Adaptive Meshfree Method of Backward Characteristics for Nonlinear Transport Equations

Jörn Behrens, Armin Iske, and Martin Käser

Technische Universität München, D-80290 München, Germany

Abstract. In previous work, a new adaptive meshfree advection scheme for numerically solving *linear* transport equations has been proposed. The scheme, being a combination of an adaptive semi-Lagrangian method and local radial basis function interpolation, is essentially a *method of backward characteristics.* The adaptivity of the meshfree advection scheme relies on customized rules for the refinement and coarsening of scattered nodes. In this paper, the method is extended to *nonlinear* transport equations. To this end, in order to be able to model shock propagation, an artificial viscosity term is added to the scheme. Moreover, the local interpolation method and the node adaption rules are modified accordingly. The good performance of the resulting method is finally shown in the numerical examples by using two specific nonlinear model problems: *Burgers equation* and the *Buckley-Leverett equation*, the latter describing a two-phase fluid flow in a porous medium.

1 Introduction

Many physical phenomena in transport processes are described by time-dependent hyperbolic conservation laws. Their governing equations have the form

$$\frac{\partial u}{\partial t} + \nabla f(u) = 0 \tag{1}$$

where for some domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, and a compact time interval $I = [0, T]$, $T > 0$, the function $u : I \times \Omega \to \mathbb{R}$ is unknown. Moreover, $f(u) = (f_1(u), \dots, f_d(u))^T$ denotes the *flux tensor.* In this paper, we consider numerically solving (1) on given initial conditions

$$u(0, x) = u_0(x), \qquad \text{for } x \in \Omega = \mathbb{R}^d, \tag{2}$$

and for *nonlinear* flux functions f.

In previous work [3,4], a new adaptive meshfree advection scheme has been proposed for numerically solving (1) for the special case where

$$f(u) = a \cdot u, \tag{3}$$

in which case we obtain the *linear* (passive) advection equation

$$\frac{\partial u}{\partial t} + a \cdot \nabla u = 0 \tag{4}$$

provided that the given velocity field

$$a = a(t, x) \in \mathbb{R}^d, \qquad t \in I, \, x \in \Omega,$$

is divergence-free.

The method in [3,4] is a combination of an adaptive semi-Lagrangian method (ASLM) [1,2] and the meshfree radial basis function interpolation. The resulting advection scheme is used for the simulation of tracer transportation in the arctic stratosphere [4]. We remark that the scheme in [3,4] is a *method of characteristics*, see [7,12]. Indeed, the characteristic curves of (4) coincide with the trajectories of fluid particles, and the meshfree ASLM in [3,4] captures the flow of particles along their characteristic curves. This is accomplished by computing backward trajectories for a finite set of current particles (nodes) at each time step, whereas the node set is adaptively modified during the simulation.

In this paper, an adaptive meshfree method of backward characteristics is designed for the purpose of numerically solving *nonlinear* equations of the form (1). In contrast to the linear case, a nonlinear flux function f usually leads to *discontinuities* in the solution u, *shocks*, as observed in many relevant applications in fluid dynamics, meteorology, astrophysics, petroleum reservoir simulation, etc. The characteristics-based method in [3,4] becomes unwieldy or impossible in nonlinear problems where the evolution of the flow along the characteristic curves may be much more complicated or characteristic curves may even be not defined, cf. [9], Subsection 6.3.1. Therefore, we apply a *vanishing viscosity* approach yielding the modified advection-diffusion equation

$$\frac{\partial u}{\partial t} + \nabla f(u) = \epsilon \cdot \Delta u, \tag{5}$$

with $\epsilon > 0$ being the artificial *diffusion coefficient.*

When it comes to extending the advection scheme of [3,4], the local interpolation scheme is to be modified accordingly. The extension of the advection scheme is subject of the discussion in Section 2. The two remaining major ingredients, local *thin plate spline* interpolation, and the customized adaption rules, are then explained in the Sections 3 and 4.

Finally, the good performance of the resulting adaptive and meshfree method of backward characteristics is shown by numerical results in Section 5, where we consider using two different nonlinear model problems: *Burgers equation*, a standard test case, where

$$f(u) = \frac{1}{2} u^2 \cdot r, \tag{6}$$

with flow direction $r \in \mathbb{R}^d$, and the *Buckley-Leverett equation*, whose flux function has the form

$$f(u) = \frac{u^2}{u^2 + \mu(1-u)^2} \cdot r. \tag{7}$$

The Buckley-Leverett equation models the saturation of a two-phase flow in a porous medium when neglecting gravitational forces or capillary effects. In this case, the value of μ in (7) is the ratio of the two different fluid's viscosities. This model problem is typically encountered in *oil reservoir modelling*. Details on this particular application are explained in the final Section 5.

2 Meshfree Method of Backward Characteristics

For the special case of passive advection (3), the scalar function u is constant along *trajectories*, *streamlines*, whose shapes are entirely and uniquely determined by the (given) velocity field $a \equiv a(t,x)$. Likewise, in the nonlinear case the solution u is constant along trajectories of fluid particles, characteristic curves. In contrast to the linear case of passive advection, these characteristic curves do, however, depend on u.

In order to numerically solve the modified equation (5), the adaptive meshfree semi-Lagrangian method of [3,4] is extended as follows. At each time step $t \to t+\tau$, with $\tau > 0$ being the time step size, the values $u(t+\tau,\xi)$ at a current finite set Ξ of nodes, each of which corresponding to a flow particle, are computed from the previous values $u(t,\xi)$, $\xi \in \Xi$. Initially, the set Ξ^0 is randomly chosen.

Starting point of the method is the Lagrangian form of (5),

$$\frac{du}{dt} = \epsilon \cdot \varDelta u,$$

where $\frac{du}{dt} = \frac{\partial u}{\partial t} + \nabla f(u)$ is the material derivative. This leads us to the discretization

$$\frac{u(t+\tau,\xi) - u(t,x^-)}{\tau} = \epsilon \varDelta u(t,x^-),$$

where $x^- \equiv x^-(\xi)$ is the *upstream location* of the node ξ. Note that a particle located at the upstream point x^- at time t moves along its trajectory and arrives at ξ at time $t+\tau$. Having computed x^- for any $\xi \in \Xi$, the desired approximation of $u(t+\tau,\xi)$ would thus be given explicitly by

$$u(t+\tau,\xi) = u(t,x^-) + \tau \cdot \epsilon \varDelta u(t,x^-), \qquad \text{for } \xi \in \Xi. \tag{8}$$

But on given $\xi \in \Xi$, the *exact* location of the upstream point x^- is usually not known. A linearized approximation of x^- is given by

$$\tilde{x} = \xi - \beta,$$

where $\beta = \tau v$ and $v = \frac{\partial f(u)}{\partial u}$ is the advection velocity, i.e.

$$v(t,x) = \dot{x} = \frac{dx}{dt}. \tag{9}$$

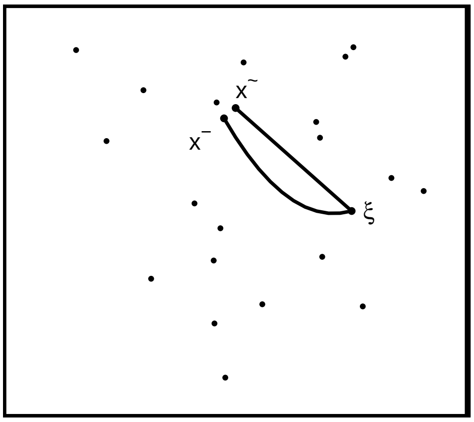

Fig. 1. Upstream point x^- of the node $\xi \in \Xi$, and its approximation $\tilde{x} \approx x^-$.

In order to compute $\tilde{x}$, we need to solve the ordinary differential equation (9). Figure 1 displays the backward trajectory of a node $\xi \in \Xi$, its corresponding upstream point x^-, along with a linear approximation of the trajectory, leading to $\tilde{x} \approx x^-$.

For computing $\tilde{x}$, our implementation utilizes a fixed point iteration, based on the midpoint rule, already used in the seminal paper on semi-Lagrangian discretization by Robert [16]:

$$\beta^{(k+1)} = \tau \cdot v\left(t + \frac{\tau}{2}, \xi - \frac{\beta^{(k)}}{2}\right). \tag{10}$$

Note that the above iteration (10) relies on the evaluation of v at the intermediate time $t + \frac{\tau}{2}$. In the situation of passive advection, this can be accomplished by the evaluation of the given wind field $a \equiv v$. But in the nonlinear case, the velocity v does also depend on the solution u. In order to compute $v\left(t + \frac{\tau}{2}, \xi - \frac{\beta^{(k)}}{2}\right)$ we employ the following extrapolation scheme.

$$v\left(t + \frac{\tau}{2}, \cdot\right) = \frac{3}{2}v(t, \cdot) - \frac{1}{2}v(t - \tau, \cdot). \tag{11}$$

Initially, in order to obtain the required values of $u(\tau, \cdot)$ from the given initial conditions (2), we use a generalized *two-level* Lax-Friedrich scheme on $\Xi^0 \equiv \Xi^\tau$.

Having computed the values $u(t+\tau, \xi)$, for all $\xi \in \Xi$, via (8), the current node set $\Xi \equiv \Xi^t$ (at time t) is finally modified by the removal (coarsening), and the insertion (refinement) of nodes, yielding a new node set $\Xi \equiv \Xi^{t+\tau}$ (at time $t+\tau$). The adaption of the nodes relies on a customized a posteriori error indicator, to be explained in Section 4.

We finally remark that the characteristics-based discretization scheme, as introduced in this section, has been theoretically analyzed by Falcone and Ferretti [10]. Their concise convergence analysis shows that the semi-Lagrangian method is of second order in time and space, provided that the interpolation method is of second order.

3 Local Interpolation using Thin Plate Splines

In this section, we are concerned with computing approximations for the values $u(t,\cdot), \varDelta u(t,\cdot)$ in the advection step (8), and $v(t,\cdot), v(t-\tau,\cdot)$ in the extrapolation (11). To this end, we work with a *local* interpolation scheme, which first collects on given $x \in \varOmega$ a set $\mathcal{N}_x \equiv \mathcal{N}_x^t$ of current neighbours (at time t) in the local neighbourhood of x, before the (known) function values of u at these neighbouring points are used for computing the approximations of $u(t,\cdot), \varDelta u(t,\cdot), v(t,\cdot), v(t-\tau,\cdot)$. But this requires some preparations. Therefore, we defer details to later in this section.

As already observed in [3,4], the interpolation is critical for the advection method's performance in terms of its efficiency and approximation quality. Indeed, the interpolation scheme does not only affect the evaluation of the model $u(x) \equiv u(t,x)$, but also the adaption rules, to be explained in the following Section 4, do heavily rely on the interpolation. Altogether, a reliable and robust interpolation scheme of good approximation quality is required.

As explained in the previous work [3,4], *thin plate splines* provide suitable and powerful meshfree methods for scattered data interpolation. But the setting in [3,4] needs to be extended here. According to the general framework of thin plate spline interpolation, dating back to Duchon [8], we work with interpolants of the form

$$s(y) = \sum_{\nu \in \mathcal{N}_x} \lambda_\nu \phi_k(\|y-\nu\|) + \sum_{|\alpha| \le k} \mu_\alpha y^\alpha, \tag{12}$$

in order to solve interpolation problems of the form

$$s(\nu) = u(\nu), \qquad \text{for all } \nu \in \mathcal{N}_x. \tag{13}$$

In (12), the *radial basis function* $\phi_k(r) = r^{2k}\log(r)$, $k \ge 1$, is referred to as *thin plate spline*, and $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^d$. Moreover, $\alpha = (\alpha_1, \ldots, \alpha_d)$ in (12) is a multi-index, a d-tuple of non-negative integers, with absolute value $|\alpha| = \alpha_1 + \ldots + \alpha_d$, and where

$$y^\alpha = y_1^{\alpha_1} \cdot \cdots \cdot y_d^{\alpha_d} \qquad \text{for } y = (y_1, \ldots, y_d)^T \in \mathbb{R}^d.$$

The above $N = \#\mathcal{N}_x$ interpolation conditions in (13) constitute a linear system of N equations with $N+Q$ unknowns in the coefficient vectors $\lambda = (\lambda_\nu)_{\nu \in \mathcal{N}_x} \in \mathbb{R}^N$ of the *major part* and $\mu = (\mu_\alpha)_{|\alpha| \le k} \in \mathbb{R}^Q$ of the *polynomial* *part* of s in (12), where $Q = \binom{k+d}{d}$ is the dimension of the linear space Π_k^d of all real-valued polynomials in d variables and of degree at most k.

In order to eliminate the Q additional degrees of freedom, the coefficients λ_ν in (12) are subject to the additional Q side conditions

$$\sum_{\nu \in \mathcal{N}_x} \lambda_\nu \nu^\alpha = 0, \qquad \text{for all } |\alpha| \le k. \tag{14}$$

Altogether, solving (13) under constraints (14) leads us to the linear system

$$\begin{bmatrix} A_{\mathcal{N}_x} & P_{\mathcal{N}_x} \\ P_{\mathcal{N}_x}^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} (u(\nu))_{\nu \in \mathcal{N}_x} \\ 0 \end{bmatrix}, \tag{15}$$

where

$$A_{\mathcal{N}_x} = (\phi_k(\|\nu - \nu'\|))_{\nu,\nu' \in \mathcal{N}_x} \in \mathbb{R}^{N \times N} \text{ and } P_{\mathcal{N}_x} = (\nu^\alpha)_{\nu \in \mathcal{N}_x; |\alpha| \le k} \in \mathbb{R}^{N \times Q}.$$

As to the well-posedness of thin plate spline interpolation, we remark that there always exists an interpolant s of the form (12) satisfying (13). Moreover, s is unique provided that the points in $\mathcal{N}_x$ are Π_k^d-unisolvent, i.e.

$$p(\nu) = 0 \quad \text{for all} \quad \nu \in \mathcal{N}_x \qquad \Longrightarrow \qquad p \equiv 0 \tag{16}$$

for $p \in \Pi_k^d$. In this case, the linear system (15) has a unique solution, and thus the thin plate spline interpolation scheme achieves to reconstruct polynomials in Π_k^d exactly. Note that (16) is a very mild side condition. E.g., for $k = 1$ this requires that the points in $\mathcal{N}_x$ must not all lie on a straight line.

For further details on thin plate spline interpolation, including their optimality properties, and alternative choices for *radial basis functions* ϕ, we refer to the recent tutorial paper [14].

Now let us finally turn to the approximation of the values $u(t,\cdot), \Delta u(t,\cdot), v(t,\cdot), v(t-\tau,\cdot)$. In order to compute an approximation for $v(t,x)$, on given $x \in \Omega$, we first consider solving the interpolation problem

$$s(\nu) = v(t,\nu), \qquad \text{for all } \nu \in \mathcal{N}_x^t,$$

by using the ansatz (12) for s and with $k = 1$. This then gives us by $s(x)$ the desired approximation of $v(t,x)$, i.e. $s(x) \approx v(t,x)$. Likewise, the approximation of $v(t-\tau,\cdot)$ is computed by solving

$$s(\nu) = v(t-\tau,\nu), \qquad \text{for all } \nu \in \mathcal{N}_x^{t-\tau},$$

using exactly the same approach, but for the previous set $\mathcal{N}_x^{t-\tau}$ of neighbours.

As to the approximation of $\Delta u(t,\tilde{x})$, this requires using a *smoother* instance of ϕ_k in (12), i.e. with $k > 1$, since $\Delta\phi_1(\|x\|)$ has a singularity at zero. We prefer to work with $\phi_2(r) = r^4 \log(r)$, whose Laplacian $\Delta\phi_2(\|x\|)$

is well-defined on all of $\mathbb{R}^d$. This amounts to first solving the interpolation problem

$$s(\nu) = u(t, \nu), \qquad \text{for all } \nu \in \mathcal{N}_{\tilde{x}}^t,$$

using the ansatz (12) with $k = 2$, which provides by

$$\Delta s(y) = \sum_{\nu \in \mathcal{N}_{\tilde{x}}^t} \lambda_\nu \Delta \phi_2(\|y - \nu\|) + 2\left(\mu_{(2,0,\ldots,0)} + \mu_{(0,\ldots,0,2)}\right)$$

the desired approximation $\Delta s(\tilde{x}) \approx \Delta u(t, \tilde{x})$. Moreover, $s(\tilde{x}) \approx u(t, \tilde{x})$. In our numerical examples, however, we kept on using the basis function ϕ_1 for computing an approximation of $u(t, \tilde{x})$. This helps to avoid undesired oscillations near the shocks.

4 Adaption Rules

4.1 Error Indication

An effective strategy for the adaptive modification of the nodes requires well-motivated refinement and coarsening rules as well as a customized error indicator. We understand the error indicator $\eta : \Xi \to [0, \infty)$ as a function of the current node set $\Xi \equiv \Xi^t$ (at time t) which serves to assign a *significance* value $\eta(\xi)$ to each $\xi \in \Xi$. The value $\eta(\xi)$ is required to reflect the local approximation quality of the interpolation around $\xi \in \Xi$. The significances $\eta(\xi)$, $\xi \in \Xi$, are then used in order to flag single nodes $\xi \in \Xi$ as “to be refined” or “to be coarsened” according to the following criteria.

Definition 1. Let $\eta^* = \max_{\xi \in \Xi} \eta(\xi)$, and let $\theta_{\text{crs}}, \theta_{\text{ref}}$ be two tolerance values satisfying $0 < \theta_{\text{crs}} < \theta_{\text{ref}} < 1$. We say that a node $\xi \in \Xi$ is `to be refined`, iff $\eta(\xi) > \theta_{\text{ref}} \cdot \eta^*$, and ξ is `to be coarsened`, iff $\eta(\xi) < \theta_{\text{crs}} \cdot \eta^*$.

In our numerical examples typical choices for the relative tolerance values are $\theta_{\text{crs}} = 0.001$ and $\theta_{\text{ref}} = 0.2$. Note that a node ξ cannot be refined and be coarsened at the same time; in fact, it may neither be refined nor be coarsened.

Now let us turn to the definition of the error indicator η. We follow along the lines of [13], where a local scheme for the detection of discontinuities of a surface from scattered data was developed, and we let

$$\eta(\xi) = |u(\xi) - s(\xi)|,$$

where the thin plate spline interpolant $s \equiv s_{\mathcal{N}}$ matches current values of $u \equiv u(t, \cdot)$ at a neighbouring set $\mathcal{N} \equiv \mathcal{N}(\xi) \subset \Xi \setminus \{\xi\}$ of current nodes, i.e. $s(\nu) = u(\nu)$ for all $\nu \in \mathcal{N}$. In our numerical examples, we preferred to use the thin plate spline $\phi_1(r) = r^2 \log(r)$, and thus the ansatz (12) with $k = 1$. This particular interpolation scheme achieves to reconstruct linear polynomials. In this case, the value $\eta(\xi)$ vanishes whenever u is linear around ξ. Moreover, the value $\eta(\xi)$ is small whenever the reproduction quality of u by s around ξ is good. In contrast, a high value of $\eta(\xi)$ typically indicates that u is subject to strong variation locally around ξ.

4.2 Coarsening and Refinement

In order to balance the approximation quality of the model against the required computational complexity we insert new nodes into regions where the value of η is high (refinement), whereas we remove nodes from Ξ in regions where the value of η is small (coarsening).

To avoid additional computational overhead and complicated data structures, effective adaption rules are required to be as simple as possible. In particular, these rules ought to be given by *local* operations on the current node set Ξ. The following coarsening rule is in fact very easy and, in combination with the refinement, it turned out to be very effective as well.

Coarsening. A node $\xi \in \Xi$ is *coarsened* by its removal from the current node set Ξ, i.e. Ξ is modified by replacing Ξ with $\Xi \setminus \{\xi\}$.

As to the refinement of a node $\xi \in \Xi$, we follow along the lines of the previous paper [4], where the effective refinement rules were motivated on the basis of available local error estimates for radial basis function interpolation. For the special case of thin plate spline interpolation, the local error estimate at $x \in \Omega$ is according to Wu and Schaback [18] of the form

$$|u(x) - s(x)| \leq C \cdot h^k_{\mathcal{N},\varrho}(x) \tag{17}$$

where $C > 0$ is a constant depending on u, and (for some radius $\varrho > 0$)

$$h_{\mathcal{N},\varrho}(x) = \sup_{\|y-x\|<\varrho} d_{\mathcal{N}}(y)$$

is the local *fill distance* of $\mathcal{N}$ around x, with

$$d_{\mathcal{N}}(y) = \min_{\nu \in \mathcal{N}} \|y - \nu\|$$

being the Euclidean distance between the point y and the set $\mathcal{N}$. We remark that for the special case $k = 1, d = 2$, the thin plate spline interpolation scheme is due to [13] locally of second order accuracy.

As suggested in [4], the reduction of the local error (17) around any $\xi \in \Xi$ is accomplished by reducing the distance function $d_{\mathcal{N}} = \min_{\nu \in \mathcal{N}} \| \cdot - \nu\|$ in a local neighbourhood of ξ.

Now recall that for a fixed node set $\Xi \subset \mathbb{R}^d$ and any $\xi \in \Xi$, the *Voronoi tile*

$$V_{\Xi}(\xi) = \left\{x \in \mathbb{R}^d \,:\, d_{\Xi}(x) = \|x - \xi\|\right\} \subset \mathbb{R}^d$$

of ξ w.r.t. Ξ contains all points in $\mathbb{R}^d$ whose nearest point in Ξ is ξ. The tile $V_{\Xi}(\xi)$ is a convex polytope whose vertices are referred to as the *Voronoi points*, forming a finite point set $\mathcal{V}_\xi$ in the neighbourhood of ξ. Figure 2 shows the Voronoi tile $V_{\Xi}(\xi)$ of a point ξ along with the set $\mathcal{V}_\xi$ of its Voronoi points. For more details on Voronoi diagrams, we refer to [15].

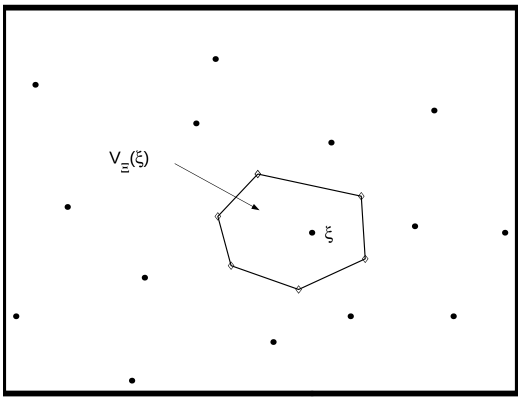

Fig. 2. Refinement of the node ξ. The Voronoi points ($\diamond$) are inserted.

Now observe that for $\xi \in \mathcal{N}$ the distance function $d_{\mathcal{N}}$ is convex on $V_\Xi(\xi)$. Moreover, it has local maxima at the Voronoi points in $\mathcal{V}_\xi$. Altogether, this gives rise to define the local refinement of nodes as follows.

Refinement. A node $\xi \in \Xi$ is *refined* by the insertion of its Voronoi points into the current node set Ξ, i.e. Ξ is modified by replacing Ξ with $\Xi \cup \mathcal{V}_\xi$.

5 Numerical Examples

We have implemented the proposed advection scheme for the special case of two dimensions, i.e. $d = 2$. In this section, the performance of the method on *nonlinear* equations (5) is shown. To this end, we considered using two model problems: *Burgers equation*, where the flux function is given by (6), and the *Buckley-Leverett equation*, whose flux function is (7).

5.1 Burgers Equation

Burgers [6] introduced the nonlinear flux function (6) in the hyperbolic conservation law (1) as a mathematical model of free turbulence in fluid dynamics. Burgers equation is nowadays a standard test case, popular mainly for the following reasons: (a) it contains the simplest form of a nonlinear advection term $u \cdot \nabla u$ simulating the physical phenomenon of wave motion, and (b) for its shock wave behaviour: As soon as the shock front occurs, there is no classical solution of the PDE and its weak solution becomes discontinuous.

In the test case, the following initial condition is used.

$$u_0(x) = \begin{cases} \exp\left(\frac{\|x-c\|^2}{\|x-c\|^2-R^2}\right) & \text{for } \|x-c\| < R \\ 0 & \text{otherwise} \end{cases}$$

with $R = 0.25$, $c = (0.3, 0.3)^T$, and we let the unit square $\tilde{\Omega} = [0,1]^2 \subset \Omega = \mathbb{R}^2$ be the computational domain, cf. [11]. Moreover, we selected the value $\epsilon = 2 * 10^{-3}$ for the diffusion coefficient in (5), and we let $r = (1,1)^T$ in (6), yielding a flow field along the diagonal of $\tilde{\Omega}$.

Initially, u_0 is sampled at 4446 randomly chosen points in the unit square, yielding the initial node set $\Xi^0 \subset \Omega$. Moreover, a constant time step size $\tau = 0.0075$ has been selected. A plot of the solution u at the three time steps $t_2 = 2 \cdot \tau, t_{80} = 80 \cdot \tau$, and $t_{160} = 160 \cdot \tau$ is shown in Figure 3.

Observe that the node adaption scheme achieves to localize the support of the solution u very effectively. Moreover, the shock front propagation is reasonably well resolved by the distribution of the current nodes, cf. the time steps $t = t_{80}, t_{160}$ in Figure 3. As already observed in the numerical examples of the previous paper [4], where we studied passive (linear) advection, this confirms the utility of the customized adaption rules yet once more.

Figure 4 shows a plot of the number of nodes per time step. The number of nodes, initially $\#\Xi^0 = \#\Xi^{t_1} = 4446$, immediately drops down to $\#\Xi^{t_2} = 1484$. This is due to *adaptivity*, starting at time $t = t_2$. Then, the number of nodes remains roughly constant for a while. Due to the growing support of u, a moderate increase of the number of nodes can be observed in the second half of the simulation, resulting in $\#\Xi^{t_{160}} = 1841$ after the final time step $t = t_{160}$.

5.2 Buckley-Leverett Equation

Buckley and Leverett [5] introduced the flux function (7) in hyperbolic conservation laws of the form (1) in order to describe the flow of two different liquids in a porous medium. This two-phase flow problem is typically encountered in applications of *oil reservoir modelling*, where specific enhanced oil recovery processes are simulated. When an underground source of oil is tapped, a certain amount of oil flows out on its own due to the high reservoir pressure. After the flow has stopped, there is usually still a large amount of oil in the reservoir pores. A standard method like waterflooding can now be used, where a fluid (e.g. water) is injected into a well in a reservoir to displace the contained hydrocarbons (e.g. oil) and produce them from another well. If gravity effects and capillary forces are neglected and the flux field is known, the two-phase flow is described by the saturation equation only, namely the *Buckley-Leverett equation.*

In our test case, we assume a single water injection well in the center of a 100% oil saturated, homogeneous, porous medium, i.e. reservoir rock, defined on the unit square. We allow open boundaries, so that the displaced oil can leave the computational domain $\tilde{\Omega} = [0,1]^2 \subset \Omega = \mathbb{R}^2$. Here, the function u quantifies the saturation of water in the reservoir pores. The values of u lie between 0 and 1, where $u \equiv 1$ denotes pure water and $u \equiv 0$ pure oil. In this

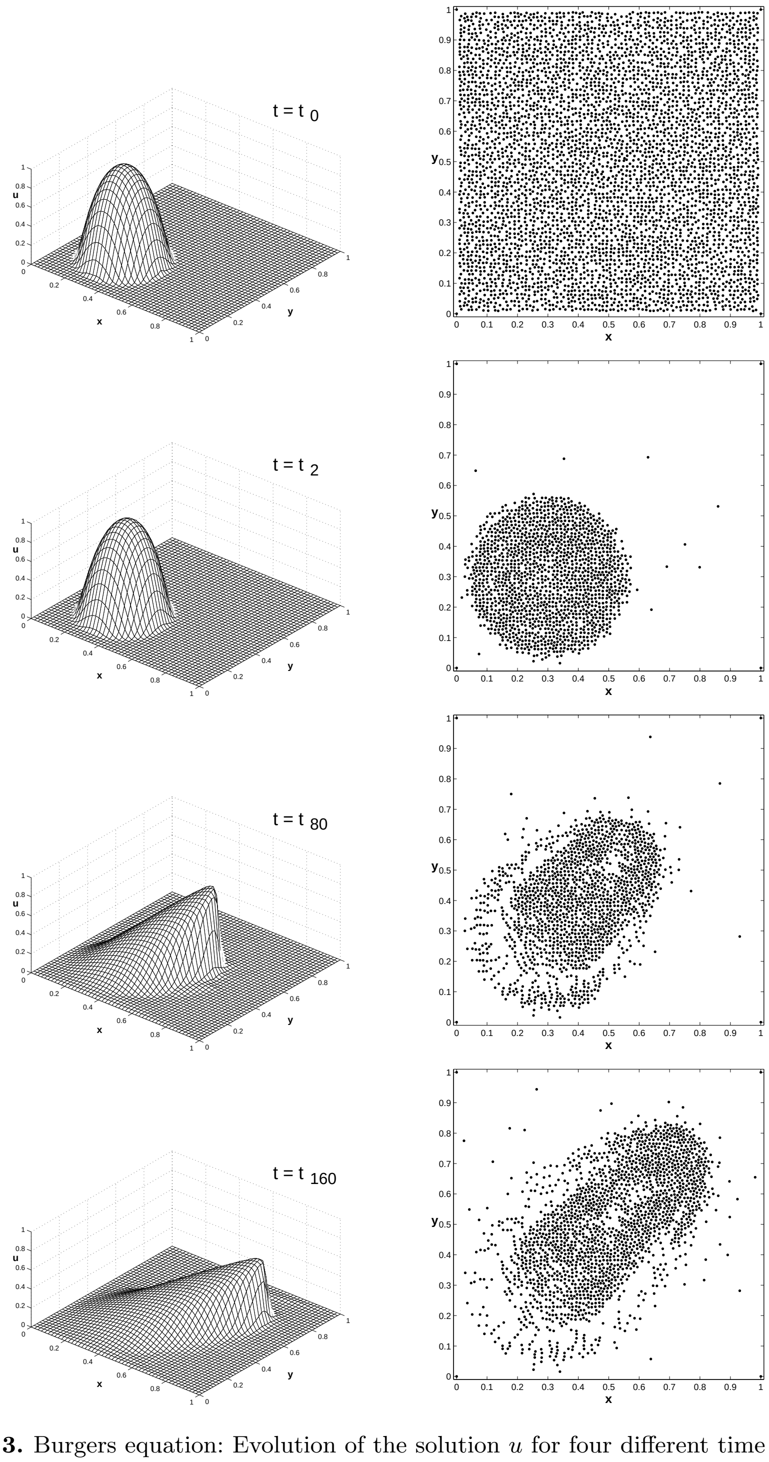

Fig. 3. Burgers equation: Evolution of the solution u for four different time steps, $t = t_0, t_2, t_{80}, t_{160}$ (left), and the corresponding node distribution (right).

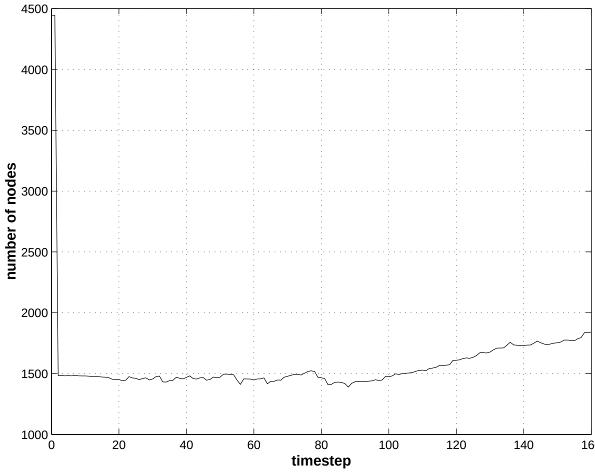

Fig. 4. Burgers equation: Number of nodes per time step.

case, the initial condition (2) is given by

$$u_0(x) = \begin{cases} 1 & \text{for } \|x - c\| \leq R \\ 0 & \text{otherwise} \end{cases}$$

with the injection well centered at $c = (0.5, 0.5)^T$ and with radius $R = 0.05$. Thus, initially the water saturation u inside the injection well is 1 and outside the well it is 0. A radial flow direction $r = (x - c)/\|x - c\|$ in (7) is assumed, so that the injected water should displace the oil radially. We decided to select a constant time step size $\tau = 0.001$, and the simulation comprises 264 time steps. Moreover, we let $\epsilon = 4 * 10^{-3}$ for the diffusion coefficient in (5). Finally, we selected the value $\mu = 0.5$ for the viscosity ratio of water and oil, appearing in the flux function (7).

The evolution of the oil's displacement by water is shown in Figure 5 (3D view and top view) and Figure 6 (side view) for four different times $t = t_0, t_2, t_{132}, t_{264}$. Initially, the function u_0 is sampled at a set Ξ^0 of $\#\Xi^0 = 4446$ of randomly distributed nodes.

Already after the second time step, the nodes are adapted to the vicinity of the well, and the number of nodes immediately drops down to $\#\Xi^{t_2} = 128$, see Figure 7. As soon as water flows into the well, a shock wave is formed, and a discontinuity in u can be observed, cf. Figures 5 and 6. In this case, a certain amount of oil is displaced immediately, whereas beyond the shock front there is a mixture of oil and water, with less oil at proceeding time. This phenomenon is referred to as the *rarefaction wave.*

Figure 5 and Figure 6 also show a comparison of the numerical and the analytic solution, the latter determined by Welge's tangent method [17]. It

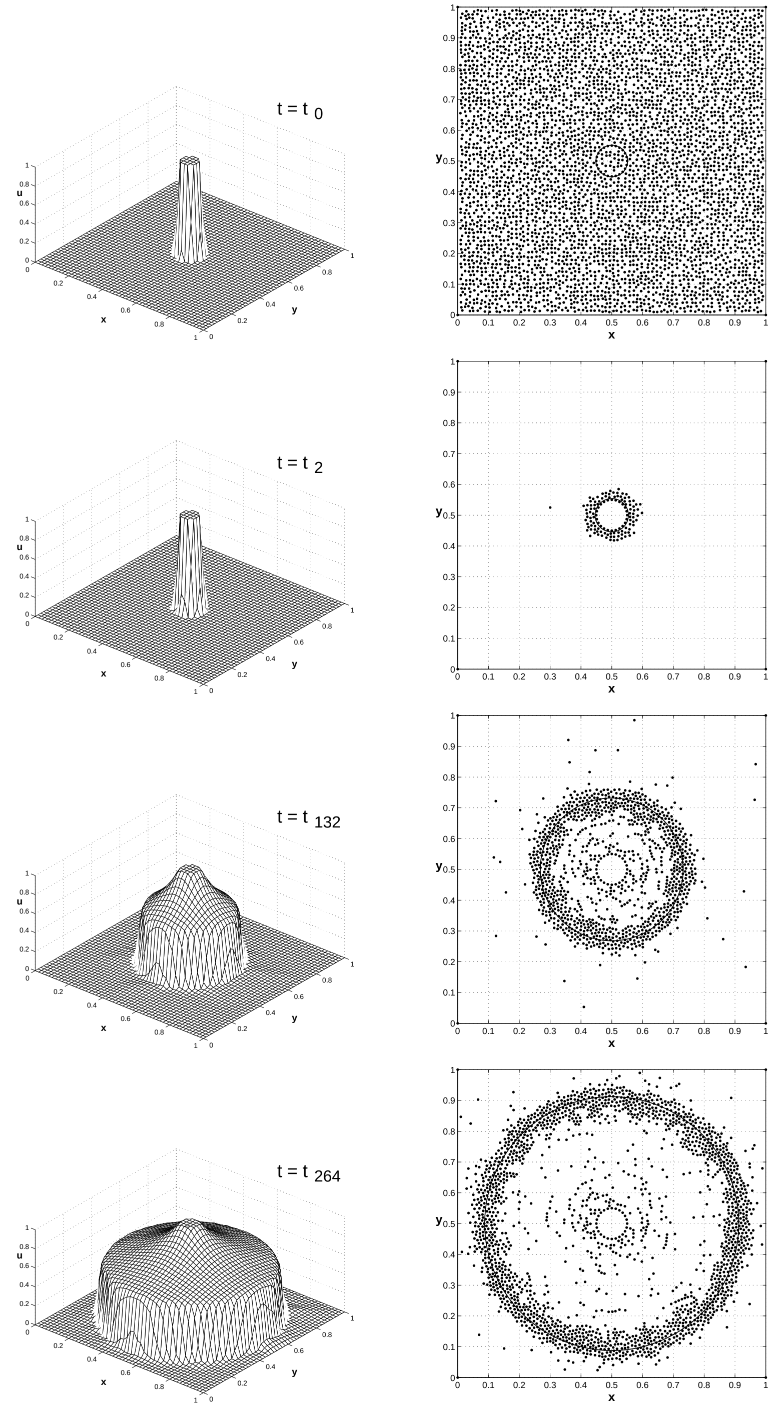

Fig. 5. Buckley-Leverett equation: The saturation u (3D view, left column; top view, right column), and the analytic solution (solid line, right column) for four different time steps $t = t_0, t_2, t_{132}, t_{264}$ (left).

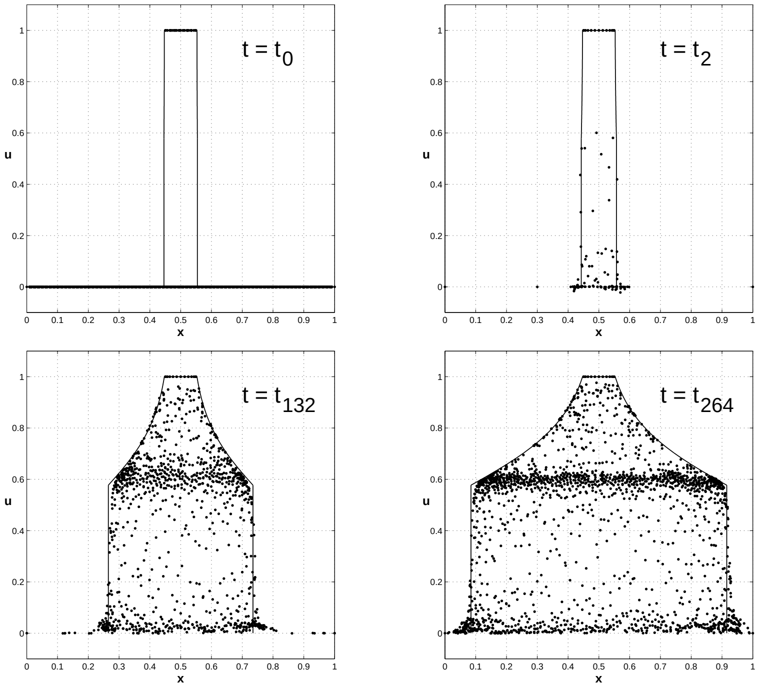

Fig. 6. Buckley-Leverett equation: The saturation u (nodes), and the analytic solution (solid line) for four different time steps $t = t_0, t_2, t_{132}, t_{264}$, side view.

can be observed, that the adaptive distribution of nodes achieves to capture the propagating shock front (the solid line in Figure 6) well. This helps to reduce the required computational costs while maintaining the accuracy, due to higher resolution around the shock. Note that since the radius of the propagating shock front increases linearly with time, the number of nodes increases linearly at proceeding time, ending up with $\#\Xi^{t_{264}} = 1967$ at the final time step $t = t_{264}$ of the simulation, see Figure 7.

6 Conclusion

The meshfree method of backward characteristics for *linear* (passive) advection from the previous work [3,4] has been extended for solving *nonlinear* transport problems. In order to avoid degeneration of characteristic curves, a vanishing viscosity approach has been added to the advection scheme. Moreover, the local interpolation scheme, using thin plate splines, has been modified accordingly. An error indicator derived from the interpolation scheme yields in combination with customized adaption rules an effective distribution

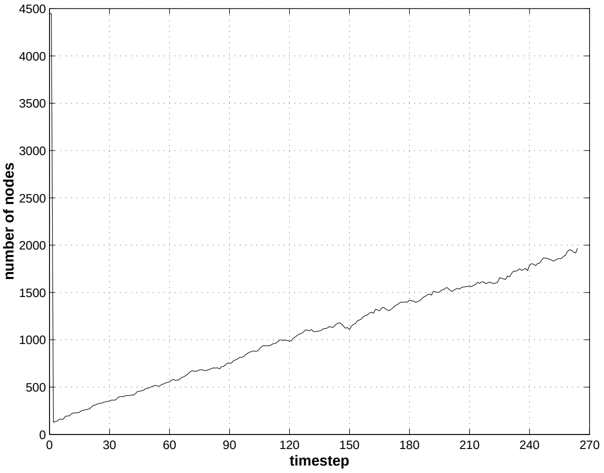

Fig. 7. Buckley-Leverett equation: Number of nodes per time step.

of nodes during the simulation. Numerical examples on nonlinear advection-dominated PDEs confirm the good performance of the proposed method.

We finally remark that the proposed method is potentially useful for higher dimensional problems. Note that our approach poses no principal restrictions to enhancements in higher dimensions. We have, however, not implemented the scheme for dimensions $d > 2$, yet.

Acknowledgment

The second and third author were partly supported by the European Union within the project NetAGES (Network for Automated Geometry Extraction from Seismic), contract no. IST-1999-29034.

References

1. Behrens J. (1996) An adaptive semi-Lagrangian advection scheme and its parallelization. Mon. Wea. Rev. **124**, 2386–2395.
2. Behrens J. (1998) Atmospheric and ocean modeling with an adaptive finite element solver for the shallow-water equations. Applied Numerical Mathematics **26**, 217–226.
3. Behrens J., and A. Iske (2001) Grid-free adaptive semi-Lagrangian advection using radial basis functions. To appear in Computers and Mathematics with Applications.
4. Behrens J., A. Iske, and St. Pöhn (2001) Effective node adaption for grid-free semi-Lagrangian advection. Discrete Modelling and Discrete Algorithms in Continuum Mechanics, Th. Sonar and I. Thomas (eds.), Logos Verlag, Berlin, 110–119.

5. Buckley, J. M., and M. C. Leverett (1942) Mechanism of fluid displacement in sands. Trans. AIME **146**, 107–116.
6. Burgers, J. M. (1940) Application of a model system to illustrate some points of the statistical theory of free turbulence. Proc. Acad. Sci. Amsterdam **43**, 2–12.
7. Courant, R., E. Isaacson, and M. Rees (1952) On the solution of nonlinear hyperbolic differential equations by finite differences. Comm. Pure Appl. Math. **5**, 243–255.
8. Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. Constructive Theory of Functions of Several Variables, W. Schempp and K. Zeller (eds.), Springer, Berlin, 85–100.
9. Durran, D. R. (1999) *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. Springer, New York.
10. Falcone, M., and R. Ferretti (1998) Convergence analysis for a class of high-order semi-Lagrangian advection schemes. SIAM J. Numer. Anal. **35:3**, 909–940.
11. Fürst, J., and Th. Sonar (2001) On meshless collocation approximations of conservation laws: positive schemes and dissipation models. ZAMM **81**, 403–415.
12. Gustafsson, B., H.-O. Kreiss, and J. Oliger (1995) *Time Dependent Problems and Difference Methods*, John Wiley and Sons, New York.
13. Gutzmer, T., and A. Iske (1997) Detection of discontinuities in scattered data approximation. Numerical Algorithms **16:2**, 155–170.
14. Iske, A. (2001) Scattered data modelling using radial basis functions. Principles of Multiresolution in Geometric Modelling, M. S. Floater, A. Iske, and E. Quak (eds.), Summer School Lecture Notes, Munich University of Technology, 271–301.
15. Preparata, F. P., and M. I. Shamos (1985) *Computational Geometry*, Springer, New York.
16. Robert, A. (1981) A stable numerical integration scheme for the primitive meteorological equations. Atmosphere-Ocean, 19, 35–46.
17. Welge, H. J. (1952) A simplified method for computing oil recovery by gas or water drive. Trans. AIME **195**, 97–108.
18. Wu, Z., and R. Schaback (1993) Local error estimates for radial basis function interpolation of scattered data. IMA J. of Numerical Analysis **13**, 13–27.