Int. Conference on Boundary and Interior Layers
BAIL 2006
G. Lube, G. Rapin (Eds)

A Discontinuous Galerkin Moving Mesh Method for Hamilton-Jacobi Equations

J.A Mackenzie and A. Nicola

Department of Mathematics,
University of Strathclyde,
26 Richmond St, Glasgow.
jam@maths.strath.ac.uk

1. Introduction

In this paper we consider the adaptive numerical solution of Hamilton-Jacobi (HJ) equations

$$\phi_t + H(\phi_{x_1}, \ldots, \phi_{x_d}) = 0, \quad \phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $t > 0$. HJ equations arise in many practical areas such as differential games, mathematical finance, image enhancement and front propagation. It is well known that solutions of (1) are Lipschitz continuous but derivatives can become discontinuous even if the initial data is smooth.

There is a close relation between HJ equations and hyperbolic conservation laws. With this in mind, it not surprising to find that many of the numerical methods used to solve HJ equations are motivated by conservative finite difference or finite volume methods for conservation laws. An increasingly popular approach to solve hyperbolic conservation laws is the discontinuous Galerkin (DG) finite element method. Recently, Hu and Shu [1] proposed a DG method to solve HJ equations by first rewriting (1) as a system of conservation laws

$$(w_i)_t + (H(\boldsymbol{w}))_{x_i} = 0, \quad i = 1, \ldots, d, \quad \boldsymbol{w}(\boldsymbol{x}, 0) = \nabla\phi_0(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{w} = \nabla\phi$. The usual DG formulation would be obtained if $\boldsymbol{w}$ belonged to a space of piecewise polynomials. However, we note that w_i, $i = 1, \ldots, d$ are not independent due to the restriction that $\boldsymbol{w} = \nabla\phi$. In [1] a least squares procedure was used to enforce this condition whereas in [4] a suitable basis was used such that $\boldsymbol{w}_h = \nabla\phi_h$

The aim of this paper is to consider the use of the DG method of Hu and Shu [1] to solve HJ equation using a moving mesh method based on the solution of moving mesh PDEs [2]. The governing equation are transformed to include the effect of the movement of the mesh and this is done in such a way that the conservation properties of the original equation are not lost. The adaptive mesh is driven by a monitor function which is shown to be non-singular in the presence of solution discontinuities.

2. DG discretisation of Hamilton-Jacobi equations

2..1. HJ equation in 1D

Let us consider the Hamilton-Jacobi equation in one dimension

$$\begin{cases} \phi_t + H(\phi_x) = 0, & (x,t) \in (a,b) \times (0,T] \\ \phi(x,0) = \phi_0(x), \end{cases} \tag{3}$$

with appropriate boundary conditions. If $u = \phi_x$ and we differentiate (3) with respect to x then u satisfies the hyperbolic conservation law

$$\begin{cases} u_t + H(u)_x = 0, & (x,t) \in (a,b) \times (0,T] \\ u(x,0) = u_0(x). \end{cases} \tag{4}$$

We assume that the physical domain $\Omega_p = [a,b]$ is the image of a computational domain $\Omega_c = [0,1]$ which is obtained via the time-dependent mapping $x = x(\xi,t)$. We will consider a discretisation of the conservative form in computational coordinates

$$\begin{cases} (\dot{x_\xi u}) + (H(u) - \dot{x}\,u)_\xi = 0, \quad (\xi,t) \in (0,1)\times(0,T], \\ u(\xi,0) = u_0(\xi). \end{cases} \tag{5}$$

The aim is to discretise (5) in space using a DG method.

2..2. The DG discretisation

For each partition $\left\{\xi_{j+\frac{1}{2}}\right\}_{j=0}^{N}$ of Ω_c, we denote

$$I_j = \left(\xi_{j-\frac{1}{2}}, \xi_{j+\frac{1}{2}}\right), \quad \Delta_j = \xi_{j+\frac{1}{2}} - \xi_{j-\frac{1}{2}}, \quad j = 1,\ldots,N.$$

We use a uniform mesh to cover the computational domain such that

$$h = \xi_{j+\frac{1}{2}} - \xi_{j-\frac{1}{2}} = \frac{1}{N}, \quad \xi_j = \frac{1}{2}\left(\xi_{j+\frac{1}{2}} + \xi_{j-\frac{1}{2}}\right), \quad j = 1,\ldots,N. \tag{6}$$

If we multiply (5) by an arbitrary smooth function v, then integrating by parts over the interval I_j we obtain

$$\int_{I_j} (\dot{x_\xi u})\, v\, d\xi - \int_{I_j} \widetilde{H}(u,\dot{x})\, v_\xi\, d\xi + \widetilde{H}(u,\dot{x})\, v\Big|_{I_j} = 0, \tag{7}$$

where

$$\widetilde{H}(u,\dot{x}) = H(u) - \dot{x}\,u, \tag{8}$$

and

$$\widetilde{H}(u,\dot{x})v\Big|_{I_j} = \widetilde{H}(u(\xi_{j+\frac{1}{2}},t), \dot{x}(\xi_{j+\frac{1}{2}}))\, v(\xi_{j+\frac{1}{2}}) - \widetilde{H}(u(\xi_{j-\frac{1}{2}},t), \dot{x}(\xi_{j-\frac{1}{2}}))\, v(\xi_{j-\frac{1}{2}}). \tag{9}$$

We will approximate the exact solution u of (7) and the smooth function v locally by polynomials of degree at most k. The numerical approximations u_h and v_h belong to the finite dimensional space

$$V_h^k = \left\{v : v|_{I_j} \in P^k(I_j), \quad j = 1,\ldots,N\right\}, \tag{10}$$

where $P^k(I_j)$ denotes the space of polynomials on I_j of degree at most k. In the discontinuous Galerkin numerical method u_h and v_h are discontinuous at the points $\xi_{j+\frac{1}{2}}$, $j = 0,\ldots,N$, and the question is how to evaluate their values in (9). We set

$$v^-_{h_{j+\frac{1}{2}}} = \lim_{\xi \to \xi^-_{j+\frac{1}{2}}} v_h(\xi), \quad \text{and} \quad v^+_{h_{j-\frac{1}{2}}} = \lim_{\xi \to \xi^+_{j-\frac{1}{2}}} v_h(\xi). \tag{11}$$

Furthermore, we replace the nonlinear flux function $\widetilde{H}(u_h,\dot{x})$ defined in (8) by a numerical flux function that depends on the values of u_h from the left and right at the point $\left(\xi_{j+\frac{1}{2}}, t\right)$, that is

$$\widetilde{H}(u_h,\dot{x})_{\xi_{j+\frac{1}{2}}}(t) = \overline{H}_{j+\frac{1}{2}}\left(u_h\left(\xi^-_{j+\frac{1}{2}}, t\right), u_h\left(\xi^+_{j+\frac{1}{2}}, t\right), \dot{x}_{\xi_{j+\frac{1}{2}}}(t)\right). \tag{12}$$

In this paper we have used the local Lax-Friedrichs flux

$$\overline{H}(p,q,\dot{x}) = \frac{1}{2}\left(\widetilde{H}(p,\dot{x}) + \widetilde{H}(q,\dot{x}) - c(q-p)\right), \tag{13}$$

where

$$c = \max_{\min(p,q)\le u\le\max(p,q)} \left|\frac{\partial \widetilde{H}}{\partial u}\right|.$$

The discontinuous Galerkin space approximation is given as the solution of the weak formulation

$$\int_{I_j} (\dot{x_\xi u_h}) v_h \, d\xi - \int_{I_j} \widetilde{H}(u_h, \dot{x})(v_h)_\xi \, d\xi + \overline{H}_{j+\frac{1}{2}} v^-_{h_{\xi_{j+\frac{1}{2}}}} - \overline{H}_{j-\frac{1}{2}} v^+_{h_{\xi_{j-\frac{1}{2}}}} = 0, \tag{14}$$

and the initial condition is given by the projection

$$\int_{I_j} u_h(\xi, 0) v_h(\xi) \, d\xi = \int_{I_j} u_0(\xi) v_h(\xi) \, d\xi, \quad j = 1, \ldots, N, \tag{15}$$

for all $v_h \in V_h$.

2..2..1 Temporal integration

The discontinuous Galerkin discretisation gives rise to a system of ODEs which can be written as

$$\begin{cases} \dfrac{d}{dt}(x_\xi \mathbf{u}_h) = L_h(\mathbf{u}_h), & \text{in } (0, T] \\ \mathbf{u}_h(0) = \mathbf{u}_{0h}, \end{cases} \tag{16}$$

where $\mathbf{u}_{0h}$ is the initial approximation for the vector of the degrees of freedom $\mathbf{u}_h$, and x_ξ depends on the meshes at time level t^n and t^{n+1} as defined earlier. To integrate (16) numerically we use the second-order TVD Runge-Kutta method

$$\begin{cases} x_\xi^{n+1} \mathbf{u}_h^{(1)} = x_\xi^n \mathbf{u}_h^n + \Delta t \, L(\mathbf{u}_h^n), \\ \\ x_\xi^{n+1} \mathbf{u}_h^{n+1} = \dfrac{1}{2} x_\xi^n \mathbf{u}_h^n + \dfrac{1}{2} x_\xi^{n+1} \mathbf{u}_h^{(1)} + \dfrac{1}{2} \Delta t \, L(\mathbf{u}_h^{(1)}). \end{cases} \tag{17}$$

2..2..2 Recovery of ϕ_h

If $(\phi_h)_x = u_h$, then ϕ_h will be determined on each interval I_j up to a constant. Following [1], the constant can be retrieved in two ways:

1. require that

$$\int_{I_j} ((\phi_h)_t + H(u_h)) \, \mathrm{d}x, \quad j = 1, \ldots, N. \tag{18}$$

2. use (18) to update only the leftmost element I_1 and then since $(\phi_h)_x = u_h$ we have

$$\phi_h(x_j, t) = \phi_h(x_1, t) + \int_{x_1}^{x_j} u_h(x, t) \, \mathrm{d}x. \tag{19}$$

2..3. HJ equations in 2D

In two-dimensions we have

$$\phi_t + H(\phi_x, \phi_y) = 0, \quad (x, y) \in \Omega_p, \quad t > 0, \tag{20}$$

where $\Omega_p \subseteq \mathbb{R}^2$ is the physical domain. To generate an adaptive mesh we have seen earlier that it is useful to regard the physical domain Ω_p as the image of a computational domain Ω_c under the invertible maps

$$x = x(\xi, \eta, t), \quad y = y(\xi, \eta, t), \quad \text{and} \quad \xi = \xi(x, y, t), \quad \eta = \eta(x, y, t), \tag{21}$$

where $\boldsymbol{x} = (x, y)$ and $\boldsymbol{\xi} = (\xi, \eta)$ are the physical and computational coordinates, respectively. A mesh covering Ω_p is obtained by applying the mapping given by (21) to a partitioning of Ω_c.

Given that the mesh can move then it is necessary to express the Eulerian ($\boldsymbol{x}$-fixed) temporal derivative in (20) in terms of the Lagrangian derivative along the trajectory of the moving mesh. If a dot denotes differentiation with respect to time with (ξ, η) fixed then (20) becomes

$$\dot{\phi} - \dot{x}\phi_x - \dot{y}\phi_y + H(\phi_x, \phi_y) \quad = \quad 0, \tag{22}$$

where $(\dot{x}, \dot{y})$ represents the mesh velocity. If we let $u = \phi_\xi$ and $v = \phi_\eta$, then by differentiating (22) with respect to ξ and η we arrive at the system

$$u_t + \bar{H}_\xi(u, v, \dot{x}, \dot{y}) = 0, \tag{23}$$

$$v_t + \bar{H}_\eta(u, v, \dot{x}, \dot{y}) = 0, \tag{24}$$

where

$$\bar{H}(u, v, \dot{x}, \dot{y}) = H(\xi_x u + \eta_x v, \xi_y u + \eta_y v) - \dot{x}(\xi_x u + \eta_x v) - \dot{y}(\xi_y u + \eta_y v). \tag{25}$$

Using the vectorial notation

$$\boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix}, \quad \boldsymbol{F}_1 = \begin{pmatrix} \bar{H} \\ 0 \end{pmatrix}, \quad \boldsymbol{F}_2 = \begin{pmatrix} 0 \\ \bar{H} \end{pmatrix},$$

we can rewrite (23) and (24) as the system of conservation laws

$$\boldsymbol{u}_t + [\boldsymbol{F}_1(\boldsymbol{u})]_\xi + [\boldsymbol{F}_2(\boldsymbol{u})]_\eta = 0. \tag{26}$$

We apply a discontinuous Galerkin method to solve this coupled system (26) for $(u, v) = \nabla_{\boldsymbol{\xi}}\phi$, where the discretisation takes place in the computational domain Ω_c. A recovery procedure will then be used to obtain an approximation of ϕ.

To obtain a weak formulation we take the inner product of (26) with a test function $\boldsymbol{v}_h \in V_h$, integrate over $K \in \mathcal{T}_h$ and replace $\boldsymbol{u}$ by its approximation $\boldsymbol{u}_h \in V_h$. That is

$$\frac{d}{dt}\int_K \boldsymbol{u}_h(\boldsymbol{\xi}, t) \cdot \boldsymbol{v}_h(\boldsymbol{\xi})\, d\boldsymbol{\xi} + \int_K [(\boldsymbol{F}_1)_\xi \cdot \boldsymbol{v}_h + (\boldsymbol{F}_2)_\eta \cdot \boldsymbol{v}_h]\, d\boldsymbol{\xi} = 0, \tag{27}$$

and using integration by parts we obtain

$$\begin{aligned} &\frac{d}{dt}\int_K \boldsymbol{u}_h(\boldsymbol{\xi}, t) \cdot \boldsymbol{v}_h(\boldsymbol{\xi})\, d\boldsymbol{\xi} \\ &+ \sum_{e \in \partial K} \int_e \left\{ (n_{e,K})_1 \boldsymbol{F}_1 \cdot \boldsymbol{v}_h(\boldsymbol{\xi}^{\text{int}(K)}) + (n_{e,K})_2 \boldsymbol{F}_2 \cdot \boldsymbol{v}_h(\boldsymbol{\xi}^{\text{int}(K)}) \right\} ds \\ &- \int_K [\boldsymbol{F}_1 \cdot (\boldsymbol{v}_h)_\xi + \boldsymbol{F}_2 \cdot (\boldsymbol{v}_h)_\eta]\, d\boldsymbol{\xi} = 0, \end{aligned} \tag{28}$$

where $n_{e,K} = [(n_{e,K})_1, (n_{e,K})_2]$ is the unit normal to the edge e of the element K, $\text{int}(K)$ denotes the value taken from the within the element K, and $\text{ext}(K)$ denotes the value taken from the exterior of the element K. Next we define the normal flux

$$\boldsymbol{F}_{e,K}(\boldsymbol{u}, \dot{x}, \dot{y}) = (n_{e,K})_1 \boldsymbol{F}_1 + (n_{e,K})_2 \boldsymbol{F}_2 = \begin{bmatrix} (n_{e,K})_1 \bar{H} \\ (n_{e,K})_2 \bar{H} \end{bmatrix}. \tag{29}$$

Note that $\boldsymbol{F}_{e,K}$ is not well defined on the element edge as $\boldsymbol{u}_h$ is discontinuous there. Therefore, we replace it by the numerical flux function

$$\widehat{\boldsymbol{F}}_{e,K}\left(\boldsymbol{u}_h(\boldsymbol{\xi}^{\text{int}(K)}), \boldsymbol{u}_h(\boldsymbol{\xi}^{\text{ext}(K)}), \dot{x}, \dot{y}\right). \tag{30}$$

Again we will use the Lax-Friedrichs flux

$$\widehat{\boldsymbol{F}}_{e,K}(\boldsymbol{a},\boldsymbol{b},\dot{x},\dot{y}) = \frac{1}{2}\left[\boldsymbol{F}_{e,K}(\boldsymbol{a}) + \boldsymbol{F}_{e,K}(\boldsymbol{b}) + \alpha_{e,K}(\boldsymbol{a}-\boldsymbol{b})\right], \tag{31}$$

where

$$\alpha_{e,K} = \gamma\rho\left((n_{e,K})_1\frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{u}} + (n_{e,K})_2\frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{u}}\right), \tag{32}$$

and $\gamma \geq 1$ and ρ is the spectral radius of the normal Jacobian matrix.

We now discuss the basis used when $\boldsymbol{u}_h$ is a discontinuous bilinear function. Following Hu and Shu [1] we assume that the approximate solution of the Hamilton-Jacobi equation ϕ_h is piecewise discontinuous and quadratic. Therefore, in the cell K_{ij} we assume

$$\begin{aligned}\phi_h(t) = \overline{\phi}(t) + \phi_\xi(t)\mu_i + \phi_\eta(t)\nu_j + \phi_{\xi\eta}(t)\mu_i\nu_j \\ +\phi_{\xi\xi}(t)\left(\mu_i^2 - \frac{1}{3}\right) + \phi_{\eta\eta}(t)\left(\nu_j^2 - \frac{1}{3}\right),\end{aligned} \tag{33}$$

where

$$\mu_i(\xi) = \frac{2(\xi-\xi_i)}{\Delta\xi}, \quad \text{and} \quad \nu_j(\eta) = \frac{2(\eta-\eta_j)}{\Delta\eta}.$$

The gradient in computational space $\nabla_{\boldsymbol{\xi}}\phi_h$ therefore takes the form

$$\nabla_{\boldsymbol{\xi}}\phi_h = \begin{bmatrix} \frac{2}{\Delta\xi}\phi_\xi(t) + \frac{2}{\Delta\xi}\phi_{\xi\eta}(t)\nu_j + \frac{4}{\Delta\xi}\phi_{\xi\xi}(t)\mu_i \\ \frac{2}{\Delta\eta}\phi_\eta(t) + \frac{2}{\Delta\eta}\phi_{\xi\eta}(t)\mu_i + \frac{4}{\Delta\eta}\phi_{\eta\eta}(t)\nu_j \end{bmatrix}, \tag{34}$$

and hence there are five unknowns that determine $\nabla_{\boldsymbol{\xi}}\phi_h$. A suitable basis for

$$V_2^1 = \{(v_1,v_2,v_3,v_4,v_5) : \boldsymbol{v}|_K = \nabla_{\boldsymbol{\xi}}\phi, \quad \phi\in P^2(K), \quad \forall K\in\mathcal{T}_h\}$$

is given by

$$\boldsymbol{b}_1 = \begin{bmatrix}1\\0\end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix}0\\1\end{bmatrix}, \quad \boldsymbol{b}_3 = \begin{bmatrix}\mu_i\\0\end{bmatrix}, \quad \boldsymbol{b}_4 = \begin{bmatrix}0\\\nu_j\end{bmatrix}, \quad \boldsymbol{b}_5 = \begin{bmatrix}\frac{\nu_j}{\Delta\xi}\\\frac{\mu_i}{\Delta\eta}\end{bmatrix}. \tag{35}$$

The local basis functions (35) can easily be shown to be orthogonal and hence the mass matrix is diagonal.

3. Moving mesh equations

Let us consider $\boldsymbol{x} = (x_1,x_2)^T$ to be a point in the physical domain Ω_p and $\boldsymbol{\xi} = (\xi_1,\xi_2)^T$ to be a point in the computational domain Ω_c. The mapping from the computational domain to the physical domain is assumed to satisfy the moving mesh PDE [2]

$$\tau\frac{\partial \boldsymbol{x}}{\partial t} = \left(\sum_{i=1}^{2} a_{i,i}^2 + b_i^2\right)^{-\frac{1}{2}}\left(\sum_{i,j=1}^{2}(\boldsymbol{a}^i\cdot G^{-1}\boldsymbol{a}^j)\frac{\partial^2\boldsymbol{x}}{\partial\xi_i\partial\xi_j} - \sum_{i,j=1}^{2}\left(\boldsymbol{a}^i\cdot\frac{\partial G^{-1}}{\partial\xi_j}\boldsymbol{a}^j\right)\frac{\partial\boldsymbol{x}}{\partial\xi_i}\right), \tag{36}$$

where $\boldsymbol{a}^i = \nabla\xi_i$ and

$$a_{i,j} = \boldsymbol{a}^i\cdot G^{-1}\boldsymbol{a}^j, \quad b_i = -\sum_{i,j=1}^{2}\left(\boldsymbol{a}^i\cdot\frac{\partial G^{-1}}{\partial\xi_j}\boldsymbol{a}^j\right). \tag{37}$$

Dirichlet boundary conditions for the above system are obtained by solving a one-dimensional MMPDE. If $\partial_p \in \partial\Omega_p$ and $\partial_c \in \partial\Omega_c$ denote the physical and computational boundary segments with arc-lengths l and l_c respectively, then the mesh on ∂_p is the solution of

$$\tau \frac{\partial s}{\partial t} = \frac{1}{\sqrt{(M^2 + (M_\zeta)^2}} \frac{\partial}{\partial \zeta}\left(M \frac{\partial s}{\partial \zeta}\right), \qquad \zeta \in (0, l_c), \tag{38}$$

with $s(0) = 0$ and $s(l_c) = l$. Here M is the one-dimensional projection of the two-dimensional monitor function along the boundary. That is, if $\boldsymbol{t}$ is a unit tangent vector along the boundary then $M(s,t) = \boldsymbol{t}^T G \boldsymbol{t}$.

3..1. Choice of monitor function

The choice of a suitable adaptivity criterion for problems which develop shocks is highly non-trivial. It is important that the mesh points move smoothly towards regions where discontinuities exist so that the $O(1)$ error at the shock is localised within one or two mesh elements. For robustness, it is also important that the rate of convergence of the method is not severely impaired by the non-uniformity of the moving mesh.

To obtain a bounded monitor function we consider the choice [3]

$$M(x,t) = \left(1 + \frac{|u_x|^2}{\alpha}\right)^{\frac{1}{2}}, \qquad \alpha = \frac{1}{\beta^2} \max_x |u_x|^2, \tag{39}$$

and β is a user-chosen parameter. It is clear that this monitor function satisfies the bounds

$$1 \le M(x,t) \le \sqrt{1+\beta^2}.$$

If the mesh is found by equidistribution of the monitor function then if $M \simeq 1$ except in the cell containing x^* then

$$\sqrt{1+\beta^2}\Delta x_{\text{min}} \simeq \frac{1}{N}\left((b-a) + \sqrt{1+\beta^2}\, \Delta x_{\text{min}}\right). \tag{40}$$

Therefore, the minimum mesh spacing is

$$\Delta x_{\text{min}} \simeq \frac{(b-a)}{(N-1)\sqrt{1+\beta^2}} \tag{41}$$

which is approximately a factor of $\sqrt{1+\beta^2}$ smaller than that using a uniform mesh.

For two dimensional problems the monitor function used is of Winslow type where

$$G = \begin{bmatrix} w & 0 \\ 0 & w \end{bmatrix}$$

and

$$w = \sqrt{1 + \frac{|\nabla u|^2}{\alpha}}, \quad \text{where} \quad \alpha = \frac{1}{\beta^2} \max_{\Omega p} |\nabla u|^2.$$

This monitor function behaves in a similar fashion to the one-dimensional monitor in the presence of solution discontinuities.

4. Numerical experiments

4..1. Example 1

The first example considered is Burgers' equation

$$\begin{cases} \phi_t + \dfrac{(\phi_x + 1)^2}{2} = 0, & -1 < x < 1,\ t > 0 \\ \phi(x,0) = -\cos(\pi(x - 0.85)), & \end{cases} \tag{42}$$

with periodic boundary conditions.

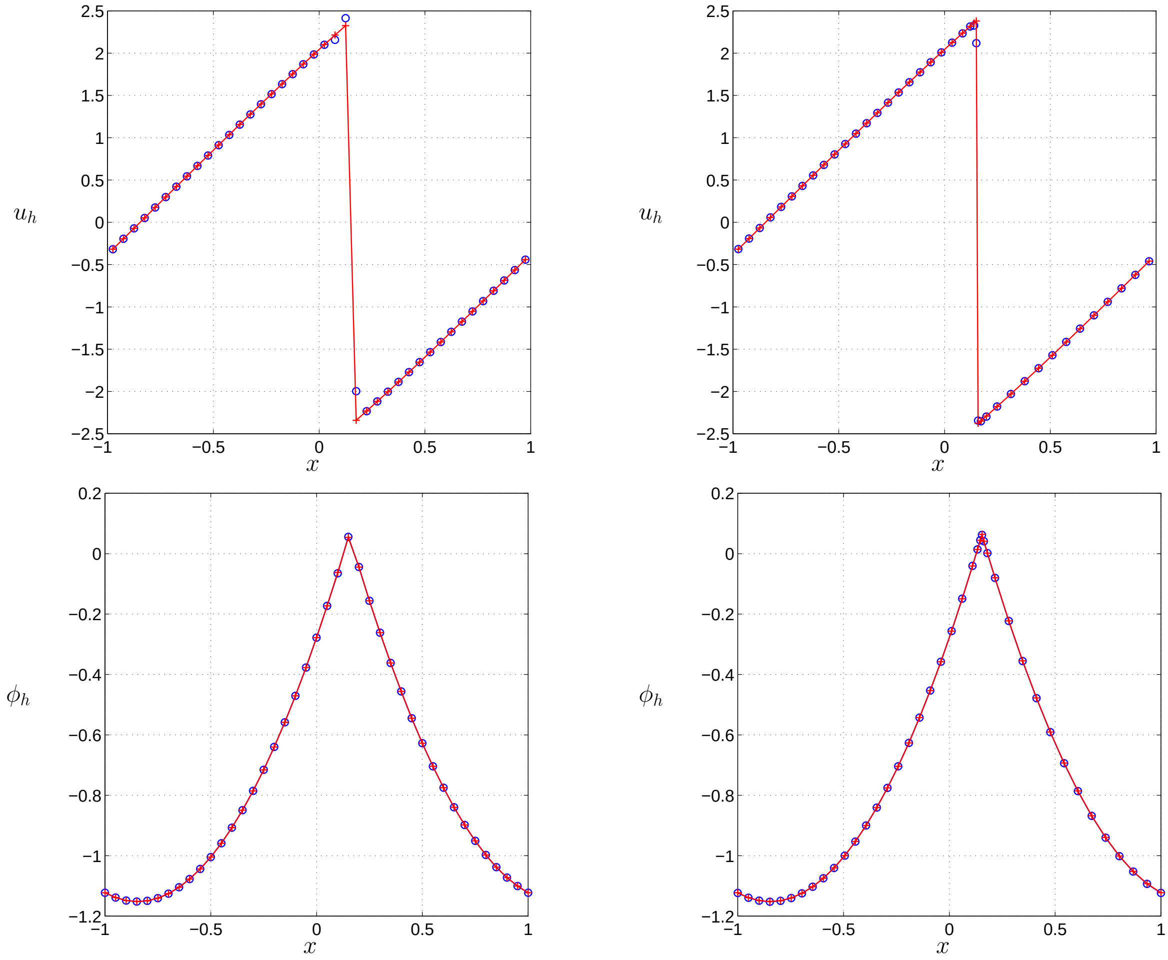

Figure 1: DG1 numerical solution (∘) and exact solution (+) obtained on a uniform stationary mesh (left), and on a moving mesh (right) at time $t = 3/\pi^2$.

4..2. Example 2

We consider the two-dimensional Burgers' equation

$$\phi_t + \frac{1}{2}(1 + \phi_x + \phi_y)^2 = 0, \quad (x, y) \in (-2, 2)^2 \tag{43}$$

$$\phi(x, y, 0) = -\cos(\pi(x + y)/2), \tag{44}$$

with periodic boundary conditions.

5. Conclusions

We have presented a DG method for the solution of nonlinear Hamilton-Jacobi equations using an adaptive moving mesh refinement strategy. A suitable mesh refinement criterion was used which does not become singular when the solution become non-smooth. Numerical experiments in one and two dimensions show that method works well and efficiently delivers high resolution solutions using relatively coarse meshes.

References

[1] C. Hu and C.W. Shu. A Discontinuous Galerkin Finite Element Method for Hamilton-Jacobi Equations. *SIAM Journal on Scientific Computing*, 21(2):666–690, 1999.

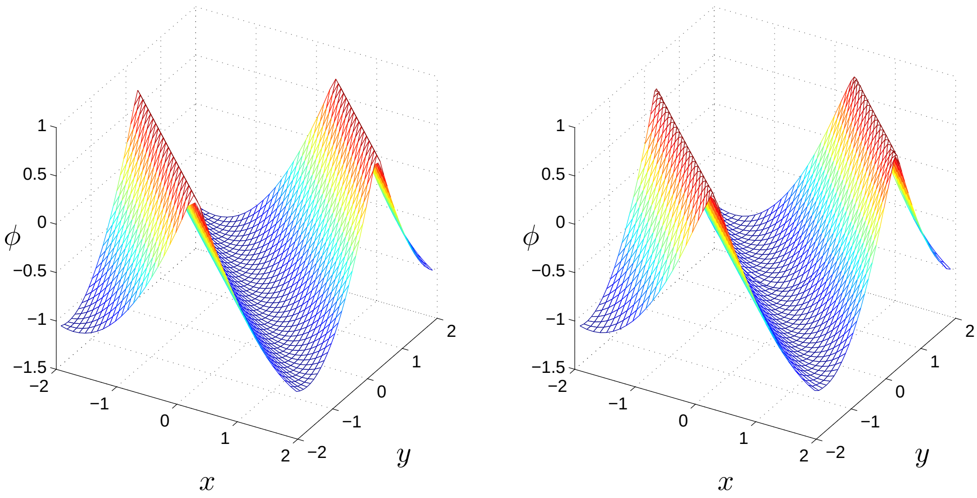

Figure 2: Approximations of ϕ_x (top) and ϕ (bottom) at $t = 1.5/\pi^2$ using uniform mesh (left) and using moving mesh (right), with $N = 40$.

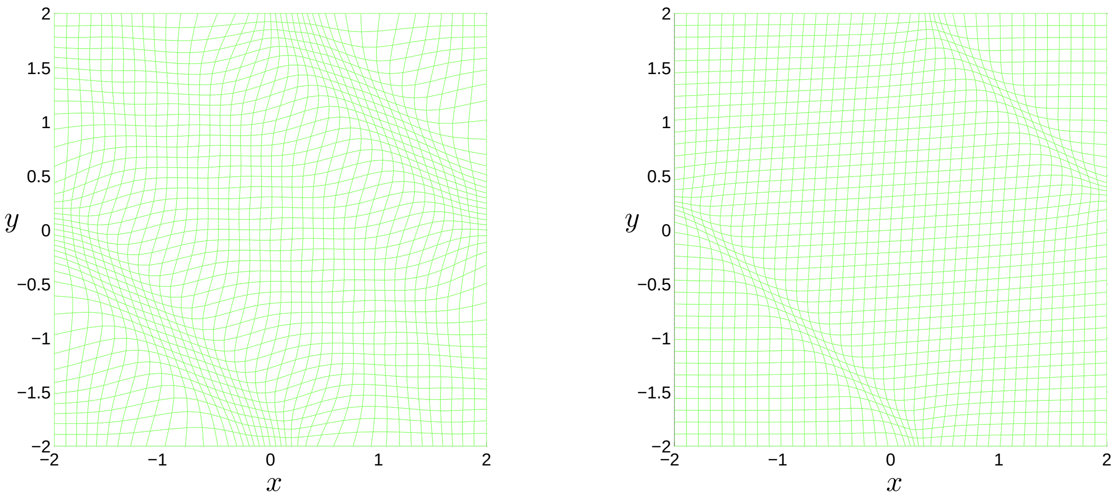

Figure 3: Adapted meshes at $t = 0.5/\pi^2$ (left), and $t = 1.5/\pi^2$ (right) obtained with $N = 40$ for Example 2.

[2] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *Journal of Computational Physics*, 171:753–775, 2001.

[3] J.M. Stockie, J.A. Mackenzie, and R.D. Russell. A Moving Mesh Method for One-Dimensional Hyperbolic Conservation Laws. *SIAM Journal on Scientific Computing*, 22(5):1791–1813, 2001.

[4] F. Li and C.W. Shu. Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. *Applied Mathematics Letters*, 18:1204–1209, 2005.