal $\hat{C}$ may be significantly larger as seen on the right plot.

[12] F. Christian and G. Santos, A posteriori estimators for nonlinear elliptic partial differential equations, J. Comput. Appl. Math., **103** (1999), 99–114.

[13] A. Cohen, W. Dahmen and R. DeVore, Adaptive wavelet methods for elliptic operator equations: convergence rates, Math. Comp., **70** (2001), no. 233, 25-75

[14] A. Dzougoutov, K.-S. Moon, A. Szepessy, E. von Schwerin and R. Tempone, Adaptive Monte Carlo Algorithms for Stopped Diffusion.

[15] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, SIAM J. Numer. Anal. **33** (1996), no. 3, 1106-1124.

[16] Edelsbrunner, H.; Grayson, D. R., Edgewise subdivision of a simplex. ACM Symposium on Computational Geometry (Miami, FL, 1999). Discrete Comput. Geom., **24** (2000), 707–719.

[17] K. Eriksson, D. Estep, P. Hansbo and C. Johnson, Introduction to adaptive methods for differential equations, Acta Numerica, (1995), 105-158.

[18] C. Johnson and A. Szepessy, Adaptive finite element methods for conservation laws based on a posteriori error estimates, Comm. Pure Appl. Math., **48** (1995), 199-234.

[19] L. Machiels, A. T. Patera, J. Peraire and Y. Maday, A general framework for finite element a posteriori error control: application to linear and nonlinear convection-dominated problems, Preprint (presentation at ICFD Conference on Numerical Methods for Fluid Dynamics, Oxford, England, March 31-April 3, 1998)

[20] Y. Maday, A. T. Patera and J. Peraire, A general formulation for a posteriori bounds for output functionals of partial differential equations; applications to the eigenvalue problem, C. R. Acad. Sci. Paris Sér. I Math., **328** (1999), 823–828.

[21] K.-S. Moon, *Convergence rates of adaptive algorithms for deterministic and stochastic differential equations*, (Licentiate thesis, ISBN 91-7283-196-0, Royal Institute of Technology, 2001) http://www.nada.kth.se/∼moon

[22] K.-S. Moon, A. Szepessy, E. von Schwerin and R. Tempone, Convergence Rates for an Adaptive Dual Weighted Residual Finite Element Algorithm, www.nada.kth.se/∼szepessy.

[23] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, Convergence rates for adaptive approximation of ordinary differential equations, Numer. Math. **96** (2003), 99–129.

[24] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, A variational principle for adaptive approximation of ordinary differential equations Numer. Math. **96** (2003), 131–152.
[25] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, Convergence rates for adaptive weak approximation of stochastic differential equations, Preprint 2002.
[26] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, Hyperbolic differential equations and adaptive numerics, in *Theory and numerics of differential equations* (Eds. J.F. Blowey, J.P. Coleman and A.W. Craig, Durham 2000, Springer Verlag, 2001)
[27] P. Morin, R. Nochetto and K.G. Siebert, Convergence of adaptive finite element methods. Revised reprint of "Data oscillation and convergence of adaptive FEM" [SIAM J. Numer. Anal. 38 (2000), no. 2, 466–488] SIAM Rev. 44 (2002), no. 4, 631–658 (electronic) (2003).
[28] R. Stevenson, *An optimal adaptive finite element method*, Preprint (2004), To appear in SIAM J. Numer. Anal.
[29] A. Szepessy, R. Tempone and G. E. Zouraris, Adaptive weak approximation of stochastic differential equations, Comm. Pure Appl. Math., **54** (2001), 1169-1214.
[30] D. Talay and L. Tubaro, *Expansion of the global error for numerical schemes solving stochastic differential equations*, Stochastic Anal. Appl., **8**, 483–509, 1990.

Department of Mathematics, University of Maryland, College Park MD 20742-4015, USA
E-mail address: moon@math.umd.edu

Institutionen för Numerisk Analys och Datalogi, Kungl. Tekniska Högskolan, S–100 44 Stockholm, Sweden.
E-mail address: schwerin@nada.kth.se

Matematiska Institutionen, Kungl. Tekniska Högskolan, S–100 44 Stockholm, Sweden.
E-mail address: szepessy@nada.kth.se

Institute for Computational and Engineering Sciences (ICES), The University of Texas at Austin, 1 University Station C0200, Austin, Texas 78712, USA.
E-mail address: rtempone@ices.utexas.edu

Paper IV

A Stochastic Phase-Field Model Computed From Coarse-Grained Molecular Dynamics

Erik von Schwerin
KTH – Computer Science and Communication
SE-100 44 Stockholm
SWEDEN

August 17, 2007

Abstract

Results are presented from numerical experiments aiming at the computation of stochastic phase-field models for phase transformations by coarse-graining molecular dynamics. The studied phase transformations occur between a solid crystal and a liquid. Nucleation and growth, sometimes dendritic, of crystal grains in a sub-cooled liquid is determined by diffusion and convection of heat, on the macroscopic level, and by interface effects, where the width of the solid–liquid interface is on an atomic length-scale. Phase-field methods are widely used in the study of the continuum level time evolution of the phase transformations; they introduce an order parameter to distinguish between the phases. The dynamics of the order parameter is modelled by an Allen–Cahn equation and coupled to an energy equation, where the latent heat at the phase transition enters as a source term. Stochastic fluctuations are sometimes added in the coupled system of partial differential equations to introduce nucleation and to get qualitatively correct behaviour of dendritic side-branching. In this report the possibility of computing some of the Allen–Cahn model functions from a micro-scale model is investigated. The microscopic model description of the material by stochastic, Smoluchowski, dynamics is considered given. A local average of contributions to the potential energy in the micro model is used to determine the local phase, and a stochastic phase-field model is computed by coarse-graining the molecular dynamics. Molecular dynamics simulations on a two phase system at the melting point are used to compute a double-well reaction term in the Allen–Cahn equation and a diffusion matrix describing the noise in the coarse-grained phase-field.

This work was supported by the Swedish Foundation for Strategic Research grant A3 02:123, "Mathematical theory and simulation tools for phase transformations is materials".

1 Introduction

Phase-field methods are widely used for modelling phase transformations in materials on the continuum level and exist in many different versions for different applications. In this

report the considered phase transformation occurs in a single component system with a solid and a liquid phase.

The phase-field model of solidification studied here is a coupled system of partial differential equations for the temperature, T, and a phase-field, ϕ, which is an order parameter used to distinguish between the solid and the liquid subdomains. Two different values, ϕ_s and ϕ_l, are equilibrium values of the phase-field in solid and liquid respectively. The phase-field varies continuously between the two values and the interface between solid and liquid, at a time t, is defined as a level surface of the phase-field; for example $\{x \in \mathbb{R}^d : \phi(x,t) = 0.5(\phi_s + \phi_l)\}$. From a computational point of view the implicit definition of the phases in the phase-field method, as in the level set method [8, 12], is an advantage over sharp interface methods, since it avoids the explicit tracking of the interface. A local change of the phase-field from ϕ_l to ϕ_s in a subdomain translates into solidification of that region with a corresponding release of latent heat and the reverse change from ϕ_s to ϕ_l means melting which requires energy. The release or absorption of latent heat is modelled as a continuous function of ϕ so that the energy released when a unit volume solidifies is $L(g(\phi_l) - g(\phi_s))$, where L is the latent heat and $g(\phi)$ is a model function, monotone with $g(\phi_s) = 0$, $g(\phi_l) = 1$, $g'(\phi_s) = 0$, and $g'(\phi_l) = 0$. Then the energy equation for a unit volume becomes a heat equation with a source term

$$\frac{\partial}{\partial t}\left(c_V T + L g(\phi)\right) = \nabla \cdot (\lambda \nabla T)\,,$$

where c_V is the heat capacity at constant volume and λ is the thermal conductivity. Here, and in the following, the usual notation for differentiation with respect to the spatial variables is applied, with ∇ and $\nabla\cdot$ denoting the gradient and the divergence respectively. The phase-field, and the related model function g, are exceptional in the energy equation in the sense that, while all the other quantities are standard physical quantities on the macroscopic level, the phase-field need not be associated with a measurable quantity. A phenomenological model of the phase change is given by the energy equation coupled to the Allen-Cahn equation

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (k_1 \nabla \phi) - k_2\Big(f'(\phi) + g'(\phi) k_3 (T_{\mathrm{M}} - T)\Big) \tag{1}$$

for the time evolution of the phase-field; here T_{M} denotes the melting point, k_1, k_2, and k_3, are positive model parameters (k_1 may be an anisotropic matrix introducing directional dependence on the growth of the solid), and the model function f is a double well potential with minima at ϕ_s and ϕ_l. Standard examples of the model functions are

$$f(\phi) = -\frac{1}{2}\phi^2 + \frac{1}{4}\phi^4, \qquad\qquad g(\phi) = \frac{15}{16}\left(\frac{1}{5}\phi^5 - \frac{2}{3}\phi^3 + \phi\right) + \frac{1}{2},$$

when $\phi_s = -1$ and $\phi_l = 1$. By construction of the model functions, the reaction term in the Allen-Cahn equation vanishes where $\phi = \phi_s$ or $\phi = \phi_l$ independently of the temperature. Since the diffusion term is zero for any constant function the two constant phase-fields $\phi \equiv \phi_s$ and $\phi \equiv \phi_l$ are stationary solutions to the Allen-Cahn equation for all temperatures. This means, for example, that nucleation of solid in a region of subcooled liquid can not occur in a phase-field modelled by the deterministic Allen-Cahn equation above. The effect of nucleation can be introduced in the model by adding a noise term in the Allen-Cahn equation, giving a stochastic partial differential equation. Simulation of dendrite growth in an subcooled liquid is another example where the deterministic system is inadequate; its

solutions fail to develop the side branches seen to form in real dendrites as the tips grow. Stochastic phase-field models where noise is added to either one, or both, of the Allen-Cahn equation and the energy equation are used to include the effect of side branching; see for example [2].

The present report contains the results from numerical experiments on a method presented and analysed in [14] and the rest of this introduction summarises the ideas from [14] needed here. That report takes the stochastic phase-field model

$$\frac{\partial}{\partial t}\left(c_V T + Lg(\phi)\right) = \nabla\cdot(\lambda\nabla T)\,, \tag{2a}$$

$$\frac{\partial \phi}{\partial t} = \nabla\cdot(k_1\nabla\phi) - k_2\Big(f'(\phi) + g'(\phi)k_3(T_{\mathrm{M}} - T)\Big) + \text{noise}, \tag{2b}$$

as its starting point and asks whether it is possible to obtain the model functions and parameters, $f(\phi)$, $g(\phi)$, k_1, k_2, k_3, and the noise, from computations on a microscale model. To answer this question the phase-field, ϕ, must be defined in terms of quantities computable on the microscale. The microscopic model used for this purpose is a molecular dynamics model of N particles in a microscopic domain D in $\mathbb{R}^3$ where the motion of the particles is given by the Smoluchowski dynamics; see for example [5]. Thus, with $X^t \in \mathbb{R}^{3N}$ denoting the positions of all particles in the system at the time t and $X_i^t \in \mathbb{R}^3$ the position of particle i, the dynamics are given by the Itô stochastic differential equations

$$dX_i^t = -\nabla_{X_i} U(X^t)\,dt + \sqrt{2k_{\mathrm{B}}T}\,dW_i^t, \qquad i = 1, 2, \ldots, N, \tag{3}$$

where U is the total potential energy of the system, ∇_{X_i} denotes the gradient with respect to the position of particle i, k_{B} is the Boltzmann constant, and $W_i = (W_{i,1}, W_{i,2}, W_{i,3})^{\mathrm{T}}$ are independent three dimensional Brownian motions, with independent components. The macroscopic temperature, T, is a constant input parameter in the microscopic model. We may identify the latent heat, in the macroscopic model, with the difference in total potential energy per unit volume of the liquid and the solid at the melting point, in the microscopic model. The idea is then to let the local contributions to the total potential energy define the phase variable. Since the potential energy decreases with the temperature even in a single phase system the equilibrium values of such a phase-field, m, unlike those of ϕ, depend on the temperature; see Figure 1. Assuming that in pure solid or pure liquid the phase-field,

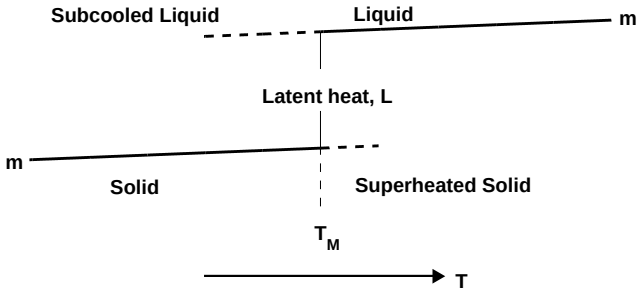


Figure 1: Schematic picture of $m(T)$ for a pure liquid (top curve) and a pure solid (bottom curve) and the latent heat as the jump in m at a phase transition.

m, varies slowly, compared to the latent heat release, with the temperature close to the

melting point, the energy equation becomes

$$\frac{\partial}{\partial t}\left(c_V T + m\right) = \nabla \cdot \left(\lambda \nabla T\right),$$

where c_V and λ are approximately the same as in (2a) for $T \approx T_{\mathrm{M}}$.

For a model where the total potential energy of the system can be naturally split into a sum of contributions arising from the interaction of individual atoms with their environment,

$$U(X) = \sum_{i=1}^{N} m_i(X), \tag{4}$$

phase-fields can be introduced on the micro level by localised averages of these contributions; a given configuration X defines a phase-field $m(\,\cdot\,; X) : D \to \mathbb{R}$ through

$$m(x; X) = \sum_{i=1}^{N} m_i(X)\eta(x - X_i), \tag{5}$$

where the choice of mollifier, η, determines the spatial smoothness of the phase-field. If, for example, the potential energy is defined entirely by pairwise interactions

$$U(X) = \frac{1}{2}\sum_{i=1}^{N}\sum_{k\neq i,k=1}^{N} \Phi(X_i - X_k),$$

as is common in simple molecular dynamics models, it is natural to let

$$m_i(X) = \frac{1}{2}\sum_{k\neq i,k=1}^{N} \Phi(X_i - X_k)$$

be particle i's contribution to the total potential energy.

With the definition (5) of the potential energy phase-field, m, and with the microscopic system defined by (3) and (4), Itô's formula gives a stochastic differential equation

$$dm(x; X^t) = \alpha(x; X^t)\, dt + \sum_{j=1}^{N}\sum_{k=1}^{3} \beta_{j,k}(x; X^t)\, dW_{j,k}^t, \tag{6}$$

for m evaluated in a point $x \in D$. The drift, $\alpha(x;\cdot)$, and the diffusions, $\beta_{j,k}(x;\cdot)$, are explicitly known functions expressed in terms of the m_i:s, the mollifier, η, and their derivatives up to second order. While m by definition is a continuous field it is still an atomic scale quantity since it is defined in terms the particle positions X^t. A macroscopic phase-field, similar to ϕ in (2), must lose both the dependence on the particle positions, X^t, and the explicit dependence on the microscale space variable x. To achieve this, a coarse-grained approximation $m_{\mathrm{cg}}(x)$ of $m(x)$ is introduced as a solution of a stochastic differential equation

$$dm_{\mathrm{cg}}^t(x) = a(m_{\mathrm{cg}}^t)(x)\, dt + \sum_{j=1}^{M} b_j(m_{\mathrm{cg}}^t)(x)\, d\widetilde{W}_j^t, \tag{7}$$

where the independent Wiener processes $\widetilde{W}_j^t$, $j = 1, 2, \ldots, M \ll N$, also are independent of the Wiener processes W_i in the micro model. Here the drift and diffusion coefficient functions, $a(m_{\text{cg}}^t)$ and $b_j(m_{\text{cg}}^t)$, may depend on more information about the coarse-grained phase-field than just the point value; compare the stochastic Allen-Cahn equation (2b), where the diffusion term in the drift contains second derivatives of the phase-field.

The choice of the coarse-grained drift and diffusion functions proceeds in two steps: first, finding a general form the coarse-grained equation where the drift and diffusion coefficient functions, defined as time averaged expected values of the microscopic drift and diffusions over simulation paths, still depend on the micro scale space variable, x; second, expressing the x dependent coarse-grained drift and diffusion coefficients by drift and diffusion functions depending only on the phase-field m_{cg}, using that m_{cg} is a smooth monotone function of x in the interface.

In the first step, a coarse-grained stochastic differential equation

$$dm_{\text{cg}}^t(x) = \overline{a}(x)\, dt + \sum_{j=1}^{M} \overline{b}_j(x)\, d\widetilde{W}_j^t,$$

is introduced by defining the drift

$$\overline{a}(x) = \frac{1}{\mathcal{T}} \mathrm{E}\left[\int_0^{\mathcal{T}} \alpha(x; X^t)\, dt \;\middle|\; X^0 = X_0 \right], \qquad x \in D, \tag{8a}$$

and choosing a diffusion matrix that fulfil

$$\sum_{j=1}^{M} \overline{b}_j(x)\overline{b}_j(x') = \frac{1}{\mathcal{T}} \mathrm{E}\left[\int_0^{\mathcal{T}} \sum_{j=1}^{N} \sum_{k=1}^{3} \beta_{j,k}(x; X^t)\beta_{j,k}(x'; X^t)\, dt \;\middle|\; X^0 = X_0 \right], \quad x, x' \in D, \tag{8b}$$

for some fixed, deterministic, initial conditions $X^0 = X_0$. The initial condition for the coarse-grained phase-field is $m_{\text{cg}}^0 = m(\cdot; X_0)$. This particular coarse-graining is motivated by the argument that the coarse-grained model will be used to compute properties on the form $\mathrm{E}\left[y(m(\cdot; X^{\mathcal{T}}))\right]$, where $y : D \to \mathbb{R}$ is a smooth function and $\mathcal{T} > 0$ is a fixed final time. The optimal coarse-grained model is the one that minimises the error in the expected value; using the conditional expected values $\overline{u}(\mu, t) = \mathrm{E}[y(m_{\text{cg}}^{\mathcal{T}}) \,|\, m_{\text{cg}}^t = \mu]$, this error can be expressed as

$$\begin{aligned}
&\mathrm{E}\left[y(m(\cdot; X^{\mathcal{T}}))\right] - \mathrm{E}\left[y(m_{\text{cg}}^{\mathcal{T}})\right] \\
&= \mathrm{E}\left[\int_0^{\mathcal{T}} \left\langle \overline{u}'(m(\cdot; X^t), t)\ ,\ \alpha(\cdot; X^t) - \overline{a}(\cdot) \right\rangle_{L^2(D)} dt \right. \\
&\left. + \frac{1}{2} \int_0^{\mathcal{T}} \left\langle \overline{u}''(m(\cdot; X^t), t)\ ,\ \sum_{j=1}^{N} \sum_{k=1}^{3} (\beta_{j,k} \otimes \beta_{j,k})(\cdot, \cdot; X^t) - \sum_{j=1}^{M} (\overline{b}_j \otimes \overline{b}_j)(\cdot, \cdot) \right\rangle_{L^2(D \times D)} dt \right],
\end{aligned}$$

where $\otimes$ denotes the tensor product $(\overline{b}_j \otimes \overline{b}_j)(x, x') = \overline{b}_j(x)\overline{b}_j(x')$, and $\overline{u}'$ and $\overline{u}''$ denote the first and second variations of $\overline{u}(\mu, t)$ with respect to μ. Assuming that $\overline{u}'$ can be expanded in powers of $\alpha - \overline{a}$, the choice (8a) cancels the leading term in the error associated with $\overline{u}'$. Similarly, (8b) corresponds to cancelling the dominating term in the expansion of $\overline{u}''$.

In a practical computation the functions α and β_j can only be evaluated in a discrete set of points $D_K = \{x^1, \ldots, x^K\} \subset D$. The right hand sides in (8a) and (8b) become a vector and a symmetric positive semidefinite K-by-K matrix, respectively. Hence $\overline{a}(x)$ becomes a vector of tabulated values for $x \in D_K$. It is natural to have one Wiener process per point x^k in the spatial discretisation, so that $K = M$. The corresponding K tabulated individual diffusion coefficient functions, $\overline{b}_j$, will be obtained by a square root factorisation of the computed matrix, by means of an eigenvector expansion; this choice of factorisation preserves the connection between the evaluation point x_k and the elements k in $\overline{b}_j$ and produces spatially localised functions, consistent with the association of individual Wiener processes and points in D_K.

In the second step, the initial configuration, X_0, in (8) is chosen so that the microscopic domain D includes a solid–liquid interface in equilibrium. Since the interface is stationary no phase transformation occurs in the simulation, and consequently the part of the reaction term in the Allen-Cahn equation (2b) relating the speed of the phase change to the deviation from the melting point, $k_2 k_3 g'(\phi)(T_\mathrm{M} - T)$, can not be obtained; the simulation must be performed at the melting point, T_M, under the given conditions. The simulation of a travelling front, off the equilibrium temperature, requires more advanced micro model simulations than the ones considered here.

The interface is assumed to be locally planar on the microscopic scale and the spatially averaged properties are expected to vary much more slowly in the directions parallel to the interface than in the direction normal to the interface. Label the direction normal to the interface as direction x_1 and let x_2, x_3 be orthogonal directions in the plane of the interface. Then the mollifier, η, in (5) can be chosen to make the averages much more localised in the x_1 direction than in the x_2 and x_3 directions. In the microscopic domain, D, the averages in the x_2 and x_3 directions are chosen to be uniform averages over the entire domain, so that the phase-fields, m and m_cg, and the drift and diffusion functions, α, $\beta_{j,k}$, $\overline{a}$, and $\overline{b}_j$, become functions of one space variable, x_1. Hence the evaluation points in D_K are only distinguished by their x_1 coordinates. As mentioned above, the drift coefficient, α, depends on the derivatives up to second order of, η, and the potential energy contributions m_i. After averaging out the x_2 and x_3 dependence, it can be written as

$$\alpha(x_1; X^t) = k_\mathrm{B} T \frac{\partial^2}{\partial x_1^2} m(x_1; X^t) + \frac{\partial}{\partial x_1} A_1(x_1; X^t) + A_0(x_1; X^t),$$

for some functions A_1 and A_0. Keeping this form in the averaging, the coarse-grained drift coefficient in (8a) can be written

$$\overline{a}(x_1) = k_\mathrm{B} T \frac{\partial^2}{\partial x_1^2} m_\mathrm{av}(x_1) + \frac{\partial}{\partial x_1} \overline{a}_1(x_1) + \overline{a}_0(x_1),$$

where the second order derivative of the averaged phase-field,

$$m_\mathrm{av}(x_1) = \frac{1}{\mathcal{T}} \mathrm{E}\left[\int_0^{\mathcal{T}} m(x_1; X^t)\, dt \right], \tag{9}$$

corresponds to the diffusion term in (2b). Assuming that the averaged phase-field m_av is a monotone function of x_1 in the interface, the explicit dependence on the spatial variable can be eliminated by inverting m_av and defining

$$a(m_\mathrm{cg}) = \overline{a}(m_\mathrm{av}^{-1}(m_\mathrm{cg})), \qquad b_j(m_\mathrm{cg}) = \overline{b}_j(m_\mathrm{av}^{-1}(m_\mathrm{cg})), \tag{10}$$

which give drift and diffusion coefficients on the form (7).

The present study is a practical test of the method described above. In particular the aims are to verify that Smoluchowski dynamics can be used in practise, in the sense that the coarse grained drift and diffusion coefficient functions can be determined together with the phase-field model potential, f, and that they seem reasonable. For this purpose simulations are performed at just one temperature and density (at the melting point) and with just two values of the angle of the stationary interface with respect to the crystal structure in the solid. An actual determination of the model functions in the phase field model would require many more simulations with varying parameters.

2 Computational Methods

The numerical computations consist of molecular dynamics computations, giving the microscopic description of the two-phase system, and the extraction of model functions for a coarse grained stochastic differential equation model.

2.1 Molecular Dynamics Models and Simulation

Two mathematical models of the material are used; both are one component molecular dynamics models where the interaction between particles is determined by a pair potential of the exponential-6 (Exp-6) type. The coarse graining is based on a stochastic model where the particle trajectories on the diffusion time scale are given by the Smoluchowski dynamics (3). The computations with this model are performed under constant volume at the melting point where a liquid and a solid phase coexist in the computational domain. The melting point is determined using constant pressure simulations of the deterministic molecular dynamics model where the particle trajectories are determined by Newton's second law with forces given the by gradients of the model potential. Both models and the corresponding simulations are described below, after a description of the potential common to the models.

2.1.1 Pair Potential Defining the Total Potential Energy

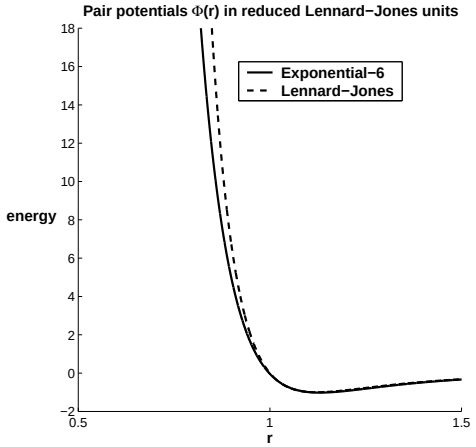
The microscopic system consists of N identical particles at positions $X = (X_1, \ldots, X_N)$ in three dimensions. The total potential energy, U, of the system is determined by the particle positions through

$$U(X) = \frac{1}{2}\sum_{i=1}^{N}\sum_{k\neq i,k=1}^{N} \Phi(X_i - X_k), \tag{11}$$

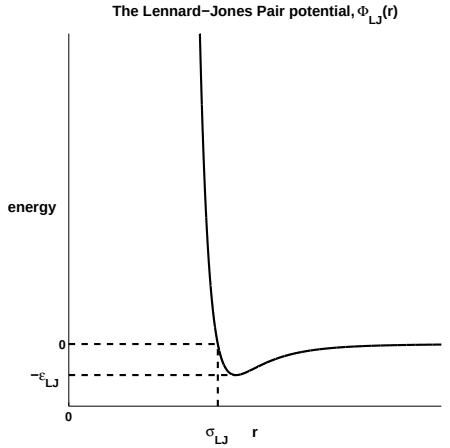
using pairwise interactions only. The pair potential is the spherically symmetric Exp-6 potential

$$\Phi(r) = A\exp(-Br) - \frac{C}{r^6}, \tag{12}$$

with r denoting the distance between two particles, and A, B, and C being positive model parameters. The Exp-6 potential, like the similar Lennard-Jones pair potential, $\Phi_{\mathrm{LJ}}(r) = 4\epsilon_{\mathrm{LJ}}\left((\sigma_{\mathrm{LJ}}/r)^{12} - (\sigma_{\mathrm{LJ}}/r)^{6}\right)$, is a short range interaction that can be used to model condensed noble gases. With the parameters used here, obtained from [11], the Exp-6 potential models Argon at high pressures. At pressures around 2 GPa, where the solid-liquid phase transition will be simulated, the Exp-6 potential with its slightly softer repulsive part describes the equation of state of Argon better than the Lennard–Jones potential does; see [11, 15]. The shapes of the two pair potentials around the global minimum of the Lennard–Jones potential can be compared in Figure 2(a); the typical inter atomic distances between nearest neighbours in both the simulated solid and liquid will be close to 1. Note that, while the Lennard–Jones pair potential tends to infinity as the interatomic



(a) Pair Potentials for Argon



(b) General form of the Lennard-Jones potential

Figure 2: (a): The Exp-6 pair potential is similar to the Lennard-Jones pair potential near the minimum, but the repulsion is slightly weaker in the Exp-6. The radius and the energy are measured in reduced Lennard-Jones units, where the Lennard-Jones parameters are $\epsilon_{\mathrm{LJ}} = k_{\mathrm{B}}120\,\mathrm{K}$ and $\sigma_{\mathrm{LJ}} = 3.405\,\text{Å}$.
(b): The parameter σ_{LJ} is the radius where the Lennard-Jones potential is 0, which is equal to the potential at infinite separation, and the parameter ϵ_{LJ} is the depth of potential minimum.

distance tends to zero, the Exp-6 pair potential, as stated in (12), reaches a global maximum before turning down and approaching minus infinity in the limit. This clearly illustrates that the model based on the Exp-6 potential breaks down if two atoms come too close, but neither one of the pair potentials is designed to describe interactions of particles much closer than the typical nearest neighbour separation.

For short range potentials, like the Exp-6 and the Lennard-Jones potentials, the potential (and its derivative) decay sufficiently fast for the combined effect on the total potential energy (and the interatomic forces) of all atom pairs separated more than a certain distance to be negligible compared to the effect of the pairs separated less than the same distance. To take advantage of this in computations a cut-off radius is introduced and all interactions between particles separated by a distance larger than the cut-off are neglected; instead of summing over all $k \neq i$ in the inner sum in (11) the sum is only taken over particles in a

spherical neighbourhood of particle i.

All the physical quantities in this report are given in the reduced Lennard-Jones units. Thus length is measured in units of σ_{LJ}, energy in units of ϵ_{LJ}, and time in units of $\sqrt{m\sigma_{\mathrm{LJ}}^2/\epsilon_{\mathrm{LJ}}}$, where m is the mass of one atom. (The time unit is the inverse of the characteristic frequency.) A list of the dimensionless units in the Argon model as well as the parameters in the Exp-6 potential can be found in Table 2.1.1. At the temperatures and pressures considered here, the stable phase of the Exp-6 potential is either the Face Centered Cubic (FCC) lattice or a liquid phase.

Quantity	Unit
Energy	$1.6568 \cdot 10^{-21}$ J
Time	$2.1557 \cdot 10^{-12}$ s
Mass	$6.6412 \cdot 10^{-26}$ kg
Length	$3.405 \cdot 10^{-10}$ m
Temperature	120 K
Pressure	$4.1968 \cdot 10^{7}$ Pa

Constant	Value
k_{B}	1

Parameter	Value
A	$3.84661 \cdot 10^{5}$
B	11.4974
C	3.9445

Table 1: Atomic units and corresponding values of physical constants and parameters in the Exp-6 model (12). Non dimensional molecular dynamics equations are obtained after normalising with the atom mass, m, and the Lennard-Jones parameters, σ_{LJ} and ϵ_{LJ}; in this Argon model $m = 6.6412 \cdot 10^{-26}$ kg (or 39.948 atomic mass units), $\sigma_{\mathrm{LJ}} = 3.405$ Å, and $\epsilon_{\mathrm{LJ}}/k_{\mathrm{B}} = 120$ K, where k_{B} is the Boltzmann constant.

2.1.2 Newtonian System Simulated at Constant Pressure

The purpose here is to approximately determine the melting point at a high fixed pressure, to be able to set up and simulate stationary (FCC-liquid) two-phase systems later. Determination of the melting point follows the two-phase method described by Belonoshko and co-authors in [1].

The mathematical model is a classical system of N identical particles where the positions, $X^t = (X_1^t, \ldots, X_N^t)$, and the velocities, $v^t = (v_1^t, \ldots, v_N^t)$, evolve in time according to Newton's equations

$$\frac{dX^t}{dt} = v^t, \tag{13a}$$

$$\frac{dv^t}{dt} = -\nabla_X U(X^t), \tag{13b}$$

where the total potential energy of the system is given by (11)-(12) using the parameter values in Table 2.1.1. Here ∇_X denotes the gradient with respect to the particle positions. The force acting on particle i is $-\nabla_{X_i} U(X^t)$ and, since all particles have unit mass in the non-dimensional units, the acceleration is equal to the force. Particle positions are restricted to a finite computational box with periodic boundary conditions, corresponding to an infinite system where the same configuration of particles is repeated periodically in all

three directions; a particle leaving the computational cell on one side enters the cell again from the opposite side and particles interact with periodic images of particles in the cell.

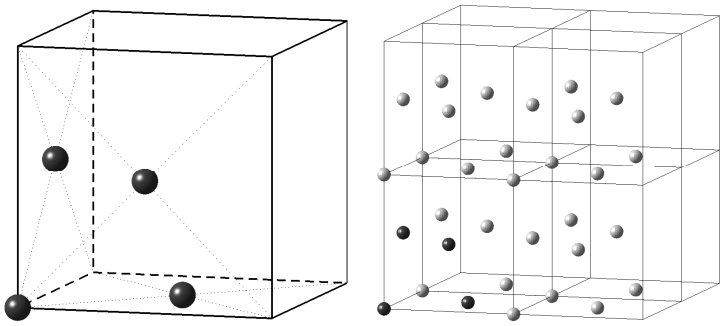
For a fixed volume of the computational cell the equations (13) will preserve the total energy, E, (the sum of potential and kinetic energy) of the system as well as the number of particles. It will approximately sample the (N, V, E) ensemble. In the determination of the melting point the simulations are instead performed in an approximation of the (N, T, P) ensemble, using a constant number of particles, N, a constant temperature, T, and a constant pressure, P. This must allow for the volume of the computational cell to change during the simulation. There must also be mechanisms for keeping the temperature and the pressure constant, thus modifying (13) so that the total energy varies.

Numerical computations of the (N,T,P) molecular dynamic simulations were performed using Keith Refson's publicly available software package Moldy, [9]. Constant temperature was enforced using the Nosé-Hoover thermostat, where the equations of motions (13) are modified, and extended, to include an additional degree of freedom modelling a thermal reservoir. The fictitious inertia associated with the thermal reservoir was 100 $\mathrm{kJ\,mol^{-1}\,ps^2}$, corresponding to 21.57 in the dimensionless equation. The pressure was kept constant using the Parinello-Rahman equation, controlling the dynamics of the vectors (three edges) that define the computational cell. The fictitious mass parameter in the Parinello-Rahman equation was 300 amu corresponding to $1.20 \cdot 10^4$ in the reduced Lennard–Jones units. A short description of the Nosé-Hoover thermostat and the Parinello-Rahman equation, with references to papers with theoretical foundations of the methods, can be found in the manual [10].

The time stepping method in Moldy is a modification of Beeman's algorithm using predictor-corrector iterations in the computation of the velocities; see [10] for details. The simulations described here used the constant time step $4.639 \cdot 10^{-5}$ and the potential cut-off 2.937.

In the two-phase method for determination of the melting point the molecular dynamics simulation starts from an initial configuration that is part solid and part liquid. As the (N, T, P) simulation proceeds the whole liquid part will solidify, if $T < T_{\mathrm{M}}$ for the given pressure, or the solid will melt, if $T > T_{\mathrm{M}}$, resulting in a single phase system. Starting from a coarse estimate of the temperature interval containing the melting temperature, that interval can be narrowed down by running simulations at temperatures in the interval and noting whether they equilibrate to an all solid or an all liquid system. The validity of this two-phase approach has been verified in [1] for determining, among other things, the melting point of a molecular dynamics model of Xenon, similar to the Argon model used here.

The initial configuration in a two-phase simulation was composed of pre-simulated solid and liquid configurations. The solid part was prepared by taking a perfect FCC configuration and performing a short molecular dynamics run at the temperature and pressure of the intended two-phase simulation to adapt the size of the computational cell. Initially the sides of the computational cell were aligned with the sides of the unit cube in the perfect FCC lattice; see Figure 3. While in general the dynamics of the cell edges in the Parinello-Rahman equations allow the cell to take the shape of any parallelepiped, here the dynamics were



(a) The FCC unit cube (b) Eight FCC unit cubes

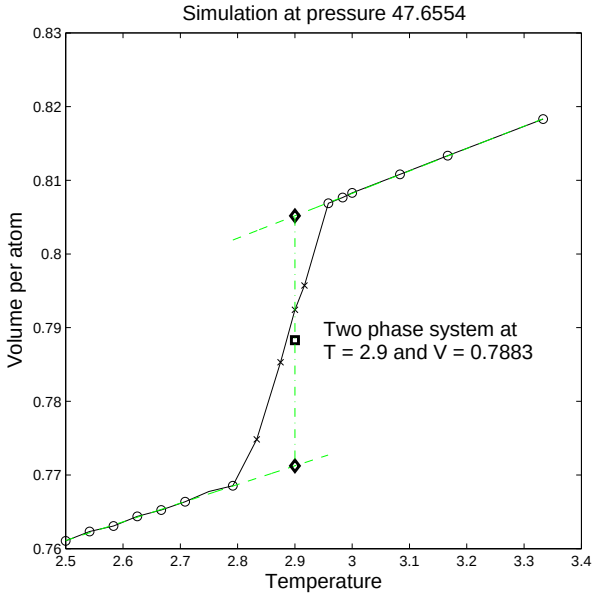
Figure 3: A perfect FCC lattice consists of FCC unit cubes, (a), stacked next to each other in three dimensions, (b). With one atom in the $(0, 0, 0)$ corner of the unit cube the three other atoms are placed at the centres of the cubic faces intersecting in $(0, 0, 0)$.

restricted to only allow rescaling, without rotation, of the three edges and thus keeping the rectangular box shape of the cell. The preparation of the liquid part started from the configuration of the already prepared FCC-solid and a run was performed at a temperature well over the estimated melting point, where the sample would melt quickly; after equilibrating at the higher temperature the sample was quenched to the temperature of the two-phase simulation. Only one side of the computational cell was allowed to change while preparing the liquid part and thus the orthogonal cross section of the simulation cell was preserved from the FCC simulation. The solid and liquid parts were joined in the two-phase initial configuration by placing them next to each other, letting the cell faces of identical shape face each other. The general appearance is similar to the configurations shown in Figure 5 on page 16, even though those configurations belong to the constant volume Smoluchowski simulations where the set up procedure is slightly modified. Periodic boundary conditions were still applied in all directions, so that each part (solid or liquid) corresponded to a semi-infinite slab surrounded on two sides by the other phase with the effect of simulating a periodic, sandwiched, material. Voids of thickness of approximately one nearest neighbour separation were introduced in both solid–liquid interfaces to make sure that no pair of particles ended up to close in the initial configuration. Since the two-phase simulations were performed at constant pressure, the voids would fill in the beginning of the run as the length of the computational cell decreased.

In the two-phase simulations the lengths of all three vectors defining the cell edges were allowed to change. Starting from an initial two-phase configuration the molecular dynamics simulation was run until the system was considered equilibrated. After equilibration the computational cell was filled with either the solid or the liquid phase. The density of the FCC solid is higher than that of the liquid phase. If the phase change was solidification of the liquid, then the volume of the computational cell would decrease during the equilibration stage before assuming an approximately constant value; if the solid was melting, the total volume would grow during equilibration. The density of the stable phase at the given pressure and temperature was obtained by time averages of the simulation after equilibration.

When the volume per particle is shown as a function of the temperature, at constant

pressure, it will display a sharp change at the melting point; see Figure 4(a) on page 12. The procedure will obtain an interval around the melting point and the accuracy can be improved by performing simulations at more temperatures to shorten the interval of uncertainty. However, the equilibration requires longer time when close to the melting point and the cost for refining the approximation grows, not only because the number of simulations grows, but more importantly because every single simulation takes longer to perform.



(a) Simulation data

	V_a	ρ
Liquid	0.8060	1.241
FCC	0.7714	1.296
Combined	0.7883	1.269

(b) Extrapolated data at $T = 2.9$

Figure 4: (a) Volume per atom as a function of temperature at a the pressure 47.6554 (or 2.0 GPa) in (N, P, T) simulations. Data points from simulations that are considered equilibrated are marked with $\circ$ and those from simulations that are not equilibrated are marked with $\times$. The two regions of equilibrated values where the volume per atom varies approximately linearly correspond to solid (FCC), at lower temperatures, and liquid, at higher temperature, respectively. The melting point at the given pressure is somewhere in between; the approximate value $T = 2.9$ is used below and in the constant volume simulations.
(b) The volume per atom of solid and liquid have been extrapolated to $T = 2.9$ by least square fits of straight lines to the simulation data and the corresponding number densities, ρ, have been computed. If $T = 2.9$ is sufficiently close to the melting point at this pressure, then the two phases will coexist in *constant volume*, (N, V, T), simulations provided that the total density is between the estimated densities of pure solid and pure liquid. The ratio of the volumes of the solid and the liquid part is determined by the total density of the combined system. The tabulated value of the density for a combined system gives approximately equal volumes of both parts at a pressure close to the one in the constant pressure simulations.

The main purpose here is to investigate the possibility of obtaining the model functions in a coarse grained phase-field model from (N, V, T) Smoluchowski dynamics simulations,

as described next. Therefor the accuracy in the determination of the melting point at the given pressure is critical only to the extent that it must be possible to perform the constant volume simulations at this temperature; that is, it must be possible to perform simulations on a two-phase system with stable interfaces between the solid and liquid parts. If the purpose were to perform computations at the melting point at this very pressure, then more computational effort would have to be spent on the accuracy of the melting point and the corresponding densities.

The numerical simulations were performed with $N = 8000$ particles; the initial solid configuration consisted of 4000 particles, corresponding to $10\times10\times10$ FCC unit cells with four atoms each, and the liquid had the same number of particles. From simulations at the pressure 47.7 in the reduced Lennard-Jones units (corresponding to 2.0 GPa) an approximate value of 2.9 for the melting point was obtained together with number densities for the liquid and solid extrapolated to this temperature; see Figure 4 on page 12. Fixing the temperature and the number density N/V, only one degree of freedom remains in the triple (N, V, T), allowing the system size to vary.

2.1.3 Smoluchowski System Simulated at Constant Volume

The constant volume and temperature Smoluchowski dynamics two-phase simulations described here were used to compute the functions (10) defining the coarse-grained phase-field dynamics (7), as described in the introduction. This meant computing time averaged quantities like the time averaged potential energy phase-field (9) and the corresponding coarse-grained drift and diffusion coefficient functions (8).

The mathematical model is that of N particles whose positions X^t follow the Smoluchowski dynamics

$$dX^t = -\nabla_X U(X^t)\, dt + \sqrt{2k_{\mathrm{B}}T}\, dW^t, \tag{14}$$

introduced on page 3. There are no velocities in the Smoluchowski dynamics. Instead the positions of all particles in the system give a complete description of the system at a particular time. Such a description, X^t, will be refered to as a configuration of the system. The particles are contained in a computational cell, shaped like a rectangular box, of fixed dimensions and the boundary conditions are periodic in all directions. Hence the volume, V, and the number of particles, N, are fixed. Without velocities there is no kinetic energy, but the temperature, T, enters directly in the dynamics. The temperature parameter is held fixed, which can be viewed as a kind of thermostat built into the dynamics.

Since the volume of the computational cell is constant, unlike in the (N, T, P) simulations above, the overall density of the system remains constant over time, which allows for stationary two-phase configurations where part of the domain is solid and part is liquid.

The numerical simulations The discrete time approximations $\bar{X}^n$ of X^{t_n}, were computed using the explicit Euler-Maruyama scheme

$$\bar{X}^n = \bar{X}^{n-1} - \nabla_X U(\bar{X}^{n-1})\, \Delta t^n + \sqrt{2k_{\mathrm{B}}T}\, \Delta W^n, \tag{15}$$

where $\Delta t^n = t^n - t^{n-1}$ is a time increment and $\Delta W^n = W(t^n) - W(t^{n-1})$ is an increment in the $3N$-dimensional Wiener process. Each run was performed using constant time step size, $\Delta t^n \equiv \Delta t$, but the time step could change between different runs depending on the purpose; in the equilibration phase the typical step size was $\Delta t = 10^{-4}$, but in the production phase the step size had to be taken smaller, as discussed later.

The computation of $\nabla_X U(\bar{X}^{n-1})$ in every time step is potentially an $\mathcal{O}(N^2)$ operation since the potential is defined by pairwise interactions. The computations described here used the potential cut-off radius 3.0, which meant that each particle only interacted directly with a relatively small number of neighbours (independent of N since the density was approximately constant). To avoid the $\mathcal{O}(N^2)$ task of computing all pairwise distances in each time step, the computational cell is divided into smaller sub cells, where the size is defined in terms of the cut-off radius so that two particles only can interact if they are in the same sub cell or in two neighbouring sub cells; information about particles migrating between sub cells is exchanged in each time step. The computations use a two dimensional grid of sub cells, where the particle positions within each sub cell are sorted with respect to the third coordinate dimension in every time step. When the particles are sorted the sweep over all particles in a sub cell can be efficiently implemented and the sorting procedure is not too expensive since the particles do not move far in one time step. A more thorough description of this algorithm can be found in [13]. The actual code used here is a modification of a parallelised code for Newtonian molecular dynamics obtained from Måns Elenius in Dzugutov's group[4]; the main modifications when adapting to Smoluchowski dynamics is the removal of velocities from the system and the introduction of a pseudo random number generator for the Brownian increments, ΔW^n.

With the cut-off radius 3.0 used in the computation and the model parameters in Table 2.1.1 on page 9, the Exp-6 pair potential and its derivatives are small at the cut-off radius. Still the potential will be discontinuous at the cut-off, unless it is slightly modified. A small linear term is added to make the potential continuously differentiable at the cut-off radius. In the practical computations, both the pair potential and the derivatives were obtained by linear interpolation from tabulated values.

The random number generator for normally distributed random variables was the Ziggurat method, described in [6], in a Fortran 90 implementation by Alan Miller, accessible from Netlib [7]. The underlying 32-bit integer pseudo random number generator is the 3-shift register SHR3. Since the purpose of the simulations only is to investigate if the coarse-graining procedure gives reasonable results just one pseudo random number generator was used, while several different random number generators ought to be used in a practical application. The generator was initialised with different seeds on different processors in the parallel computations, but it does not have distinct cycles simulating independent random variables. The hope is that the nature of the molecular dynamics simulations is enough to avoid the danger of correlated random numbers on the different processors, but this could be tested by comparing with other pseudo random generators that actually simulate independent random variables on different processors.

The two-phase systems for the Smoluchowski dynamics simulations were set up to obtain a two-phase system at temperature $T = 2.90$ with approximately equal volumes of solid and liquid and with stationary interfaces. To achieve this two equal volumes of FCC-

solid and liquid were pre-simulated with the densities tabulated in Figure 4, on page 12. The preparation of the initial configurations for the Smoluchowski dynamics two-phase simulations was similar to the procedure described above, but some adjustments must be made because of the constant volume restriction. The shape of the computational cell used when generating the solid part was chosen to match the periodic structure of the FCC lattice at the tabulated density for the FCC part. A short equilibration run, at $T = 2.90$, starting from a perfect FCC lattice at this density gave the initial solid configuration. The computational cell for the initial liquid part was chosen to be the same as the one in FCC simulation and the initial configuration when pre-simulating the liquid part was obtained from the FCC configuration by distributing vacancies to get the correct density in the liquid. In a simulation of (15) using a temperature, T, above the melting point, T_{M}, the sample was melted and equilibrated. Afterwards the liquid was cooled to desired temperature using a subsequent simulation with $T = T_{\mathrm{M}}$.

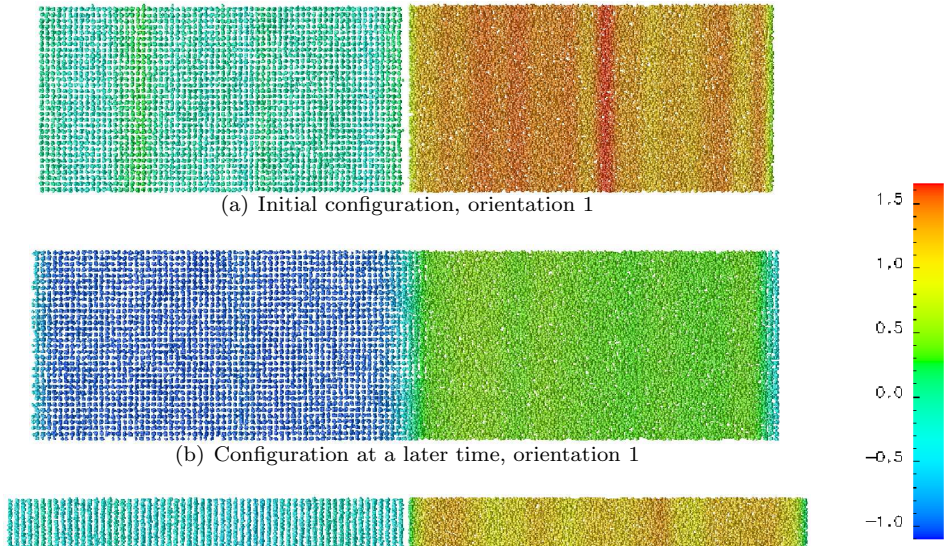
Since no pair of atoms can be too close in the initial configuration, gaps had to be introduced between the solid and liquid parts, but the voids could not be introduced as additional volumes in the computational cell; the individual parts were equilibrated at (N, V, T) corresponding to the expected densities for solid and liquid in the combined system, so increasing the total volume would reduce the overall density, resulting in partial or total melting of the solid part. To make room for the voids both the solid and the liquid parts were compressed slightly in the direction normal to the solid–liquid interfaces, before inserting them in their respective volumes in the computational cell for the two-phase simulation. Initial configurations obtained by this procedure are shown as configurations (a) and (c) in Figure 5, on page 16. The orientation of the solid–liquid interfaces with respect to the FCC lattice differ between the two initial configurations shown, and these orientations with the corresponding numerical simulations will be labelled Orientation 1 (O1) and Orientation 2 (O2) in the following. The shaded plane in Figure 6(b) shows the orientation of the interface in O1 and the shaded plane in Figure 6(c) shows the orientation in O2.

Even though the compression in one direction was small, it introduced an artificial internal stress in the system. The higher value of the phase-field in the subfigures (a) and (c) in Figure 5 compared to the corresponding regions in the subfigures (b) and (d) is an effect of the compression. In the initial phase of the equilibration of the two-phase system, the compressed parts expand to fill the voids. The phase-fields in the interiors of the solid and liquid parts in subfigures (b) and (d) have reached the levels seen in the corresponding single phase systems, which shows at least that the local potential energy contributions had returned to normal before the production runs started.

As a test of the two-phase configuration serving as initial data in the production run, the radial distribution functions in the interior of the two phases were computed. The radial distribution function, $g(r)$, is useful for identifying the phase of a single-phase system. For a single component system $g(r)$, where $r \in \mathbb{R}^+$, is implicitly defined by the condition that the average number of atoms in a spherical shell between the radii r_1 and r_2 from the centre of any atom is

$$\rho \int_{r_1}^{r_2} g(r) 4\pi r^2 \, dr,$$

where ρ is the global particle density. In other words, the radial distribution function is the average particle density, as a function of the separation r, normalised by overall density. Figure 7, on page 18, shows good agreement for simulation O2 between $g(r)$ corresponding



(a) Initial configuration, orientation 1

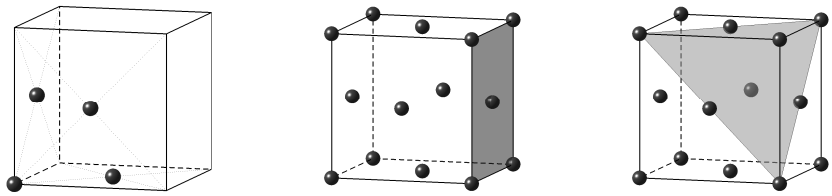
(b) Configuration at a later time, orientation 1

(c) Initial configuration, orientation 2

(d) Configuration at a later time, orientation 2

Figure 5: Snapshots of the process of setting up initial configurations for the two-phase simulations O1 and O2. The left part is solid (FCC) and the right part liquid. In the initial configurations, (a) and (c), the individual parts have been equilibrated at T_{melt} (for the combined system), and slightly compressed in one direction (to allow for two gaps). Subfigures (b) and (d) show configurations at later times when the parts have expanded to fill the voids and form two interfaces. The atoms are coloured according to a computed phase variable; in (a) and (b) the phase variable is just the instantaneous field $m(x_1; X^0)$, whereas (b) and (d) use discrete time averages approximating $\frac{1}{t_2-t_1}\int_{t_1}^{t_2} m(x_1; X^t)\, dt$.

Simulation O1 used 64131 particles in a computational cell of dimensions $93.17 \times 23.29 \times 23.29$, while simulation O2 used 78911 particles in a cell of dimensions $100.86 \times 24.71 \times 24.96$.



(a) Unit cell (b) Orientation 1 (c) Orientation 2

Figure 6: The shaded planes in (b) and (c) show the two orientations of the solid-liquid interface with respect to the FCC lattice treated in the numerical simulations.

to single phase solid and liquid configurations and $g(r)$ computed in the interior of the two phases, excluding two intervals of length 10.0 in the interface regions.

An effect of the finite size of the computational cell is that periodic boundary conditions may interact with the solid and affect the results; here the computational cell was chosen to match the FCC structure in a specific orientation with respect to the box and thus stabilises the structure and orientation. It is important to know that the density in the FCC part (and hence the box cross section) is consistent with constant pressure simulations close to the melting point. A related question is whether the length of the computational box is large enough for properties around the interfaces in the infinitely layered structure to be good approximations of those near an interface between a solid and liquid on the macroscopic scale.

2.2 Computation Of the Coarse-Grained Model Functions

The coefficient functions (10) in the stochastic differential equation (7) for the coarse-grained phase-field are defined in terms of the time averaged expected values (8) and (9) on the form

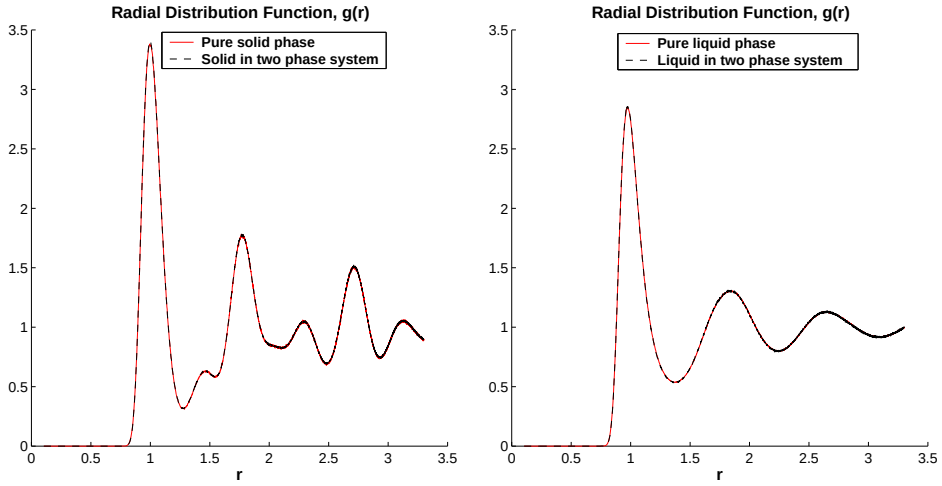
$$\frac{1}{\mathcal{T}}\mathrm{E}\left[\int_0^{\mathcal{T}} \psi(\cdot; X^t) \;\middle|\; X^0 = X_0\right],$$

where X_0 is a configuration of a stationary two-phase system. By setting up an initial configuration, X_0, as described in the previous section, and simulating discrete sample trajectories using the Euler-Maruyama method (15), a sequence of configurations $\{\bar{X}^k\}_{k=1}^K$ approximating the sequence $\{X^{t_k}\}_{k=1}^K$ for some times $0 < t_1 < \cdots < t_K = \mathcal{T}$, is obtained. In a post processing step a set of configurations $\mathcal{S} \subseteq \{\bar{X}^k\}_{k=1}^K$ is selected and averages

$$\mathcal{A}_{\mathcal{S}}(\psi) = \sum_{X \in \mathcal{S}} \psi(\cdot; X) w_X,$$

consistently weighted with weights w_X, are computed as approximations of the corresponding expected values in the continuous time model. It is usually more efficient not to include every configuration in the averages. This will be discussed in Section 3.

As described in the introduction, the averages are functions of the coordinate direction x_1, normal to the planar interface, since the mollifier in the definition (5) of the microscale



(a) FCC (b) Liquid

Figure 7: The radial distribution function, $g(r)$, computed from several configurations, separated in time, in the process of setting up the two-phase system in simulation O2. The solid curve shows $g(r)$ computed as an average over all particles in the computational cell used while pre-simulating the solid and the liquid part, in subfigure (a) and (b) respectively. The dashed curves show $g(r)$ computed as an average over particles in two slices of the computational cell of the two-phase system; subfigure (a) shows $g(r)$ obtained from the slice $5.0 \leq x_1 \leq 45.43$, inside the solid phase, and subfigure (b) shows $g(r)$ from the slice $55.43 \leq x_1 \leq 95.86$, inside the liquid phase. The configurations are taken from an equilibration run, after the closing of the initial gaps between the pre-simulated phases, but before the "production" run. The radial distribution functions show good agreement between the single phase systems and the corresponding solid and liquid subdomains away from the interface.

phase-field, m, is chosen to take uniform averages in the planes parallel to the interface. The mollifier used in the computations is

$$\eta(x) = \eta(x_1) = c \exp\left(-\frac{1}{2}\left(\frac{x_1}{\epsilon}\right)^2\right)\mathbf{1}_{|x_1|<R_c}, \tag{16}$$

where c is a normalising constant, ϵ is a smoothing parameter, and R_c is a cut-off. The smoothing parameter is on the order of typical nearest neighbour distances, $\epsilon \approx 1$, and $R_c = 6\epsilon$, for all choices of ϵ, which gives $\eta(R_c) \approx 1.5 \cdot 10^{-8}\eta(0)$; the shape of η can be seen in Figure 25(a), on page 41.

An explicit derivation of expressions for the drift and the diffusion is given in Appendix A. Separating the drift in terms containing two, one, and zero, derivatives of the mollifier, the right hand side of (8a) is approximated by

$$k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}\mathcal{A}_{\mathcal{S}}\left(m\right) + \frac{\partial}{\partial x_1}\mathcal{A}_{\mathcal{S}}\left(a_1\right) + \mathcal{A}_{\mathcal{S}}\left(a_0\right),$$

where

$$a_1(x;X) = \sum_{j=1}^{N}(k_{\mathrm{B}}T - m_j(X))[F_j(X)]_1\eta(x - X_j) \tag{17}$$

and

$$\begin{aligned} a_0(x;X) = &-\sum_{j=1}^{N}\left(k_{\mathrm{B}}T\nabla_{X_j}\cdot F_j(X) + \frac{1}{2}||F_j(X)||^2\right)\eta(x - X_j) \\ &- \frac{1}{2}\sum_{j=1}^{N}\sum_{i\neq j,i=1}^{N} f_{ij}(X)\cdot F_j(X)\eta(x - X_i). \end{aligned} \tag{18}$$

Here F_j is the total force acting on particle j, $[F_j(X)]_1$ is the x_1-component of the force, and f_{ij} are the contributions from individual pairs,

$$F_j(X) = -\nabla_{X_j}U(X) = \sum_{i\neq j,i=1}^{N}\Phi'(||X_i - X_j||)\frac{X_i - X_j}{||X_i - X_j||} = \sum_{i\neq j,i=1}^{N} f_{ij}(X).$$

The right hand side in equation (8b), for the coarse grained diffusion, is approximated by

$$\overline{B}(\cdot,\cdot) = \mathcal{A}_{\mathcal{S}}\left(2k_{\mathrm{B}}T\sum_{j=1}^{N}\big(p_j(\cdot,\cdot;X) + q_j(\cdot,\cdot;X)\big)\right), \tag{19}$$

where

$$\begin{aligned} p_j(x,y;X) = &\left(\frac{m_j(X)}{\epsilon^2}\right)^2\left[x - X_j\right]_1\left[y - X_j\right]_1\eta(x - X_j)\eta(y - X_j) \\ &- \frac{m_j(X)}{2\epsilon^2}[x - X_j]_1\eta(x - X_j)\left([F_j(X)]_1\eta(y - X_j) + \sum_{i\neq j,i=1}^{N}[f_{ij}(X)]_1\eta(y - X_i)\right) \\ &- \frac{m_j(X)}{2\epsilon^2}[y - X_j]_1\eta(y - X_j)\left([F_j(X)]_1\eta(x - X_j) + \sum_{i\neq j,i=1}^{N}[f_{ij}(X)]_1\eta(x - X_i)\right) \end{aligned}$$

and

$$q_j(x,y;X) = \frac{1}{4}\left(F_j(X)\eta(x-X_j) + \sum_{i\neq j, i=1}^{N} f_{ij}(X)\eta(x-X_i)\right) \cdot \left(F_j(X)\eta(y-X_j) + \sum_{i\neq j, i=1}^{N} f_{ij}(X)\eta(y-X_i)\right).$$

The functions $\mathcal{A}_{\mathcal{S}}(\psi)$ are computed in a discrete set of points $D_K = \{x_1^i\}_{i=1}^K$ along the x_1 axis of the molecular dynamics domain. This makes the computed components, $\mathcal{A}_{\mathcal{S}}(m)$, $\mathcal{A}_{\mathcal{S}}(a_1)$, and $\mathcal{A}_{\mathcal{S}}(a_0)$, of the drift coefficient function K-vectors and the computed $\overline{B}$ a K-by-K matrix. The individual diffusion coefficient functions $\bar{b}_j$ are obtained by taking the square root of the computed diffusion matrix, $\overline{B} = \overline{B}^{1/2}(\overline{B}^{1/2})^{\mathrm{T}}$, and letting the j:th column of $\overline{B}^{1/2}$ define $\bar{b}_j$. While an exact computation would produce a symmetric positive semi definite matrix $\overline{B}$, finite precision effects make some computed eigenvalues negative, but small in absolute value. In an eigenvector factorisation of $\overline{B}$, let Λ denote a diagonal matrix with all eigenvalues of $\overline{B}$ and Λ_+ a smaller diagonal matrix containing the dominant, possibly all, of the positive eigenvalues but no negative ones. Let V and V_+ be the matrices of the corresponding eigenvectors. Then the square root of the matrix Λ_+ is a real diagonal matrix which can be used in the approximation

$$\overline{B} = V\Lambda V^{\mathrm{T}} \approx V_+\Lambda_+{V_+}^{\mathrm{T}} = \left(V_+{\Lambda_+}^{1/2}{V_+}^{\mathrm{T}}\right)\left(V_+{\Lambda_+}^{1/2}{V_+}^{\mathrm{T}}\right)^{\mathrm{T}} =: BB^{\mathrm{T}}. \tag{20}$$

With one Wiener process $\widetilde{W}_j$ in the coarse-grained stochastic differential equation (7) per evaluation point, $K = M$, the component vectors, $\bar{b}_j$, of the diffusion in coarse-grained equation can be defined as the column vectors of the matrix B, to obtain

$$\sum_{j=1}^{M} \bar{b}_j \bar{b}_j^{\mathrm{T}} \approx \overline{B}.$$

If two grid points, x_1 and y_1, are further apart than twice the sum of the cut-off in the potential and the cut-off in the mollifier, then $p_j(x,y;\cdot)$ and $q_j(x,y;\cdot)$ is zero; hence a natural ordering $x_1^1 < x_1^2 < \cdots < x_1^K$ of the grid points makes $\overline{B}$ a band matrix. The definition of B in (20) preserves the connection between grid points and diffusion functions and the dominating terms in a tabulated vector $\bar{b}_j$ are those of nearby grid points.

3 Results

This section describes results from numerical simulations performed to compute the coarse-grained model functions. The value of the smoothing parameter ϵ in the mollifier is 1.0, unless another value is specified.

3.1 The averaged phase-field $m_{\text{av}} \approx \mathcal{A}_{\mathcal{S}}(m)$

The first observation is that during the time intervals of the molecular dynamics simulations, the interfaces between the solid and the liquid subdomains were sufficiently stable for the averaged potential energy phase-fields, $\mathcal{A}_{\mathcal{S}}(m)$, to appear qualitatively right. The phase-field appears to have two distinct equilibrium values, corresponding to the solid and liquid subdomains, and the transitions between the two regions are smooth and occur over distances of a few nearest neighbour distances; see Figure 8. Figure 9(b) shows that the computational cells in the molecular dynamics simulations are large enough for the phase-field in the interior of the two phases to attain values similar to the values in the corresponding single phase simulations. In simulations with a cubic, $23.29 \times 23.29 \times 23.29$, computational cell the gap between the phase-field levels in the solid and the liquid was significantly smaller, which indicates that the length of the computational cell can not be taken much smaller than in simulations O1 and O2. It is still possible that further increasing the size of the computational cell may affect the results.

3.2 The averaged drift $\overline{a} \approx \mathcal{A}_{\mathcal{S}}(\alpha)$

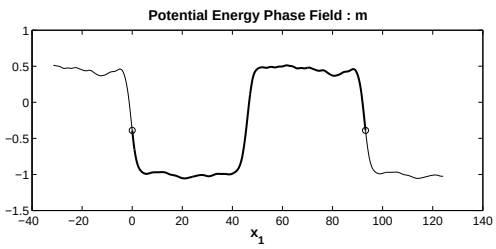
The average $\mathcal{A}_{\mathcal{S}}(m)$ approximates the expected time average (9). The next expected value to study is the one defining the coarse grained drift in (8a). In a stationary situation, where the interfaces do not move during the simulation and the averaged phase-field converges to a stationary profile, the average total drift in the stochastic differential equation describing the phase-field variable must converge to zero. Still the time averaged total drift corresponding to the simulation O2, whose averaged phase-field was discussed above, is far from zero; see Figure 10. The computed time averaged drift

$$\mathcal{A}_{\mathcal{S}}\left(\alpha(x;\bar{X}^n)\right) \approx \frac{1}{\mathcal{T}}\mathrm{E}\left[\int_0^{\mathcal{T}} \alpha(x;X^t)\,dt \;\middle|\; X^0 = X_0\right]$$

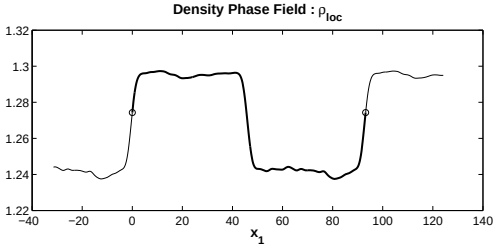
depends both on the length of the time interval where the average is computed, the number of configurations used in the average, and on the discrete approximation $\bar{X}^n$ of X^{t_n}; these potential error sources must be analysed to explain the result.

3.2.1 The effect of discrete time dynamics

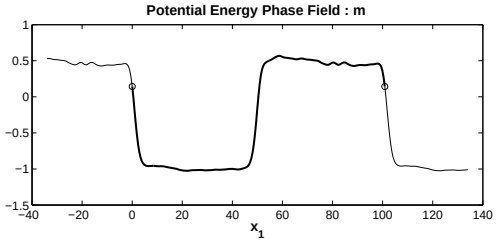
First consider the error associated with the discrete dynamics. The explicit form of the drift is derived for the continuous time mathematical model with the Smoluchowski dynamics (14), and not the discrete time Euler-Maruyama dynamics (15) that is used in the numerical simulations. For a fixed size of the time step this means that, even if the state of the numerical simulation is stationary on the time scale of the simulation so that time averaged phase-field converges to an equilibrium profile, the time averaged total drift will not go zero because of the time discretisation error. Figure 11 shows that the computed radial distribution functions, here from single phase solid configurations, are close when the time steps used vary from 10^{-7} to 10^{-4}; still the larger time steps give average computed drifts $\mathcal{A}_{\mathcal{S}}\left(\alpha(x;\bar{X}^n)\right)$ that are inconsistent with the observed time evolution of the average phase-field $\mathcal{A}_{\mathcal{S}}\left(m(x;\bar{X}^n)\right)$. As shown in Figure 13, the time step $\Delta t = 1 \cdot 10^{-5}$ gives an



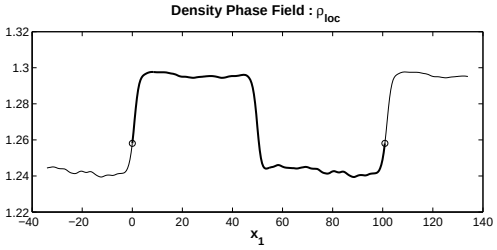
(a) Orientation 1



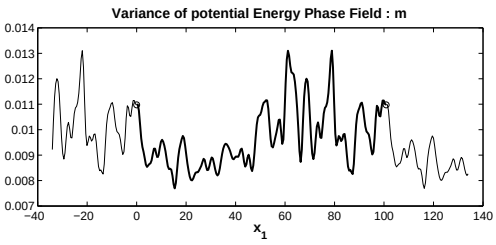
(b) Orientation 1



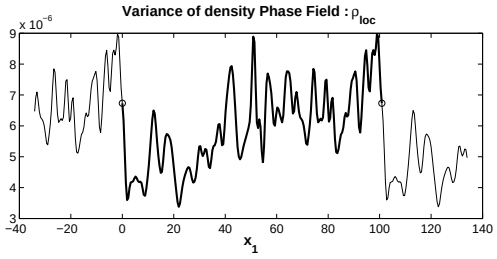
(c) Orientation 2



(d) Orientation 2

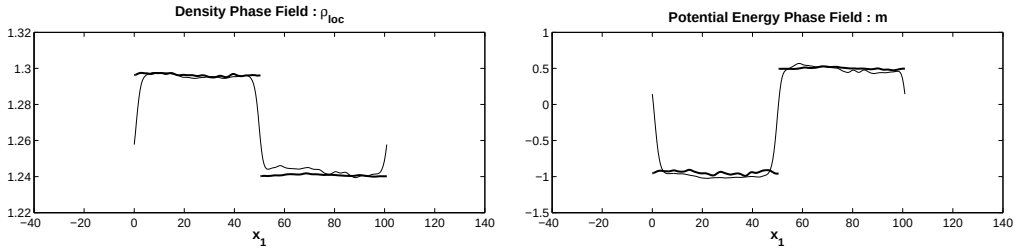


(e) Orientation 2



(f) Orientation 2

Figure 8: Subfigures (a) and (c) show the potential energy phase-field, $\mathcal{A}_{\mathcal{S}}(m)$, computed from simulations O1 and O2, respectively. Subfigures (b) and (d) show the corresponding spatially averaged particle densities. Subfigures (e) and (f) show the pointwise sample variance associated with the averages in (c) and (d). The thick parts of the curves show the computed functions in molecular dynamics cell. The thinner parts show the periodic continuations across the boundaries of the cell, marked by circles. The averages in simulation O1, and O2, were formed over 1721, and 1775, configurations separated in time by $5 \cdot 10^{-4}$, so that the total time from first to last configuration was 0.860, and 0.8875, respectively. The high frequency fluctuations are small after averaging on this time scale, but larger fluctuations remain in both phases. This suggests that the two phase system is not yet equilibrated. Still the computed phase-fields appear qualitatively correct.



(a) Density Phase-Field, O2

(b) Potential Energy Phase-Field, O2

Figure 9: The computational cell in the molecular dynamics simulations must be sufficiently large for the infinitely layered structure to resemble a system with a single solid–liquid interface on the macroscopic scale. In simulation O2 the total length of the computational cell was 100.86; subfigure (b) shows that this was sufficient for the averaged phase-field, $\mathcal{A}_{\mathcal{S}}(m)$, to obtain values in the interior of each phase that are similar to the functions, marked by thick curves, obtained in the single phase configurations simulated during the setup of simulation O2.

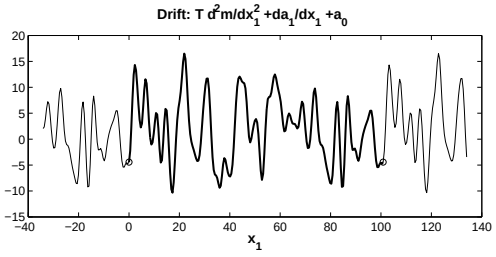


Figure 10: The average total drift, $\mathcal{A}_{\mathcal{S}}(\alpha)$, based on the same 1775 configurations from simulation O2 as $\mathcal{A}_{\mathcal{S}}(m)$ in Figure 8(c), is still dominated by large oscillations.

average drift that oscillates between -100 and -250, even when the computed phase-field $\mathcal{A}_{\mathcal{S}}\left(m(x;\bar{X}^n)\right)$ is approximately constant over times of the order 10. For this reason, the time step used in simulations O1 and O2, generating configurations for the computation of $\mathcal{A}_{\mathcal{S}}\left(m(x;\bar{X}^n)\right)$ and $\mathcal{A}_{\mathcal{S}}\left(\alpha(x;\bar{X}^n)\right)$, was $\Delta t = 5\cdot 10^{-7}$, while the time step used in the setup of the initial configurations often was a thousand times larger. With this small time step the fluctuations in the computed average drift outweighs the deviation from the expected zero mean; see Figure 10.

The choice of the time step size $\Delta t = 5 \cdot 10^{-7}$ was guided by a rough error estimate, taking into account the maximal absolute value of second order derivatives of the Smoluchowski drift $-\nabla_{X_j} U(X^t)$ when the nearest neighbours don't come closer than approximately 0.8, as indicated by Figure 11. Then the time step was adjusted so that the slow convergence of the time averaged drift in terms of $\mathcal{T}$ and the number of configurations, $\bar{X}^n$, was the dominating error source in the results. This over-killing of the time discretisation error in the molecular dynamics wastes computer power and could possibly be avoided by more accurate error estimates, allowing a matching of the different error contributions. Using a reasonable number of grid points, K, in the computation of the drift coefficient K-vectors and the diffusion K-by-K matrix $\overline{B}$, in (19), the computational cost for obtaining $\overline{B}$ in particular, far exceeds the cost of actually making a time step in the molecular dynamics simulation. Hence the additional cost of over-killing the time step error is not very significant, provided that not every configuration in the time stepping is included in the averages $\mathcal{A}_{\mathcal{S}}(m)$, $\mathcal{A}_{\mathcal{S}}(\alpha)$, and $\overline{B}$. In the averages shown in Figure 8 and Figure 10, for example, the configurations were sampled at time intervals $5 \cdot 10^{-4}$, corresponding to 1000 time steps in the molecular dynamics simulation.

A further improvement may be to incorporate finite step-size effects in the expressions for the components of the drift. The higher order derivatives of the pair potential attain large values when two particles come closer than 1; see Figure 12. Hence the time step must be taken very small for Itô's formula to be a good approximation of the dynamics of the discrete system. Instead of a direct application of Itô's formula in the derivation of the drift and diffusion terms in (26) and (27) on page 44 one could include higher order terms in the expansion to improve the accuracy of the computed drift.

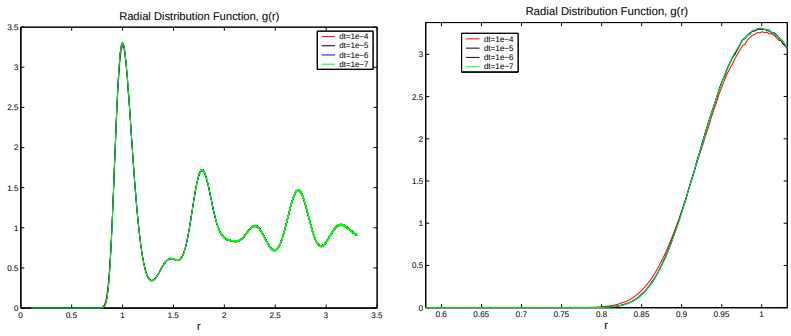
3.2.2 Dependence on the length of the time averaging interval

Next consider the dependence of the computed coarse-grained drift coefficient function on the length of the time interval $\mathcal{T}$. Introducing the time averaged drift over a sample path as

$$\overline{A}_{\mathcal{T}} = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} \alpha(\cdot; X^t)\, dt,$$

the coarse-grained drift (8a) is $\overline{a} = \mathrm{E}[\overline{A}_{\mathcal{T}}]$. The rate of convergence of $\overline{a}$, as $\mathcal{T} \to \infty$, in the continuous time mathematical model can be estimated by integration of the stochastic differential equation (6) for the phase-field m. Integrating from 0 to $\mathcal{T}$ gives

$$m(\cdot; X^{\mathcal{T}}) - m(\cdot; X^0) = \int_0^{\mathcal{T}} \alpha(\cdot; X^t)\, dt + \int_0^{\mathcal{T}} \sum_{j=1}^{N} \sum_{k=1}^{3} \beta_{j,k}(\cdot; X^t)\, dW_{j,k}^t, \qquad (21)$$



(a) $g(r)$, FCC, different Δt (b) Detail: first slope

Figure 11: The radial distribution function, $g(r)$, computed using four different step sizes in a single phase FCC simulation. The difference between the curves is small (a), even if the one obtained for $\Delta t = 10^{-4}$ differs visibly from the others in the first peak (b). In spite of the good approximation in the radial distribution function, the larger step sizes give very poor results in the computed dynamics of m.

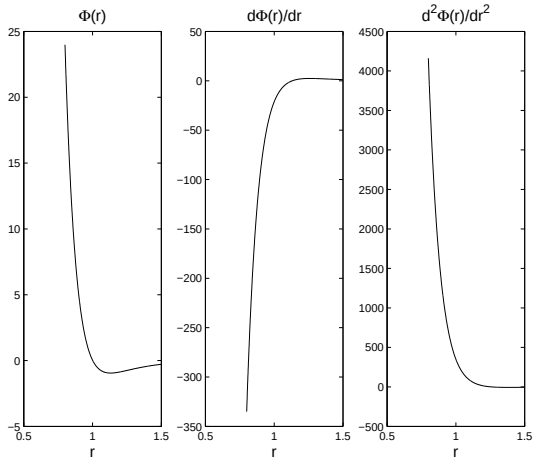
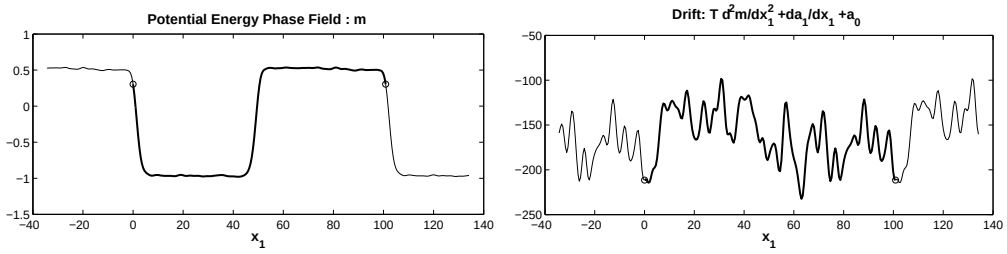
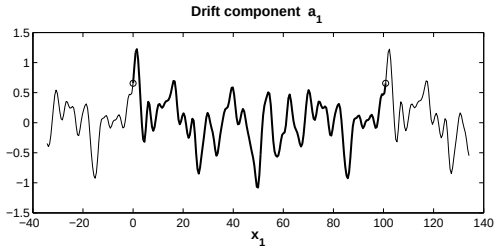


Figure 12: The absolute value of the Exp-6 potential and its derivatives grow very quickly with decreasing r, in the range with positive $g(r)$ in Figure 11(b). The potential and its two first derivatives using the model parameters in Table 2.1.1, on page 9, are shown here.

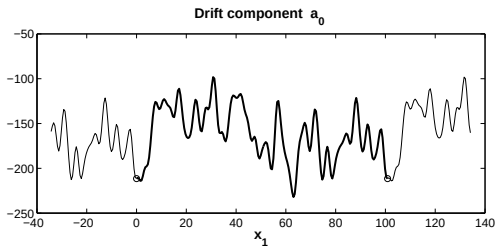


(a) $\mathcal{A}_{\mathcal{S}}\,(m)$

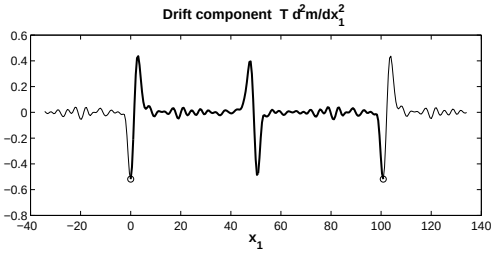
(b) $\mathcal{A}_{\mathcal{S}}\,(\alpha)$



(c) $\mathcal{A}_{\mathcal{S}}\,(a_1)$

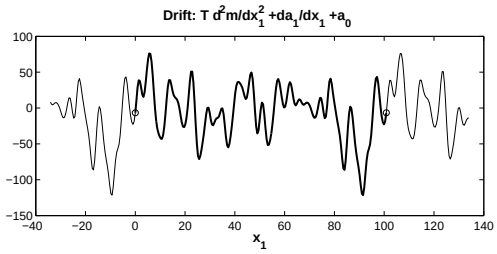


(d) $\mathcal{A}_{\mathcal{S}}\,(a_0)$

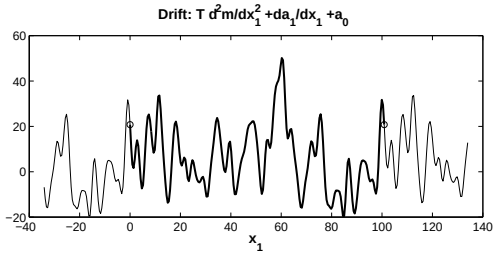


(e) $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}\mathcal{A}_{\mathcal{S}}\,(m)$

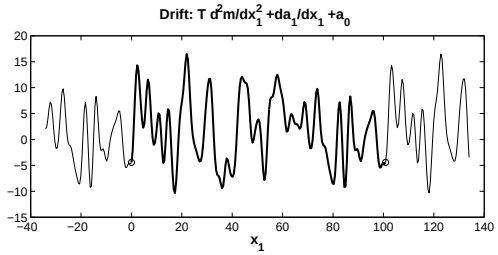
Figure 13: Using the step size $\Delta t = 1 \cdot 10^{-5}$ in the Euler-Maruyama scheme, the computed average phase-field $\mathcal{A}_{\mathcal{S}}\,(m)$ is approximately stationary during the time interval of the averaging. In subfigure (a) the average is based on 123 configurations, sampled at every ten thousandth time step, corresponding to a total time interval of 12.3. Still, the computed average drift $\mathcal{A}_{\mathcal{S}}\,(\alpha)$ is far from zero during this time interval. The large deviation from zero is entirely due to the term $\mathcal{A}_{\mathcal{S}}\,(a_0)$.



(a) Mean based on 111 configurations, $\mathcal{T} = 0.0555$



(b) Mean based on 444 configurations, $\mathcal{T} = 0.2220$



(c) Mean based on 1775 configurations, $\mathcal{T} = 0.8875$

	$\mathcal{T} = 0.0555$	$\mathcal{T} = 0.2220$	$\mathcal{T} = 0.8750$
111 cfgs.	-5.7 $(1.3 \cdot 10^3)$	-10.3 $(1.2 \cdot 10^3)$	4.3 $(1.1 \cdot 10^3)$
444 cfgs.		6.1 $(2.1 \cdot 10^2)$	0.67 $(2.7 \cdot 10^2)$
1775 cfgs.			1.9 $(3.8 \cdot 10^1)$

(d) The spatial (x_1) mean and, within parentheses, variance of $\mathcal{A}_{\mathcal{S}}(\alpha)$

Figure 14: The total drift $\mathcal{A}_{\mathcal{S}}(\alpha)$, decays slightly faster with $\mathcal{T}$ than the predicted $1/\sqrt{\mathcal{T}}$ in the examples (a), (b), and (c) above. Here the number of configurations in the averages grows with $\mathcal{T}$ and the means and variances of $\mathcal{A}_{\mathcal{S}}(\alpha)$ tabulated in (d) suggest that the number of configurations still restricts the rate of convergence. The average in subfigure (c) is based on the same 1775 configurations from simulation O2 as $\mathcal{A}_{\mathcal{S}}(m)$ in Figure 8(c). The averages in subfigures (a) and (b) are based on the first 111 and 444 configurations, respectively.

so that, by taking the expectation and using that, since X^t is W^t-adapted, the expectations of the Itô-integrals vanish

$$\mathrm{E}\left[\int_0^{\mathcal{T}} \alpha(\cdot; X^t)\, dt\right] = \mathrm{E}\left[m(\cdot; X^{\mathcal{T}}) - m(\cdot; X^0)\right]. \tag{22}$$

Hence, if the phase-field is stationary, then the expected mean drift over time is zero. Normalising (21) and (22) by $\mathcal{T}$,

$$\overline{A}_{\mathcal{T}} - \mathrm{E}\left[\overline{A}_{\mathcal{T}}\right] = \\ \frac{1}{\mathcal{T}}\left(m(\cdot; X^{\mathcal{T}}) - m(\cdot; X^0) - \mathrm{E}\left[m(\cdot; X^{\mathcal{T}}) - m(\cdot; X^0)\right] - \sum_{j=1}^{N}\sum_{k=1}^{3}\int_0^{\mathcal{T}} \beta_{j,k}(\cdot; X^t)\, dW_{j,k}^t\right)$$

and the variance of $\overline{A}_{\mathcal{T}}$ is obtained as

$$\begin{aligned}\mathrm{Var}[\overline{A}_{\mathcal{T}}] &= \mathrm{E}\left[\left(\overline{A}_{\mathcal{T}} - \mathrm{E}\left[\overline{A}_{\mathcal{T}}\right]\right)^2\right] \\ &= \frac{1}{\mathcal{T}^2}\mathrm{Var}\left[m(\cdot; X^{\mathcal{T}}) - m(\cdot; X^0)\right] \\ &\quad + \frac{1}{\mathcal{T}^2}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{E}\left[\left(\int_0^{\mathcal{T}} \beta_{j,k}(\cdot; X^t)\, dW_{j,k}^t,\right)^2\right] \\ &\quad - \frac{2}{\mathcal{T}^2}\mathrm{E}\left[\left(m(\cdot; X^{\mathcal{T}}) - m(\cdot; X^0)\right)\left(\sum_{j=1}^{N}\sum_{k=1}^{3}\int_0^{\mathcal{T}} \beta_{j,k}(\cdot; X^t)\, dW_{j,k}^t\right)\right],\end{aligned}$$

where last expression was simplified using the independence of the different components of W^t, and the zero expected value of Itô integrals. Assuming that both the phase-field and all the diffusion coefficients are bounded, the dominating term in the expression for the variance is

$$\frac{1}{\mathcal{T}^2}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{E}\left[\left(\int_0^{\mathcal{T}} \beta_{j,k}(\cdot; X^t)\, dW_{j,k}^t,\right)^2\right] = \mathcal{O}\left(\frac{1}{\mathcal{T}}\right).$$

In the two phase simulations considered here, the values of the computed phase-field varies between a lower level in the solid a higher in the liquid. Because of the small positive probability for two particles, with trajectories computed using the Euler-Maruyama dynamics (15), to get within an arbitrarily small distance of each other, there is no guarantee that computed phase-field always will stay in this range. However, if the minimum interatomic distance becomes to small, that is a breakdown of the whole microscopic model and not just a problem when computing the drift; this situation has not been observed to happen in the simulations here and the observed values of the phase-field are all in the range $(-1.5, 1.0)$. Hence the assumption that m is bounded seems reasonable here; a bound on the absolute value of the diffusion coefficients $\beta_{j,k}$ is less certain, and it will have to be larger than the bound on m.

For the average drift to be small compared to the stationary values of the phase-field itself, it must be at least a factor 100 smaller than the computed average shown in Figure 10. Based

on the rough analysis above, the expected time average of the total drift can be expected to decay as $1/\sqrt{\mathcal{T}}$ with a large constant factor. When the computed drift $\mathcal{A}_{\mathcal{S}}\left(\alpha\right)$ in Figure 10 is compared to averages computed using two smaller subsequences of configurations, the convergence to zero appears to be slightly faster than $1/\sqrt{\mathcal{T}}$; see Figure 14. Even when extrapolating with the measured convergence rate, decreasing the average drift by a factor 100 would require increasing the averaging time interval by more than a factor 1000, which is beyond reach within the present project. With increasing accuracy in the time average, eventually the time step in the molecular dynamics simulations must be decreased, further increasing the computational cost.

Since the total drift coefficient function, $\overline{a}(x_1) \approx \mathcal{A}_{\mathcal{S}}\left(\alpha(x_1;\cdot)\right)$, where

$$\mathcal{A}_{\mathcal{S}}\left(\alpha(x_1;\cdot)\right) = k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}\mathcal{A}_{\mathcal{S}}\left(m(x_1;\cdot)\right) + \frac{\partial}{\partial x_1}\mathcal{A}_{\mathcal{S}}\left(a_1(x_1;\cdot)\right) + \mathcal{A}_{\mathcal{S}}\left(a_0(x_1;\cdot)\right), \tag{23}$$

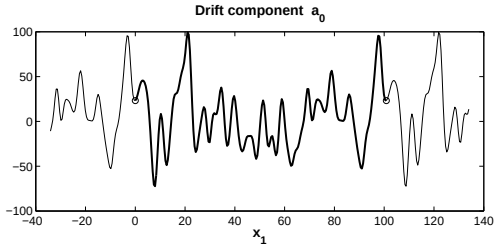
in the coarse grained model is expected to be zero in a stationary situation, a more accurate computation would serve primarily as a consistency test. On the other hand, the individual terms in the right hand side are not all expected to vanish independently. Indeed, it is clear from the results on $\mathcal{A}_{\mathcal{S}}\left(m(x_1;\cdot)\right)$ in Section 3.1 that the term with two differentiations with respect to x_1 will not be identically zero. This also shows that while the total drift is far from $\mathcal{A}_{\mathcal{S}}\left(\alpha(x_1;\cdot)\right)$ converged, at least one term is reasonably accurate.

A closer look on the terms of the drift, reveals that the different terms are of different orders of magnitude. The term $\mathcal{A}_{\mathcal{S}}\left(a_0(x_1;\cdot)\right)$, with a_0 defined in (18), contains both second order differentials of the potential with respect to the particle positions and second powers of first order differentials. These terms, as illustrated in Figure 12, attain much larger values than the potential itself and cancellation is required to reduce $\mathcal{A}_{\mathcal{S}}\left(a_0(x_1;\cdot)\right)$ to a size comparable with the two other terms in the drift. Figure 15(e) shows an individual $a_0(x_1;\cdot)$ computed from one configuration; in the length of the computational cell, the values range from approximately -500 to +500, whereas the phase-field, $m(x_1;\cdot)$, is of the order 1, and $a_1(x_1;\cdot)$ is of intermediate magnitude. A comparison between the computed averages $\mathcal{A}_{\mathcal{S}}\left(\alpha(x_1;\cdot)\right)$ in Figure 14 and $\mathcal{A}_{\mathcal{S}}\left(a_0(x_1;\cdot)\right)$ in Figure 15 shows that $\mathcal{A}_{\mathcal{S}}\left(a_0(x_1;\cdot)\right)$ is the dominates the other two terms completely here.

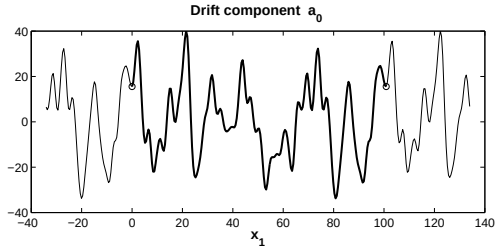
The average $\mathcal{A}_{\mathcal{S}}\left(a_1(x_1;\cdot)\right)$, contains first order differentials of the potential, but only to the first power. The convergence of is faster than that of $\mathcal{A}_{\mathcal{S}}\left(a_0(x_1;\cdot)\right)$, but the computed averages in Figure 16 still show significant fluctuations. The final term in $\mathcal{A}_{\mathcal{S}}\left(\alpha(x_1;\cdot)\right)$ is $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}\mathcal{A}_{\mathcal{S}}\left(m(x_1;\cdot)\right)$, which only depends on the potential and not its derivatives. This average converges faster than the other two and, even after two differentiations with respect to x_1, the fluctuations are small compared to the distinct structures at the interfaces; see Figure 17.

3.2.3 Obtaining the phase-field double-well potential from the drift

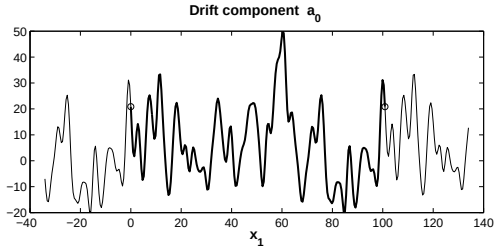
When defining a phase-field variable in terms the potential energy in the microscale model in Section 1, the goal was to compute a reaction–diffusion equation, like the Allen-Cahn equation (2b), for the coarse-grained phase-field. In a one dimensional problem, with $T \equiv$



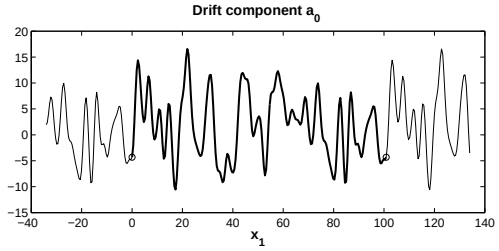
(a) Mean based on 111 configurations, $\mathcal{T} = 0.8875$



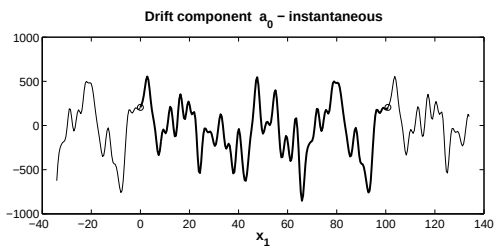
(b) Mean based on 444 configurations, $\mathcal{T} = 0.8875$



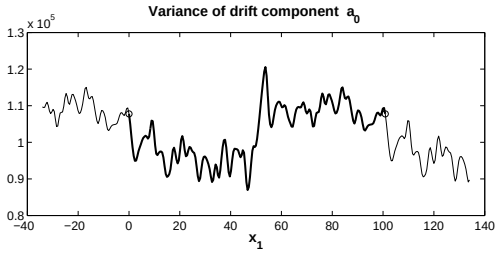
(c) Mean based on 444 configurations, $\mathcal{T} = 0.2220$



(d) Mean based on 1775 configurations, $\mathcal{T} = 0.8875$

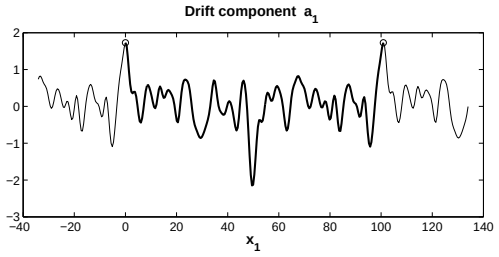


(e) One single configuration, X^t

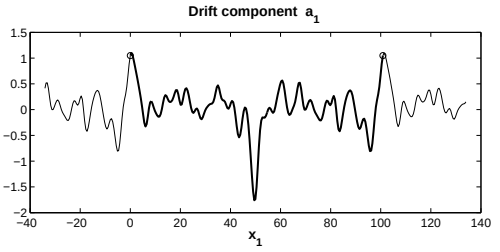


(f) Variance based on the configurations in (d)

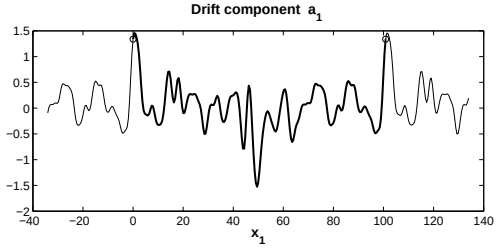
Figure 15: The term $\mathcal{A}_{\mathcal{S}}(a_0)$ is the slowest converging average in the drift average; a comparison with Figure 14 shows that this term dominates the total drift average. This explicit form of the term, given in (18) is a sum over all particles of terms that are second order in the particle forces and a term containing the divergence of the particle force; in the molecular dynamics simulation, these terms are large and so is the function a_0, when computed from a single configuration, as in (e). Eventually the average must decrease to order 1 through cancellation, but for the number of configurations available here fluctuations dominate the computed averages $\mathcal{A}_{\mathcal{S}}(a_0)$.



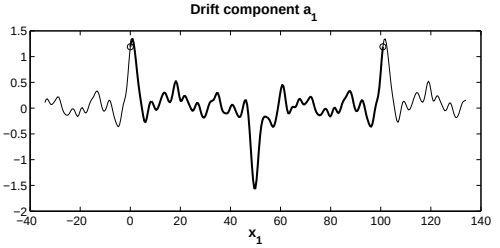
(a) Mean based on 111 configurations, $\mathcal{T} = 0.8875$



(b) Mean based on 444 configurations, $\mathcal{T} = 0.8875$



(c) Mean based on 444 configurations, $\mathcal{T} = 0.2220$



(d) Mean based on 1775 configurations, $\mathcal{T} = 0.8875$

Figure 16: The term $\mathcal{A}_{\mathcal{S}}(a_1)$ is supposed to approach zero as the number of configurations, and $\mathcal{T}$, increases, provided that the interfaces are stationary. Though the fluctuations are large here, they are much smaller than in Figure 15. When the fluctuations decrease a pattern appears with peaks at the two interfaces. This supports the observation, from the computed $\mathcal{A}_{\mathcal{S}}(m)$ in Figure 8, that the two phase system is not in equilibrium yet and the interfaces are not really stationary on the time scale of the average.

T_{M} and k_1 constant, the Allen-Cahn equation reduces to

$$\frac{\partial \phi}{\partial t} = k_1 \frac{\partial^2}{\partial x_1^2}\phi - k_2 f'(\phi) + noise, \tag{24}$$

where the derivative of the double-well potential f gives the reaction part in this reaction–diffusion equation. Now, the coarse-grained equation

$$dm_{\mathrm{cg}}^t(x_1) = \left(k_{\mathrm{B}}T \frac{\partial^2}{\partial x_1^2} m_{\mathrm{cg}}^t(x_1) + \frac{\partial}{\partial x_1}\overline{a}_1(x_1) + \overline{a}_0(x_1) \right) dt + \sum_{j=1}^{M} \overline{b}_j(x)\, d\widetilde{W}_j^t,$$

where

$$\overline{a}_1(x_1) = \mathcal{A}_{\mathcal{S}}\,(a_1)(x_1), \qquad \overline{a}_0(x_1) = \mathcal{A}_{\mathcal{S}}\,(a_0)(x_1) \qquad \text{, for } x_1 \in D_K,$$

and the diffusion coefficient vectors, $\overline{b}_j$, are obtained from the factorisation (20), is a stochastic convection–reaction–diffusion equation. As the described above the time averaged drift is zero in a stationary situation, but in the computations presented here the fluctuations are still too large. In the ideal situation for a stationary interface, when all three components in the drift average have converged, the convection should vanish, that is

$$\frac{\partial}{\partial x_1}\overline{a}_1 \equiv 0,$$

and the reaction and diffusion parts should cancel each other, so that

$$0 = k_{\mathrm{B}}T \frac{\partial^2}{\partial x_1^2} m_{\mathrm{cg}}^t(x_1) + \overline{a}_0(x_1). \tag{25}$$

The second best thing, when some of the computed averages contain too large errors, is to extract information from the most accurate part, that is $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}m_{\mathrm{av}}(x_1)$. Assuming that this computed average already is close to what it would be in the ideal situation, an approximation of the reaction term can be obtained from (25).

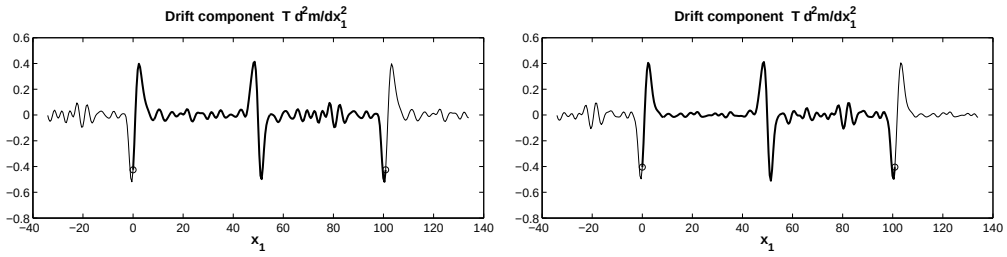
The expression of the drift in the coarse-grained equation (7) as a function of the coarse-grained phase-field m_{cg} in the interface regions, instead of the space variable x_1, assumes monotonicity of the phase-field near the interfaces to allow the inversion in (10). Figure 18 shows $m_{\mathrm{av}}(x_1)$ and $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}m_{\mathrm{av}}(x_1)$ in the interval of monotonicity for $m_{\mathrm{av}}(x_1)$ in the simulation O2. Using the computed $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}m_{\mathrm{av}}(x_1)$ in (25), gives

$$\overline{a}_0(x_1) = -k_{\mathrm{B}}T \frac{\partial^2}{\partial x_1^2} m_{\mathrm{av}}(x_1).$$

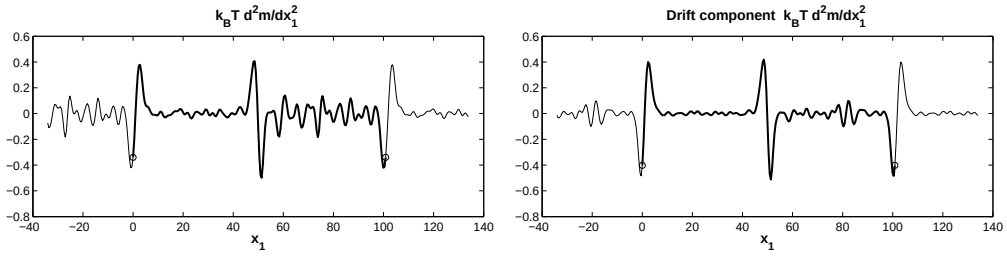
Inverting the computed function $m_{\mathrm{av}}(x_1)$ in the interface intervals, the derivative of the double-well potential f can be identified as

$$f'(m_{\mathrm{cg}}) = \overline{a}_0(m_{\mathrm{av}}^{-1}(m_{\mathrm{cg}})).$$

Integration with respect to m_{cg} in the interval between $m_{\mathrm{cg}_{\mathrm{solid}}}$ and $m_{\mathrm{cg}_{\mathrm{liquid}}}$ gives the double-well potentials shown in Figure 19(a). As expected the potentials obtained from the two different simulations O1 and O2 are slightly different. However, the potentials obtained from the two different interfaces in one molecular dynamics simulation cell also differ slightly and it is not possible to say that difference between simulations O1 and O2 depend on the orientation of the interfaces with respect to the crystal lattice. The computed double wells seem to be qualitatively right.



(a) Mean based on 111 configurations, $\mathcal{T} = 0.8875$ (b) Mean based on 444 configurations, $\mathcal{T} = 0.8875$



(c) Mean based on 444 configurations, $\mathcal{T} = 0.2220$ (d) Mean based on 1775 configurations, $\mathcal{T} = 0.8875$

Figure 17: The average $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}\mathcal{A}_{\mathcal{S}}\left(a_1\right)$ converges faster than the two other terms in $\mathcal{A}_{\mathcal{S}}\left(\alpha\right)$. The fluctuations are larger in subfigures (b) than in (c), which indicates that the error is dominated by the length of the averaging time interval rather than the number of configurations sampled within the time interval.

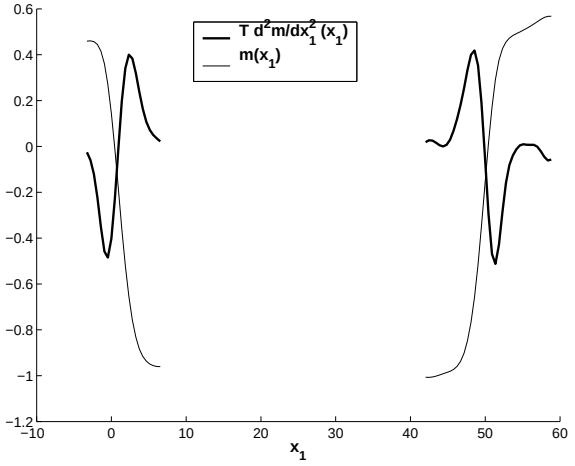


Figure 18: The computed $m_{\mathrm{av}}(x_1)$ in its monotone intervals in the interfaces together with the corresponding diffusion part of the drift $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}\mathcal{A}_{\mathcal{S}}\left(a_1\right)$. The curves shown are part of the those in Figure 8(d) and Figure 17(d).

3.3 The averaged diffusion matrix $\overline{B}$ and the coarse-grained diffusion coefficients $\bar{b}_j$.

The final component to extract in the coarse-grained model is the diffusion in the stochastic differential equation for m^t_{cg}. Using $\epsilon = 1.0$ and the same 1775 configurations that were used in the computation of the averaged phase-field and drift for simulation O2, the averaged diffusion matrix $\overline{B}$, has been computed, with the result shown in Figure 20(a). As described in Section 2.2, the square root of $\overline{B}$is computed by an eigenvector decomposition where all negative eigenvalues are set to zero; the result is shown in Figure 20(b). The negative eigenvalues are very small in absolute value, compared to the dominating positive ones, so the error made by neglecting them is insignificant when BB^{T} is compared to $\overline{B}$. By choosing the diffusion coefficients $\bar{b}_j$ in the coarse-grained stochastic differential equation as the columns of B, they become localised in space; see Figure 20(c). With $\epsilon = 1.0$ the observed difference between the diffusion in the solid part and the liquid part is small, as shown in Figure 21.

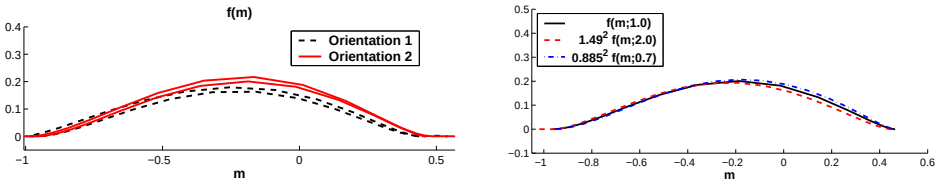
3.4 Dependence on the smoothing parameter

The mollifier η includes a parameter, ϵ, determining the scale on which the local average is taken. This is in itself an ad hoc variable in the micro model and it is important to analyse its effects on the computed quantities.

A lower limit on ϵ is set by the demand that the phase-field be approximately constant in the solid in spite of the periodic structure. If the solid structure is aligned with the computational domain in such a way that the global spatial averages are taken parallel to atomic layers, then the parameter ϵ controlling the width of the average in the orthogonal direction must be large enough to smooth the gaps between the atomic layers. In the numerical simulations the orientations of the FCC lattice with respect to the solid–liquid interface, and hence the planes of averaging, are precisely such that averages are computed parallel to atomic planes, as illustrated in Figure 22. In the present case the distance to the nearest neighbours in the FCC-lattice is around 1.02; with η on the form (16) the parameter ϵ must be taken greater than 0.43 to ensure that η decreases with at most a factor $1/2$ in half the distance to the nearest neighbour, which seems a reasonable demand. Figure 23, presenting computed phase-fields based on local averages of the density and the potential energy using $\epsilon = 0.45$, shows that the smoothing parameter has to be larger than this to avoid oscillations in the solid part. The phase-fields based on $\epsilon = 0.70$ in Figure 24 do not show these oscillations on the length scale smaller than the distance between atom layers.

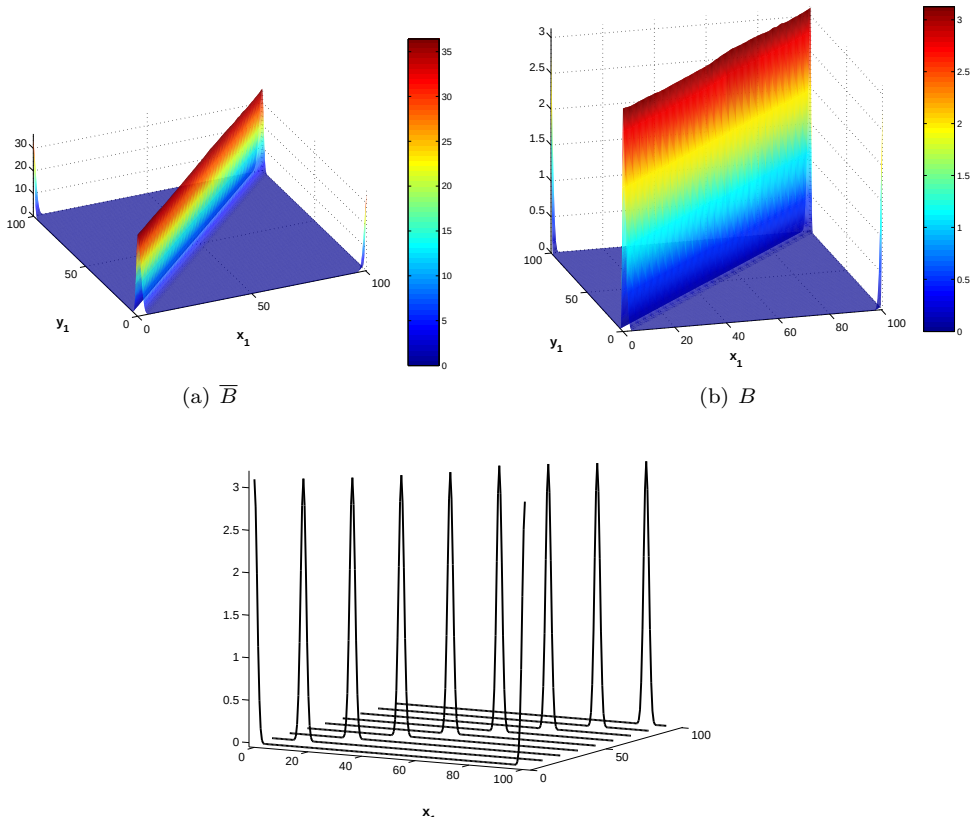
For the method to be reasonable, the lower bound on ϵ must not hide an interface width in the phase-field that is sharp even on the atomic scale. In addition to the computations with $\epsilon = 1.0$, the phase field has been computed for $\epsilon = 0.45$, 0.70, and 2.0. The computed phase-fields in the regions around the interfaces, for both orientation 1 and 2, are shown in Figure 25. The comparison shows that the interface width varies with the smoothing parameter. It would not, however, become infinitely sharp in the limit when ϵ goes to zero, even if the lower bound on ϵ were disregarded. This is clear from the results presented in Figure 26 where, in addition to the values of ϵ above, a phase-field obtained with $\epsilon = 0.05$, violating the lower bound, is shown around one of the interfaces in O1. This value of the

smoothing parameter, and the corresponding mollifier cutoff, $R_c = 6 \cdot 0.05 = 0.3$, is so small that the contribution to the phase-field of an individual atom in the FCC lattice is restricted to an interval extending less than half way to the next atom layer in either direction. Still the change in the phase-field, from strong oscillations in the solid to decaying oscillations around the average in the pure liquid, occurs gradually on a length scale corresponding to at least several atom layers and thus several times the artificial smoothing introduced by ϵ. Figure 26 also shows that the interface region of the phase-field obtained with $\epsilon = 0.45$, 0.70, and 1.0 is wider than the transition region of a step function, representing an infinitely sharp interface, smoothed by a convolution with the mollifier using the corresponding ϵ. For $\epsilon = 2.0$ the interface is very close to that of a mollified step function in both width and profile. The interface width of the smoothed step function is proportional to ϵ and it is expected that the same will hold for the phase-field, m, if the smoothing parameter is increased beyond the present range.



(a) Double well potential, $\epsilon = 1.0$ (b) Orientation 2, different ϵ

Figure 19:
(a) The computed double well potentials from both simulation O1 and O2 using m_{av} shown in Figure 8 and the corresponding $k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}m_{\mathrm{av}}(x_1)$.
(b) The computed double well potentials from one of the interfaces in O2, using three different values of the smoothing parameter ϵ in the mollifier. Since the interface width varies with ϵ the height of the potential barriers vary with ϵ. Here double-wells have been rescaled with factors obtained in the analysis of the ϵ-dependence in Figure 27 to compare the shape of the curves.

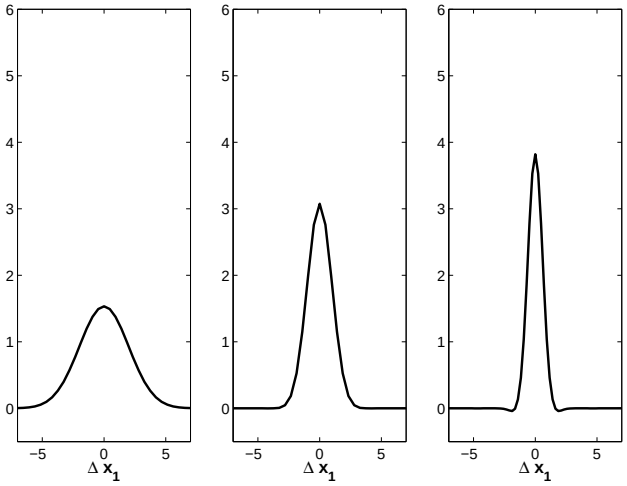


(a) $\overline{B}$

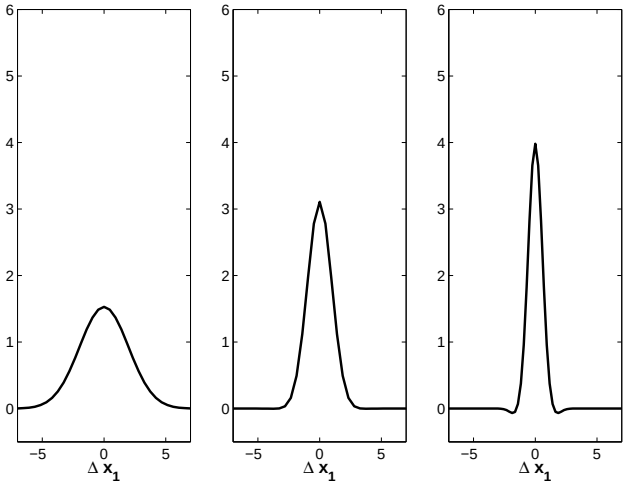
(b) B

(c) Some $\bar{b}_j$:s

Figure 20: The computed average diffusion matrix $\overline{B}$, for $\epsilon = 1.0$, using the same configurations from simulation O2 as in Figure 8(d) and Figure 17(d), is shown in (a). The square root B of $\overline{B}$, as defined in (20) is shown in (b). The individual columns in B are the diffusion coefficient functions, $\bar{b}_j$, in the stochastic differential equation for the coarse-grained phase-field m^t. Some of these column vectors have been plotted as functions of the space variable x_1 in (c). The support of each $\bar{b}_j$ is centred around the grid point x_1^j.

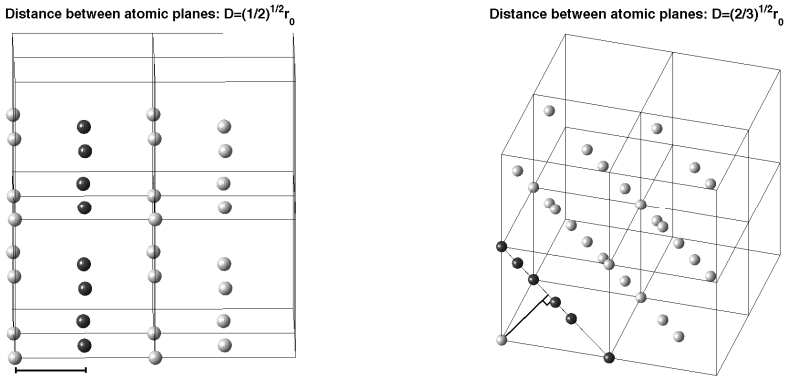


(a) FCC, ϵ=2.0 (b) FCC, ϵ=1.0 (c) FCC, ϵ=0.55



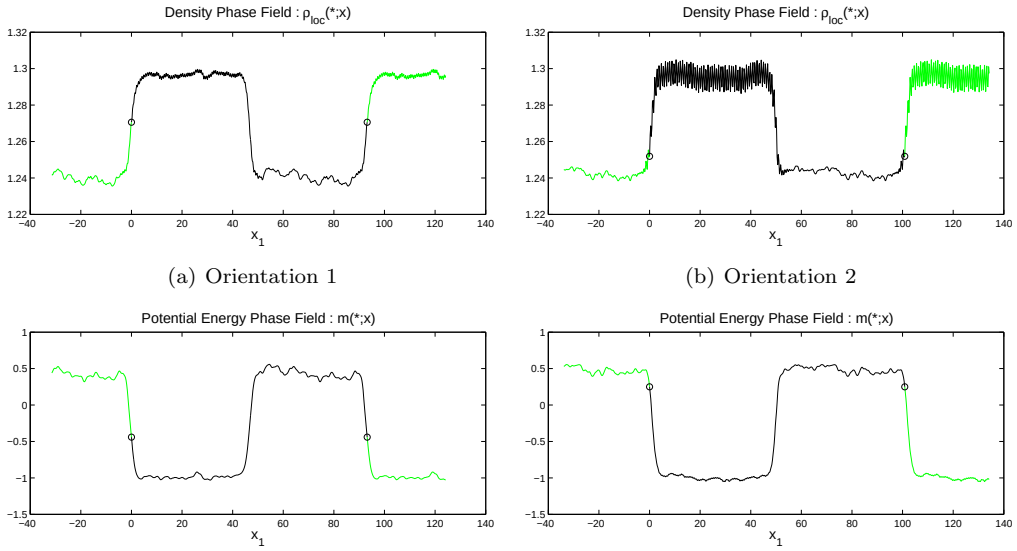
(d) Liquid, ϵ=2.0 (e) Liquid, ϵ=1.0 (f) Liquid, ϵ=0.55

Figure 21: The average diffusion coefficient functions, $\tilde{\bar{b}}(\Delta x_1) = \text{mean}\left\{\bar{b}_j(x_1^j + \Delta x_1)\right\}$ have been computed for different values of ϵ, with the mean taken over points x_1^j in the interior of the solid and the liquid domains, respectively. The configurations used are the same as in Figure 20. The difference between the solid and liquid parts is small.



(a) Orientation 1 (b) Orientation 2

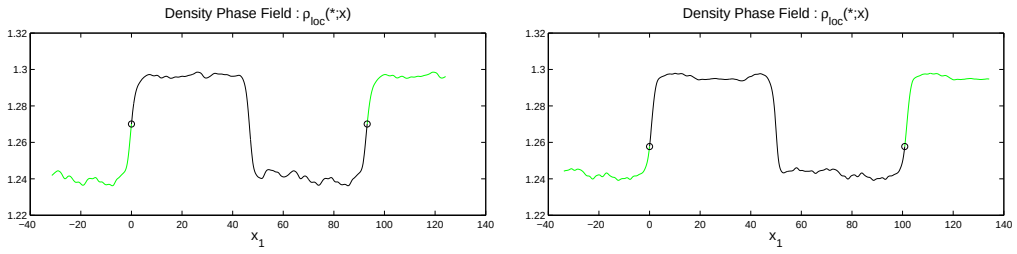
Figure 22: The distance between two adjacent atom layers in a perfect FCC lattice is $\sqrt{1/2}\,r_0$ in orientation 1 and $\sqrt{2/3}\,r_0$ in orientation 2, where r_0 is the nearest neighbour distance.



(a) Orientation 1 (b) Orientation 2

(c) Orientation 1 (d) Orientation 2

Figure 23: Computed density, ρ_{loc}, and potential energy phase fields for simulations O1 and O2 using $\epsilon = 0.45$.



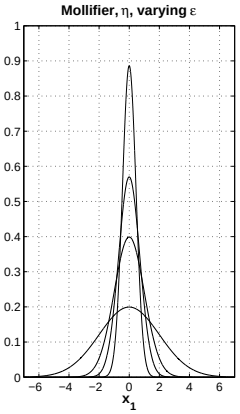
(a) Orientation 1

(b) Orientation 2

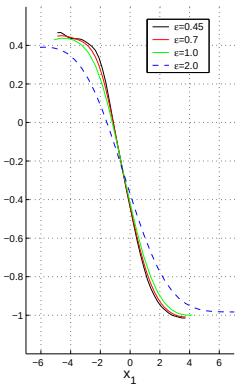
(c) Orientation 1

(d) Orientation 2

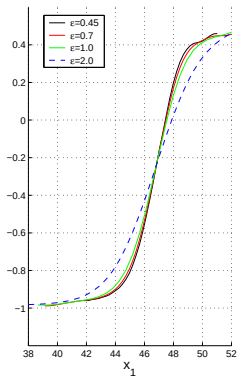
Figure 24: Computed density, ρ_{loc}, and potential energy phase fields for simulations O1 and O2 using $\epsilon = 0.70$.



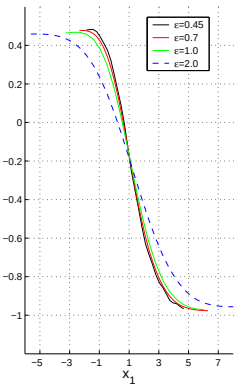
(a) Mollifier, η



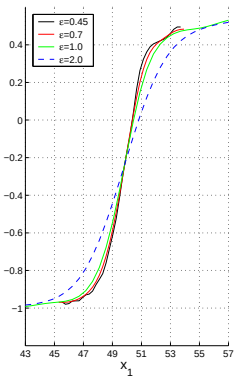
(b) Orientation 1



(c) Orientation 1

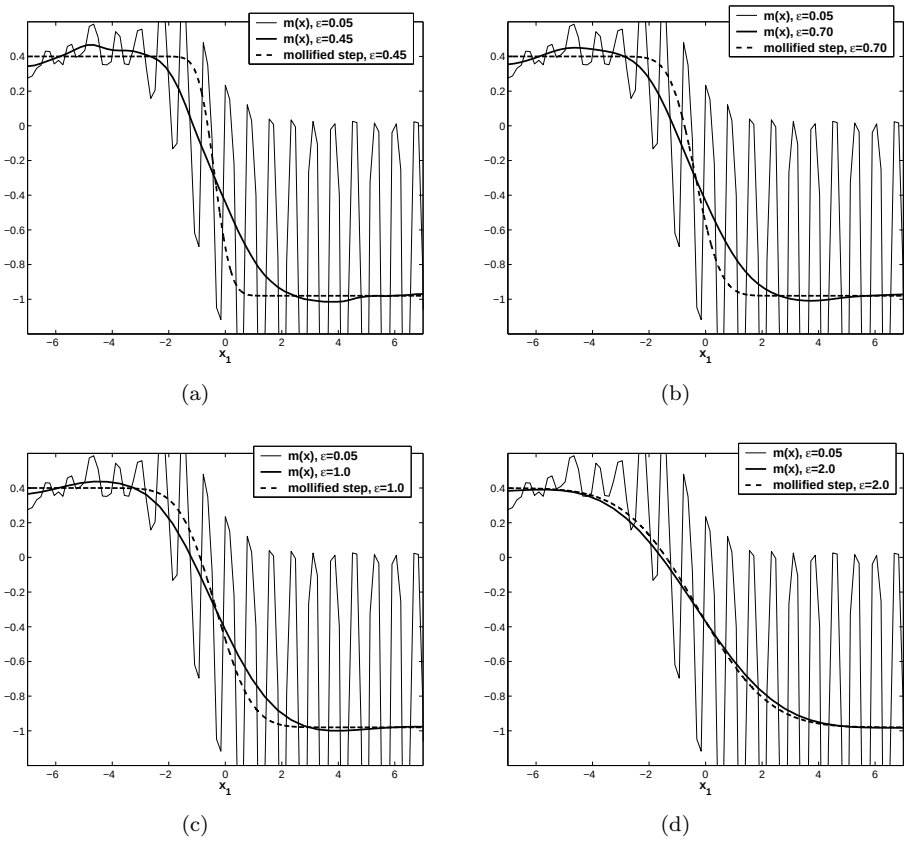


(d) Orientation 2



(e) Orientation 2

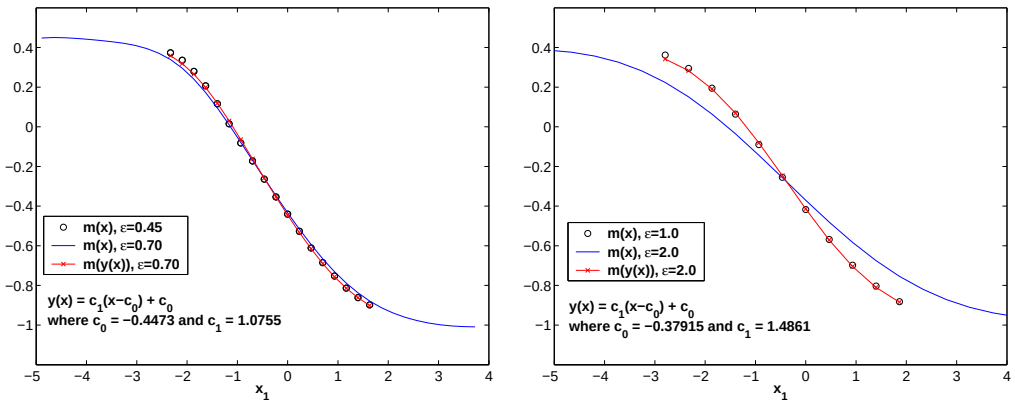
Figure 25: The mollifier, η, in the definition of the phase field, m, depends on the model parameter ϵ. The width of the averaging is proportional to ϵ, as illustrated in (a) which shows η for $\epsilon = 0.45, 0.7, 1.0, 2.0$.
The phase field, m, in the interface regions has been computed from 174 configurations with the four ϵ-values listed above. In (b) and (c) the configurations are taken from simulation O1, and in (d) and (e) from simulation O2. In each case the time interval between two successive configurations is $2.5 \cdot 10^{-3}$, corresponding to $5 \cdot 10^3$ time steps. Though the interface width in the computed phase-fields varies with ϵ, it is not proportional to ϵ in this range.



(a) (b)

(c) (d)

Figure 26: For the phase field based on local contributions to the potential energy the transition from solid to liquid occurs on a length scale of at least several nearest neighbour distances for any choice of the smoothing parameter ϵ.
The four subfigures are based on the same configurations from simulation O1 as were Figure 25(b)–25(c). The oscillating curve present in all subfigures is the computed phase-field, m, using $\epsilon = 0.05$ with a cutoff of η at 0.3. The nearest neighbour distance is approximately 1 and, for the present orientation of the FCC structure with respect to the x_1-axis, the x_1-distance between the atomic layers becomes approximately $1/\sqrt{2}$. Since the cutoff is less than half the distance between the atomic layers the phase-field would be exactly zero at the middle distance if the crystal were perfect and it is very close to zero here. The transition from the stable oscillation pattern in the solid to diminishing oscillations around the mean in the liquid is extended over a distance corresponding to at least four or five atomic layers in the solid.
The phase-field, m, for $\epsilon = 0.45$, 0.70, 1.0, and 2.0 is shown as the heavy solid curve in subfigures (a)–(d). For reference the convolutions $\int_{-\infty}^{\infty} f(y)\eta(x-y)\,dy$ of a sharp interface, given by the step function $f(y) = m_{\text{liq}}\mathbf{1}_{\mathbb{R}^-}(y) - m_{\text{FCC}}\mathbf{1}_{\mathbb{R}^+}(y)$, and the mollifier using the respective ϵ-value is included as the heavy dashed curve. For the smaller ϵ-values the mollified step function is significantly sharper than the corresponding phase-field.



(a) (b)

		approximating ϵ		
		0.70	1.0	2.0
reference ϵ	0.45	1.09 (1.56)	1.23 (2.22)	1.82 (4.44)
	0.70		1.13 (1.43)	1.68 (2.86)
	1.0			1.49 (2.00)

(c) Rescaling factors – the accuracy is approximately ± 0.05.

Figure 27: For an interface given by the convolution of a sharp step function and the mollifier, as in Figure 26, the interface width is directly proportional to ϵ, since interface profiles corresponding to different ϵ are identical up to affine coordinate transformations around the interface, x_{if}, that is: $\phi_{\epsilon_2}(\frac{\epsilon_2}{\epsilon_1}(x - x_{\text{if}}) + x_{\text{if}}) = \phi_{\epsilon_1}(x)$. On a sufficiently large scale the same scaling of the interface width can be expected from the phase-field m obtained from MD simulations. This is not the case when ϵ is of the order of the nearest neighbour distance; then the interface width grows more slowly than ϵ. One way to quantify this statement is to consider the tabulated phase-field, m_{ϵ_1}, using the parameter value ϵ_1, as given data to be approximated by the phase-field, m_{ϵ_2}, based on the parameter value ϵ_2; the allowed approximations use affine coordinate transformations $y(x) = c_1(x - c_0) + c_0$ of the independent coordinate. The data points $((x_k), m_{\epsilon_1}(x_k))$ are taken from the interior of an interface, $m_{\text{solid}} < m_0 \leq m_{\epsilon_1}(x_k) \leq m_1 < m_{\text{liquid}}$, and the function m_{ϵ_2} is defined by linear interpolation between tabulated values. A least squares approximation of the overdetermined system $m_{\epsilon_2}(y(x_k)) = m_{\epsilon_1}(x_k)$ for c_0 and c_1 gives a value of the scaling factor c_1 to be compared to ϵ_2/ϵ_1.
Subfigures (a) and (b) show two examples for the interface in Figure 25(b). The circles, $\circ$, denote the reference data points, the solid line shows the linear interpolation of the tabulated values for the approximating phase-field, and the line marked with crosses, $\times$, is the least square approximation.
The table (c) shows the scaling constants obtained after averaging over all four interfaces in Figure 25(b)–25(e). The corresponding quotients $\frac{\epsilon_2}{\epsilon_1}$ are included in parenthesis for reference.

A Explicit Calculation of Drift and Diffusion Functions

Let the total potential energy be

$$U(X^t) = \sum_{i=1}^{N} m_i(X),$$

where

$$m_i(X) = \frac{1}{2} \sum_{k \neq i, k=1}^{N} \Phi(||X_i - X_k||).$$

For the phase-field

$$m(x; X) = \sum_{i=1}^{N} m_i(X) \eta(x - X_i),$$

where the particle positions $X \in \mathbb{R}^{3N}$ solve the Itô stochastic differential equation

$$dX^t = -\nabla_X U(X^t)\, dt + \sqrt{2k_{\mathrm{B}}T}\, dW^t,$$

Itô's formula gives

$$dm(x; X^t) = \sum_{j=1}^{N} \alpha_j(x; X^t)\, dt + \sum_{j=1}^{N} \sum_{k=1}^{3} \beta_{j,k}(x; X^t)\, dW_{j,k}^t,$$

with

$$\alpha_j(x; X) = -\nabla_{X_j} m(x; X) \cdot \nabla_{X_j} U(X) + k_{\mathrm{B}}T \nabla_{X_j} \cdot \nabla_{X_j} m(x; X) \tag{26}$$

and

$$\beta_{j,\cdot}(x; X) = \sqrt{2k_{\mathrm{B}}T} \nabla_{X_j} m(x; X). \tag{27}$$

Introducing the total force, F_j, acting on particle j, and the contributions from individual pairs, f_{ij},

$$F_j(X) = -\nabla_{X_j} U(X) = \sum_{i \neq j, i=1}^{N} f_{ij}(X),$$
$$f_{ij}(X) = \Phi'(||X_i - X_j||) \frac{X_i - X_j}{||X_i - X_j||},$$

the gradient of m_i with respect to the position of particle j is

$$\begin{aligned}
\nabla_{X_j} m_i(X) &= \frac{1}{2} \sum_{k \neq i, k=1}^{N} \nabla_{X_j} \Phi(||X_i - X_k||) \\
&= \delta_{ij} \frac{1}{2} \sum_{k \neq j, k=1}^{N} \nabla_{X_j} \Phi(||X_j - X_k||) + (1 - \delta_{ij}) \frac{1}{2} \nabla_{X_j} \Phi(||X_i - X_j||) \\
&= -\delta_{ij} \frac{1}{2} F_j(X) - (1 - \delta_{ij}) \frac{1}{2} f_{ij}(X),
\end{aligned}$$

where δ_{ij} is the Kronecker delta: $\delta_{ij} = 1, \text{if } i = j, \delta_{ij} = 0, \text{if } i \neq j$. The gradient of the phase-field variable with respect to the position of particle j is

$$\begin{aligned}
\nabla_{X_j} m(x;X) &= m_j(X)\nabla_{X_j}\eta(x-X_j) + \sum_{i=1}^{N}\nabla_{X_j} m_i(X)\eta(x-X_i) \\
&= -m_j(X)\nabla_x\eta(x-X_j) \\
&\quad - \frac{1}{2}\sum_{i=1}^{N}\delta_{ij}F_j(X)\eta(x-X_i) - \frac{1}{2}\sum_{i=1}^{N}(1-\delta_{ij})f_{ij}(X)\eta(x-X_i) \\
&= -\nabla_x(m_j(X)\eta(x-X_j)) \\
&\quad - \frac{1}{2}F_j(X)\eta(x-X_j) - \frac{1}{2}\sum_{i\neq j,i=1}^{N} f_{ij}(X)\eta(x-X_i).
\end{aligned}$$

Introducing the notation $-G_j$ for the divergence of the force F_j with respect to X_j and the notation g_{ij} for the individual contributions,

$$\begin{aligned}
G_j(X) &= -\nabla_{X_j}\cdot F_j(X) = -\sum_{i\neq j,i=1}^{N}\nabla_{X_j}\cdot f_{ij}(X) = \sum_{i\neq j,i=1}^{N} g_{ij}(X), \\
g_{ij}(X) &= \Phi''(||X_i - X_j||) + \Phi'(||X_i - X_j||)\frac{2}{||X_i - X_j||},
\end{aligned}$$

the divergence of gradient of phase field variable with respect to the position of particle j becomes

$$\begin{aligned}
\nabla_{X_j}\cdot\nabla_{X_j} m(x;X) &= -\nabla_{X_j}\cdot\big(m_j(X)\nabla_x\eta(x-X_j)\big) \\
&\quad - \frac{1}{2}\nabla_{X_j}\cdot\big(F_j(X)\eta(x-X_j)\big) - \frac{1}{2}\sum_{i\neq j,i=1}^{N}\nabla_{X_j}\cdot(f_{ij}(X)\eta(x-X_i)) \\
&= -\nabla_{X_j} m_j(X)\cdot\nabla_x\eta(x-X_j) - m_j(X)\nabla_{X_j}\cdot\nabla_x\eta(x-X_j) \\
&\quad - \frac{1}{2}\nabla_{X_j}\cdot F_j(X)\eta(x-X_j) - \frac{1}{2}F_j(X)\cdot\nabla_{X_j}\eta(x-X_j) \\
&\quad - \frac{1}{2}\sum_{i\neq j,i=1}^{N}\nabla_{X_j}\cdot f_{ij}(X)\eta(x-X_i) \\
&= \frac{1}{2}F_j(X)\cdot\nabla_x\eta(x-X_j) + m_j(X)\nabla_x\cdot\nabla_x\eta(x-X_j) \\
&\quad + \frac{1}{2}G_j(X)\eta(x-X_j) + \frac{1}{2}F_j(X)\cdot\nabla_x\eta(x-X_j) \\
&\quad + \frac{1}{2}\sum_{i\neq j,i=1}^{N} g_{ij}(X)\eta(x-X_i) \\
&= \nabla_x\cdot\nabla_x\big(m_j(X)\eta(x-X_j)\big) + \nabla_x\cdot\big(F_j(X)\eta(x-X_j)\big) \\
&\quad + \frac{1}{2}G_j(X)\eta(x-X_j) + \frac{1}{2}\sum_{i\neq j,i=1}^{N} g_{ij}(X)\eta(x-X_i).
\end{aligned}$$

Using the explicit expressions for $\nabla_{X_j} m(x;X)$ and $\nabla_{X_j}\cdot\nabla_{X_j} m(x;X)$, the components (26)

of the drift become

$$\begin{aligned}
\alpha_j(x;X) &= \nabla_x(m_j(X)\eta(x-X_j))\cdot(-F_j(X)) + \frac{1}{2}F_j(X)\eta(x-X_j)\cdot(-F_j(X)) \\
&\quad + \frac{1}{2}\sum_{i\neq j,i=1}^{N} f_{ij}(X)\eta(x-X_i)\cdot(-F_j(X)) \\
&\quad + k_{\mathrm{B}}T\nabla_{X_j}\cdot\nabla_{X_j}m(x;X) \\
&= -\nabla_x\cdot(m_j(X)F_j(X)\eta(x-X_j)) - \frac{1}{2}||F_j(X)||^2\eta(x-X_j) \\
&\quad - \frac{1}{2}\sum_{i\neq j,i=1}^{N} f_{ij}(X)\cdot F_j(X)\eta(x-X_i) \\
&\quad + k_{\mathrm{B}}T\nabla_x\cdot\nabla_x\big(m_j(X)\eta(x-X_j)\big) + k_{\mathrm{B}}T\nabla_x\cdot\big(F_j(X)\eta(x-X_j)\big) \\
&\quad + k_{\mathrm{B}}T\frac{1}{2}G_j(X)\eta(x-X_j) + k_{\mathrm{B}}T\frac{1}{2}\sum_{i\neq j,i=1}^{N} g_{ij}(X)\eta(x-X_i). \\
&= k_{\mathrm{B}}T\nabla_x\cdot\nabla_x\big(m_j(X)\eta(x-X_j)\big) \\
&\quad + \nabla_x\cdot\Big((k_{\mathrm{B}}T - m_j(X))F_j(X)\eta(x-X_j)\Big) \\
&\quad + \frac{1}{2}\big(k_{\mathrm{B}}TG_j(X) - ||F_j(X)||^2\big)\eta(x-X_j) \\
&\quad + \frac{1}{2}\sum_{i\neq j,i=1}^{N}\big(k_{\mathrm{B}}Tg_{ij}(X) - f_{ij}(X)\cdot F_j(X)\big)\eta(x-X_i)
\end{aligned}$$

so that, after summing over j,

$$\alpha(x;X) = k_{\mathrm{B}}T\nabla_x\cdot\nabla_x m(x;X) + \nabla_x\cdot\widetilde{a_1}(x;X) + a_0(x;X)$$

with

$$\widetilde{a_1}(x;X) = \sum_{j=1}^{N}(k_{\mathrm{B}}T - m_j(X))F_j(X)\eta(x-X_j)$$

and

$$\begin{aligned}
a_0(x;X) &= \sum_{j=1}^{N}\left(k_{\mathrm{B}}TG_j(X) - \frac{1}{2}||F_j(X)||^2\right)\eta(x-X_j) \\
&\quad - \frac{1}{2}\sum_{j=1}^{N}\sum_{i\neq j,i=1}^{N} f_{ij}(X)\cdot F_j(X)\eta(x-X_i).
\end{aligned}$$

Using the one-dimensional mollifier

$$\eta(x) = \eta(x_1) = \text{constant}\cdot\exp\left(-\frac{1}{2}\left(\frac{x_1}{\epsilon}\right)^2\right), \tag{28}$$

that only varies in the x_1-direction, the expression for the drift reduces to

$$\alpha(x;X) = k_{\mathrm{B}}T\frac{\partial^2}{\partial x_1^2}m(x;X) + \frac{\partial}{\partial x_1}a_1(x;X) + a_0(x;X)$$

with

$$a_1(x;X) = \sum_{j=1}^{N}(k_{\mathrm{B}}T - m_j(X))[F_j(X)]_1\eta(x - X_j),$$

where $[F_j(X)]_1$ is the x_1 component of $F_j(X)$.

For the purpose of computing an approximation of

$$\frac{1}{\mathcal{T}}\mathrm{E}\left[\int_1^{\mathcal{T}}\sum_{j=1}^{N}\sum_{k=1}^{3}\beta_{j,k}\otimes\beta_{j,k}\right]$$

it is not practical to postpone the differentiation of the mollifier with respect to the space varible. Using the choice (28), the gradient of the mollifier can be expressed in terms of the mollifier itself as

$$\nabla_x\eta(x - X_j) = \frac{-1}{\epsilon^2}\eta(x - X_j)\left(\left[x - X_j\right]_1, 0, 0\right)^T.$$

Then the expression for $\nabla_{X_j}m(x;X)$ becomes

$$\begin{aligned}\nabla_{X_j}m(x;X) = &\left(\frac{m_j(X)}{\epsilon^2}\left(\left[x - X_j\right]_1, 0, 0\right)^T - \frac{1}{2}F_j(X)\right)\eta(x - X_j)\\ &- \frac{1}{2}\sum_{i\neq j,i=1}^{N} f_{ij}(X)\eta(x - X_i)\end{aligned}$$

and, using the diffusion component (27),

$$\sum_{k=1}^{3}\beta_{j,k}(x;X)\beta_{j,k}(y;X) = 2k_{\mathrm{B}}T\Big(p_j(x,y;X) + q_j(x,y;X)\Big),$$

where

$$\begin{aligned}p_j(x,y;X) = &\left(\frac{m_j(X)}{\epsilon^2}\right)^2\left[x - X_j\right]_1\left[y - X_j\right]_1\eta(x - X_j)\eta(y - X_j)\\ &- \frac{m_j(X)}{2\epsilon^2}[x - X_j]_1\eta(x - X_j)\Big([F_j(X)]_1\eta(y - X_j) + \sum_{i\neq j,i=1}^{N}[f_{ij}(X)]_1\eta(y - X_i)\Big)\\ &- \frac{m_j(X)}{2\epsilon^2}[y - X_j]_1\eta(y - X_j)\Big([F_j(X)]_1\eta(x - X_j) + \sum_{i\neq j,i=1}^{N}[f_{ij}(X)]_1\eta(x - X_i)\Big)\end{aligned}$$

and

$$\begin{aligned}q_j(x,y;X) = &\frac{1}{4}\Big(F_j(X)\eta(x - X_j) + \sum_{i\neq j,i=1}^{N} f_{ij}(X)\eta(x - X_i)\Big)\\ &\cdot\Big(F_j(X)\eta(y - X_j) + \sum_{i\neq j,i=1}^{N} f_{ij}(X)\eta(y - X_i)\Big).\end{aligned}$$

References

[1] A. B. Belonoshko, O. LeBacq, R. Ahuja, and B. Johansson, *Molecular dynamics study of phase transitions in Xe*, J. Chem. Phys. **117** (2002), no. 15, 7233–7244.

[2] W. J. Boettinger, J. A. Warren, C. Beckermann, and A. Karma, *Phase-Field Simulation of Solidification*, Annu. Rev. Mater. Res. **32** (2002), 163–194.

[3] E. Cancès, F. Legoll, and G. Stoltz, *Theoretical and Numerical Comparison of Some Sampling Methods for Molecular Dynamics*, Preprint IMA 2040 (2005).

[4] M. Dzugutov, `mik@pdc.kth.se`

[5] A. J. Majda and P. R. Kramer, *Stochastic Mode Reduction for Particle-Based Simulation Methods for Complex Microfluid Systems*, SIAM Journal on Applied Mathematics, **64** (2004), no. 2, 401–422.

[6] G. Marsaglia and W. W. Tsang, *The ziggurat method for generating random variables*, J. Statist. Software, **5** (2000), no. 8, 1–7.

[7] Netlib is a collection of mathematical software, papers, and databases. The Netlib collection of pseudo random number generators is accessible from `http://www.netlib.org/random/`.

[8] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Applied Mathematical Sciences 153, Springer–Verlag, New York, 2003.

[9] K. Refson, *Moldy: a portable molecular dynamics simulation program for serial and parallel computers*, Comput. Phys. Commun., **126** (2000), no. 3, 310–329.

[10] K. Refson, MOLDY, Release 2.16, 2004, a general-purpose molecular dynamics code. Available free at `http://www.ccp5.ac.uk/librar.shtml`

[11] M. Ross, *The repulsive forces in dense argon* J. Chem. Phys. **73** (1980), no. 9, 4445–4450.

[12] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*, Cambridge University Press, Cambridge, 1999.

[13] S. I. Simdyankin and M. Dzugutov, *Case Study: Computational Physics – The Molecular Dynamics Method*, Technical Report, TRITA-PDC-2003:1, ISSN 1401-2731, Royal Institute of Technology, Stockholm, 2003.

[14] A. Szepessy, *Atomistic and Continuum Models for Phase Change Dynamics*, pp. 1563–1582 in *Proceedings of the International Congress of Mathematicians Madrid, August 22–30, 2006, Volume III*, 2007, EMS Ph.

[15] K. V. Tretiakov and S. Scandolo, *Thermal conductivity of solid argon at high pressure and high temperature: A molecular dynamics study*, J. Chem. Phys. **121** (2004), no. 22, 11177–11182.