A PARTICLE–PARTITION OF UNITY METHOD FOR THE SOLUTION OF ELLIPTIC, PARABOLIC AND HYPERBOLIC PDES

MICHAEL GRIEBEL† AND MARC ALEXANDER SCHWEITZER†

Abstract. In this paper, we present a meshless discretization technique for instationary convection-diffusion problems. It is based on operator splitting, the method of characteristics and a generalized partition of unity method. We focus on the discretization process and its quality. The method may be used as an h- or p-version. Even for general particle distributions, the convergence behavior of the different versions corresponds to that of the respective version of the finite element method on a uniform grid. We discuss the implementational aspects of the proposed method. Furthermore, we present the results of numerical examples, where we considered instationary convection-diffusion, instationary diffusion, linear advection and elliptic problems.

Key words. meshless methods, gridless discretizations, particle methods, Galerkin methods, partition of unity methods, Lagrange multipliers, h-version, p-version, advection, convection-diffusion

AMS subject classifications. 65M60, 65M25, 65C05, 76N99, 65N30, 65N99

1. Introduction. The discretization of partial differential equations is usually obtained by covering the domain of interest by a suitable grid. Then, after time discretization, linearization and space discretization (by finite elements, finite differences or finite volumes), a linear system of discrete equations is set up and solved. Here, beside uniform structured grids which are most simple to handle, also block structured grids and unstructured grids are employed successfully. Furthermore, for dealing with non-smooth data or solutions, adaptive refinement is a must for efficiency reasons, see [7, 8, 9, 10, 13, 22, 23, 39, 64].

However, all grid- or mesh-based methods are quite involved when it comes to time-dependent problems with complicated geometries, free moving boundaries and interfaces. Then the geometry of the domain may change over time, the non-smooth part of the data or the non-smooth part of the solution changes with time or the location of singularities of the solution may vary time-dependently. Examples for such problems are crack propagation in elasticity theory [12], free surface flow or multiphase flow in fluid dynamics [65] or accretion disks, jets and cloud simulations in magneto-hydrodynamics [20].

In the Eulerian approach, an adaptive mesh technique (h-version, hp-version) must follow the time-dependent features of the data or solution by local refinement and coarsening of the mesh [7, 22, 39, 46, 64]. But time-dependent adaptive mesh refinement and coarsening is not simple, especially for 3D problems. It is quite involved, programming is complicated, data structures are not easy to handle and the storage overhead is significant. Besides, good local and global error estimators are necessary. Therefore, there exist only a few unstructured adaptive programs which are able to handle three-dimensional application-oriented problems with time-dependent change of the geometry, the data or the solution.

The Lagrangian viewpoint allows the mesh itself to be moved (r-method) [6, 53, 54]. But an implementation is still cumbersome, since the mesh may become tangled and twisted, elements may collapse or angles of some elements might degenerate over time due to the movement of the nodes. The proper treatment of these problems is

†Sonderforschungsbereich 256 *Nonlinear Partial Differential Equations*, Project D *Meshless numerical methods for the simulation of 3D flows with free boundaries*, Institut für Angewandte Mathematik, Universität Bonn, Wegelerstr. 6, D-53115 Bonn, {griebel, schweitz}@iam.uni-bonn.de

not an easy task, especially in 3D applications. At least for the 2D case, the so-called monotone mesh method [65] can cope with these problems to some extent.

Besides, in real life engineering applications, a very time consuming portion of the overall computation is the mesh generation from CAD input data. Typically more than 70 percent of the overall computing time is spent by mesh generators.

Hence, especially within the engineering community, there is growing interest in other discretization methods which involve no mesh at all. These approaches are summarized under the term meshless or gridless methods. The main idea is to consider points only, i.e. we omit any fixed relation between the nodes such as element boundaries, and to move just these points in a time-dependent setting. Here, the location of the points and the distribution of the points account for the description of the changing geometry, the change in data and the time-dependent changing variation of the solution or its gradient.

Generally speaking, there are two different types of meshless approaches; first, the classical particle methods [57, 58, 60, 61] and second, gridless discretizations based on data fitting techniques [11, 26]. Traditional particle methods stem from physics applications like Boltzmann equations [1, 33]. They are truly Lagrangian methods, i.e. they are based upon a time-dependent formulation or conservation law. In a particle method we use a discrete set of points to discretize the domain of interest and the solution at a certain time. The PDE is transformed into equations of motion for the discrete set of particles such that the particles can be moved via these equations. After time discretization of the equations of motion we obtain a certain particle distribution for every time step. Therefore, we get an approximate solution to the PDE via the definition of a density function for these particle distributions. These methods are easy to implement. However, they exhibit in general relatively poor convergence properties in weak norms.

The so-called gridless methods follow a different approach. Here, patches or volumina are attached to each point whose union form an open covering of the domain. Then, local shape functions are constructed with the help of methods from data fitting. These shape functions are used in a Galerkin or collocation discretization process to set up a linear system of equations. Finally this system must be solved efficiently. In contrast to particle methods, such gridless discretizations may also be applied to stationary and elliptic problems. According to the data fitting method involved we can distinguish basically the following three approaches: Shepard's method [69] which has a consistency of first order only, the moving least squares method (MLSM) [44, 45] which generalizes Shepard's approach implicitly to the case of higher order shape functions, and the partition of unity p-version method which generalizes Shepard's approach explicitly to higher consistency orders. Meanwhile, different realizations of these approaches exist. First, there is the smoothed particle hydrodynamics (SPH) technique of Lucy and Monaghan [31, 32, 51, 55, 56, 71] which resembles (up to area weighted scaling) Shepard's method. Then, Duarte and Oden [26, 27] used in their hp-cloud approach the moving least squares (MLS) idea. Belytschko and coworkers [11, 12] apply similar techniques based on the moving least squares approach to engineering problems. Furthermore, Dilts [24] used the MLS technique to extend the SPH method to the so-called moving least squares particle hydrodynamics (MLSPH) method. Babuška and Melenk [3, 4] proposed the so-called partition of unity method (PUM) which is mainly applied to uniform point distributions up to now. Liu et al. [50] proposed variants of the SPH method based on the idea of reproducing kernels of higher order and wavelets. There exist also generalizations of the finite difference approach to the gridless setting [49]. Furthermore, Kansa [40, 41], Schaback and Franke [28, 29] and Wendland [73] used the radial basis approach from approximation theory to construct meshless methods for the discretization of PDEs. The mass-packet method of Yserantant [74, 75] is somewhat different from the classical particle methods. Here, the particles are not considered in the sense of statistical mechanics but they are understood as comparatively big mass-packets, the conservation of mass is automatically guaranteed by this ansatz. For an overview on meshless methods see [76] and the references therein.

All these data fitting approaches do not depend (at least to a great extent) upon a mesh or any fixed relation between gridpoints (particles). However the realization and implementation of such a method is not so simple in general: There are often problems with stability and consistency. Furthermore, in a Galerkin method, the discretization of the differential operator, i.e. the integration of the stiffness matrix entries, is in general quite involved in comparison with the conventional grid-based approach. Another challenging task is the discrete formulation of Dirichlet boundary conditions, since the constructed shape functions are in general non-interpolatory. Nevertheless, the different variants of gridless methods are interesting from both the practical and the theoretical point of view. These methods, which are up to now merely in an experimental premature state, possess some potential and might have an interesting future.

The meshless method we propose in this paper is based on the Lagrangian approach for time-dependent problems used within classical particle methods and the ideas of Babuška and Melenk [3, 4], but allows for a random point distribution. We consider time-dependent convection-diffusion problems, which we decompose into a parabolic and a hyperbolic subproblem using a simple operator splitting. The hyperbolic problem is discretized via a Lagrange–type particle method, i.e. we define equations of motion for the particles corresponding to the problem and move the particles accordingly. We discretize the remaining parabolic subproblem with a two step Particle–Galerkin method. In the first step we move the particles $\{x_i\}$ via a Monte Carlo step to adapt the particle distribution to the diffusion process. Then, an implicit Eulerian time discretization reduces the parabolic problem to an elliptic problem in every time step. For its discretization in space we employ the PUM approach to randomly distributed points $\{x_i\}$. Here, following ideas from scattered data approximation [21, 69], we set up a partition of unity $\{\varphi_i\}$ over the domain Ω using a localized version of Shepard's method. Since a partition of unity achieves first order consistency only, we need to improve on the approximation quality of the constructed space $V = \text{span}\,(\{\varphi_i\})$ to allow its use in a collocation or Galerkin method. Thus, we expand the functions φ_i by multiplication with a local polynomial ψ^i of degree p_i, defined on $\text{supp}(\varphi_i)$, and define the space $V = \text{span}\,(\{\varphi_i \psi^i\})$. We use these functions as trial and test functions in a Galerkin method. Altogether we obtain an analogue of the h-, p- and hp-version of the finite element method for general particle distributions. Furthermore, the results of our numerical experiments show that the convergence properties of the proposed method are similar to those of the finite element method.

The remainder of this paper is organized as follows: In §2, we develop a Lagrangian discretization of the instationary convection-diffusion problem using a simple operator splitting. The resulting hyperbolic subproblem is discretized via a particle method, which is presented in §3. For the remaining parabolic problem we use a simple time stepping technique. It reduces the parabolic problem to a sequence of elliptic problems. In §4, we discuss their discretization using a generalized Particle–Galerkin method based on the PUM. Then, in §5, we present the results of our numerical experiments. Finally, we give some concluding remarks.

2. An Operator Splitting Method for instationary Convection-Diffusion Problems. A widely used approach for the discretization of instationary convection-diffusion problems

$$\begin{array}{rcll} \nabla A \nabla u + v \nabla u & = & u_t & \text{for all } (x,t) \in \Omega \subset \mathbb{R}^d \times (0,T), \\ u(x,0) & = & u_0(x) & \text{for all } x \in \Omega \subset \mathbb{R}^d, \\ Bu(x,t) & = & g(x,t) & \text{for all } (x,t) \in \partial\Omega \times (0,T) \end{array} \tag{2.1}$$

is the operator splitting method [70]. This method can be summarized as follows: Let $S(t)$ be the solution operator for the hyperbolic problem

$$v \nabla u^A = u_t^A, \quad u^A(x,0) = u_0^A(x), \tag{2.2}$$

such that $u^A(x,t) = S(t) u_0^A(x)$ is the entropy solution to (2.2), and let $H(t)$ be the solution operator for the parabolic problem

$$\nabla A \nabla u^D = u_t^D, \quad u^D(x,0) = u_0^D(x), \quad Bu^D(x,t) = g(x,t)\,|_{\partial\Omega}, \tag{2.3}$$

such that $u^D(x,t) = H(t) u_0^D(x)$ is the solution to (2.3). A combination of these two continuous solution operators is used as an approximative solution operator for (2.1). This continuous solution operator is evaluated at certain times, but is still kept continuous in space, i.e.

$$u_n(x) = u(x, n\Delta t) \approx [H(\Delta t) S(\Delta t)]^n u_0(x), \tag{2.4}$$

is the semi-discrete approximative solution to (2.1) at time $n\Delta t$. Based on this principle, different variants of the method have been developed, cf. [37, 38, 48]. The splitting (2.4) inherits its approximation properties from $H(t)$ and $S(t)$, depending on the respective time discretization. In this paper we stick to the most straightforward approach (2.4) for reasons of simplicity.

Now the next step in the discretization process of (2.1), is the choice of approximations to the solution operators $S(t)$ and $H(t)$. The hyperbolic problem (2.2) can be solved analytically with the help of the method of characteristics. Therefore, a natural choice for the discretization of (2.2) is a method which can exploit this knowledge about a continuous solution to (2.2). Such a method would be a Lagrangian particle method, since here a characteristic through a given point x_i^A in space may be interpreted as path the point will follow over time. This particle method works as follows: First, we sample the initial value u_0 in appropriate points $\{x_i^A\}$, this gives a set of pairs $\{(x_i^A, u_0^i)\}$. Then, we construct the characteristic through every particle x_i^A at time t and move each particle along its corresponding characteristic to its new position at time $t + \Delta t$. Finally, we define an approximate solution $\tilde{u}$ at time $t + \Delta t$ using shape functions that are constructed with respect to the particle distribution. This process is covered in more detail in §3.

Now, the discretization of the parabolic problem (2.3) should be compatible to the discretization of (2.2). Since we use a Lagrangian particle method for (2.2), we need a discretization method for (2.3) which can handle (almost) random distributions of nodes (particles). Moreover we want to treat (2.3) also from the Lagrangian point of view. In a pure particle method a parabolic problem like (2.3) is discretized using a

Monte Carlo method. Here the shape functions themselves—not only the particles—are moved across the domain without being changed which causes the poor accuracy of these methods. In our approach we also move the particles. The shape functions though are not only moved across the domain. We construct new shape functions in every time step based on the new particle distribution.

We move the particles across the computational domain to simulate the effect of the diffusive transport of (2.3) on the shape functions and approximation space which adapts the distribution of the spatial degrees of freedom only. An approximate solution of (2.3) is then computed via a variational approach with the newly generated approximation space. Therefore the proposed method inherits the advantages of particle methods in adapting the particle distribution via a Monte Carlo step but does not inherit the poor approximation quality of a Monte Carlo method.

Overall we propose the following three-step approximation of (2.3):

1. Time discretization of (2.3) and $u(x,t)$:

$$\begin{aligned} (0,T) &\rightarrow (t_0,\dots,t_n), \quad t_k = k\Delta t \\ u(x,t) &\rightarrow u(x,t_k) = u^k(x) \\ \nabla A \nabla u^D = u^D_t &\rightarrow -\Delta t \nabla A \nabla u^D_{m+1} + u^D_{m+1} = u^D_m. \end{aligned}$$

2. Use a Monte Carlo step to translate the diffusion equation (2.3) into equations of motion for the particles $\{x_i^A\}$, i.e. use a random walk process

$$x_i^A \rightarrow x_i^A + \sqrt{2\Delta t} A^{\frac{1}{2}} \xi = x_i^D,$$

where ξ are independent random numbers with Gaussian distribution, vanishing mean and variance one.

3. Now that the particle distribution $\{x_i^D\}$ is suitable for the approximation of the solution of (2.3) in the next time step we approximate the remaining elliptic problem

$$-\Delta t \nabla A \nabla u^D_{m+1} + u^D_{m+1} = u^D_m \tag{2.5}$$

using a spatial discretization method which allows the use of a random node arrangement. Here, a gridless method seems to be the most natural choice.

In §4 we present a generalized partition of unity method for elliptic problems, give a short discussion on the implementational difficulties—which are often neglected in other papers—and give some ideas on how to overcome these difficulties.

3. A Lagrangian Discretization of Hyperbolic Problems using a Particle Method. The main problem of an Eulerian discretization is the following: The initial mesh may be adapted to the initial data but may not be appropriate for the resolution of the solution in future time steps. Therefore we have to use expensive coarsening and refinement strategies—or even complete re-meshing—to adapt the mesh to the solution over time. In contrast to that, a Lagrangian discretization uses the given PDE itself to define a transformation that maps the distribution of the spatial degrees of freedom to appropriate positions over time. Hence, the degrees of freedom follow the solution over time.

We consider the two-dimensional transport problem

$$\frac{\partial u}{\partial t}(p,q,t) + a(p,q,t,u)\nabla_{(p,q)} u(p,q,t) = 0 \tag{3.1}$$

in $\Omega \times (0,T)$ with $\Omega \subset \mathbb{R}^2$ and the general transformation

$$T : (\xi, \eta, \tau) \mapsto (p, q, t) = (p(\xi, \eta, \tau), q(\xi, \eta, \tau), t(\xi, \eta, \tau)).$$

Now we suppose that the function $u(p,q,t)$ is carried into the function $\hat{u}(\xi, \eta, \tau)$ under the transformation T, i.e.

$$u(p, q, t) = u \circ T(\xi, \eta, \tau) = \hat{u}(\xi, \eta, \tau). \tag{3.2}$$

Using the chain rule, we obtain the τ-derivative of $\hat{u}$

$$\begin{aligned} \frac{\partial \hat{u}(\xi,\eta,\tau)}{\partial \tau} = & \frac{\partial u(p,q,t)}{\partial p} \frac{\partial p(\xi,\eta,\tau)}{\partial \tau} + \frac{\partial u(p,q,t)}{\partial q} \frac{\partial q(\xi,\eta,\tau)}{\partial \tau} \\ & + \frac{\partial u(p,q,t)}{\partial t} \frac{\partial t(\xi,\eta,\tau)}{\partial \tau} . \end{aligned} \tag{3.3}$$

If we restrict ourselves to transformations T with $t(\xi, \eta, \tau) = \tau$, we have $\partial t / \partial \tau = 1$. Simple reordering of (3.3) gives

$$\frac{\partial u(p,q,t)}{\partial t} = \frac{\partial \hat{u}(\xi,\eta,\tau)}{\partial \tau} - \frac{\partial u(p,q,t)}{\partial p} \frac{\partial p(\xi,\eta,\tau)}{\partial \tau} - \frac{\partial u(p,q,t)}{\partial q} \frac{\partial q(\xi,\eta,\tau)}{\partial \tau} .$$

Now we plug this relation into (3.1) and obtain the so-called Lagrangian form [6] of our PDE (3.1)

$$\frac{\partial \hat{u}}{\partial \tau} + \frac{\partial u}{\partial p} \left(a_1 - \frac{\partial p}{\partial \tau} \right) + \frac{\partial u}{\partial q} \left(a_2 - \frac{\partial q}{\partial \tau} \right) = 0 . \tag{3.4}$$

A solution of (3.4) independent of $\nabla_{(p,q)} u$ may now be obtained from the system of ODEs

$$\frac{\partial p(\xi,\eta,\tau)}{\partial \tau} = a_1(p,q,t,u) , \quad \frac{\partial q(\xi,\eta,\tau)}{\partial \tau} = a_2(p,q,t,u) , \quad \text{and} \quad \frac{\partial \hat{u}(\xi,\eta,\tau)}{\partial \tau} = 0 .$$

Note that this is essentially the method of characteristics. The solution $(p, q, \hat{u})$ can now be approximated by a numerical ODE solver like an Euler or Runge–Kutta method, i.e. the transformation T of (3.2) is approximated by a transformation $\tilde{T}$ which is found by numerical integration. The solution to (3.1) can then be approximated by plugging this transformation $\tilde{T}$ into (3.2) which expresses the well known fact that a solution of (3.1) is constant along characteristics, i.e. on the images of T, giving

$$\tilde{u}(p, q, t) = u \circ \tilde{T}(\xi, \eta, \tau) \simeq \hat{u}(\xi, \eta, \tau). \tag{3.5}$$

We can evaluate the solution at the current time by evaluating the solution at a former time at the corresponding position on the characteristic. Therefore, we use a backward integration $I^B_{\Delta t}$ to compute $\tilde{T}$. But the particles $\{x_i^t\}$ are propagated forward in time along the approximated characteristic since they have to provide the spatial resolution of the current solution, i.e. a new particle distribution $\{x_i^{(t+\Delta t)}\}$ is defined in every time step. A spatial approximation $\tilde{u}_S(x, t + \Delta t) = \sum_i u_i^{(t+\Delta t)} \phi_i^{(t+\Delta t)}(x)$ to (3.5) is set up with the use of some shape functions $\{\phi_i^{(t+\Delta t)}\}$ which are centered in these new particle positions $\{x_i^{(t+\Delta t)}\}$. Here, the specific choice of the shape functions

$\{\phi_i^{(t+\Delta t)}\}$ determines the respective particle method. Since we want to combine the particle method for the hyperbolic subproblem (2.2) with our gridless discretization method for the elliptic subproblem (2.5) we use the shape functions employed there, see §4. Again, in a pure particle method the shape functions are fixed over time, i.e. they are attached to a certain particle and never changed during the computation. Hence, the current particle position and overall particle distribution are not taken into account. This is one of the reasons for the poor approximation quality of these methods. In our approach we move the particles and construct new shape functions based on the newly generated particle distribution. Therefore, the shape functions are not fixed over time but rather always adapted to the current particle distribution.

Altogether we obtain a discrete two-step particle scheme:

1. Time discretization:
   - treat non-linearities of a, choose ODE solver(s) $I_{\Delta t}^{F/B}$ and time intervals
$$(0,T) \to (t_0, \dots, t_n), \quad t_k = k\Delta t.$$
2. Spatial discretization:
   - propagate particles $\{x_i^{k-1}\}$ forward along their characteristic
$$x_i^k = I_{\Delta t}^F(a, x_i^{k-1}),$$
   - project the approximate solution $\tilde{u}_S^{k-1} \circ I_{\Delta t}^B(a,\cdot)$ onto new approximation space
$$\tilde{u}_S^k(x) = \sum u_i^k \phi_i^k(x) = \tilde{u}_S^{k-1} \circ I_{\Delta t}^B(a,x) = \sum u_i^{k-1}\phi_i^{k-1} \circ I_{\Delta t}^B(a,x).$$

The shape functions $\{\phi_i^k\}$ are constructed in every time step k according to the new particle positions $\{x_i^k\}$ (see §4), hence the shape functions themselves are propagated along the characteristic. We compute an approximate solution $\tilde{u}_S^k$ by testing a transformed of the solution $\tilde{u}_S^{k-1}$ from the former time step with the new basis functions $\{\phi_i^k\}$, i.e. we have to solve the mass matrix problem for the new shape functions $\{\phi_i^k\}$.

4. Gridless Discretization of Elliptic Problems. In the following, we describe our approach for a gridless discretization of an elliptic problem. The approach is roughly as follows: The discretization is stated in terms of the points only. To obtain test and trial spaces, patches or volumina $\omega_i \subset \mathbb{R}^d$ are attached to each point x_i whose union form an open cover $\{\omega_i\}$ of the domain Ω, i.e. $\Omega \subset \bigcup \omega_i$. Now, from given weight functions W_i, local shape functions are constructed by Shepard's method. They form a partition of unity $\{\varphi_i\}$. Then, each shape function φ_i is multiplied with a sequence of local polynomials $\{\psi_i^k\}$ to gain higher degree shape functions. These are finally plugged into the weak form to set up a linear system of equations.

4.1. Construction of trial and test space using PUM. Necessary conditions for a trial and test space to perform well in a Galerkin method are local approximability and inter-element continuity. Here, local approximability means that the shape functions can approximate the exact solution well locally, and inter-element continuity means that any linear combination of shape functions satisfies some global continuity condition. In the finite element method (FEM) we achieve the local approximability via the choice of local polynomials and fulfill the condition of inter-element continuity by imposing restrictions onto these polynomials on the element edges. The PUM approach [3, 4] instead focuses on the fulfillment of the condition of inter-element continuity via the choice of an appropriate partition of unity (PU) $\{\varphi_i\}$ subordinate to a cover $\{\omega_i\}$. Local expansion of the functions φ_i by the multiplication with local approximation spaces $V_i(\omega_i)$ causes the generated space

$$V := \left\{ \sum_{i=1}^{N} \varphi_i v_i \,|\, v_i \in V_i \right\}$$

to fulfill the condition of local approximability. Theorem 4.3 (see below) states that the global approximation space V inherits the approximation quality of the local spaces V_i. Furthermore the space V inherits the smoothness of the PU (and the local spaces V_i). Here, the approximation property of the space V may either be achieved by the smallness of the patches (h-version) or by the approximation quality of V_i (p-version).

4.1.1. Open covering of the domain. The starting point for any gridless discretization approach is a collection of N points

$$\{x_i \in \mathbb{R}^d : x_i \in \overline{\Omega}, i = 1, \ldots, N\}. \tag{4.1a}$$

Then, to each point x_i, a patch

$$\omega_i = \{x \in \mathbb{R}^d : ||x_i - x|| \leq h_i\} \subset \mathbb{R}^d. \tag{4.1b}$$

is attached. Here, $h_i \in \mathbb{R}$ is the half of the diameter of the patch ω_i. The norm $||\cdot||$ is the Euclidean distance for circles or balls and it is the $||\cdot||_\infty$-norm for quadratic or cube type patches. It is easy to see that this concept can be generalized to patches of more general form: If we allow individual non-uniform sizes $h_i = (h_i^{(1)}, h_i^{(2)}, \ldots, h_i^{(d)})$ in the different coordinate directions and accordingly generalized additive norms $||\cdot||$, we obtain also patches with ellipsoid shape and rectangular or brick type objects[1]. Figure 4.1 shows typical examples of the shapes of a patch ω_i.

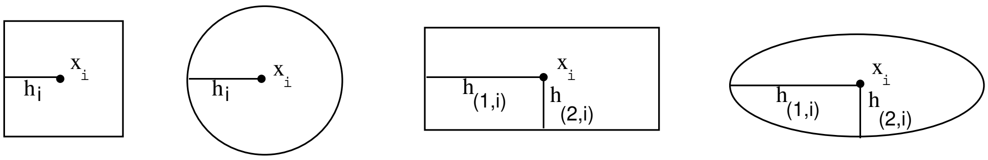

FIGURE 4.1. *Typical examples of the shapes of a patch ω_i.*

The construction of the patches $\{\omega_i\}$ from a given set of grid points $\{x_i\}$ is a first crucial step in the discretization process. Keeping in mind that these patches will be the supports of the trial and test functions in a Galerkin method, the most basic property these patches have to fulfill is that they cover[2] the complete domain $\bigcup_{i=1}^N \omega_i \supset \overline{\Omega}$. In other words, for each point $x \in \overline{\Omega}$ there exists at least one patch ω_i which contains x. Figure 4.2 gives an example of an open covering of the domain Ω. A naive approach towards the construction of such a cover $\{\omega_i\}$ would be the design

[1]We performed experiments using different norms and thus different shapes of the local patches ω_i. It turned out that circles or balls are computationally much more difficult to handle than supports of quadratic or rectangular shape without giving substantial advantage. Therefore, we decided to stick to rectangular shapes in our gridless method.

[2]Other gridless methods like SPH allow for holes in the domain Ω. Methods based on the MLSM [24, 27] have to impose more severe geometric conditions onto the cover $\{\omega_i\}$.

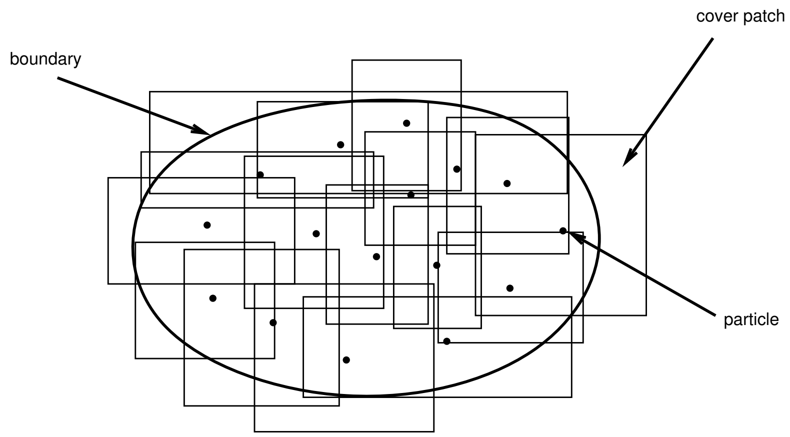

FIGURE 4.2. *Example of an open covering of* Ω.

of patches $\tilde{\omega_i}$ in such a way that every particle $x_i \in \tilde{\omega_j}$ for some $j \neq i$. But this procedure (in general) does not lead to a cover $\{\tilde{\omega_j}\}$ of the complete domain Ω, i.e. $\bigcup_j \tilde{\omega_j} \not\supset \overline{\Omega}$, since the particles $\{x_i\}$ may not be uniformly distributed in the domain Ω. Therefore, we have to use a more sophisticated algorithm. We use the following variant of an algorithm developed in [27] which resolves the problem mentioned above by using a set P of particles x_i and pseudo-particles ξ_k which we choose to be the nodes of a coarse mesh. The pseudo-particles ξ_k are introduced to guarantee that the patches ω_i completely cover the entire domain Ω.

1. For all $i = 1, \ldots, N$: Set ω_i such that $\operatorname{diam}(\omega_i) = 0$.
2. For all $y \in P$:
   - Evaluate the set $S_{y,R}$ of all particles x_i that fall within a searching square B_R which is centered in y and whose side length is equal to $2R$. If $S_{y,R} = \emptyset$ or $S_{y,R} = y$ in the case of $y = x_i$, increase the size of the searching square, i.e. R, and try again.
   - Compute the distances $h_i^y = \|y - x_i\|$ for all $x_i \in S_{y,R}$ with $x_i \neq y$.
   - Determine the particle $x_j \neq y$ with $h_j^y = \min_i h_i^y$.
   - If $y \notin \omega_j$ increase ω_j appropriately such that $y \in \omega_j$ holds.
3. For all $i = 1, \ldots, N$: Set $\operatorname{diam}(\omega_i) = \alpha \operatorname{diam}(\omega_i)$.

Crucial for the efficient implementation of the above algorithm is the evaluation of the sets $S_{y,R}$. Note that a similar search problem appears in the SPH. There an efficient algorithm for such problems was developed in [71]. Other algorithms for such a cover construction can be derived from tree-algorithms, such as quadtrees or AVL-trees [42]. The overall complexity of the above algorithm is $O(\operatorname{card}(P)\log(N))$.

Since the entries $a_{ij} = a(v_j, v_i)$ of the stiffness matrix A will be nonzero only for functions v_i, v_j with $\operatorname{supp}(v_i) \cap \operatorname{supp}(v_j) \neq \emptyset$, we want to control the number of patches overlapping the same point $x \in \Omega$, i.e. the number of overlapping patches $\{\omega_i\}$ should be minimal. The amount of overlap of the patches $\{\omega_i\}$ though has a significant effect on the smoothness of our shape functions. Therefore, the overlap $\omega_i \cap \omega_j$ of two patches ω_i and ω_j should be sufficiently large (if it is not empty). This motivates the stretch parameter α in step 3 of the algorithm above which is used to control the interplay between the smoothness of the shape functions and the density of the stiffness matrix.

4.1.2. Weight functions and partition of unity. Now we associate a certain weight function $W_i(x)$ to each subdomain ω_i. Since we decided to use rectangular patches only, i.e. $\omega_i = \bigotimes_{k=1}^d \omega_i^{(k)}$, $\omega_i^{(k)} = \{x^{(k)} \in \mathbb{R}, |x_i^{(k)} - x^{(k)}| \leq h_i^{(k)}\}$, the most natural choice is to use a product approach of one-dimensional local functions, i.e. $W_i(x) = \bigotimes_{k=1}^d W_i^{(k)}(x^{(k)})$ where $\operatorname{supp}(W_i) = \overline{\omega_i}$. If we use $[-1,1]$ as reference interval and define the affine linear mapping $T_i^{(k)} : [-1,1] \to \omega_i^{(k)}$, $x^{(k)} \to T_i^{(k)}(x^{(k)})$, we can define the respective weight functions as $W_i^{(k)}(x^{(k)}) = W(T_i^{(k)}(x^{(k)}))$ where the one-dimensional weight function W can be any non-negative function. Furthermore, we obtain the mapping T_i from the reference element $[-1,1]^d$ to ω_i as the product $T_i : [-1,1]^d \to \omega_i$, $x \to T_i(x) = \bigotimes_{k=1}^d T_i^{(k)}(x^{(k)})$. The weight functions[3] W we use in this paper are the well known linear, quadratic and cubic B-splines.

Now, in the next step, we construct shape functions φ_i from these given weight functions W_i with the help of data fitting techniques. In general, a data fitting method [36] produces an approximation $\tilde{u}$ of a function u by

$$\tilde{u}(x) = \sum_{i=1}^{N} u_i \cdot \varphi_i(x)$$

where u_i are given data or are derived from that. Shepard's method uses the idea of inverse distance weighting, which leads to shape functions

$$\varphi_i(x) = \frac{W_i(x)}{\sum_{j=1}^N W_j(x)}, \tag{4.2}$$

where $W_i(x) = \|x - x_i\|^{-\beta}$. But since such shape functions have global support, they would lead to a dense stiffness matrix and a quadratic complexity of the method. We therefore use a localized version of Shepard's approach. There are basically two variants: In [45] a locally supported singular weight function such as

$$W_i(x) = L_i(x)\|x - x_i\|^{-\beta}, \quad \text{where } L_i \in \mathcal{C}^\infty \quad \text{and } \operatorname{supp}(L_i) = \omega_i$$

is used. This approach generates an interpolatory partition of unity, i.e. $\varphi_i(x_j) = \delta_{ij}$. Another approach is to employ a locally supported smooth weight function, e.g. W_i is chosen to be a B-spline[4] [36]. We use the latter approach to circumvent the evaluation of a quotient of singular functions close to their singularity during the numerical integration of the stiffness matrix entries.

Since we postulate that the union of the ω_i covers the domain Ω, by (4.2), we are at least able to reproduce constant functions exactly, i.e. the functions (4.2) form a *partition of unity*. Interestingly, this is valid for any choice of weight function, particle distribution and topology of the cover. Thus, we obtain a consistency order[5] of one

[3] Besides these weight functions also the thin-plate splines, Gaussians and especially the so-called SPH-spline [11, 26, 31, 51] are used in other meshless methods. Note that circular patch shapes would allow also for radial functions as weight functions W_i [28, 29, 40, 41, 73].

[4] Note that, together with collocation and area weighting of the resulting shape functions, this gives basically the smoothed particle hydrodynamics method SPH, which was first proposed in [51] and further elaborated in [32, 55, 56, 71].

[5] Consistency orders are usually given as exponents of h, i.e. $\|u - \tilde{u}\| = O(h^\alpha)$ with α being the consistency order. However, this formulation is applicable to uniform node arrangements only. Hence, we have to use a different notation in a meshless method. Taking into account that we have $\|x_i - x_j\| = O(N^{-1/d})$ for equi-distributed particles $\{x_i\}$ and that $h = N^{-1/d}$ for uniform node arrangements, it seems natural to define the consistency order α of a meshless method as $\alpha := (d \log(\|u - \tilde{u}\|)) / \log(N)$.

in the L^2-norm. The particle distribution $\{x_i\}$ and the cover $\{\omega_i\}$ only effect the computational effort necessary to evaluate the functions φ_i, since the sum in (4.2) has to be taken over all patches ω_j with $\omega_i \cap \omega_j \neq \emptyset$. Furthermore, the amount of overlap of the cover patches and the choice of weight functions W_i effect the smoothness of the Shepard functions φ_i. On the one hand, the partition of unity inherits the smoothness of the weight functions W_i since $\Big(W_i \in \mathcal{C}^k(\mathbb{R}^d) \quad \wedge \quad \forall\, x\ \sum_i W_i(x) \neq 0\Big) \Rightarrow \varphi_i \in \mathcal{C}^k(\mathbb{R}^d)$ holds. On the other hand, the amount of overlap affects the smoothness of φ_i. If the cover is minimal, i.e. there is exactly one patch ω_j for every $x \in \overline{\Omega}$ with $x \in \overline{\omega_j}$, the partition of unity degenerates to the characteristic functions $\varphi_i = \chi_{\omega_i}$ independent of the chosen weight functions W_i. Thus we see that small overlaps will cause very large gradients of φ_i close to the boundary of the respective support ω_i.

First order consistency is in general not sufficient. Hence, some effort is necessary to improve the consistency order. Here, the moving least squares method (MLSM) [11, 24, 26, 27] allows the construction of shape functions with higher reproduction and consistency order but increases the computational effort dramatically. Furthermore one has to impose severe geometric restrictions onto the cover [27] to make the method work at all. Therefore we use a different approach. We use the partition of unity to collect local approximation spaces V_i defined on the cover patches ω_i, thus generating a global approximation space V on Ω (see the following section). This space may also reproduce the constant only, but it was shown in [3, 4] that the consistency order of the global space V is nevertheless the same as the consistency order of the local spaces V_i (see §4.1.4).

4.1.3. Local polynomial expansion. The partition of unity functions φ_i are able to reproduce the constant function. However, for the discretization of a PDE by a Galerkin method, they are not yet sufficient. First, depending on the computed cover $\{\omega_i\}$ and the choice of weight functions W_i, their derivatives can be unboundedand the entries of the stiffness matrix can not be evaluated in a stable way. We can circumvent these difficulties by the construction of cover patches ω_i with sufficient overlap, but still the consistency error of the discretization would be only of first order in the L^2-norm.

Therefore, we additionally use a hierarchy of local polynomial basis functions of higher degree. To this end, we multiply the partition of unity functions φ_i locally with polynomials. Since we use rectangular patches ω_i only, a local tensor-product space is the most natural choice.

Consider on the reference interval $[-1,1]$ the Legendre polynomials

$$L_p(x) = \frac{2p+1}{p} x L_{p-1}(x) - \frac{p-1}{p} L_{p-2}(x) \quad p = 2, 3, \dots \tag{4.3}$$

with $L_0(x) = 1$ and $L_1(x) = x$. Now, we can define a multi-dimensional hierarchical basis $\{\mathcal{L}_k\}$ with $\mathcal{L}_k(x) = \bigotimes_{j=1}^d L_{k_j}(x^{(j)})$, where $k = (k_1, \dots, k_d) \in \mathbb{N}^d$ is a multi-index. Then via the affine map T_i from $\omega_i = \bigotimes_{j=1}^d [x_i^{(j)} - h_i^{(j)}, x_i^{(j)} + h_i^{(j)}]$ to $[-1,1]^d$, with $T_i = \bigotimes_{j=1}^d T_i^{(j)}$, where $T_i^{(j)} : [x_i^{(j)} - h_i^{(j)}, x_i^{(j)} + h_i^{(j)}] \to [-1,1]$, we can define a local approximation space V_i on ω_i $V_i = \text{span}\ \{\mathcal{L}_k \circ T_i, \quad k \in I_i\}$. Here $I_i \subset \mathbb{N}_0^d$ is some set of multi-indices $k = (k_1, \dots, k_d)$, e.g. the full tensor product set $I_i^T = \{k \in \mathbb{N}_0^d : |k|_\infty \leq p_i\}$ or the set $I_i^C = \{k \in \mathbb{N}_0^d : |k|_1 \leq p_i\}$. The use of the set I_i^C leads to the space V_i^C of complete polynomials of degree p_i, which was also used for the p-version of the finite element method [5].

The global approximation space V is then defined as

$$V = \sum_i \varphi_i V_i \,. \tag{4.4}$$

Note that the local spaces V_i are independent, i.e. there are no compatibility requirements among them. Therefore, the degrees p_i for the sets I_i can not only vary locally, but even different basis functions, e.g. monomials, Taylor polynomials or harmonic functions [3, 4], might be used on some patches ω_i. In the following, we stick to the Legendre polynomials (4.3). Note that in a general situation with varying sets I_i and bases $\{\psi_k^i\}$ there are some further conditions on the amount of overlap of the cover $\{\omega_i\}$ to ensure that the products $\{\varphi_i\psi_k^i\}$ of the partition of unity functions $\{\varphi_i\}$ and the local basis functions $\{\psi_k^i\}$ are a basis of the global space (4.4), see [4, 68].

4.1.4. Basic Convergence Theory for PUM. In this section we state the notation and basic theory of the PUM as developed by Babuška and Melenk in [3, 4]. Crucial to the theory of the partition of unity method is the notion of a (M, C_∞, C_∇) partition of unity. The conditions formulated in the following definitions allow to show the basic approximation properties of the PUM space V of (4.4) as given in Theorem 4.3.

DEFINITION 4.1 (Partition of Unity). Let $\Omega \subset \mathbb{R}^d$ be an open set, let $\{\omega_i\}$ be an open cover of Ω satisfying a point-wise overlap condition

$$\exists M \in \mathbb{N} \quad \forall x \in \Omega \quad \operatorname{card}\{i \,|\, x \in \omega_i\} \leq M \,.$$

Let $\{\varphi_i\}$ be a Lipschitz partition of unity subordinate to the cover $\{\omega_i\}$ satisfying

$$\begin{array}{rclcrcl} \operatorname{supp}(\varphi_i) & \subset & \overline{\omega_i} \quad \text{for all } i\,, & \qquad & \sum_i \varphi_i & \equiv & 1 \text{ on } \Omega\,, \\ \|\varphi_i\|_{L^\infty(\mathbb{R}^d)} & \leq & C_\infty\,, & & \|\nabla\varphi_i\|_{L^\infty(\mathbb{R}^d)} & \leq & \frac{C_\nabla}{\operatorname{diam}(\omega_i)}\,, \end{array}$$

where C_∞ and C_∇ are two constants. Then $\{\varphi_i\}$ is called a (M, C_∞, C_∇) partition of unity subordinate to the cover $\{\omega_i\}$. The partition of unity is said to be of degree $k \in \mathbb{N}_0$ if $\varphi_i \in \mathcal{C}^k(\mathbb{R}^d)$ for all i. The covering sets ω_i are called patches.

DEFINITION 4.2 (PUM space). Let $\{\omega_i\}$ be an open cover of $\Omega \subset \mathbb{R}^d$ and let $\{\varphi_i\}$ be a (M, C_∞, C_∇) partition of unity subordinate to $\{\omega_i\}$. Let $V_i \subset H^1(\Omega \cap \omega_i)$ be given. Then the space

$$V := \left\{ \sum_{i=1}^n \varphi_i v_i \,|\, v_i \in V_i \right\}$$

is called a PUM space. The PUM space V is said to be of degree $k \in \mathbb{N}$ if $V \subset \mathcal{C}^k(\Omega)$. The spaces V_i are referred to as the local approximation spaces.

THEOREM 4.3 (Approximation Property). *Let $\Omega \subset \mathbb{R}^d$ be given. Let $\{\omega_i\}$, $\{\varphi_i\}$ and V_i be as in Definitions 4.1, 4.2. Let $u \in H^1(\Omega)$ be the function to be approximated. Assume that the local approximation spaces V_i have the following approximation properties : On each patch $\Omega \cap \omega_i$, the function u can be approximated by a function $v_i \in V_i$ such that $\|u - v_i\|_{L^2(\Omega\cap\omega_i)} \leq \epsilon_1(i)$, and $\|\nabla(u - v_i)\|_{L^2(\Omega\cap\omega_i)} \leq \epsilon_2(i)$ hold. Then the function $u_{\mathrm{ap}} := \sum_i \varphi_i v_i \in V \subset H^1(\Omega)$ satisfies*

$$\begin{array}{rcl} \|u - u_{\mathrm{ap}}\|_{L^2(\Omega)} & \leq & \sqrt{M} C_\infty \left(\sum_i \epsilon_1^2(i)\right)^{1/2}\,, \\ \|\nabla(u - u_{\mathrm{ap}})\|_{L^2(\Omega)} & \leq & \sqrt{2M} \left(\sum_i \left(\frac{C_\nabla}{\operatorname{diam}(\omega_i)}\right)^2 \epsilon_1^2(i) + C_\infty^2 \epsilon_2^2(i)\right)^{1/2}\,. \end{array} \tag{4.5}$$

Proof: see [3, 4].

Due to this theorem we can use the PUM as an h-version, a p-version or even an hp-version. From (4.5) we can see that the approximation property of the approximation space V may be improved via the reduction of the diameters of the cover patches $\{\omega_i\}$ (the insertion of particles $\{x_i\}$) or via the enhancement of the approximation qualities of the local spaces V_i (the increment of the polynomial order p_i). These refinement strategies are independent, thus we may use both at the same time. Assume that we have a local error estimate $\epsilon_1(i) \leq c_i(\mathrm{diam}\,(\omega_i))^{\nu_i} p_i^{-\mu_i} \|u\|_{L^2(\Omega\cap\omega_i)}$ for some μ_i, ν_i. Then the error estimates (4.5) of Theorem 4.3 take the form

$$\|u - u_{\mathrm{ap}}\|_{L^2(\Omega)} \leq M C_\infty \max_i \{c_i(\mathrm{diam}\,(\omega_i))^{\nu_i} p_i^{-\mu_i}\} \|u\|_{L^2(\Omega)} ,$$

demonstrating the hp-like behavior of the method.

Since the PUM only employs a scattered data set, an adaptive h-refinement (by the insertion of new particles $\{x_i\}$) does not have to cope with problems caused by the grid-structure like hanging nodes, etc. Furthermore, there are no compatibility restrictions on the local spaces V_i. Thus we may also employ an adaptive p-refinement on the local spaces V_i. Hence the development of an appropriate error estimator (or at least indicator) is of great interest and is a main topic for future research.

Theorem 4.3 gives an error estimate for u_{ap} which is the linear combination of the locally optimal solutions v_i. The Galerkin method though constructs a different approximate solution u_{G}. Here, we do not use the local information from V_i only, but rather all information from the spaces V_j with $\omega_j \cap \omega_i \neq \emptyset$. Therefore we sometimes may expect an even better convergence than Theorem 4.3 implies.

Note that the partition of unity $\{\varphi_i\}$ with φ_i from (4.2) and the space V in (4.4) fulfill Definitions 4.1 and 4.2. Consequently, we obtain the approximation properties and error estimates as stated in Theorem 4.3 for our approximation space V. In the following we use the space V in a Galerkin discretization of our elliptic subproblem (2.3).

4.2. Galerkin Method with the PUM space. We want to solve elliptic boundary value problems of the type

$$\begin{array}{rcll} Lu & = & f & \text{in } \Omega \subset \mathbb{R}^d , \\ Bu & = & g & \text{on } \partial\Omega , \end{array} \tag{4.6}$$

where L is a symmetric partial differential operator of second order and B expresses suitable boundary conditions.

Here, the following two major questions come up: How can we efficiently evaluate the integrals which arise in the Galerkin discretization for the stiffness matrix and the right hand side? And how can we deal with boundary conditions properly?

4.2.1. Construction of appropriate Gauss quadrature formulae. In the following let $a(\cdot,\cdot)$ be the continuous and elliptic bilinear form induced by L on $H^1(\Omega)$. We discretize the partial differential equation using Galerkin's method. Then, we have to compute the stiffness matrix

$$A = (a_{ij}) , \quad \text{with } a_{ij} = a(v_j, v_i) ,$$

and the right hand side vector

$$\hat{f} = (f_i) , \quad \text{with } f_i = \int_\Omega f v_i ,$$

where $\{v_i\}$ is a basis of the PUM space V.

The partition of unity functions $\{\varphi_i\}$ are (in general) not of the same shape due to the use of a scattered particle set $\{x_i\}$ and the varying overlap of the cover patches $\{\omega_i\}$. Therefore the trial and test functions $\varphi_i\psi_k^i$ are also not of the same shape and we cannot simply use a transformation onto a reference element to compute the integrals (as it is done in the FEM). Furthermore we cannot employ a straightforward integration scheme to compute the integrals because of the low order global continuity of the trial and test functions $\varphi_i\psi_k^i$ and their derivatives.

Assume that the functions $\{\varphi_i\psi_k^i\}$ form a basis of V. If we restrict ourselves for reasons of simplicity to the case $L = -\Delta$ we have to compute the integrals $\int_\Omega \varphi_i\psi_k^i f$ for the right hand side and the integrals $\int_\Omega \nabla(\varphi_i\psi_k^i)\nabla(\varphi_j\psi_l^j)$ for the stiffness matrix. Recall that φ_i is defined by (4.2). Now we carry out the differentiation. With the notation

$$\mathcal{G}_i := \left(\nabla W_i \sum_m W_m - W_i \sum_m \nabla W_m\right) ,$$

we end up with the integrals

$$\begin{aligned} &\int_\Omega \left(\frac{1}{\sum_m W_m}\right)^4 \mathcal{G}_i\psi_k^i\mathcal{G}_j\psi_l^j , \qquad \int_\Omega \left(\frac{1}{\sum_m W_m}\right)^2 W_i\nabla\psi_k^i W_j\nabla\psi_l^j , \\ &\int_\Omega \left(\frac{1}{\sum_m W_m}\right)^3 \mathcal{G}_i\psi_k^i W_j\nabla\psi_l^j , \qquad \int_\Omega \left(\frac{1}{\sum_m W_m}\right)^3 W_i\nabla\psi_k^i\mathcal{G}_j\psi_l^j \end{aligned} \tag{4.7}$$

for the stiffness matrix and the integrals

$$\int_\Omega \frac{1}{\sum_m W_m} W_i\psi_k^i f \tag{4.8}$$

for the right hand side. Since we use piecewise polynomial weights W_i the functions $\mathcal{G}_i$ may have quite a number of jumps of significant size within their support ω_i. Therefore, the integrals (4.7) should not be computed by a simple quadrature formula which does not respect these discontinuities.

All of the integrals (4.7) and (4.8) are of the form $\int_\Omega \left(\sum_m W_m\right)^{-p}\chi$ which leads to the idea of constructing Gauss quadrature formulae with respect to the weight functions $\left(\sum_m W_m\right)^{-p}$. Since the functions χ have support $\Omega\cap\omega_i$ or even $\Omega\cap\omega_i\cap\omega_j$ the construction of global quadrature formulae on Ω with respect to each of these weights is not advisable. However the construction of Gauss quadrature formulae for each integration domain $\Omega\cap\omega_i\cap\omega_j$ and each integration weight is rather expensive. Therefore, we developed a different approach which is an improvement of the integration scheme introduced in [68]. Here, a decomposition of the integration domain induced by the employed shape functions was proposed. More recently a similar yet simpler approach was proposed by Dolbow and Belytschko in [25]. They also make use of the tensorproduct structure of the supports of their shape functions but they construct a *global* decomposition of the domain Ω. In our approach, we decompose each *local* integration domain $\Omega\cap\omega_i\cap\omega_j$ independent of all ω_k with $\omega_k\cap(\Omega\cap\omega_i\cap\omega_j) = \emptyset$. Furthermore, the number of cells of such a local decomposition[6] is almost minimal.

[6]The decomposition is closely connected to the cover construction. The decomposition may be computed during the construction of the cover with only a slight increase of computational work.

We exploit the tensor product structure of the weight functions W_i and the cover patches ω_i to decompose the integration domain into subpatches with simple geometry on which the functions φ_i are *rational*. Here, we also use the fact that the employed weights W_i are tensor products of B-splines, i.e. they are piecewise polynomial. Consider the integration domain $\Omega_{ij} = \omega_i \cap \omega_j \subset \Omega$. The intersection Ω_{ij} of two cover

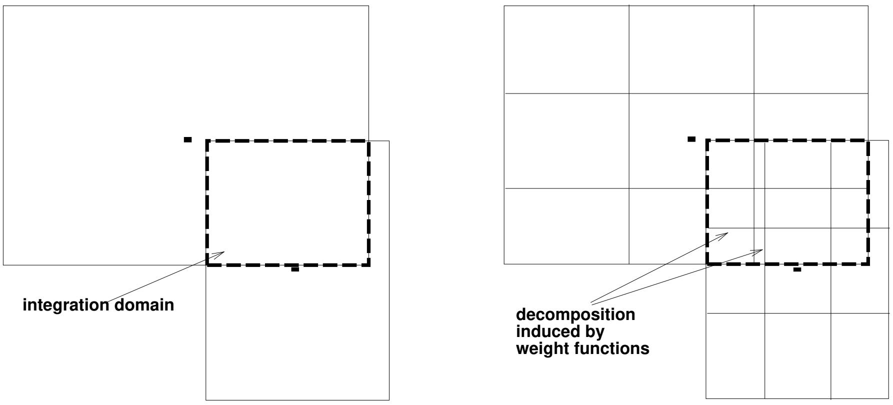

FIGURE 4.3. *Integration domain $\Omega_{ij} = \omega_i \cap \omega_j$ (left). Decomposition $\{C_n\}$ of the integration domain Ω_{ij} via the subdivision induced by the weight functions W_i and W_j (right). Here, the weights are tensor products of quadratic B-splines.*

patches ω_i, ω_j which are tensor products of intervals is also a tensor product of intervals, see Figure 4.3 (left). Furthermore, the employed weight functions W_k are tensor products of B-splines of order l, i.e. they are piecewise polynomials of degree l. Therefore, these weight functions W_k induce a subdivision of each cover patch ω_k into $(l+1) \times (l+1)$ subpatches $\{\omega_{kp}\}$ on which $W_k\big|_{\omega_{kp}}$ is polynomial. Furthermore, these subpatches $\{\omega_{kp}\}$ are also tensor products of intervals. With the help of these subpatches $\{\omega_{ip}\}$, $\{\omega_{jp}\}$ we can define a decomposition $\{C_n\}$ of Ω_{ij}, see Figure 4.3 (right). On the cells C_n of this decomposition we have that $W_i\big|_{C_n}$ and $W_j\big|_{C_n}$ are polynomials of degree l, but all other $W_m\big|_{C_n}$ may still be piecewise rational only. Therefore, we refine the decomposition $\{C_n\}$ by further subdividing the cells C_n with

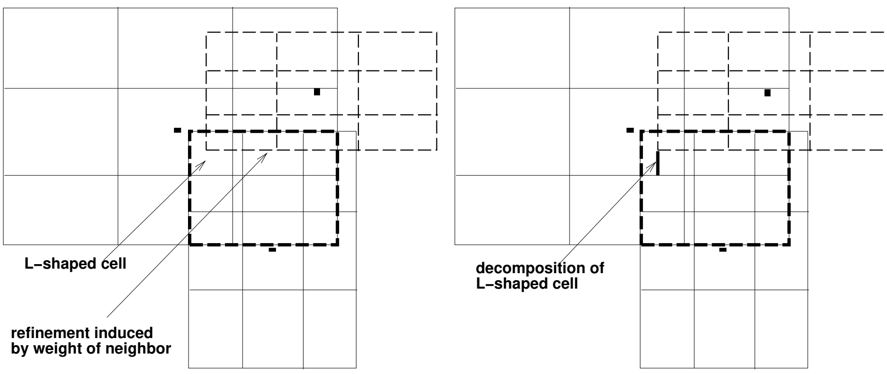

FIGURE 4.4. *Refinement of the decomposition $\{C_n\}$ of the integration domain Ω_{ij} via the subdivision induced by the weight function W_k (tensor product of quadratic B-splines) of one neighboring particle x_k (left). Further decomposition of L-shaped cells of this refinement step (right).*

the help of the $\{\omega_{kp}\}$ subpatches for all k with $\Omega_{ij} \cap \omega_k \neq \emptyset$, see Figure 4.4 (left). The cells $\hat{C}_n$ of this decomposition may be L-shaped, see Figure 4.4 (left). Such an L-shaped cell $\hat{C}_n$ is further subdivided into two rectangular cells $\tilde{C}_n^1$, $\tilde{C}_n^2$ (see Figure 4.4 (right)) to allow the use of a tensor product quadrature scheme on each cell of the decomposition. The resulting decomposition[7] $\{\tilde{C}_n\}$ consists of rectangular cells $\tilde{C}_n$ on which all weight functions $W_m|_{\tilde{C}_n}$ are polynomials of degree l. Therefore, the functions $\mathcal{G}_i$ and the integration weights $(\sum_m W_m)^{-p}$ are rational on these cells. Due to the restrictions imposed in §4.1.4 on the partition of unity $\{\varphi_i\}$ and on the cover $\{\omega_i\}$, the functions $(\sum_m W_m)^{-p}$ are nonsingular on the cells $\tilde{C}_n$. Hence, we can use any standard numerical integration[8] scheme to compute the integrals on the cells $\tilde{C}_n$.

According to Strang's second lemma we have to integrate all entries of the stiffness matrix sufficiently accurately to maintain the convergence order of the discretization error. The shape functions $\varphi_i\psi_k^i$ with $\psi_k^i \neq 1$ are smoother than the partition of unity function φ_i, i.e. a function $\varphi_i\psi_k^i$ is the sum of a polynomial and a rational function whereas φ_i is a rational function only. In a naive approach we could try to exploit this smoothness to reduce the computational costs of the numerical integration of the integrals involving $\varphi_i\psi_k^i$ with $\psi_k^i \neq 1$ by using a quadrature scheme which has a lower order than the quadrature scheme we used for the integrals involving the function φ_i only. But this might lead to the loss of the wanted convergence rate of the discretization error since we now compute the entries of the stiffness matrix with varying accuracy only.

Therefore, we have to find a different way to reduce the computational costs of the numerical integration which allows for a uniform error bound in k and l, i.e.

$$\left\| \int_\Omega \nabla(\varphi_i\psi_k^i)\nabla(\varphi_j\psi_l^j) - I_\Omega^h\left(\nabla(\varphi_i\psi_k^i)\nabla(\varphi_j\psi_l^j)\right) \right\| \leq Ch^q \quad \text{for all } k,l\,.$$

In any case, we have to evaluate the functions φ_i and their derivatives $\nabla\varphi_i$ in the quadrature points of a sufficiently high order scheme to compute the integrals of the non-smooth functions φ_i. Furthermore, it is quite clear that an evaluation of φ_i and $\nabla\varphi_i$ is far more expensive than an evaluation of a local basis function ψ_k^i and its derivatives $\nabla\psi_k^i$. Taking this and the product structure of the approximation space into account we can reduce the computational costs by using the already computed values of the functions φ_i, φ_j and their derivatives for the evaluation of the integrals

$$\int_\Omega \nabla(\varphi_i\psi_k^i)\nabla(\varphi_j\psi_l^j) \quad \text{for all } k,l\,. \tag{4.9}$$

We carry out the differentiation in (4.9) and use the product structure of the shape functions $\{\varphi_i\psi_k^i\}$ to obtain the following integrals

$$\int_\Omega \nabla\varphi_i\nabla\varphi_j\psi_k^i\psi_l^j + \nabla\varphi_i\varphi_j\psi_k^i\nabla\psi_l^j + \varphi_i\nabla\varphi_j\nabla\psi_k^i\psi_l^j + \varphi_i\varphi_j\nabla\psi_k^i\nabla\psi_l^j \tag{4.10}$$

[7] Note, that this approach is not restricted to domains Ω which are unions of rectangles but rather applicable to general domains. For integration domains $\omega_i \cap \omega_j \cap \Omega$ with $\omega_i \cap \omega_j \not\subset \Omega$ we apply this construction to the fictitious integration domain $\omega_i \cap \omega_j$. The cells $\tilde{C}_n$ with $\tilde{C}_n \cap \partial\Omega \neq \emptyset$ of this decomposition $\{\tilde{C}_n\}$ are then further subdivided using a polygonal approximation to the boundary $\partial\Omega$ of the computational domain Ω. Hence the final cell decomposition for integrals close to the boundary of general domains consists of rectangles and triangles.

[8] Since we use polynomials as local spaces V_i we may even compute the integrals analytically, i.e. by symbolic computation, if we consider for example the Laplacian or more general operators involving polynomial coefficient functions [68].

for all k, l. Now we have to integrate products of partition of unity functions φ_i, φ_j, their gradients $\nabla\varphi_i$, $\nabla\varphi_j$, local basis functions ψ_k^i, ψ_l^j and gradients $\nabla\psi_k^i$, $\nabla\psi_l^j$ of local basis functions. Therefore, we can reuse the computed values of φ_i, φ_j and their gradients $\nabla\varphi_i$, $\nabla\varphi_j$ to compute the integrals (4.10) for all k, l. Here, we evaluate the partition of unity functions φ_i and its gradient $\nabla\varphi_i$ only once per quadrature point for the complete local approximation space V_i, consequently reducing the costs but still computing all integrals with the same order of error. Hence, we have to consider the maximal polynomial degrees p_i, p_j of the local functions $\{\psi_k^i\}$, $\{\psi_l^j\}$ and the regularity of the functions φ_i, φ_j for the selection of the order q of the numerical integration scheme on the cells $\tilde{C}_n$.

Note that a further reduction of the costs of the integration can be achieved by using so-called sparse grid integration schemes [30, 63] instead of tensor product formulas on the subpatches.

4.2.2. Boundary conditions and Lagrangian multipliers. Our PUM shape functions $\{\varphi_i\psi_k^i\}$ are non-interpolatory since the partition of unity functions $\{\varphi_i\}$ are (in general) non-interpolatory, i.e. $\varphi_i(x_j) \neq \delta_{ij}$. Furthermore the usage of local approximation spaces V_i with $\dim(V_i) > 1$ generates an approximation space $V = \sum_i \varphi_i V_i$ with more degrees of freedom than interpolation nodes $\{x_i\}$. Thus, we have to cope with the problem: How to fulfill boundary conditions?

First consider (4.6) with Neumann boundary conditions $Bu = \partial u/\partial n_L = g$ on $\Gamma := \partial\Omega$. We learn from the variational formulation

$$F(v) = \frac{1}{2}a(v,v) - (f,v) - \int_\Gamma gv d\Gamma \to \min\{v \in H^1(\Omega)\}, \tag{4.11}$$

that the trial functions v only have to be from the definition space $H^1(\Omega)$ of the differential operator L in its weak form. The functions v do not have to fulfill any additional condition such as boundary values. Thus, the basis of a finite-dimensional subspace V of $H^1(\Omega)$ used to approximate the solution of (4.11) may be compiled of any function $v \in H^1(\Omega)$ and does not need to be interpolatory. Hence, we may directly use our functions $\varphi_i\psi_k^i$ as trial and test functions in the Galerkin procedure without problems.

However, Dirichlet boundary conditions $u = g$ on Γ explicitly impose the values of the solution u on the boundary Γ. Thus, the trial space of the usual weak formulation

$$\text{Find } u \in H^1(\Omega) \cap \{u = g \text{ on } \Gamma\} : a(u,v) = (f,v) \text{ for all } v \in H_0^1(\Omega) \tag{4.12}$$

is not the complete space $H^1(\Omega)$. Therefore, any finite-dimensional space V used to approximate the solution of (4.12) has to consist of functions $\tilde{u}$ fulfilling the boundary condition $\tilde{u} = g$ on Γ. This is usually achieved in the FEM by interpolation of the boundary value $\tilde{u} = \tilde{g}$. Since our shape functions are non-interpolatory, this approach can not be pursued.

For Dirichlet problems, we use a result from the theory of Lagrange multipliers which states: A solution of (4.12) creates a stationary point of

$$F(v,q) = \frac{1}{2}a(v,v) - (f,v) - \int_\Gamma q(v-g)d\Gamma\,. \tag{4.13}$$

In this formulation the functions v do not have to satisfy the boundary conditions $v = g$ on Γ explicitly, since the additional term enforces the boundary conditions.

The respective weak form of (4.13) is: Find $(u,q) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ with

$$\begin{array}{rclcl} a(u,v) & + & (\gamma(v),p) & = & (f,v) \text{ for all } v \in H^1(\Omega) \\ (\gamma(u),q) & & & = & (g,q) \text{ for all } q \in H^{-1/2}(\Gamma) \end{array}, \tag{4.14}$$

where $\gamma : H^1(\Omega) \to H^{1/2}(\Gamma)$ is the trace operator. We can use the shape functions $\{\varphi_i \psi_k^i\}$ from the construction presented in §4.1 for the discretization of the *saddle point problem* (4.14) since it only employs the space $H^1(\Omega)$ as trial and test space instead of the restricted spaces $H^1(\Omega) \cap \{u = g \text{ on } \Gamma\}$ as trial and $H_0^1(\Omega)$ as test space for problem (4.12). Thus, there is no need for interpolatory basis functions. The Lagrange multiplier approach in the finite element context can be found in [2, 15], for applications with wavelets, see [43], and within the fictitious domain and mortar element approach compare [52, 59].

Now we have to construct an appropriate finite dimensional subspace $Q \subset H^{-1/2}$ for the discretization of (4.14). Here, we use the particles $\{x_i \in \overline{\Omega}\}$ from (4.1) to construct a new particle distribution $\{x_j^\Gamma\}$ on the boundary Γ. The particles $\{x_i \in \overline{\Omega}\}$ with $\omega_i \cap \Gamma \neq \emptyset$ are projected in direction of the outer normal onto the boundary Γ and give a new set of particles $\{x_j^\Gamma\}$, which live on the boundary Γ. Then, following the construction described in the previous sections, we set up the finite dimensional PUM space Q on the boundary using the PUM approach: From the newly generated particles $\{x_j^\Gamma\}$ we construct a cover $\{\omega_j^\Gamma\}$ of the $(d-1)$-dimensional manifold Γ, define a partition of unity using Shepard's approach and choose polynomials on the patches $\{\omega_j^\Gamma\}$ to define a PUM space Q on the boundary. Note that this procedure has to be done with respect to the interior PUM space V. Necessary for the existence and uniqueness of a solution of the saddle point problem

$$\begin{array}{rclcll} a(u,v) & + & (\gamma(v),p) & = & (f,v) & \text{for all } v \in V \subset H^1(\Omega)\,, \\ (\gamma(u),q) & & & = & (g,q) & \text{for all } q \in Q \subset H^{-1/2}(\Gamma)\,. \end{array} \tag{4.15}$$

is the fulfillment of the discrete Ladyzhenskaya–Babuška–Brezzi (LBB) condition [19]

$$\inf_{q \in Q, q \neq 0} \sup_{v \in V, v \neq 0} \frac{|(\gamma(v),q)|}{\|v\|_V \|q\|_Q} \geq c_0\,. \tag{4.16}$$

According to the results in [15] the boundary space Q has to have a coarse enough resolution compared with the resolution of V in the interior to fulfill the discrete LBB condition (4.16). This coarser resolution of the space Q may be achieved by properly[9] thinning out the particle distribution, i.e. using only an appropriate subset of $\{x_j^\Gamma\}$, or by the choice of lower order polynomials on the boundary than in the interior. Both of these approaches have shown good results in our numerical experiments, see [68].

4.3. Solution of the discrete saddle point problem. Now we have to solve a linear system with saddle point structure, i.e.

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ q \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{4.17}$$

[9] We propose the use of a particle distribution with fewer particles but approximately the same density distribution as the complete particle distribution $\{x_j^\Gamma\}$.

where $B \in \mathbb{R}^{\dim(Q)} \times \mathbb{R}^{\dim(V)}$, $q \in \mathbb{R}^{\dim(Q)}$, $u \in \mathbb{R}^{\dim(V)}$ and $A \in \mathbb{R}^{\dim(V)} \times \mathbb{R}^{\dim(V)}$ is symmetric and positive definite and thus invertible. Block Gaussian elimination directly leads to the Schur complement system

$$BA^{-1}B^Tq = BA^{-1}f - g. \tag{4.18}$$

The unique solvability of (4.17) is given when (4.16) holds.

A possible approach for the solution of (4.17) is now to iteratively solve (4.18) by a (preconditioned) conjugate gradient method and then to obtain u by backward substitution, i.e. $u = A^{-1}(f - B^Tq)$. A realization of this method with a linear iteration is the (preconditioned) Uzawa algorithm, see [14, 47]. It reads

$$\begin{array}{rcl} u_{i+1} & = & u_i + A^{-1}(f - (Au_i + B^Tq_i)) \\ q_{i+1} & = & q_i + \omega_B C_B(Bu_{i+1} - g) \end{array} \tag{4.19}$$

where ω_B is an iteration parameter and C_B is a preconditioner for the Schur complement $BA^{-1}B^T$. The drawback of this method is that the action of A^{-1} must be computed in each iteration step, which makes the method expensive.

If we now replace A^{-1} by some steps of an iterative method, most preferable a multi-grid method or a multilevel preconditioner, we employ only an approximation C_A to A^{-1} and obtain the so-called inexact Uzawa algorithm [16, 17]

$$\begin{array}{rcl} u_{i+1} & = & u_i + \omega_A C_A(f - (Au_i + B^Tq_i)) \\ q_{i+1} & = & q_i + \omega_B C_B(Bu_{i+1} - g) \end{array} \tag{4.20}$$

with iteration parameters ω_A, ω_B. Altogether, in this algorithm only the action of the matrices A, B, B^T and preconditioners C_A, C_B onto vectors must be programmed.

Here, we can exploit the block-structure of the matrices A and B, which is induced by the product structure of the shape functions $\varphi_i\psi_k^i$. The block corresponding to all the φ_i's can be solved efficiently with an algebraic multi-grid solver (AMG) [34, 66, 67] and this solver may further be used as a preconditioner on the diagonal blocks corresponding to $\psi_k^i \neq 1$. Altogether, this procedure gives an efficient preconditioner C_A for the inexact Uzawa algorithm.

Now we have to find an efficient iterative solution method or preconditioner C_B for the Schur complement $BA^{-1}B^T$. Currently we use only $C_B = I$, therefore the iterative solution of (4.17) is not yet optimal, i.e. the condition number of the system matrix still depends on $h \simeq N^{-1/d}$. In the future, effective Schur complement preconditioners C_B have to be constructed to make our gridless method competitive. An approach could be the generalization of the methods developed in [18, 43] to matrices from gridless discretizations.

5. Numerical experiments. In this section we present the results of our numerical experiments. We apply the particle-partition of unity method to a linear advection problem, several elliptic problems, the heat-equation and an instationary convection-diffusion problem.

5.1. Hyperbolic Problem. In the following we consider a standard test case for linear advection problems, the so-called Molenkamp problem [72]:

$$\frac{\partial u}{\partial t} + \begin{pmatrix} -y \\ x \end{pmatrix} \nabla u = 0 \text{ in } \Omega := (-1,1)^2 \times (0,T) \tag{5.1}$$

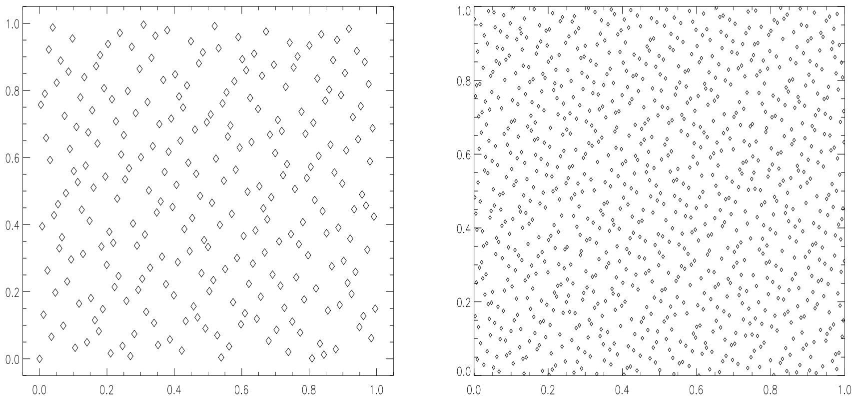

FIGURE 5.1. *Particle distributions with* 256 *points (left) and* 1024 *points (right) generated via a* (2, 3) *Halton–sequence.*

with an initial value $u(x, y, 0) = \exp(-100((x-\frac{1}{4})^2 + (y-\frac{1}{4})^2))$. For the exact solution $u(x, y, 2l\pi) = u(x, y, 0)$ for $l \in \mathbb{N}$ holds and mass $\int_\Omega u$ and energy $\|u\|^2_{L^2}$ are conserved.

We discretize (5.1) using the particle method described in §3 with the shape functions from the particle-PUM. For the resolution of the initial value we choose a simple block-structure approach. Since $u(x, y, 0)$ almost vanishes outside of some $\hat{\Omega} \subset \Omega$, we need only a few particles $\{\xi_i^B\}$ in $\Omega \setminus \hat{\Omega}$ for the construction of the shape functions. We distribute these particles $\{\xi_i^B\}$ in $\Omega \setminus \hat{\Omega}$ using a $(2, 3)$ Halton–sequence[10] (see Figure 5.1) with about 200 points and another $(2, 3)$ Halton–sequence with 256 particles $\{\xi_i^I\}$ is used to resolve the initial value $u(x, y, 0)$ in $\hat{\Omega}$ (see Figure 5.2). Note that a further adaptation of the initial particle distribution can be achieved following the ideas of [35].

Some of the particles $\{\xi_i\}$ leave the domain during the computation since we propagate the particles along their corresponding characteristic. To cope with the particle loss, we embed the domain Ω in a larger domain $\tilde{\Omega}$ and distribute a set of background particles $\{\xi_i^B\}$ rather in $\tilde{\Omega} \setminus \hat{\Omega}$ than $\Omega \setminus \hat{\Omega}$. We move all particles $\{\xi_i^B\} \cup \{\xi_i^I\}$ along their characteristic, but a PUM shape function is constructed in those particles $\xi_i \in \{\xi_i^B\} \cup \{\xi_i^I\}$ with $\xi_i \in \Omega$ only. Hence, some particles $\{\xi_i\}$ still leave the domain, but there also are some particles $\{\xi_i^B\}$ (re-)entering Ω. Therefore, we have an almost constant number of particles in Ω and $\hat{\Omega}$ over time.

We use a stretch parameter $\alpha = 1.25$, the well known hat function and complete quadratic Legendre polynomials V_i^C for the construction of the shape functions in this experiment. Hence, the approximation space has about 2700 degrees of freedom, of which we have 1500 within the approximate support $\hat{\Omega}$. As ODE solver in this experiment we use a simple Euler and a fourth order Runge–Kutta method. The increase of computational work by applying a Runge–Kutta method instead of the Euler method is negligible since we only have more evaluations of the velocity field which is independent of the solution of (5.1).

[10]Halton–sequences are pseudo Monte Carlo sequences, which are used in sampling and numerical integration [62]. Consider $n \in \mathbb{N}_0$ given as $\sum_j n_j p^j = n$ for some prime p. We can define the transformation H_p from $\mathbb{N}_0$ to $[0, 1]$ with $n \mapsto H_p(n) = \sum_j n_j p^{-j-1}$. Then, the (p, q) Halton–sequence with N points is defined as $\{(H_p(n), H_q(n))\}$ where $n = 0, \ldots, N-1$.

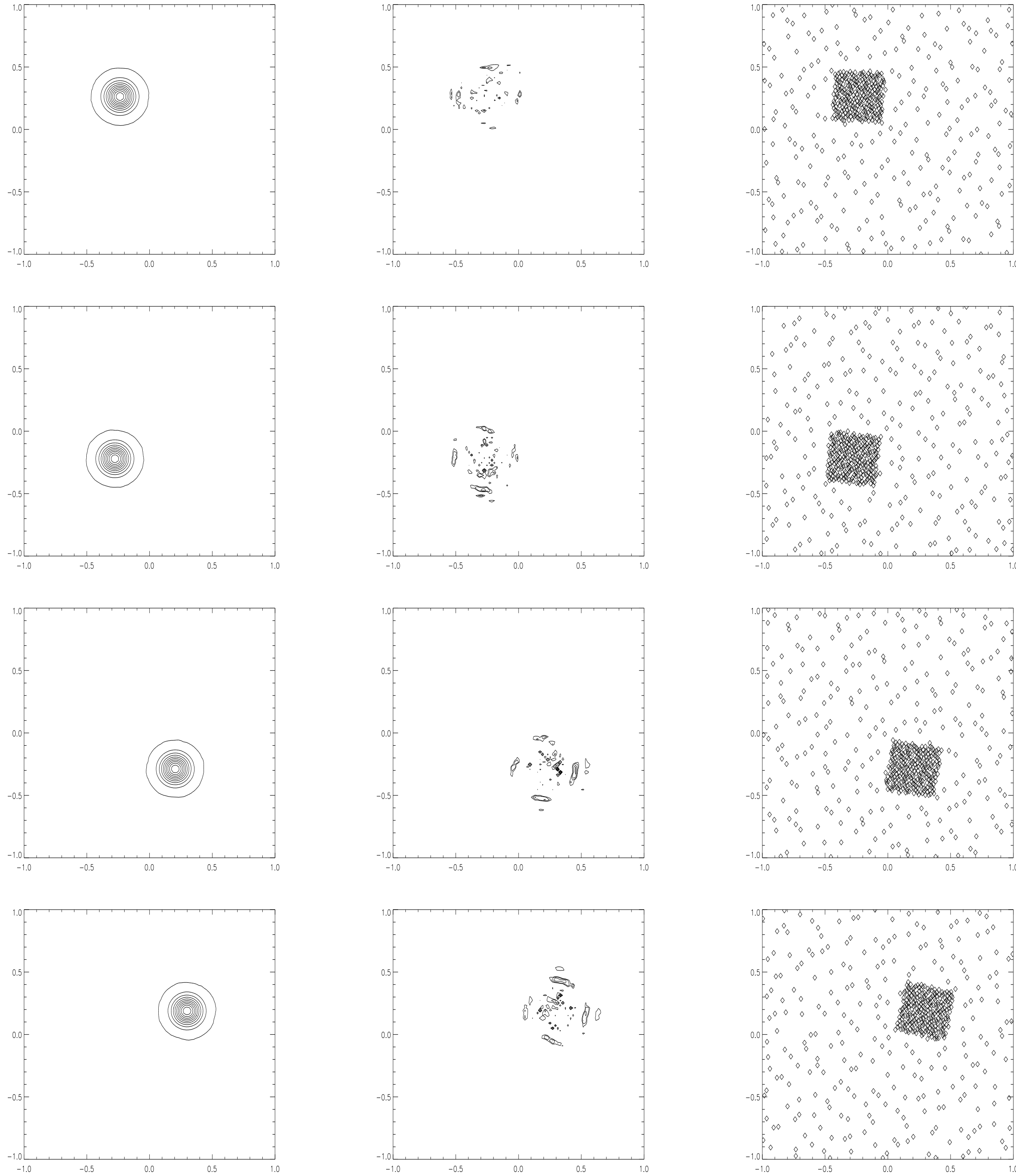

FIGURE 5.2. *Isolines* $(0.005, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ *of the computed solution (left), isolines* $(0.0005, 0.001, 0.002, 0.003, 0.004, 0.005)$ *of the error (middle) and particle distributions (right) for (5.1) using the Runge–Kutta method at times* $(6\Delta t, 12\Delta t, 18\Delta t, 24\Delta t)$ *with* $\Delta t = 2\pi/25$.

We solve the mass matrix problem using a preconditioned CG method with a tolerance of 10^{-8}. The error $u - u_{\mathrm{ap}}$ in the L^∞-norm and the oscillations

$$u_{\mathrm{ap}}^{\mathrm{osc}} = \begin{cases} \min u - \min u_{\mathrm{ap}} & : \quad |\min u_{\mathrm{ap}} - \min u| > |\max u - \max u_{\mathrm{ap}}| \\ \max u_{\mathrm{ap}} - \max u & : \quad |\min u_{\mathrm{ap}} - \min u| \leq |\max u - \max u_{\mathrm{ap}}| \end{cases}$$

(both computed on a 500×500 grid), the mass defect $d_{\mathrm{M}} = 1 - \int_\Omega u_{\mathrm{ap}} / \int_\Omega u$ and the energy defect $d_{\mathrm{E}} = 1 - \|u_{\mathrm{ap}}\|_{L^2}^2 / \|u\|_{L^2}^2$ for the computed solution u_{ap} at various times t for several values of Δt are given in Table 5.1 for the Euler method. The corresponding values for the Runge–Kutta method are displayed in Table 5.2. From these numbers we see the anticipated convergence behavior in the L^∞-norm, i.e. $O(\Delta t)$ for the Euler and $O(\Delta t^4)$ for the Runge–Kutta method.

Δt	$\|e\|_{L^\infty}$	$u_{\mathrm{ap}}^{\mathrm{osc}}$	d_{M}	d_{E}	$\|e\|_{L^\infty}$	$u_{\mathrm{ap}}^{\mathrm{osc}}$	d_{M}	d_{E}	$\|e\|_{L^\infty}$	$u_{\mathrm{ap}}^{\mathrm{osc}}$	d_{M}	d_{E}
	$t = 1$				$t = 2$				$t = 4$			
4									4.17_{-1}	1.21_{-3}	2.03_{-1}	2.03_{-1}
2					1.15_{-1}	1.59_{-3}	5.99_{-2}	6.00_{-2}	2.27_{-1}	2.41_{-3}	1.16_{-1}	1.16_{-1}
1	2.93_{-2}	1.23_{-3}	1.57_{-2}	1.57_{-2}	5.89_{-2}	1.40_{-3}	3.11_{-2}	3.11_{-2}	1.17_{-1}	2.06_{-3}	6.13_{-2}	6.13_{-2}
	$t = 8$				$t = 16$				$t = 32$			
8	9.62_{-1}	1.34_{-3}	5.05_{-1}	5.05_{-1}	1.00_{0}	3.44_{-3}	7.55_{-1}	7.55_{-1}	1.00_{0}	-1.66_{-1}	9.40_{-1}	9.43_{-1}
4	7.42_{-1}	6.83_{-4}	3.65_{-1}	3.65_{-1}	9.90_{-1}	1.68_{-3}	5.97_{-1}	5.97_{-1}	1.00_{0}	9.89_{-3}	8.38_{-1}	8.38_{-1}
2	4.37_{-1}	1.23_{-3}	2.19_{-1}	2.19_{-1}	7.69_{-1}	1.00_{-3}	3.90_{-1}	3.90_{-1}	9.94_{-1}	3.00_{-3}	6.28_{-1}	6.28_{-1}
1	2.31_{-1}	1.23_{-3}	1.19_{-1}	1.19_{-1}	4.44_{-1}	9.31_{-4}	2.24_{-1}	2.24_{-1}	7.77_{-1}	2.84_{-3}	3.97_{-1}	3.97_{-1}
	$t = 64$				$t = 128$				$t = 256$			
8	1.00_{0}	-9.40_{-1}	9.96_{-1}	1.00_{0}	1.00_{0}	-1.00_{0}	1.00_{0}	1.00_{0}	1.00_{0}	-1.00_{0}	1.00_{0}	1.00_{0}
4	1.00_{0}	-4.28_{-1}	9.74_{-1}	9.82_{-1}	1.00_{0}	-9.91_{-1}	9.99_{-1}	1.00_{0}	1.00_{0}	-1.00_{0}	1.00_{0}	1.00_{0}
2	1.00_{0}	4.86_{-3}	8.62_{-1}	8.62_{-1}	1.00_{0}	-7.87_{-1}	9.81_{-1}	9.93_{-1}				
1	9.96_{-1}	5.34_{-3}	6.37_{-1}	6.37_{-1}								

TABLE 5.1

Errors in the L^∞-norm, oscillations $u_{\mathrm{ap}}^{\mathrm{osc}}$, mass d_{M} and energy d_{E} defects for the Molenkamp problem (5.1) using the Euler method, t and Δt given in $2\pi/50$.

Δt	$\|e\|_{L^\infty}$	$u_{\mathrm{ap}}^{\mathrm{osc}}$	d_{M}	d_{E}	$\|e\|_{L^\infty}$	$u_{\mathrm{ap}}^{\mathrm{osc}}$	d_{M}	d_{E}	$\|e\|_{L^\infty}$	$u_{\mathrm{ap}}^{\mathrm{osc}}$	d_{M}	d_{E}
	$t = 1$				$t = 2$				$t = 4$			
4									5.14_{-3}	1.72_{-3}	-2.23_{-4}	-2.20_{-4}
2					4.62_{-3}	1.61_{-3}	-3.91_{-6}	-5.88_{-7}	4.31_{-3}	1.84_{-3}	-9.12_{-6}	9.98_{-7}
1	4.40_{-3}	1.41_{-3}	5.56_{-7}	4.27_{-6}	5.25_{-3}	1.57_{-3}	2.30_{-7}	7.57_{-6}	5.05_{-3}	1.68_{-3}	-2.40_{-7}	1.58_{-5}
	$t = 8$				$t = 16$				$t = 32$			
8	3.16_{-2}	7.62_{-4}	-1.30_{-2}	-1.30_{-2}	6.18_{-2}	1.44_{-3}	-2.62_{-2}	-2.62_{-2}	1.21_{-1}	1.97_{-3}	-5.32_{-2}	-5.32_{-2}
4	4.30_{-3}	1.70_{-3}	-4.48_{-4}	-4.40_{-4}	8.13_{-3}	1.97_{-3}	-8.94_{-4}	-8.77_{-4}	8.48_{-3}	1.69_{-3}	-1.79_{-3}	-1.75_{-3}
2	4.08_{-3}	1.41_{-3}	-1.51_{-5}	1.32_{-6}	4.60_{-3}	1.73_{-3}	-3.05_{-5}	5.67_{-6}	4.72_{-3}	1.20_{-3}	-5.65_{-5}	7.97_{-6}
1	3.80_{-3}	1.05_{-3}	4.40_{-7}	2.72_{-5}	6.20_{-3}	1.36_{-3}	1.40_{-6}	5.61_{-5}	6.96_{-3}	1.84_{-3}	2.65_{-6}	1.03_{-4}
	$t = 64$				$t = 128$				$t = 256$			
8	2.40_{-1}	1.02_{-3}	-1.09_{-1}	-1.09_{-1}	4.63_{-1}	4.93_{-4}	-2.30_{-1}	-2.30_{-1}	7.99_{-1}	2.63_{-3}	-5.14_{-1}	-5.13_{-1}
4	1.53_{-2}	1.14_{-3}	-3.58_{-3}	-3.51_{-3}	3.05_{-2}	1.99_{-3}	-7.17_{-3}	-7.05_{-3}	5.90_{-2}	2.72_{-3}	-1.44_{-2}	-1.42_{-2}
2	8.49_{-3}	2.45_{-3}	-1.08_{-4}	1.06_{-5}	9.90_{-3}	3.14_{-3}	-2.08_{-4}	-5.46_{-6}				
1	1.02_{-2}	3.29_{-3}	5.13_{-6}	1.90_{-4}								

TABLE 5.2

Errors in the L^∞-norm, oscillations $u_{\mathrm{ap}}^{\mathrm{osc}}$, mass d_{M} and energy d_{E} defects for the Molenkamp problem (5.1) using using a fourth order Runge–Kutta method, t and Δt given in $2\pi/50$.

We compute an approximate solution u_{ap} in every time step k by projecting a transformed former solution onto the current approximation space.

$$u_{\mathrm{ap}}^k = \Pi^k \left(u_{\mathrm{ap}}^{k-1} \circ \tilde{T}^{\Delta t} \right) \tag{5.2}$$

Hence, for the defects in mass d_{M} and energy d_{E} we get the equivalences

$$d_{\mathrm{M}}^{k} = 1 - \frac{\int_{\Omega} \Pi^{k}\left(u_{\mathrm{ap}}^{k-1} \circ \tilde{T}^{\Delta t}\right)}{\int_{\Omega} u^{0}}, \qquad d_{\mathrm{E}}^{k} = 1 - \frac{\int_{\Omega} \|\Pi^{k}\left(u_{\mathrm{ap}}^{k-1} \circ \tilde{T}^{\Delta t}\right)\|^{2}}{\int_{\Omega} \|u^{0}\|^{2}} \tag{5.3}$$

where $u(\cdot, k\Delta t) = u^{k} = u^{0} \circ T^{k\Delta t}$ and $T^{k\Delta t}$ is the transformation induced by the exact characteristic at $k\Delta t$ with $|\det J_{T^{k\Delta t}}| = 1$. Not only the transformation $\tilde{T}^{\Delta t}$ but also the projection Π^{k}, i.e. the approximation space, depends on the applied ODE solver since the particles which define the approximation space are propagated forward along the characteristic. Therefore, we have two sources for a time dependent error component in (5.3), $|\det J_{\tilde{T}^{\Delta t}}|$ and Π^{k}. A convergence behavior similar to that of the applied ODE solver can be perceived from the measured defects $d_{\mathrm{M/E}}$. As one can expect, for both ODE solvers we have $d_{\mathrm{M}} = d_{\mathrm{E}}$ if the defects are large compared with the spatial resolution, i.e. the displayed defects $d_{\mathrm{M/E}}$ are significant only if the local errors can be resolved (beyond oscillations) by the approximation space. For the Euler method we have a loss of energy and mass over time which is due to the fact that the backward Euler integration contracts the support of the approximate solution, i.e. the characteristics are approximated by a contracting spiral. For the particle distribution though we approximate the characteristic by forward integration. Hence, the path of a single particle is a widening spiral and the particles are propaged away from the support of the solution. The particles may even leave the domain of interest whereas the approximate solution will move to the center with a contracted support. Therefore, the approximate solution may vanish once its support is contracted beyond the spatial resolution of the approximation space (see Table 5.1). Hence, we have an additional loss of spatial resolution due to the difference in forward-backward time integration. This error affects the quality of the projection Π^{k} which is the main source for the oscillations $u_{\mathrm{ap}}^{\mathrm{osc}}$. The displayed oscillations $u_{\mathrm{ap}}^{\mathrm{osc}}$ are also a measure for peak attenuation since $u_{\mathrm{ap}}^{\mathrm{osc}} = \min u - \min u_{\mathrm{ap}}$ in all our experiments.

In Figure 5.2 the isolines of the approximate solution, the isolines of the error and the corresponding particle distribution at times $(6\Delta t, 12\Delta t, 18\Delta t, 24\Delta t)$ with $\Delta t = 2\pi/25$ are displayed. We clearly see that the spatial degrees of freedom follow the evolution of the solution. From these figures we cannot see a difference in the forward-backward time integration for the Runge–Kutta method, i.e. the particles and the solution are propaged along the "same" path. This can also be observed from the oscillations displayed in Table 5.2 since $u_{\mathrm{ap}}^{\mathrm{osc}}$ is (almost) independent of Δt and t.

From the displayed isolines we notice the "conservation" of the peak of the solution. Moreover, we see that the overall error is dominated by the error in space rather than in time for small values of Δt. Further experiments showed the anticipated "third order" convergence in space (see §5.2). Here, the error is located in a small region around the approximate support $\hat{\Omega}$ where the resolution of the approximation space is significantly less than in $\hat{\Omega}$. Hence, the error could probably be reduced by using an initial particle distribution which is more adapted to the initial value. The isolines of the computed solution after (slightly more than) 3 and 6 full revolutions using the Runge–Kutta method with $\Delta t = 4\pi/25$ are displayed in Figure 5.3. Again, we see the conservation of the peak. But here we can also observe the mass and energy defect. The diameter of every isoline grows, i.e. we have an increase in mass and energy over time. Furthermore, the isolines of the corresponding errors are displayed which show the dominance of the error in time rather than in space for large values of Δt.

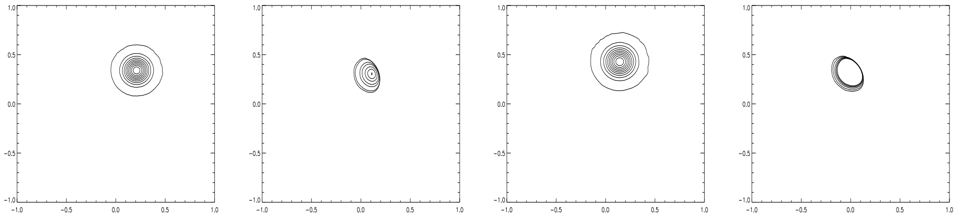

FIGURE 5.3. *Isolines* $(0.005, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ *of the computed solution (left) and isolines* $(0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4)$ *of the error (right) for (5.1) using a fourth order Runge–Kutta method after 3* $t = 19\Delta t$ *(left) and 6* $t = 38\Delta t$ *(right) revolutions with* $\Delta t = 4\pi/25$.

5.2. Elliptic Problems. In the following we consider the model problem

$$-\nabla A \nabla u + bu = f \text{ in } \Omega := (0,1)^2 \tag{5.4}$$

with suitable boundary conditions $Bu = g$ on $\Gamma := \partial\Omega$. We discretize this elliptic problem using our particle-PUM as described in §4. The particle distribution we use is generated either via a Halton–sequence or a graded uniform distribution.

According to the results of [3, 4] (see §4.1.4), we can estimate the global error with respect to $\max_i \operatorname{diam}(\omega_i)$ only. We can expect erratic fluctuations in the convergence rates if we measure the convergence based on such error estimates since we use a pseudo Monte Carlo sequence to generate the particle distribution. Furthermore, we may get a convergence behavior of the global space V which is superior to the one of the local spaces V_i due to the overlapping of the cover patches, see §4.1.4. Therefore, we measure convergence rates ρ with respect to the number of degrees of freedom $\mathsf{dof} = \sum_{i=1}^N \dim(V_i)$, i.e. we propose an algebraic error estimate

$$\|u - \tilde{u}\| = O(\mathsf{dof}^\rho).$$

We compute ρ from the measured errors $\|u - \tilde{u}_l\|$ and $\|u - \tilde{u}_{l-1}\|$ on two successive levels l and $l-1$ via the relation

$$\rho = \frac{\log\left(\frac{\|e_l\|}{\|e_{l-1}\|}\right)}{\log\left(\frac{\mathsf{dof}_l}{\mathsf{dof}_{l-1}}\right)}, \quad \text{where } e_l = u - \tilde{u}_l. \tag{5.5}$$

These convergence rates ρ can be translated into the well-accepted h^α notation in the case of a uniform particle distribution and $\dim(V_i) = \mathsf{const}$. Here, we have $h \sim N^{-1/d} \sim \mathsf{dof}^{-1/d}$. Therefore we get the relation $-\alpha/d \sim \rho$. For example, the analogue of a convergence of the order $O(h^2)$ in the L^2-norm is achieved if we measure $\rho_{L^2} = -1$ for a 2D problem.

Example 1 (h-version of particle-PUM, smooth solution). Our first elliptic example is the simple Helmholtz equation

$$-\Delta u + u = f \text{ in } \Omega = (0,1)^2 \tag{5.6}$$

with Neumann boundary conditions $\partial u/\partial n = g$ on $\Gamma = \partial\Omega$. We choose f and g such that the continuous solution to the problem is given as

$$u(x,y) = \arctan\left(100\left(\frac{x+y}{\sqrt{2}} - 0.8\right)(x-x^2)(y-y^2)\right). \tag{5.7}$$

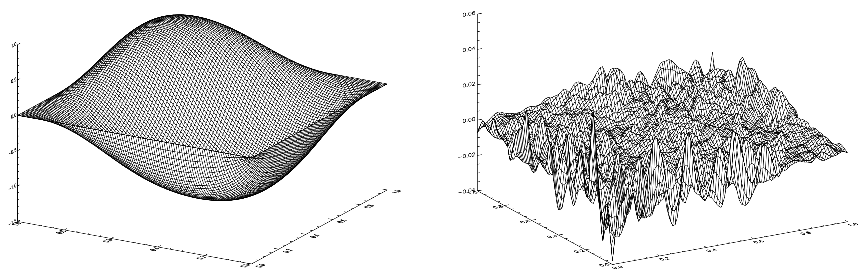

FIGURE 5.4. *Graph of solution (5.7) and graph of error for Example 1 using* 256 *particles.*

The weight function W_i in this experiment is the quadratic B-spline, the stretch parameter $\alpha = 1.5$, the local approximation spaces V_i are the linear Legendre polynomials and the underlying particle distribution is the pseudo Monte Carlo distribution generated via a $(2, 3)$ Halton–sequence. Therefore, we expect to measure convergence rates $\rho_{L^2} \sim -1$ in the L^2-norm and $\rho_{H^1} \sim -0.5$ in the H^1-norm. The solution (5.7) and the error for the experiment with 256 particles are displayed in Figure 5.4.

N	p	dof	$\|e\|_{L^\infty}$	ρ_{L^∞}	$\|e\|_{L^2}$	ρ_{L^2}	$\|e\|_{H^1}$	ρ_{H^1}
64	1	192	1.318_{-1}	$***$	2.819_{-2}	$***$	9.826_{-1}	$***$
256	1	768	4.002_{-2}	-0.860	5.472_{-3}	-1.183	4.056_{-1}	-0.638
1024	1	3072	1.119_{-2}	-0.919	1.629_{-3}	-0.874	2.223_{-1}	-0.434
4096	1	12288	3.004_{-3}	-0.949	3.833_{-4}	-1.044	1.044_{-1}	-0.535
16384	1	49152	7.797_{-4}	-0.973	9.439_{-5}	-1.011	5.230_{-2}	-0.499

TABLE 5.3
Errors and convergence rates for Example 1.

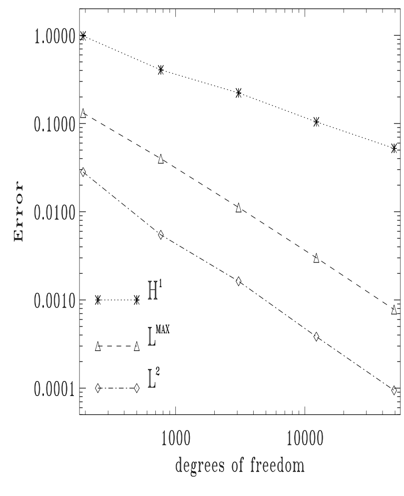

FIGURE 5.5. *Convergence history for Example 1.*

Table 5.3 and Figure 5.4 give the error in different norms and the associated convergence rates for varying particle numbers. We clearly see a rate of -1 in the L^2- and L^∞-norm and a rate of -0.5 for the H^1-norm (i.e. a convergence of the order $O(h^2)$ in the L^2- and L^∞-norm and a convergence of the order $O(h)$ in the H^1-norm). These results demonstrate that the particle-PUM based on a pseudo Monte Carlo particle distribution matches the convergence behavior of the h-version of the finite element method on a uniform grid. Thus, our method incorporates the advantages of a particle method, but preserves at the same time the usual convergence properties of the finite element method.

Example 2 (h-version of particle-PUM, singular solution). In our second experiment with an elliptic problem we consider the Laplace equation

$$-\Delta u = f \text{ in } \Omega = (0,1) \times (-0.5, 0.5) \tag{5.8}$$

with Dirichlet boundary conditions $u = g$ on $\Gamma = \partial\Omega$. We choose f and g such that

$$u(x,y) = u(z) = \mathrm{Re}(z^{1/2}) = (x^2+y^2)^{1/4} \cos\left(\frac{\arctan(\frac{x}{y})}{2}\right) \tag{5.9}$$

is the continuous solution to the problem. Besides a uniform particle distribution

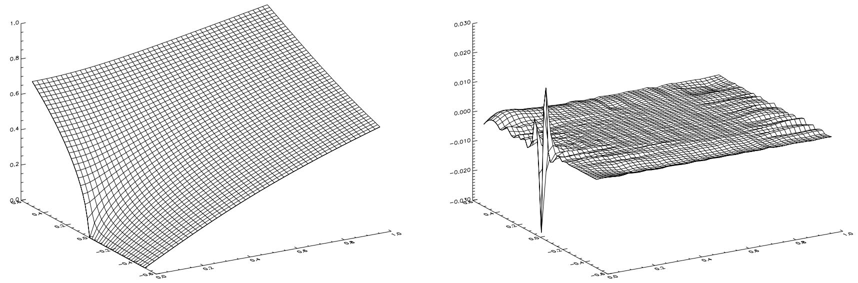

FIGURE 5.6. *Graph of solution (5.9) and graph of error for Example 2 using* 256 *particles.*

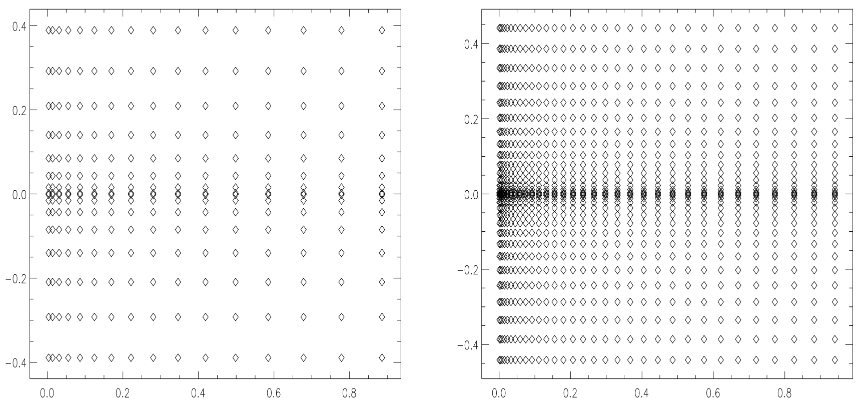

FIGURE 5.7. *Graded particle distribution via transformation:* $(x,y) \mapsto (x^2, \pm 2y^2)$.

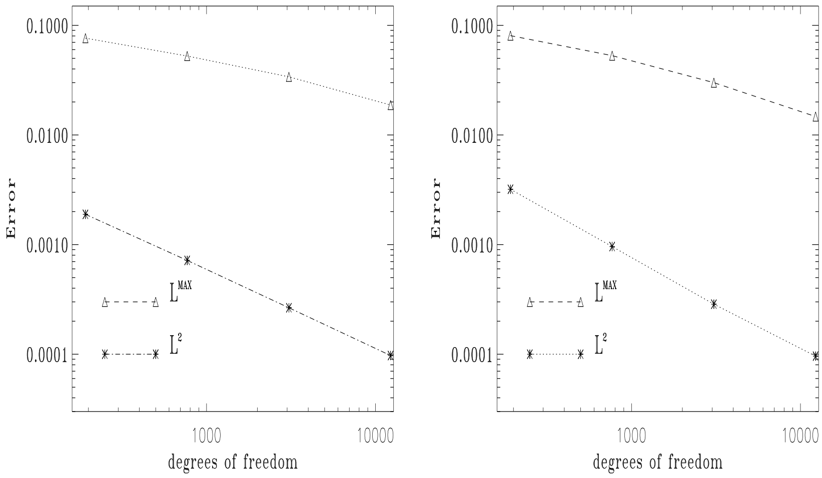

FIGURE 5.8. *Convergence history for Example 2 for the uniform particle distribution (left) and the graded particle distribution (right).*

we also use a graded particle sequence (see Figure 5.7) to cope with the singularity of the solution (5.9). Note that all particles ξ_i of the uniform distribution are interior points of the domain Ω, i.e. $\xi_i \in \Omega \setminus \partial\Omega$. We use the same weight functions and local approximation spaces as in Example 1 and a stretch parameter $\alpha = 1.25$ for the uniform and the graded particle distribution. The Dirichlet boundary conditions are enforced by Lagrangian multipliers which are constructed according to §4.2.2. Here, we are using the constant function only as local approximation spaces for the multiplier space Q_h to have a coarser resolution on the boundary than in the interior so that the discrete Ladyzhenskaya–Babuška–Brezzi condition (4.16) holds. We measure a convergence rate ρ_{L^∞} of about -0.35 for the uniform particle distribution (see Table 5.4). This is slightly better than the well-known convergence of the order $O(h^{1/2})$ of the finite element method in the L^∞-norm for this problem. Since the pointwise convergence of the finite element method in points sufficiently far away from the singularity is of the order $O(h^{3/2})$, we can expect a convergence rate ρ_{L^2} in the L^2-norm less than -0.75. We measure a convergence rate ρ_{L^2} which is close to -0.75 for the uniform particle distribution. Figure 5.6 shows the solution (5.9) and the error for 256 particles of the graded particle distribution. From Figure 5.8 and Table 5.4 we see that the use of a graded particle distribution improves the convergence behavior of the method. Now, we measure convergence rates ρ_{L^2} and ρ_{L^∞} of about

$\|e\|_{L^\infty}$	ρ_{L^∞}	$\|e\|_{L^2}$	ρ_{L^2}	N	p	dof	$\|e\|_{L^\infty}$	ρ_{L^∞}	$\|e\|_{L^2}$	ρ_{L^2}
7.639_{-2}	$***$	1.889_{-3}	$***$	64	1	192	8.059_{-2}	$***$	3.196_{-3}	$***$
5.251_{-2}	-0.270	7.171_{-4}	-0.699	256	1	768	5.293_{-2}	-0.303	9.593_{-4}	-0.868
3.391_{-2}	-0.315	2.653_{-4}	-0.717	1024	1	3072	3.001_{-2}	-0.409	2.860_{-4}	-0.873
1.875_{-2}	-0.427	9.727_{-5}	-0.724	4096	1	12288	1.473_{-3}	-0.513	9.603_{-5}	-0.787

TABLE 5.4

Errors and convergence rates for Example 2 for the uniform particle distribution (left) and the graded particle distribution (right).

-0.8 in the L^2-norm and -0.5 in the L^∞-norm. Since the grading we employ to the particle distribution (see Figure 5.7) is not optimal, we are not able to completely recover $\rho_{L^2,L^\infty} \sim -1$. Here, error estimators for adaptive particle refinement must be developed in the future to fully exploit the capabilities of the method.

The results from Examples 1 and 2 demonstrate that the particle-PUM does work on irregular or scattered data with roughly the same convergence behavior as the finite element method on a uniform grid. Furthermore, Example 2 indicates that we might also resolve singularities of the solution by local refinement, i.e. by increasing the particle density close to the singularity. Thus, we can achieve a similar convergence as we would obtain with an adaptive finite element method but can circumvent the geometric constraints we have to face there (angle conditions, hanging nodes, proper local grid refinement, quasi-uniformity of the grid, etc.). With the particle-PUM we may insert particles (almost) anywhere near the singularity without worrying much about the neighboring particles and their respective patch size.

Example 3 (p-version of particle-PUM, smooth solution). We again consider the continuous problem from Example 1, but we will now apply the p-version of our particle-PUM. Here, the particle distribution is fixed, we use 256 particles generated via a $(2, 3)$ Halton–sequence and a stretch parameter $\alpha = 1.5$. The weight function used in this experiment is the well known hat function. Furthermore, we use the complete polynomials V_i^C as local approximation spaces with degrees $p_i = p$ and p ranging from one to five. We choose tensor products of Legendre polynomials as local basis functions.

N	p	dof	$\|e\|_{L^\infty}$	ρ_{L^∞}	$\|e\|_{L^2}$	ρ_{L^2}	$\|e\|_{H^1}$	ρ_{H^1}
256	1	768	6.292_1	$***$	1.761_0	$***$	1.514_2	$***$
256	2	1536	5.094_0	-3.627	1.512_{-1}	-3.542	1.507_1	-3.329
256	3	2560	3.423_{-1}	-5.286	6.628_{-3}	-6.122	9.738_{-1}	-5.362
256	4	3840	1.837_{-2}	-7.207	2.920_{-4}	-7.700	5.732_{-2}	-6.986
256	5	5376	9.130_{-4}	-8.921	1.392_{-5}	-9.044	3.007_{-3}	-8.761

TABLE 5.5
Errors and convergence rates for Example 3.

For analytic solutions u we may estimate the error $u - \tilde{u}$ on every cover patch ω_i using the Taylor–series expansion, cf. §4.1.4. We obtain the local estimate

$$\|u - \tilde{u}\| = O\left(\|D^{(p+1)}u\| \frac{h^{p+1}}{(p+1)!}\right) \tag{5.10}$$

for an approximation $\tilde{u}$ of degree p, which leads to an exponential error estimate

$$\|u - \tilde{u}\| = O\left(\exp(-b\sqrt{\mathsf{dof}_p})\right) \tag{5.11}$$

on every cover patch ω_i. In our first experiment with the p-version we choose f and g such that the continuous solution to the problem is given as

$$u(x, y) = \exp(4(x + y)). \tag{5.12}$$

This solution is sufficiently smooth for (5.11) to hold. Therefore, we expect to see an exponential convergence in this experiment. The plots of the measured errors against the number of degrees of freedom are displayed in Figure 5.9 (left). Their gradients ρ as defined in (5.5) are given in Table 5.5. These results clearly show the expected exponential convergence behavior.

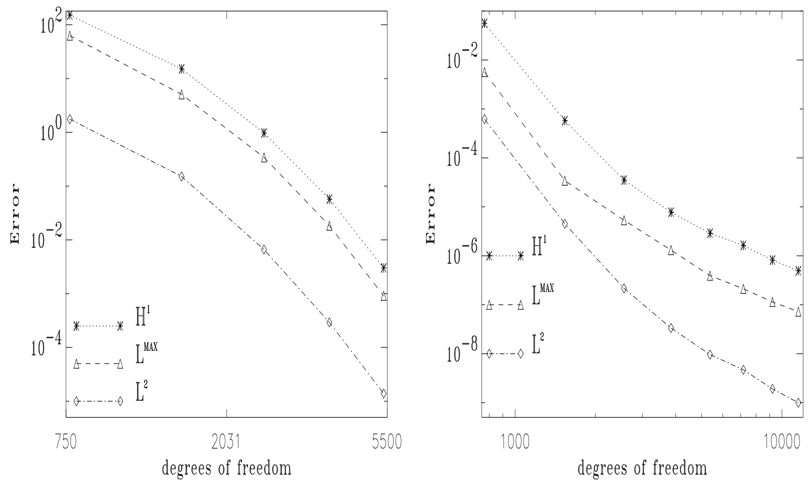

FIGURE 5.9. *Convergence history for Example 3 (left), Example 4(right).*

N	p	dof	$\|e\|_{L^\infty}$	ρ_{L^∞}	$\|e\|_{L^2}$	ρ_{L^2}	$\|e\|_{H^1}$	ρ_{H^1}
256	1	768	5.694_{-3}	$***$	6.208_{-4}	$***$	4.073_{-1}	$***$
256	2	1536	3.429_{-5}	-7.375	4.516_{-6}	-7.103	5.614_{-2}	-6.605
256	3	2560	5.335_{-6}	-3.642	2.165_{-7}	-5.947	5.765_{-4}	-5.479
256	4	3840	1.311_{-6}	-3.461	3.346_{-8}	-4.605	3.510_{-5}	-3.726
256	5	5376	3.931_{-7}	-3.580	9.608_{-9}	-3.708	7.747_{-6}	-2.926
256	6	7168	2.116_{-7}	-2.153	4.696_{-9}	-2.489	2.894_{-6}	-1.999
256	7	9216	1.141_{-7}	-2.460	1.908_{-9}	-3.583	8.173_{-7}	-2.743
256	8	11520	7.254_{-8}	-2.028	9.972_{-10}	-2.909	4.935_{-7}	-2.261

TABLE 5.6
Errors and convergence rates for Example 4.

Example 4 (p-version of particle-PUM, limited smoothness of solution). In a second experiment with the p-version of our particle-PUM we choose f and g such that the continuous solution to the problem is given by

$$u(x, y) = (x^2 + y^2)^{5/4} \,. \tag{5.13}$$

Here, the estimate (5.11) does not hold since (5.13) is not analytic. Therefore, we may not expect an exponential convergence behavior of the p-version of our particle-PUM in this experiment.

We use the same particle distribution, the same cover and the same local approximation spaces V_i as in Example 3 with p now ranging from one to eight. Figure 5.10 shows the solution (5.13) and the error for $p = 5$. The measured errors are displayed in Figure 5.9 (right). The corresponding rates ρ and the measured errors are given in Table 5.6. From these results we clearly see an algebraic convergence behavior of the p-version of our particle-PUM for problems with solutions with limited smoothness properties. A similar behavior can be observed when we employ a finite element discretization with the p-version on a fixed uniform grid. In fact, a convergence rate of $2 \cdot 5/4$ ($\rho_{H^1} = -2.5$) in the H^1-norm is well-known for the classical p-version of the FEM [5] for this problem and we (almost) recover this convergence behavior with our particle-PUM.

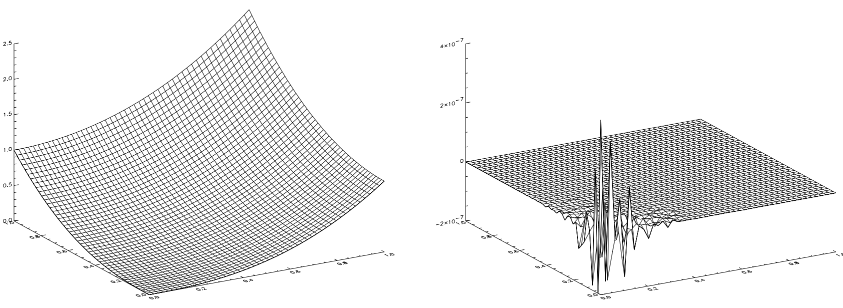

FIGURE 5.10. *Graph of solution (5.13) and graph of error for Example 4 with $p = 5$.*

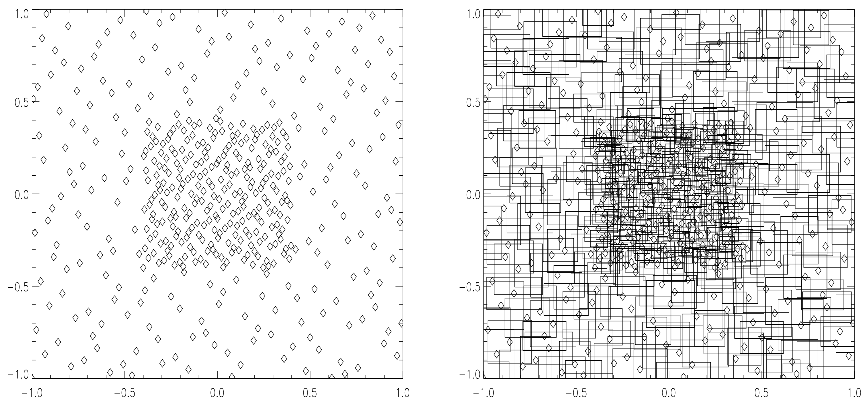

FIGURE 5.11. *Initial particle distribution (left) and the constructed cover (right) for (5.14).*

This experiment again demonstrates that the p-version of the particle-PUM based on a pseudo Monte Carlo particle distribution converges as well as the corresponding p-version of the finite element method.

In summary the results of our numerical experiments with the h- and p-version of our particle-PUM applied to elliptic problems clearly show that the method possesses the advantages of a particle method. At the same time, it exhibits a convergence behavior similar to that of the respective version of the finite element method.

5.3. Parabolic Problem. In the following we consider the heat-equation

$$\frac{\partial u}{\partial t} - \Delta u = 0 \text{ in } \Omega := (-0.5, 0.5)^2 \times (0, T) \tag{5.14}$$

with vanishing Neumann boundary conditions $\partial u / \partial n = 0$ on $\Gamma := \partial\Omega$ and initial value $u(x, y, 0) = \exp(-100(x^2 + y^2))$ in $\Omega := (-0.5, 0.5)^2$. For this parabolic model problem we use a random walk process as described in §2 to simulate the diffusive transport on the particle positions. Furthermore, we reduce the parabolic problem to a sequence of elliptic problems via a standard implicit Eulerian time discretization which gives the usual $O(\Delta t)$ convergence behavior in time. We employ the shape functions from the PUM based on the particle positions after the random walk process to discretize the elliptic problem in every time step. Here, the weight functions W_i

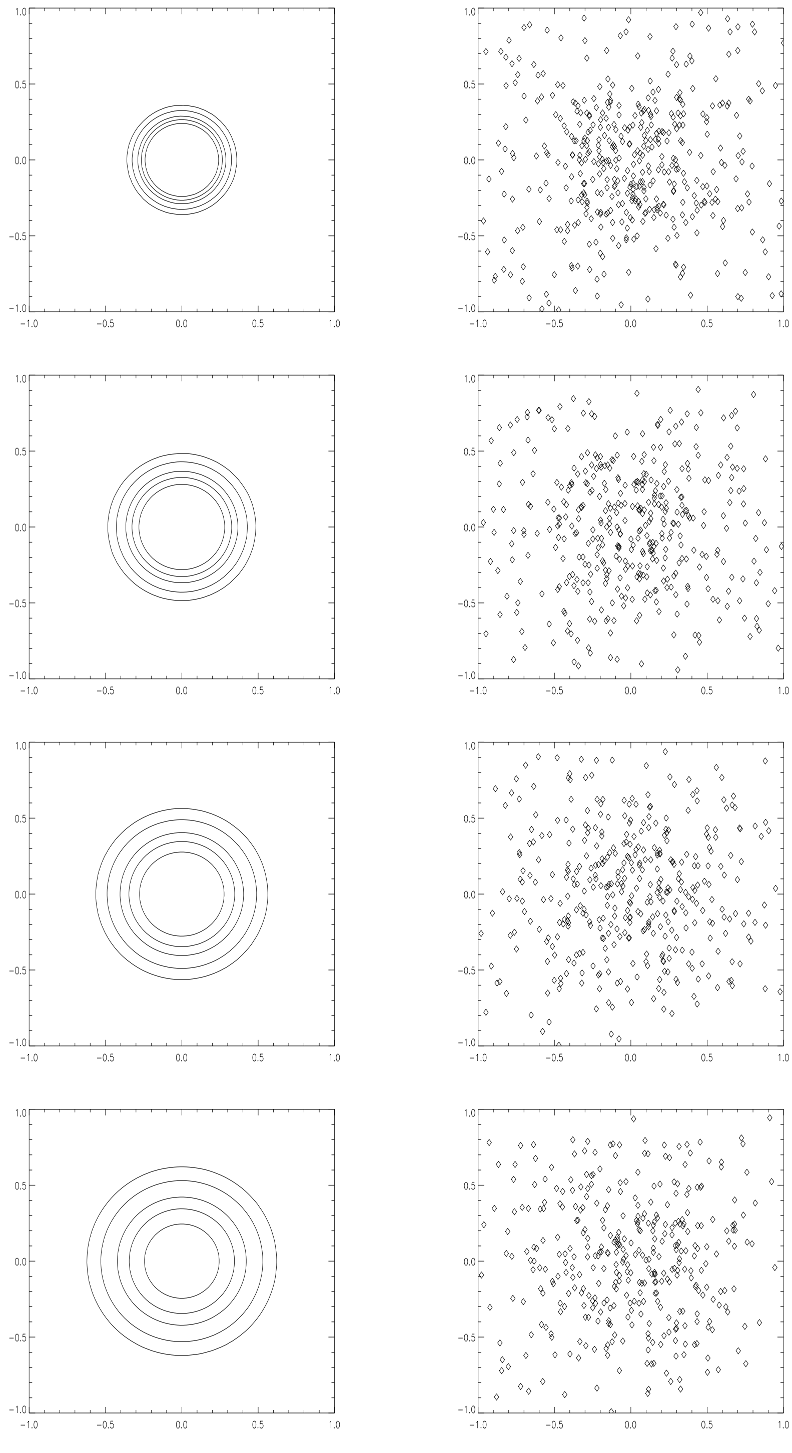

FIGURE 5.12. *Particle distributions and isolines* $(0.005, 0.01, 0.02, 0.03, 0.045)$ *of the approximate solution for (5.14) with* $\Delta t = 10^{-3}$ *at times* $(0.005, 0.015, 0.025, .035)$.

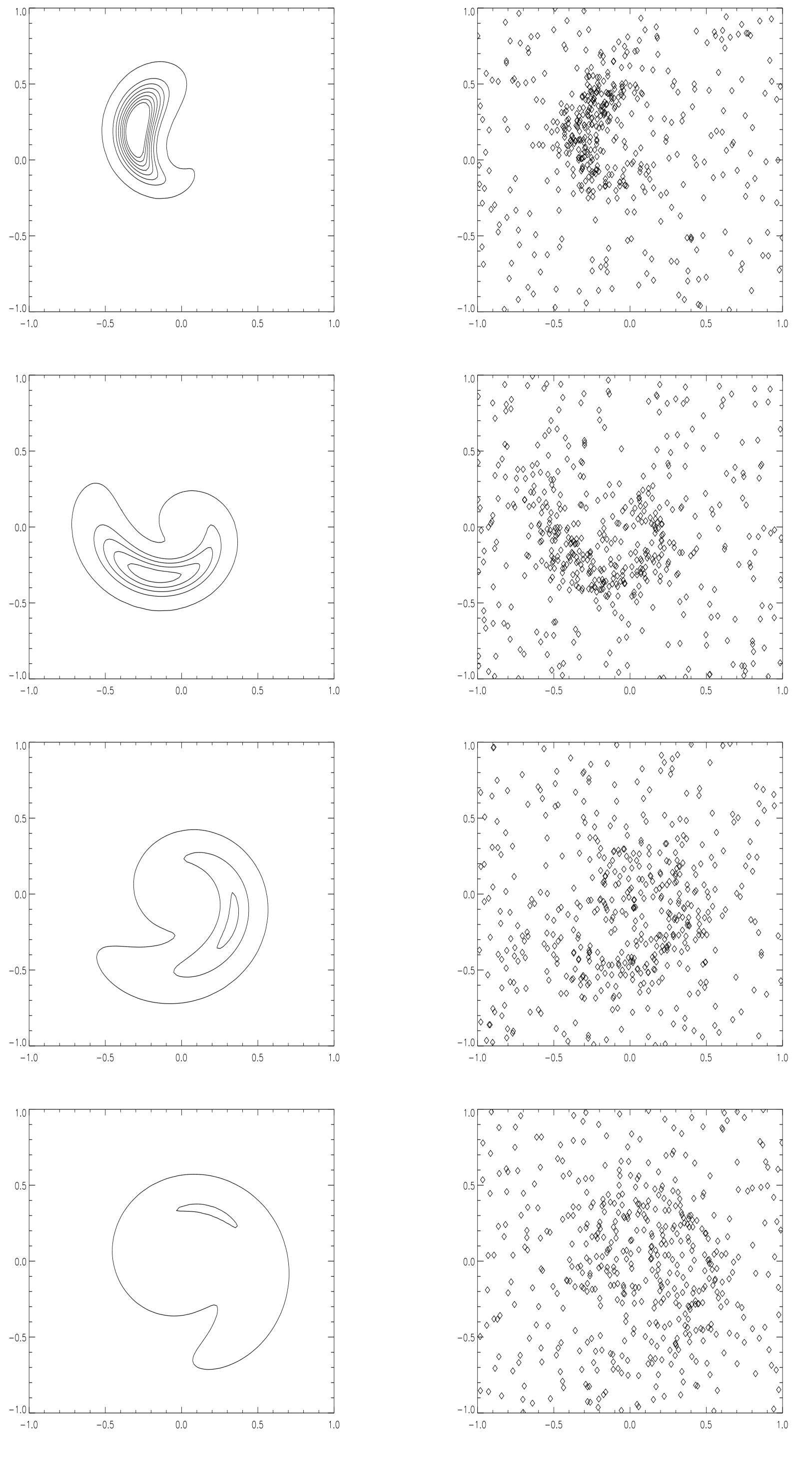

FIGURE 5.13. *Particle distributions and isolines* $(0.01, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2)$ *of the approximate solution for (5.15) with* $\Delta t = 2\pi/50$ *at times* $(5\Delta t, 10\Delta t, 15\Delta t, 20\Delta t)$.

are the hat functions. The stretch parameter in this experiment is $\alpha = 1.25$. As local approximation spaces V_i we use the complete quadratic polynomials V_i^C. We use a block-structure approach using an approximate support $\hat{\Omega}$ of the initial value as described in §5.1 to resolve the initial value. The particle distributions and isolines of the solution at times $(5\Delta t, 15\Delta t, 25\Delta t, 35\Delta t)$ with $\Delta t = 0.001$ are displayed in Figure 5.12. Here, we clearly see the diffusive particle transport and the reduction of the total number of particles over time. The diffusive particle transport adapts the distribution of the spatial degrees of freedom appropriately to the evolution of the solution. The solution gets smoother over time due to the diffusion. Here, the solution can be resolved to the same accuracy as the initial value with fewer unknowns, i.e. a successively reduced number of particles is sufficient to give a good approximation on Ω. We start with 469 particles (see Figure 5.11) and after 35 time steps there are only 374 left (see Figure 5.12). But the number of particles within the approximate support $\hat{\Omega}$ (while $\hat{\Omega} \subset \Omega$) is almost constant over time, i.e. the solution in every time step is represented by an almost constant number of degrees of freedom in space. This is due to the Monte Carlo simulation of the diffusion onto the particle distribution and the Neumann boundary conditions. Overall, we only reduce the computational work by applying the Monte Carlo step but the spatial resolution in every time step is sufficiently high to give a good approximation of the solution to (5.14). Due to use of quadratic polynomials V_i^C we have a "third order" convergence in space.

5.4. Convection-Diffusion Problem. In the following we consider the model problem

$$\frac{\partial u}{\partial t} - \frac{1}{125}\Delta u + \frac{1}{\sqrt{x^2+y^2}} \begin{pmatrix} -y \\ x \end{pmatrix} \nabla u = 0 \text{ in } \Omega := (-1,1)^2 \times (0,T) \tag{5.15}$$

with vanishing Neumann boundary conditions $\partial u/\partial n = 0$ on $\Gamma := \partial\Omega$ and initial value $u(x,y,0) = \exp(-100((x-\frac{1}{4})^2 + (y-\frac{1}{4})^2))$.

We discretize (5.15) via the particle method presented in §2. First, we split the operator $\frac{\partial}{\partial t} - \nu\Delta + v\nabla$ into a hyperbolic operator $\frac{\partial}{\partial t} + v\nabla$ and a parabolic operator $\frac{\partial}{\partial t} - \nu\Delta$. Then, we apply the corresponding particle method to the resulting hyperbolic problem (see §3, §5.1) with a fourth order Runge–Kutta method, and to the remaining parabolic problem (see §2, §5.3). Again, we use $\alpha = 1.25$, hat functions as weight functions W_i for the construction of the partition of unity $\{\varphi_i\}$ and a block-structure approach to resolve the initial value. As local approximation spaces V_i we use the complete quadratic polynomials V_i^C.

The particle distributions and isolines of the solution at $(5\Delta t, 10\Delta t, 15\Delta t, 20\Delta t)$ with $\Delta t = 2\pi/50$ are displayed in Figure 5.13. Here, we see that the diffusive particle movement together with the particle movement induced by the Lagrangian discretization of the convective terms of (5.15) causes the particles to follow the solution over time. Thus, we have roughly the same number of degrees of freedom for the resolution of the solution in every time step. Note that the Lagrangian treatment of the convection part circumvents the stability problems we might experience with a finite element discretization for the convective terms of (5.15). Further experiments showed the expected discretization error of "third order" in space and the expected $O(\Delta t)$ discretization error in time which is due to the first order implicit Eulerian time discretization for the parabolic subproblem.

6. Concluding remarks. We presented a meshless Lagrangian discretization method for instationary convection-diffusion problems. The method combines a particle approach and a meshless method for the generation of shape functions and the adaptation of the distribution of nodes (particles). We gave the details of the implementation and results of our numerical experiments. They showed that the convergence properties of the h- and p-version of our particle-PUM for elliptic problems are comparable to those of the h- and p-version of the FEM respectively. The convergence behavior in time of the proposed method for hyperbolic problems corresponds to the convergence behavior of the employed ODE solver. Furthermore, a fewer number of degrees of freedom in space than with a uniform Semi-Lagrangian method or even Euler method may be used since the degrees of freedom follow the solution over time. Hence, we do not have to resolve the complete computational domain with the same accuracy. Adaptive Semi-Lagrangian or Euler methods would have to apply refinement and coarsening in every time step whereas our method will transform an adaptively refined particle distribution for the initial value into an adaptively refined particle distribution in every time step without applying explicit refinement and coarsening.

Up to now, the method is in an experimental state and is surely not yet competitive to existing grid-based methods. The efficiency of the method has to be improved. The most room for improvement is during the construction of a cover, the integration of the shape functions and the iterative solution of the resulting linear system, especially in the case of an elliptic problem with Dirichlet boundary conditions where we have to solve a saddle point problem. Furthermore, an adaptive refinement, i.e. the proper insertion of particles (h-version) as well as the local increment of the degree of the polynomials used (p-version) or both together (hp-version), needs the development of reliable local error indicators and estimators. Moreover, an adaptive control of the time step size, the use of local time stepping methods and the choice of more sophisticated operator splittings could further improve the performance of the method for time-dependent problems.

In summary, much additional work remains to be done, but the results presented in this paper demonstrate the promising properties of the particle-PUM and encourage its further development.

REFERENCES

[1] H. Babovsky, *Die Boltzmann–Gleichung*, B. G. Teubner, Stuttgart–Berlin, 1998.

[2] I. Babuška, *The Finite Element Method with Lagrangian multipliers*, Numer. Math., 20 (1973), pp. 179–192.

[3] I. Babuška and J. M. Melenk, *The Partition of Unity Finite Element Method: Basic Theory and Applications*, Comput. Meths. Appl. Mech. Engrg, 139 (1996), pp. 289–314. Special Issue on Meshless Methods.

[4] ———, *The Partition of Unity Method*, Int. J. Numer. Meths. Eng., 40 (1997), pp. 727–758.

[5] I. Babuška and M. Suri, *The p and hp Versions of the Finite Element Method, Basic Principles and Properties*, SIAM Review, 36 (1994), pp. 578–632.

[6] M. J. Baines, *Moving Finite Elements*, Monographs on Numerical Analysis, Oxford Science Publications, 1994.

[7] R. E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations – Users' Guide 7.0*, vol. 15 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1994.

[8] P. Bastian, *Parallele Adaptive Mehrgitterverfahren*, Skripten zur Numerik, B. G. Teubner, Stuttgart, 1996.

[9] P. Bastian, K. Birken, K. Johannsen, S. Lang, K. Eckstein, N. Neuss, H. Rentz-Reichert, and C. Wieners, *UG - A Flexible Software Toolbox for Solving Partial Differential Equations*, Comput. and Visual. in Sci., 1 (1997), pp. 27–40.

[10] R. Beck, B. Erdmann, and R. Roitzsch, *Kaskade 3.0 Users Guide*, Tech. Rep. 95-11, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1995.

[11] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, *Meshless Methods:*

An Overview and Recent Developments, Comput. Meths. Appl. Mech. Engrg., 139 (1996), pp. 3–47. Special Issue on Meshless Methods.

[12] T. Belytschko, Y. Y. Lu, and L. Gu, *Element-free Galerkin methods*, Int. J. Numer. Meths. Eng., 37 (1994), pp. 229–256.

[13] M. J. Berger and J. Oliger, *An Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, J. Comput. Phys., 53 (1984), pp. 484–512.

[14] D. Braess, *Finite Elemente*, Springer, Berlin–Heidelberg, 2nd ed., 1997.

[15] J. H. Bramble, *The Lagrange Multiplier Method for Dirichlet's Problem*, Math. Comput., 37 (1981), pp. 1–11.

[16] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the Inexact Uzawa Algorithm for Saddle Point Problems*, J. Numer. Anal., 34 (1997), pp. 1072–1092.

[17] ———, *Uzawa Type Algorithms for Nonsymmetric Saddle Point Problems*. to appear Math. Comp., 1997.

[18] J. H. Bramble, J. E. Pasciak, and P. S. Vassilevski, *Computational Scales of Sobolev Norms with Application to Preconditioning*. to appear in Math. Comp.

[19] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, Berlin–Heidelberg, 1991.

[20] W. Dai and P. Woodward, *Extension of the Piecewise Parabolic Method to Multidimensional Ideal Magnetohydrodynamic*, J. Comput. Physics, 115 (1994), pp. 485–514.

[21] P. J. Davis, *Interpolation and Approximation*, Dover, 1975.

[22] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, *Toward a univeral hp Adaptive Finite Element Strategy, part 1. Constrained Approximation and Data Structure*, Comput. Meth. Appl. Mech. Engrg., 77 (1989), pp. 79–112.

[23] P. Deufelhard, P. Leinen, and H. Yserantant, *Concepts of an Adaptive Hierarchical Finite Element Code*, IMPACT of Comput. in Sci. and Eng., 1 (1989), pp. 3–35.

[24] G. A. Dilts, *Moving-Least-Square-Particle Hydrodynamics I: Consistency and Stability*. submitted to Int. J. Numer. Meths. Eng., 1996.

[25] J. Dolbow and T. Belytschko, *Numerical Integration of the Galerkin Weak Form in Meshfree Methods*, Computational Mechanics, (1999), pp. 219–230.

[26] C. A. M. Duarte, *A Review of Some Meshless Methods to Solve Partial Differtial Equations*, Tech. Rep. 95-06, TICAM, University of Texas, 1995.

[27] C. A. M. Duarte and J. T. Oden, *Hp clouds – A Meshless Method to Solve Boundary Value Problems*, Num. Meth. for PDE, 12 (1996), pp. 673–705. also as Tech. Rep. 95-05, TICAM, University of Texas, 1995.

[28] C. Franke and R. Schaback, *Convergence Orders of Meshless Collocation Methods using Radial Basis Functions*, Adv. in Comput. Math., 8 (1998), pp. 381–399.

[29] ———, *Solving Partial Differential Equations by Collocation using Radial Basis Functions*, Appl. Math. and Comput., 93 (1998), pp. 73–82.

[30] T. Gerstner and M. Griebel, *Numerical Integration using Sparse Grids*, Num. Alg., 18 (1998), pp. 209–232. also as Tech. Rep. 553 SFB 256, Universität Bonn, 1998.

[31] R. A. Gingold and J. J. Monaghan, *Smoothed Particle Hydrodynamics: Theory and Application to non-spherical stars*, Mon. Not. R. Astr. Soc., 181 (1977), pp. 375–389.

[32] ———, *Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics*, Journ. of Comput. Physics, 46 (1982), pp. 429–453.

[33] R. T. Glassey, *The Cauchy Problem in Kinetic Theory*, SIAM, Philadelphia, 1996.

[34] T. Grauschopf, M. Griebel, and H. Regler, *Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic Multigrid Coarsening for Second Order Elliptic PDEs*, App. Num. Math., 23 (1997). also as Tech. Rep. SFB 342/02/96A, Institut fr Informatik, TU München, 1996.

[35] E. Hlawka and R. Mück, *Über eine Transformation von gleichverteilten Folgen II*, Computing, 9 (1972), pp. 127–138.

[36] J. Hoschek and D. Lasser, *Grundlagen der geometrischen Datenverarbeitung*, B. G. Teubner, Stuttgart, 1992.

[37] K. Hvistendahl Karlsen, K. Brusdal, H. K. Dahle, S. Evje, and K.-A. Lie, *The Corrected Operator Splitting Approach applied to an Advection-Diffusion Problem*. to appear in Comput. Meth. Appl. Mech. Engrg.

[38] K. Hvistendahl Karlsen and N. H. Risebro, *An Operator Splitting Method for Nonlinear Convection-Diffusion Equations*, Numer. Math., 77 (1997), pp. 365–382.

[39] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1987.

[40] E. J. Kansa, *Multiquadratics—A Scattered Data Approximation Scheme with Applications to Computational Fluid-Dynamics—I Surface Approximations and Partial Derivative Esti-*

mates, Comput. and Math. w. Appl., 19 (1990), pp. 127–145.

[41] ———, *Multiquadratics—A Scattered Data Approximation Scheme with Applications to Computational Fluid-Dynamics—II Solutions to Parabolic, Hyperbolic and Elliptic Partial Differential Equations*, Comput. and Math. w. Appl., 19 (1990), pp. 147–161.

[42] D. E. Knuth, *The Art of Computer Programming*, vol. 3 Searching and Sorting, Addison Wesley, Reading, Massachusetts, Second ed., 1998.

[43] A. Kunoth, *Multilevel Preconditioning – Appending Boundary Conditions by Lagrange Multipliers*, Adv. Comput. Math., 4 (1995), pp. 145–170. Special Issue on Multiscale Methods.

[44] P. Lancaster, *Moving Weighted Least–Squares Methods*, in Polynomial and Spline Approximation, B. N. Sahney, ed., vol. 49 of NATO Adv. Stud. Series C, Dordrecht Boston London, 1979, D. Reidel, pp. 103–120. Proc. 1978 NATO Adv. Stud. Inst.

[45] P. Lancaster and K. Salkauskas, *Surfaces Generated by Moving Least Squares Methods*, Math. Comput., 37 (1981), pp. 141–158.

[46] J. Lang, *Adaptive Finite Element Methods for Reaction-Diffusion Equations*, App. Num. Math., (1998), pp. 105–116.

[47] U. Langer and W. Queck, *Preconditioned Uzawa–type Iterative Methods for Solving Mixed Finite Element Equations*, Tech. Rep. 3/1987, Technische Universität Karl–Marx–Stadt, 1987.

[48] D. Lanser and J. G. Verwer, *Analysis of Operator Splitting for Advection-Diffusion-Reaction Problems from Air Pollution Modelling*, CWI Report MAS-9805, Centrum voor Wiskunde en Informatica, Amsterdam, 1998.

[49] T. Liska and J. Orkisz, *The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics*, Comput. and Struct., 11 (1980), pp. 83–95.

[50] W. K. Liu, S. Jun, and Y. F. Zhang, *Reproducing Kernel Particle Methods*, Int. J. for Numer. Meths. in Fluids, 20 (1995), pp. 1081–1106.

[51] L. B. Lucy, *A Numerical Approach to the Testing of the Fission Hypothesis*, Astronomical Journal, 82 (1977), pp. 1013–1024.

[52] Y. Maday, C. Mavriplis, and A. T. Patera, *Nonconforming Mortar Element Methods: Application to Spectral Discretizations*, in Proc. Domain Decomposition Methods 2, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1989, SIAM, pp. 392–418.

[53] K. Miller, *Moving Finite Elements II*, SIAM J. Num. Anal., 18 (1981), pp. 1033–1057.

[54] K. Miller and R. N. Miller, *Moving Finite Elements I*, SIAM J. Num. Anal., 18 (1981), pp. 1019–1032.

[55] J. J. Monaghan, *An Introduction to SPH*, Comput. Phys. Comm., 48 (1977), pp. 89–96.

[56] ———, *Why Particle Methods Work*, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 422–433.

[57] K. Nanbu, *Direct Simulation Scheme derived from the Boltzmann Equation*, J. Phys. Soc. Japan, 49 (1980), pp. 20–49.

[58] ———, *Theoretical Basis on the Direct Simulation Monte Carlo Method*, in Rarefied Gas Dynamics, V. Boffi and C. Cercignani, eds., vol. 1, Teubner, Stuttgart, 1986.

[59] S. V. Nepomnyaschikh, *Decomposition and Fictitious Domain Methods for Elliptic Boundary Value Problems*, in Proc. Domain Decomposition Methods 5, T. F. Chan, D. E. Keyes, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, eds., SIAM, 1992.

[60] H. Neunzert, A. Klar, and J. Struckmeier, *Particle Methods: Theory and Applications*, Tech. Rep. 95-153, Arbeitsgruppe Technomathematik, Universität Kaiserslautern, 1995.

[61] H. Neunzert and J. Struckmeier, *Particle Methods for the Boltzmann Equation*, Acta Numerica, (1995), pp. 417–457.

[62] H. Niederreiter, *Random Number Generation and Quasi–Monte Carlo Methods*, SIAM, Philadelphia, 1992.

[63] E. Novak and K. Ritter, *High Dimensional Integration of Smooth Functions over Cubes*, Numer. Math., 75 (1996), pp. 79–97.

[64] J. T. Oden, T. Strouboulis, and P. Devloo, *Adaptive Finite Element Methods for the Analysis of Inviscid Compressible Flow: Part I Fast Refinement/Unrefinement and Moving Mesh Methods for Unstructured Meshes*, Comput. Meths. Appl. Mech. Engrg., 59 (1986), pp. 327–362.

[65] E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow*, Elsevier, 1987.

[66] J. Ruge and K. Stüben, *Efficient Solution of Finite Difference Equations by Algebraic Multigrid*, Arbeitspapiere 89, GMD, Bonn, 1984.

[67] ———, *Algebraic Multigrid*, Arbeitspapiere 210, GMD, Bonn, 1986.

[68] M. A. Schweitzer, *Ein Partikel–Galerkin–Verfahren mit Ansatzfunktionen der Partition of Unity Method*, Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, 1997.

[69] D. Shepard, *A Two-Dimensional Interpolation Function for Irregularly Spaced Data*, in Proceedings 1968 ACM Nat. Conf., ACM, 1968, pp. 517–524.
[70] G. Strang, *On the Construction and Comparison of Difference Schemes*, J. Numer. Anal., 5 (1968), pp. 506–517.
[71] J. W. Swegle, S. W. Attaway, F. J. Mello, and D. L. Hicks, *An Analysis of the Smoothed Particle Hydrodynamics*, Tech. Rep. SAND93-2513 UC-705, Sandia National Laboratories, 1994.
[72] C. B. Vreugdenhil and B. Koren, eds., *Numerical Methods for Advection–Diffusion Problems*, vol. 45 of Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig, Wiesbaden, 1993.
[73] H. Wendland, *Meshless Galerkin Methods using Radial Basis Functions*, Math. of Comput., 68 (1999), pp. 1521–1531.
[74] H. Yserantant, *A New Class of Particle Methods*, Numer. Math., 76 (1997), pp. 87–109.
[75] ———, *A Particle Model of Compressible Fluids*, Numer. Math., 76 (1997), pp. 111–142.
[76] *Special Issue on Meshless Methods*, vol. 139, Comput. Meths. Appl. Mech. Engrg, 1996.