Discontinuous Galerkin method for Hamilton-Jacobi equations and front propagation with obstacles

Chi-Wang Shu

▶ **To cite this version:**

Chi-Wang Shu. Discontinuous Galerkin method for Hamilton-Jacobi equations and front propagation with obstacles. NETCO 2014 - New Trends in Optimal Control, Jun 2014, Tours, France. hal-01024613

HAL Id: hal-01024613
https://inria.hal.science/hal-01024613

Submitted on 16 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discontinuous Galerkin method for Hamilton-Jacobi equations and front propagation with obstacles

Chi-Wang Shu

Division of Applied Mathematics

Brown University

Joint work with Olivier Bokanowski, Yingda Cheng, Tao Xiong and Mengping Zhang

Outline

- Introduction
- A discontinuous Galerkin method for Hamilton-Jacobi equations
- Front propagation problems in the presence of obstacles
- Numerical examples

Introduction

We are interested in solving the following HJ equation

$$\varphi_t + H(\varphi_{x_1}, ..., \varphi_{x_d}) = 0, \qquad \varphi(x, 0) = \varphi^0(x). \tag{1}$$

- H could also depend on φ, x and t in some applications, however the main difficulty for numerical solutions is the nonlinear dependency of H on the gradient of φ.

- HJ equations appear often in many applications: image processing and computer vision, control and differential games, level set methods for multiphase flows, etc.
- Strong C^1 solution generically does not exist for all time t even if the initial condition $\varphi^0(x)$ is smooth. We must consider viscosity solutions among Lipschitz continuous functions.

Discontinuous Galerkin method

How does the DG method work — an example

To solve a hyperbolic conservation law:

$$u_t + f(u)_x = 0 \tag{2}$$

Multiplying with a test function v, integrate over a cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, and integrate by parts:

$$\int_{I_j} u_t v dx - \int_{I_j} f(u) v_x dx + f(u_{j+\frac{1}{2}}) v_{j+\frac{1}{2}} - f(u_{j-\frac{1}{2}}) v_{j-\frac{1}{2}} = 0$$

Now assume both the solution u and the test function v come from a finite dimensional approximation space V_h, which is usually taken as the space of piecewise polynomials of degree up to k:

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

However, the boundary terms $f\left(u_{j+\frac{1}{2}}\right)$, $v_{j+\frac{1}{2}}$ etc. are not well defined when u and v are in this space, as they are discontinuous at the cell interfaces.

From the conservation and stability (upwinding) considerations, we take

- A single valued monotone numerical flux to replace $f\left(u_{j+\frac{1}{2}}\right)$:

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}})$$

  where $\hat{f}(u,u) = f(u)$ (consistency); $\hat{f}(\uparrow,\downarrow)$ (monotonicity) and $\hat{f}$ is Lipschitz continuous with respect to both arguments.

- Values from inside I_j for the test function v

$$v^-_{j+\frac{1}{2}}, \qquad v^+_{j-\frac{1}{2}}$$

Hence the DG scheme is: find $u \in V_h$ such that

$$\int_{I_j} u_t v dx - \int_{I_j} f(u) v_x dx + \hat{f}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0 \qquad (3)$$

for all $v \in V_h$.

DG method for Hamilton-Jacobi equations

Difficulty in designing DG schemes for HJ equations: the nonlinear Hamiltonian H is outside the derivative, preventing a direct application of integration by parts for a weak formulation.

Different DG methods for solving HJ equations:

- The method of Hu and Shu (SISC 1999):
  - Use the DG scheme to solve for the derivatives $u = \varphi_x$ and $v = \varphi_y$, which satisfy a weakly hyperbolic system.
  - Then use a least square procedure to recover φ from u and v. The missing information can be obtained through evolving the cell averages.

Later, Li and Shu (Appl. Math. Letters 2005) gave a reinterpretation of this DG method, leading to a mathematically equivalent but practically easier to implement version:

- Use the DG scheme to solve for the derivatives $u = \varphi_x$ and $v = \varphi_y$, which satisfy a weakly hyperbolic system, in a locally curl-free subspace.
- Then there is a unique polynomial φ satisfying $u = \varphi_x$ and $v = \varphi_y$, hence the least square procedure is not needed. The missing information can again be obtained through evolving the cell averages.

- The DG method of Cheng and Shu (JCP 2007) to be discussed later. Also the central DG version of this method by Li and Yakovlev (JSC 2010).
- The local DG method of Yan and Osher (JCP 2011): one first solves two copies of $u = \varphi_x$ by fluxes upwinding from the left and upwinding from the right, denoted by u^- and u^+ respectively, then approximate $H(\varphi_x)$ by $\hat{H}(u^-, u^+)$ where $\hat{H}$ is any monotone numerical Hamiltonian (numerical flux).

In the following we consider the DG scheme of Cheng and Shu JCP 2007, see also Bokanowski, Cheng and Shu SISC 2011.

Motivation: look at a simple linear HJ equation

$$\varphi_t + a(x)\varphi_x = 0$$

with $a'(x) \geq 0$. It can be rewritten as a conservation law with a source term

$$\varphi_t + (a(x)\varphi)_x = a'(x)\varphi \tag{4}$$

The usual DG method can be applied to (4) as finding $\varphi_h \in V_h$ such that

$$\begin{aligned}
&\int_{I_j} \partial_t \varphi_h(x,t) v_h(x) dx - \int_{I_j} a(x) \varphi_h(x,t) \partial_x v_h(x) dx \\
&+a(x_{j+\frac{1}{2}}) \varphi_h(x^-_{j+\frac{1}{2}}, t) v_h(x^-_{j+\frac{1}{2}}) - a(x_{j-\frac{1}{2}}) \varphi_h(x^-_{j-\frac{1}{2}}, t) v_h(x^+_{j-\frac{1}{2}}) \\
= &\int_{I_j} a'(x) \varphi_h(x,t) v_h(x) dx
\end{aligned} \tag{5}$$

holds for all test functions $v_h(x) \in V_h$. The scheme (5) can be rewritten as

$$\begin{aligned}
&\int_{I_j} \left(\partial_t \varphi_h(x,t) + a(x) \partial_x \varphi_h(x,t) \right) v_h(x) dx \\
&+a(x_{j-\frac{1}{2}}) \left(\varphi_h(x^+_{j-\frac{1}{2}}, t) - \varphi_h(x^-_{j-\frac{1}{2}}, t) \right) v_h(x^+_{j-\frac{1}{2}}) = 0
\end{aligned}$$

This motivates the following definition of our DG scheme for general HJ equations: Find $\varphi_h \in V_h$, such that

$$\begin{aligned}
& \int_{I_j} (\partial_t \varphi_h(x,t) + H(\partial_x \varphi_h(x,t), x)) v_h(x) dx \\
+ \quad & \frac{1}{2} \left(\min_{x \in I_{j+\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j+\frac{1}{2}}) - \left| \min_{x \in I_{j+\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j+\frac{1}{2}}) \right| \right) [\varphi_h]_{j+\frac{1}{2}} (v_h \\
+ \quad & \frac{1}{2} \left(\max_{x \in I_{j-\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j-\frac{1}{2}}) + \left| \max_{x \in I_{j-\frac{1}{2}}} H_1(\partial_x \varphi_h, x_{j-\frac{1}{2}}) \right| \right) [\varphi_h]_{j-\frac{1}{2}} (v_h \\
= \quad & 0, \qquad j = 1, \ldots, N
\end{aligned}$$

holds for any $v_h \in V_h$.

A priori error estimates Xiong, Shu and Zhang, IJNAM 2013:

Theorem: For a smooth solution φ of the nonlinear Hamilton-Jacobi equation, the DG solution φ_h, using piecewise polynomials of degree k in 1D or piecewise tensor product polynomials of degree k in 2D on rectangular meshes, satisfies the optimal L^2 error estimates:

$$\|\varphi - \varphi_h\| \leq Ch^{k+1}.$$

Front propagation problems in the presence of obstacles

Bokanowski, Cheng and Shu, Num. Math. 2014; Math. Comp. in revision

We consider the following equation

$$\min(u_t + H(x, \nabla u),\ u - g(x)) = 0, \quad x \in \mathbb{R}^d, \quad t > 0, \tag{6}$$

together with an initial condition. Here $g(x)$ is the "obstacle function", and (6) is referred as the "obstacle equation".

Recently it was remarked in (Bokanowski, Forcadel and Zidani, SICON 2010) that (6) could be used to code the reachable sets of optimal control problems by using u as a level set function. It can be used to recover various objects such as the minimal time function.

We propose fully discrete and explicit DG methods for (6), with second (RK2) and third order (RK3) explicit Runge Kutta time discretization.

Compared to traditional finite element methods for such problems, the DG scheme proposed does not require solving a nonlinear equation at each time step. Rather, the obstacles are incorporated by a simple projection step given explicitly through a comparison with the obstacle functions at Gaussian quadrature points.

We derive stability and $O(\sqrt{h})$ error estimates for these fully discrete schemes, in the particular case where $H(x, \nabla u)$ is linear in ∇u (although the equation is nonlinear because of the obstacle term), leading to an equation of the form

$$\min(u_t + f(x) \cdot \nabla u,\ u - g(x)) = 0, \quad x \in \mathbb{R}^d \tag{7}$$

with Lipschitz continuous solutions.

Our strategy is mainly to use stability and error estimates of Zhang and Shu SINUM 2010 for fully discrete RKDG schemes for linear equations, and extend them to the obstacle case (under standard CFL condition $\tau/h \leq const$, where τ and h are the time and space steps respectively, for the RK2 and RK3 schemes).

A variational formulation

We have

$$(u_t + H(., \nabla u),\ u - g) = 0. \tag{8}$$

Let V^g be defined by

$$V^g = \{v \in L^2(\mathbb{R}^d),\ v(x) \geq g(x) \text{ a.e.}\}$$

Let v be in V^g. Using the previous identity, we obtain

$$(u_t + H(., \nabla u),\ v - u) = (u_t + H(., \nabla u),\ v - g) \geq 0 \tag{9}$$

Hence we obtain a variational formulation: to find u such that, for a.e. $t > 0$,

$$u(t,.) \geq g, \quad \text{and} \quad (u_t + H(., \nabla u),\, v - u) \geq 0, \quad \forall v \in V^g, \tag{10}$$

and

$$u(0,.) = u_0. \tag{11}$$

A DG scheme

We consider the following one-dimensional model problem on the interval I with periodic boundary conditions:

$$\min(u_t + u_x,\ u - g(x)) = 0. \tag{12}$$

For the unconstrained problem $u_t + u_x = 0$, the standard DG scheme with forward Euler time discretization can be written as: to find $u_h^{n+1} \in V_h$, such that

$$(u_h^{n+1} - u_h^n, v_h) - \tau \mathcal{H}(u_h^n, v_h) = 0, \quad \forall v_h \in V_h. \tag{13}$$

Here τ is the time step, $(\phi, \varphi) = \int_I \phi\varphi dx$ and

$$\mathcal{H}(\phi, \varphi) = \int_I \phi\, \varphi_x\, dx + \sum_j \phi^-_{j+\frac{1}{2}} [\varphi]_{j+\frac{1}{2}}, \tag{14}$$

where $[\varphi] = \varphi^+ - \varphi^-$ denotes the jump of the function φ at cell interface.

Now we propose a DG scheme for (12) with forward Euler discretization as follows: to find $u_h^{n+1} \in V_h^g$, such that

$$(u_h^{n+1} - u_h^n, v_h - u_h^{n+1}) - \tau \mathcal{H}(u_h^n, v_h - u_h^{n+1}) \geq 0, \quad \forall\, v_h \in V_h^g. \text{ (15)}$$

Here, the space $V_h^g := \{v_h \in V_h,\ , v_h(x_\alpha^j) \geq g(x_\alpha^j),\ \forall\, j, \alpha\}$, where $\{x_\alpha^j\}_{\alpha=1}^{k+1}$ are the $(k+1)$ Gaussian points on cell I_j, $j = 1, \ldots, N$.

Clearly, (15) can be viewed as a discrete version of (10).

The scheme (15) is now well defined but hard to implement because of the implicitness, the inequality and the non-standard space V_h^g. To address this difficulty, the main idea of our approach is to rewrite (15) in an equivalent form that is closely related to the unconstrained scheme (13).

Now we define $\varphi_\alpha^j(\cdot)$ to be Lagrange polynomials on I_j, such that $\varphi_\alpha^j(x_\beta^j) = \delta_{\alpha\beta}$. Then $u_h^n|_{I_j} = \sum_{\alpha=1}^{k+1} u_\alpha^{n,j}\, \varphi_\alpha^j(x)$, where $u_\alpha^{n,j}$ is the point value of u_h^n at x_α^j. Define the vector $U^{n,j} = \{u_1^{n,j}, u_2^{n,j}, \ldots, u_{k+1}^{n,j}\}$, and $U^n = \{U^{n,1}, U^{n,2}, \ldots, U^{n,N}\}$ then the DG scheme for the standard linear equation (13) can be written equivalently as:

$$\forall V \in \mathbb{R}^{k+1}, \quad (U^{n+1,j} M - U^{n,j} A - U^{n,j-1} B,\, V) = 0 \qquad (16)$$

with $(V, W) := \sum_{\alpha=1}^{k+1} V_j W_j$. M is the mass matrix and $M_{\alpha\beta} = (\varphi_\alpha^j, \varphi_\beta^j) = w_\alpha \delta_{\alpha\beta}$ where $w_\alpha > 0$ is the Gaussian weight. A and B are $(k+1) \times (k+1)$ matrices.

Since (16) holds for any $V \in \mathbb{R}^{k+1}$, we obtain the "finite difference version" of the DG scheme

$$U^{n+1,j} M - U^{n,j} A - U^{n,j-1} B = 0,$$

which can be equivalently written as

$$U^{n+1} = F(U^n).$$

Now, we consider the scheme (15). Denote $G_\alpha^j = g(x_\alpha^j)$, then

$$V_h^g = \{\varphi \in V_h, \text{ such that } \varphi(x_\alpha^j) \geq G_\alpha^j \text{ for all } j,\ \alpha\}.$$

We define a vector $V \geq G$ if all the element of V is greater than the corresponding element in G.

The scheme (15) for the obstacle problem can be written in the vector form as: to find $U^{n+1} \geq G$ such that

$$\forall\, V \geq G, \quad (U^{n+1,j} M - U^{n,j} A - U^{n,j-1} B,\, V - U^{n+1,j}) \geq 0 \quad (17)$$

Using the fact that M is positive and diagonal, taking $V = U^{n+1}$ except on interval I_j and for an index α, we obtain

$$\forall\, V_\alpha^j \geq G_\alpha^j, \quad (w_j\, U_\alpha^{n+1,j} - (U^{n,j} A + U^{n,j-1} B)_\alpha^j)\,(V_\alpha^j - U_\alpha^{n+1,j}) \geq 0. \quad (18$$

Hence $w_j\, U_\alpha^{n+1,j} - (U^{n,j} A + U^{n,j-1} B)_\alpha^j \geq 0$, for all α, j, by taking V_α^j large enough. Since $U^{n+1} \geq G$, we have

$$(w_j\, U_\alpha^{n+1,j} - (U^{n,j} A + U^{n,j-1} B)_\alpha^j)\,(G_\alpha^j - U_\alpha^{n+1,j}) \leq 0.$$

On the other hand, let $V_\alpha^j = G_\alpha^j$ in (18),

$$(w_j\, U_\alpha^{n+1,j} - (U^{n,j} A + U^{n,j-1} B)_\alpha^j)\,(G_\alpha^j - U_\alpha^{n+1,j}) \geq 0.$$

Hence,

$$(w_j\, U_\alpha^{n+1,j} - (U^{n,j} A + U^{n,j-1} B)_\alpha^j)\,(G_\alpha^j - U_\alpha^{n+1,j}) = 0.$$

This implies $U_\alpha^{n+1,j} \geq \frac{1}{w_j}(U^{n,j} A + U^{n,j-1} B)_\alpha^j = F(U^n)_\alpha^j$ and $U_\alpha^{n+1,j} \geq G_\alpha^j$ and one of the two inequalities must be an equality. Thus,

$$U_\alpha^{n+1,j} = \max(F(U^n)_\alpha^j, G_\alpha^j). \tag{19}$$

Conversely this last equation implies also (17). In conclusion, the DG scheme in (15) is equivalent to (19).

The DG scheme is thus:

1. compute $\tilde{u}_h^{n+1}$ from solving the unconstrained problem $u_t + u_x = 0$ by the standard DG method, namely to solve

$$(\tilde{u}_h^{n+1} - u_h^n, v_h) - \tau \mathcal{H}(u_h^n, v_h) = 0, \quad \forall\, v_h \in V_h$$

2. *"The Projection Step"*. Take the maximum of $\tilde{u}_h^{n+1}$ and $g(x)$ at the Gaussian points,

$$u_\alpha^{n+1,j} := \max\left((\tilde{u}_h)_\alpha^{n+1,j}, G_\alpha^j\right).$$

   Then recover u_h^{n+1} from those point values.

We can clearly see that the construction of the equivalent scheme (19) relies heavily on the fact that M is a diagonal matrix. In general, if M is not diagonal, we can not deduce a simple form as in (19). This justifies the choice of Gaussian points in the finite element space V_h^g.

A DG finite element method for the general obstacle problems (6) can be formulated similarly.

Numerical examples

Example 1.

We consider

$$f(t,x,y) := sign\left(\frac{T}{2} - t\right) \begin{pmatrix} -2\pi\, y \\ 2\pi\, x \end{pmatrix} \max(1 - \|\mathbf{x}\|_2, 0).$$

where $\|\mathbf{x}\|_2 := \sqrt{x^2 + y^2}$ and with a Lipschitz continuous initial data φ:

$$\varphi^0(x,y) = \min(\max(y,-1),1). \tag{20}$$

The function φ^0 has a 0-level set which is the x axis. In this example the front evolves up to time $t = T/2$ then it comes back to the initial data at time $t = T$. $T/2$ represents the number of turns.

Computations have been done up to time $T = 10$.

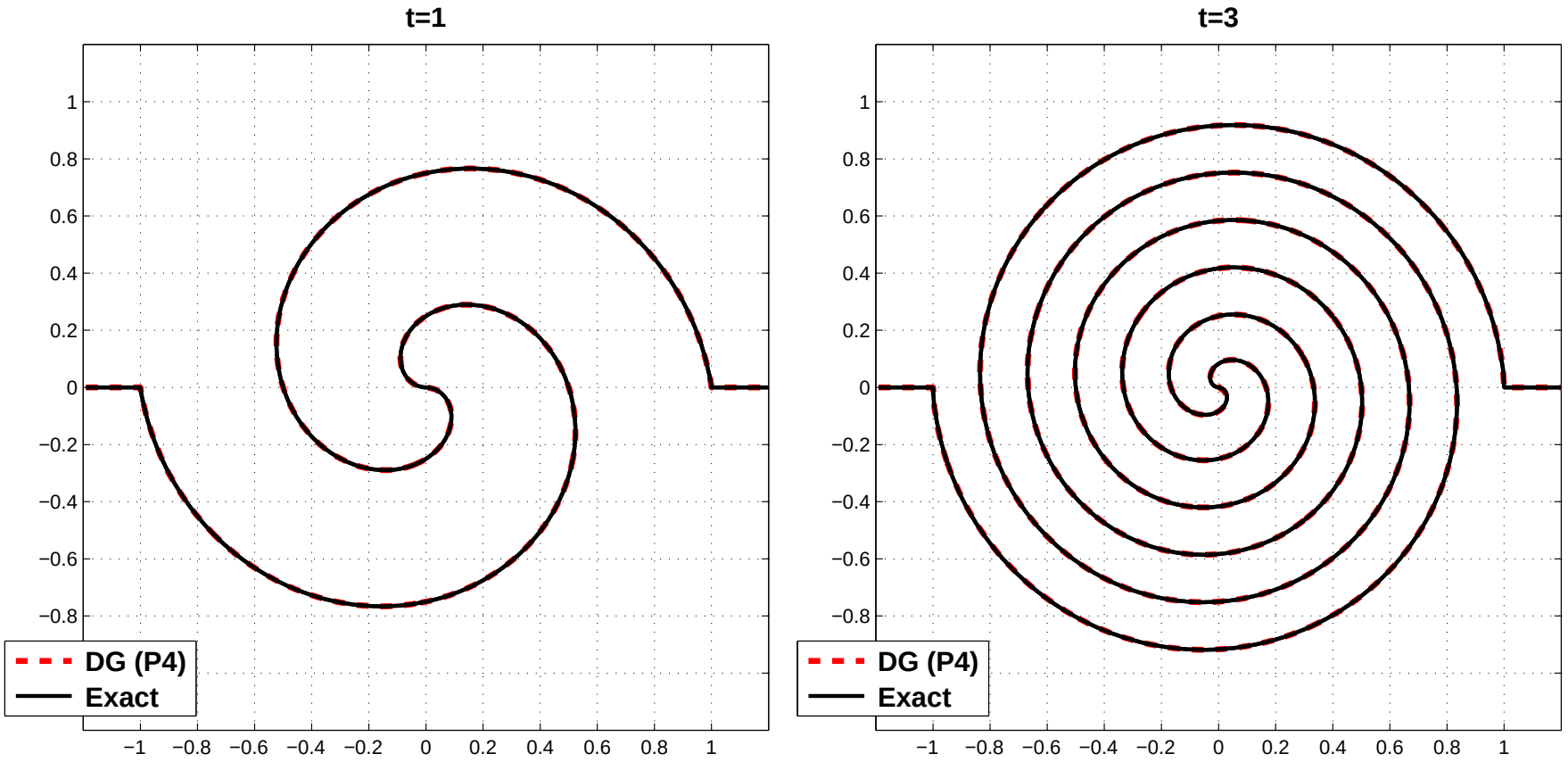

Figure 1: Example 1. Plots at times $t = 1$ and $t = 3$ with P^4 and 24×24 mesh cells.

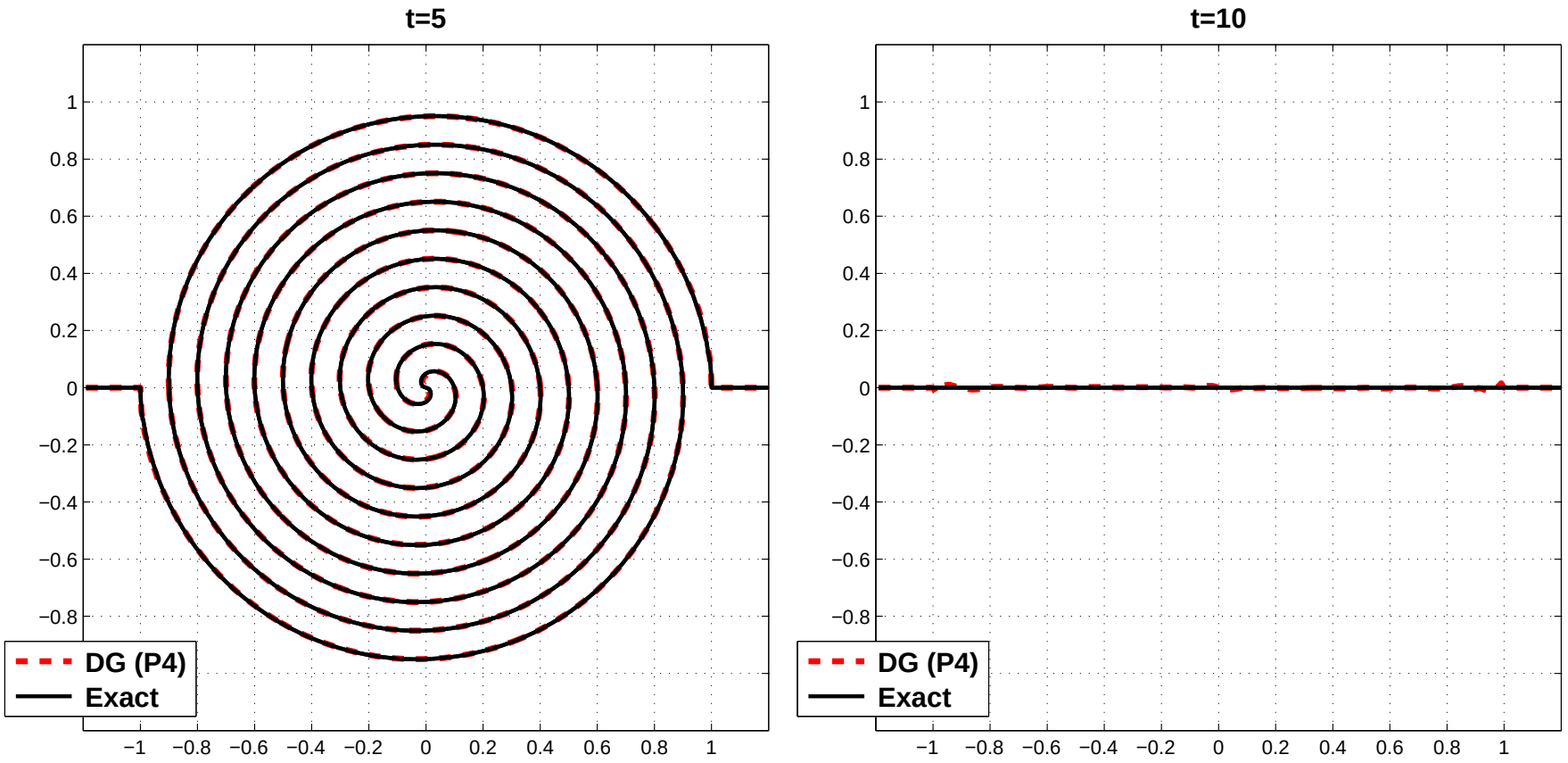

Figure 2: Example 1. Plots at times $t = 5$ and $t = 10$ (return to initial data), with P^4 and 24×24 mesh cells.

Example 2.

We first consider a one-dimensional test:

$$\min(u_t + u_x, u - g(x)) = 0, \quad t > 0,\ x \in [-1, 1], \tag{21}$$

$$u(0, x) = u_0(x), \quad x \in [-1, 1], \tag{22}$$

with periodic boundary conditions and $g(x) := \sin(\pi x)$,
$u_0(x) := 0.5 + \sin(\pi x)$.

In Table 1 we show the numerical errors away from the singular points of the solution $u(t, .)$. We observe the optimal third order convergence rate for P^2 elements. In Figure 3 we show the numerical solution which agrees well with the exact solution everywhere.

Table 1: Example 2. $t = 0.5$. P^2 elements.

N_x	Δx	L^1-error	order	L^2-error	order	L^∞-error	order
40	5.00e-2	3.34e-05	2.41	1.01e-04	1.98	7.02e-04	2.20
80	2.50e-2	1.77e-06	4.24	3.64e-06	4.79	2.82e-05	4.64
160	1.25e-2	1.78e-07	3.31	2.91e-07	3.64	2.40e-06	3.55
320	6.25e-3	2.13e-08	3.06	3.43e-08	3.08	1.28e-07	4.23
640	3.13e-3	2.66e-09	3.00	4.28e-09	3.00	1.60e-08	3.00
1280	1.56e-3	3.32e-10	3.00	5.35e-10	3.00	2.00e-09	3.00

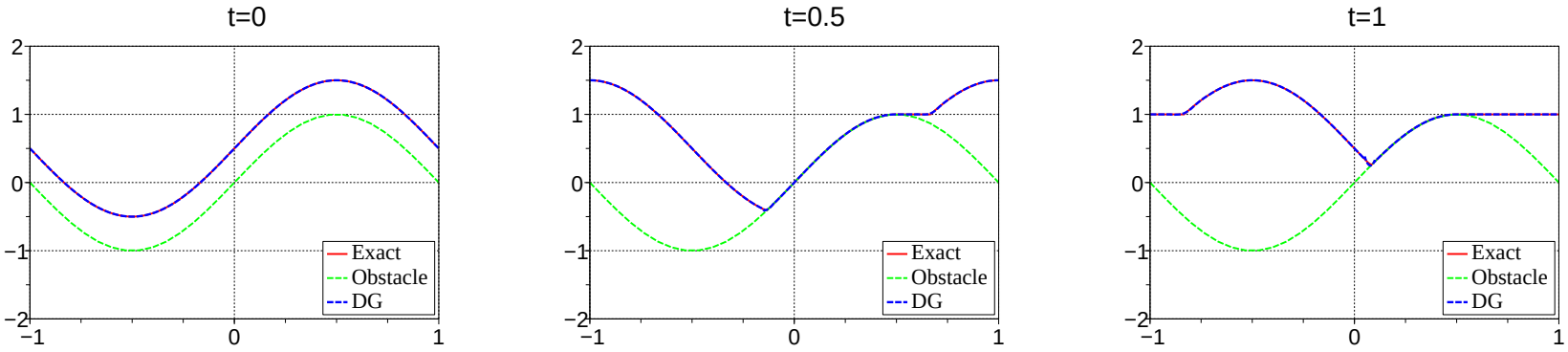

Figure 3: Example 2, times $t = 0$ (initial data), $t = 0.5$ and $t = 1$, using P^2 elements with $N_x = 20$ mesh cells (obstacle : green dotted line)

Example 3.

We consider a one-dimensional test with a nonlinear Hamiltonian:

$$\min(u_t + |u_x|, u - g(x)) = 0, \quad t > 0,\ x \in [-1, 1], \tag{23}$$

$$u(0, x) = u_0(x), \quad x \in \Omega, \tag{24}$$

with periodic boundary conditions and $g(x) := \sin(\pi x)$,
$u_0(x) := 0.5 + \sin(\pi x)$.

In Figure 4 we show the numerical solution for times $t \in \{0.2, 0.4\}$,
which agrees well with the exact solution.

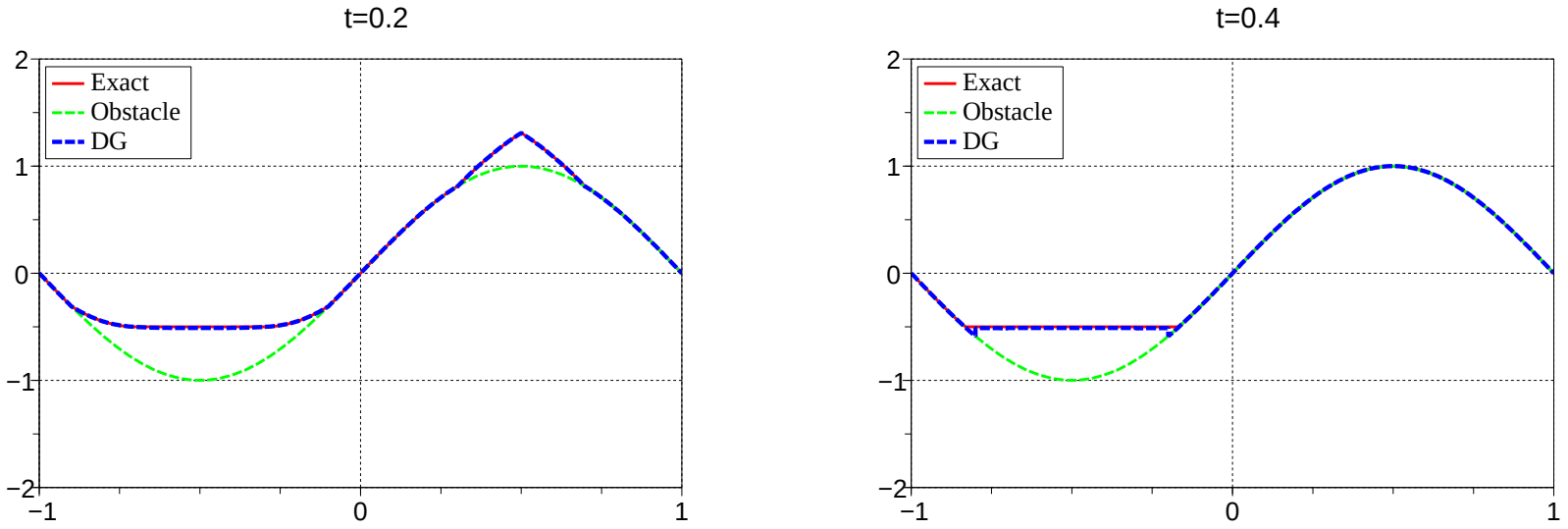

Figure 4: Example 3, numerical and exact solutions at times $t = 0.2$ and $t = 0.4$, $N_x = 20$, using P^2 (obstacle : green dotted line).

Example 4.

The initial data is $u_0(\mathbf{x}) := \|\mathbf{x} - (-0.5, 0)\|_\infty - 0.3$, with $\mathbf{x} = (x, y)$. The obstacle is coded by $g(\mathbf{x}) := 0.25 - \|\mathbf{x} - (0, 0.25)\|_\infty$. The equation solved is

$$\min(u_t + u_x, u - g(x, y)) = 0, \quad t > 0, \ (x, y) \in \Omega, \tag{25}$$

$$u(0, x, y) = u_0(x, y), \quad (x, y) \in \Omega, \tag{26}$$

and the domain is $\Omega := [-1, 1]^2$ (with periodic boundary conditions).

Results are shown in Fig. 5 at three different times, which agree well with the exact solution.

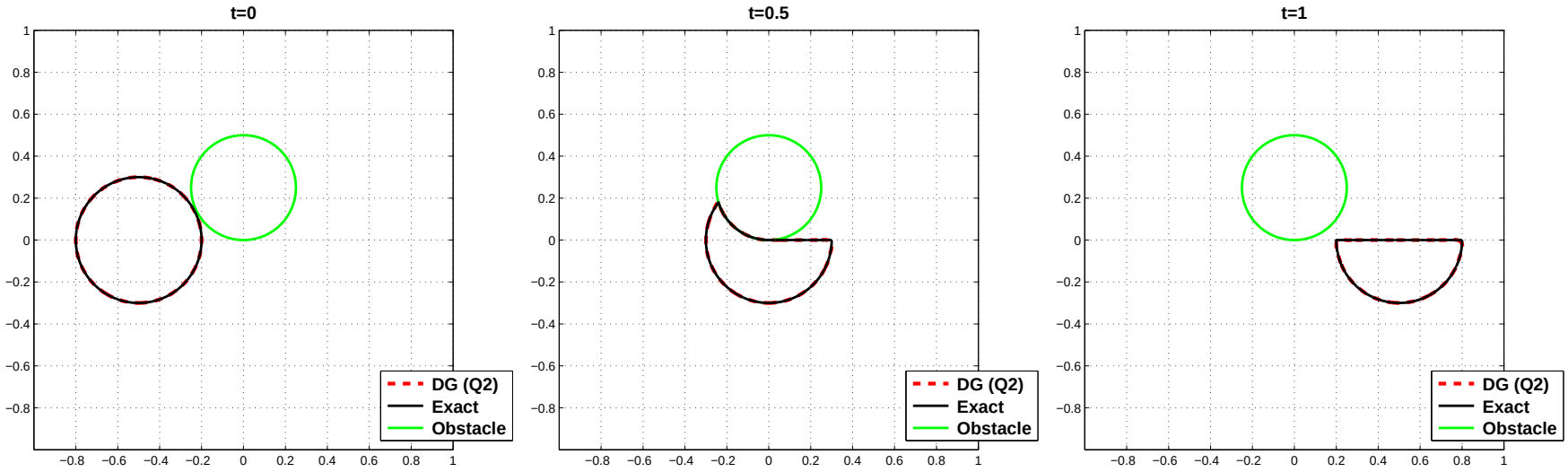

Figure 5: Example 4 ($N_x = N_y = 40$), times $t \in \{0,\ 0.5,\ 1\}$

Example 5.

We consider 2–d linear with variable coefficients

$$f(x,y) := \begin{pmatrix} -2\pi\ y \\ 2\pi\ x \end{pmatrix} \max(1 - \|\mathbf{x}\|_2, 0).$$

where $\|\mathbf{x}\|_2 := \sqrt{x^2 + y^2}$ and with a Lipschitz continuous initial data u_0:

$$u_0(x,y) = \min(\max(y, -1), 1). \tag{27}$$

The obstacle constraint is well taken into account (the square obstacle prevents the front to evolve between two arcs tangent to the square).

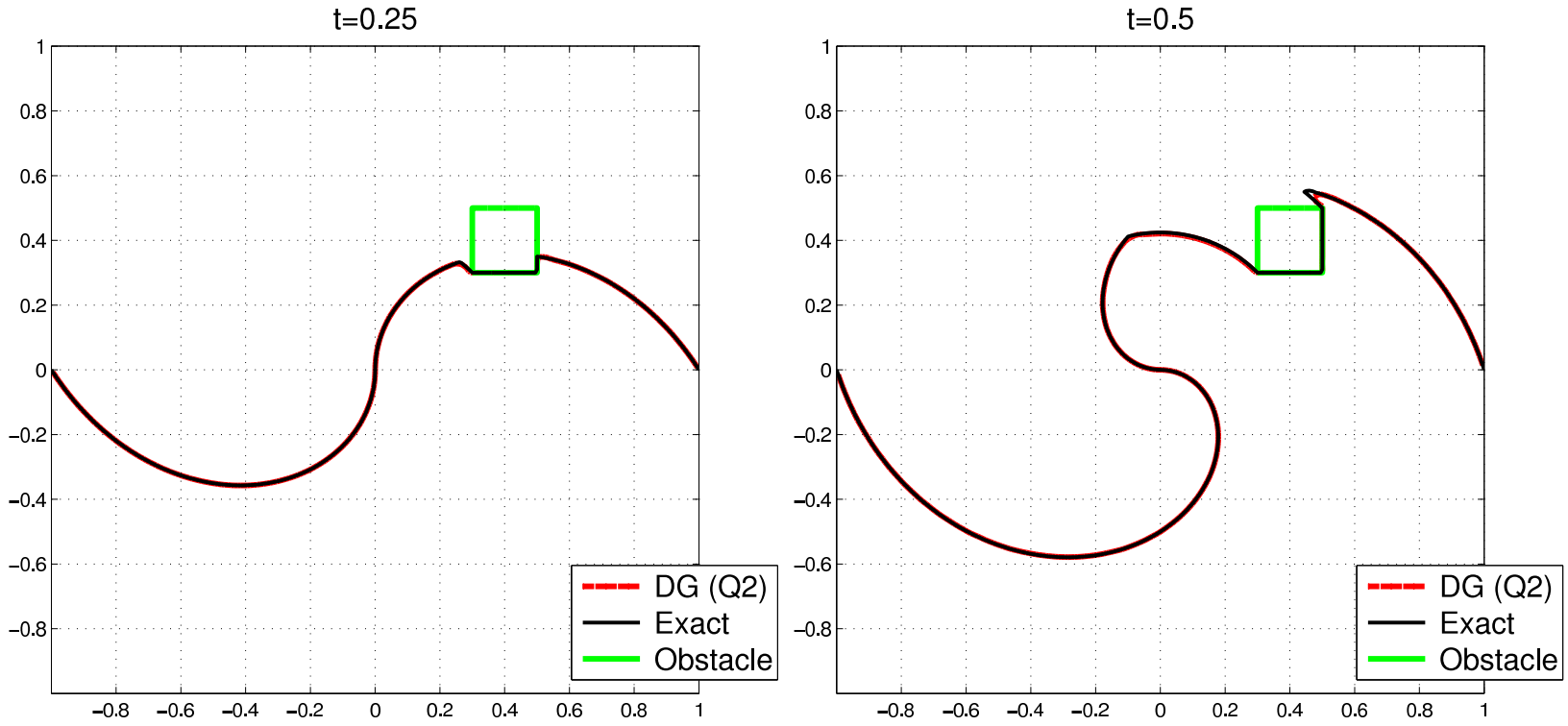

Figure 6: Example 5, plots at times $t \in \{0.25,\ 0.5,\ 0.75,\ 1\}$, with Q^2 and 40×40 mesh cells.

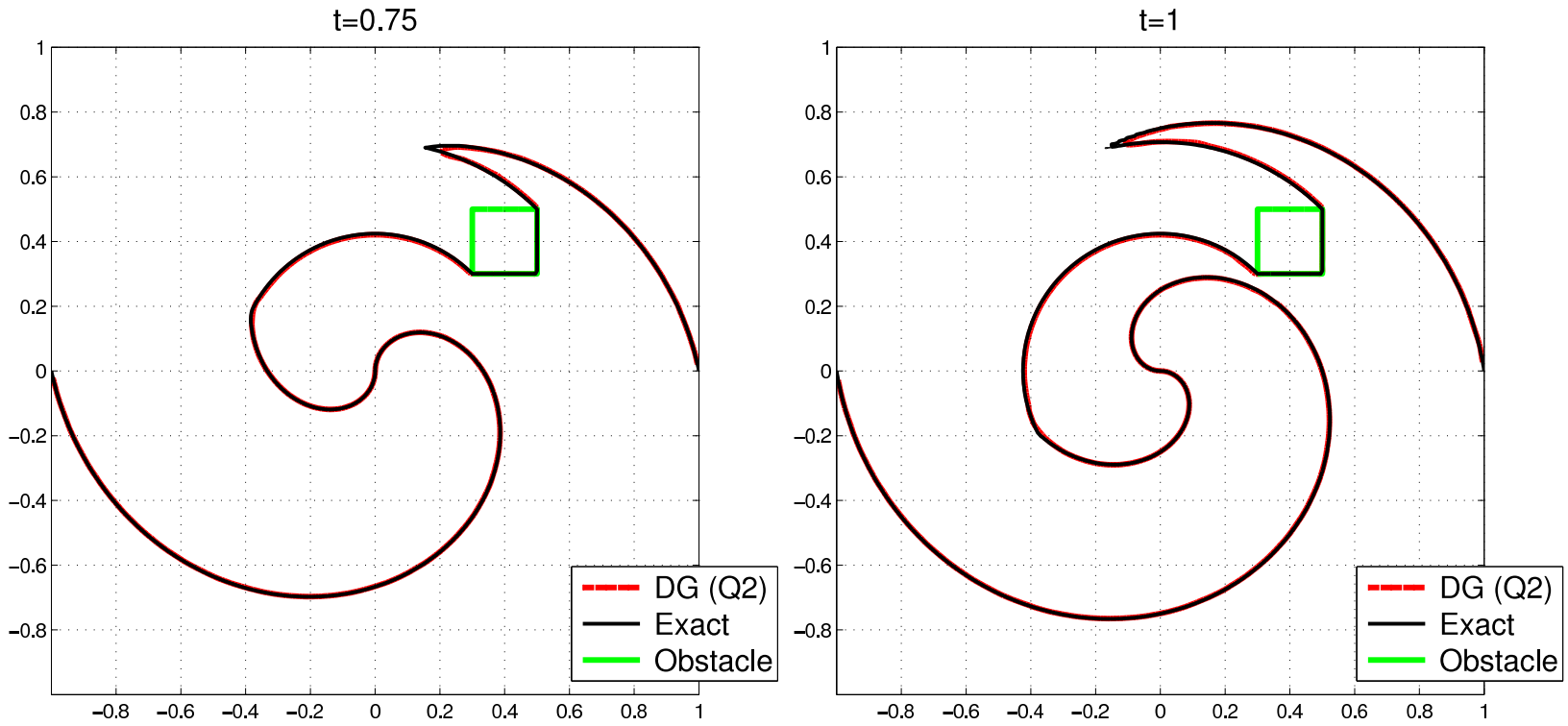

Figure 7: Example 5 (continued).

Example 6.

In this example we consider an initial data $u_0(x,y) := \|(x,y)-(1,0)\|_\infty - 0.5$, an obstacle coded by $g(x,y) := 0.5 - \|(x,y)-(0,\ 0.5)\|_\infty$, and the problem

$$\min(u_t + \max\left(0, 2\pi(-y,x)\cdot\nabla u\right), u - g(x,y)) = 0, \quad t > 0,\ (x,y)$$
$$u(0,x,y) = u_0(x,y), \quad (x,y) \in \Omega,$$

The domain is $\Omega := [-2,2]^2$. Thus we want to compute the backward reachable set associated to the dynamics $f(x,y) = -2\pi(-y,x)$ and the target $\mathcal{T} = \{(x,y), u_0(x,y) \leq 0\}$, together with an obstacle or forbidden zone represented by $\{(x,y), g(x,y) \geq 0\}$.

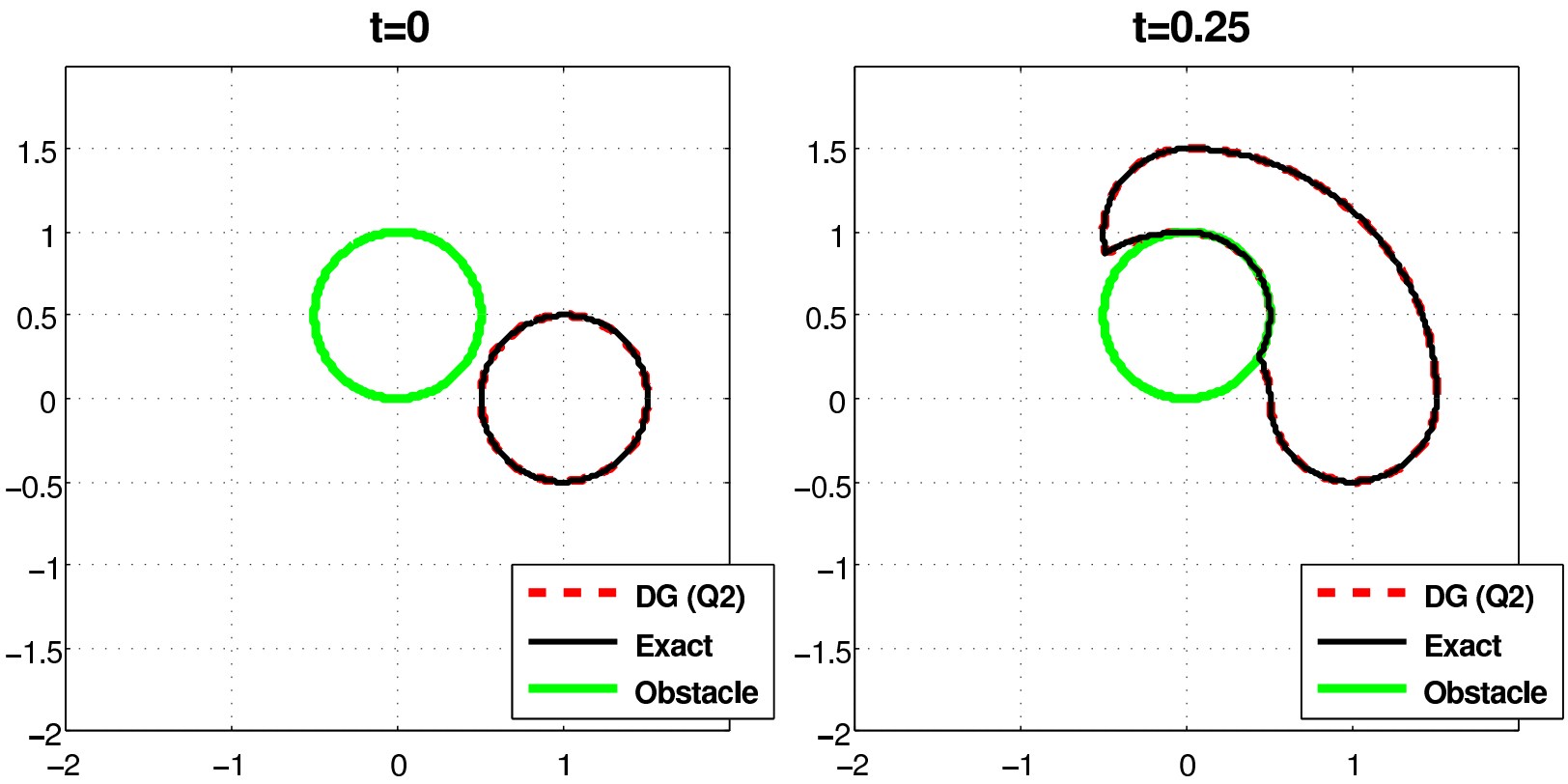

Figure 8: Example 6. Plots at times $t \in \{0,\ 0.25,\ 0.5,\ 0.75\}$, with Q^2 and 80×80 mesh cells.

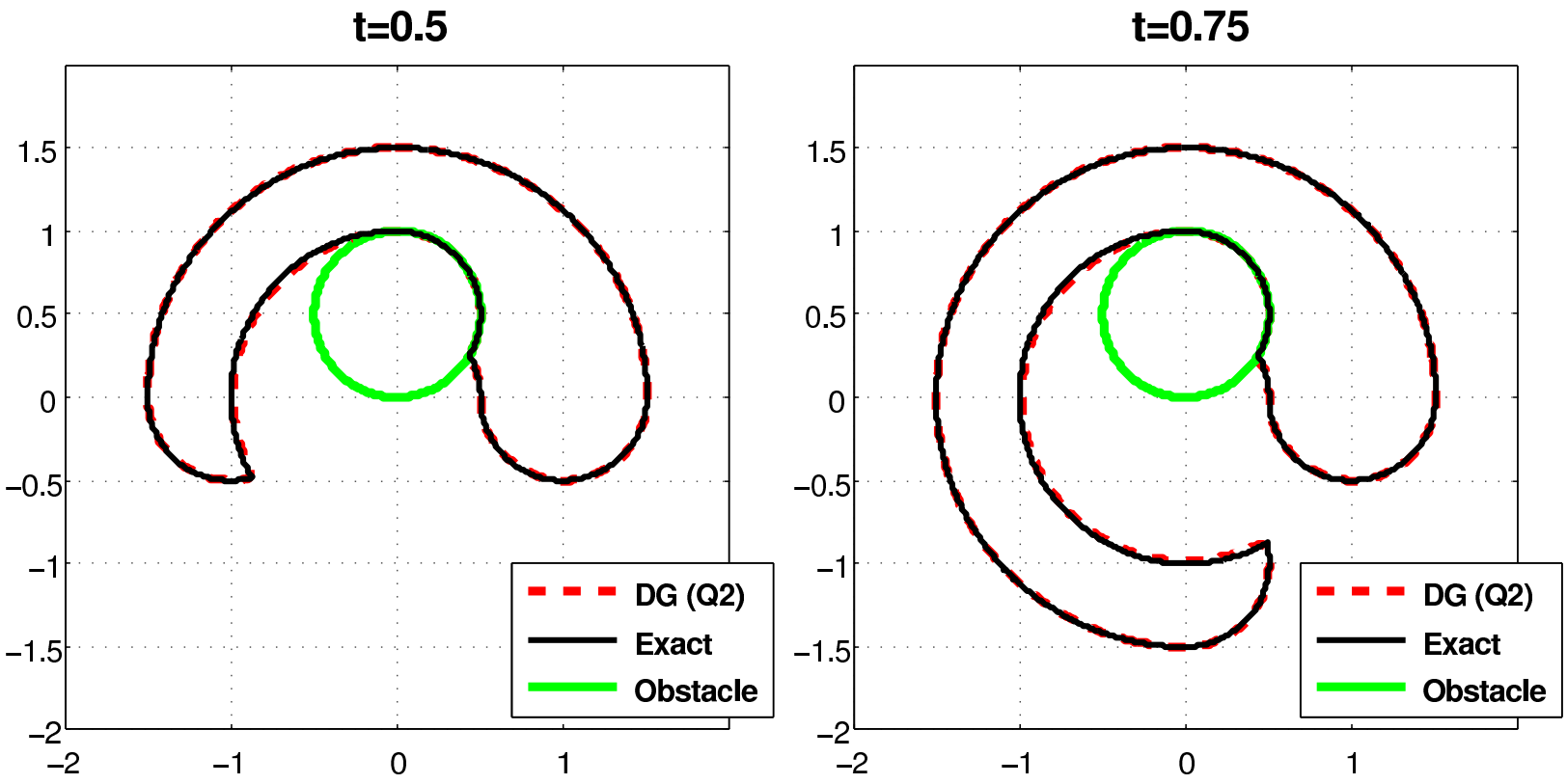

Figure 9: Example 6 (continued)

Example 7.

We consider the problem

$$\min(u_t + \max\left(0, u_x + \frac{1}{2}|u_y|\right), u - g(x,y)) = 0, \quad t > 0,\ (x,y) \in ($$

$$u(0,x,y) = u_0(x,y), \quad x \in \Omega, \qquad ($$

with $u_0(\mathbf{x}) := \|\mathbf{x} - (-1.0,\ 0)\|_\infty - 0.5$ and
$g(\mathbf{x}) := \min\left(0.25, \|\mathbf{x} - (0.2,\ 0)\|_2 - 0.5\right)$, corresponding to a square initial data and a disk obstacle.

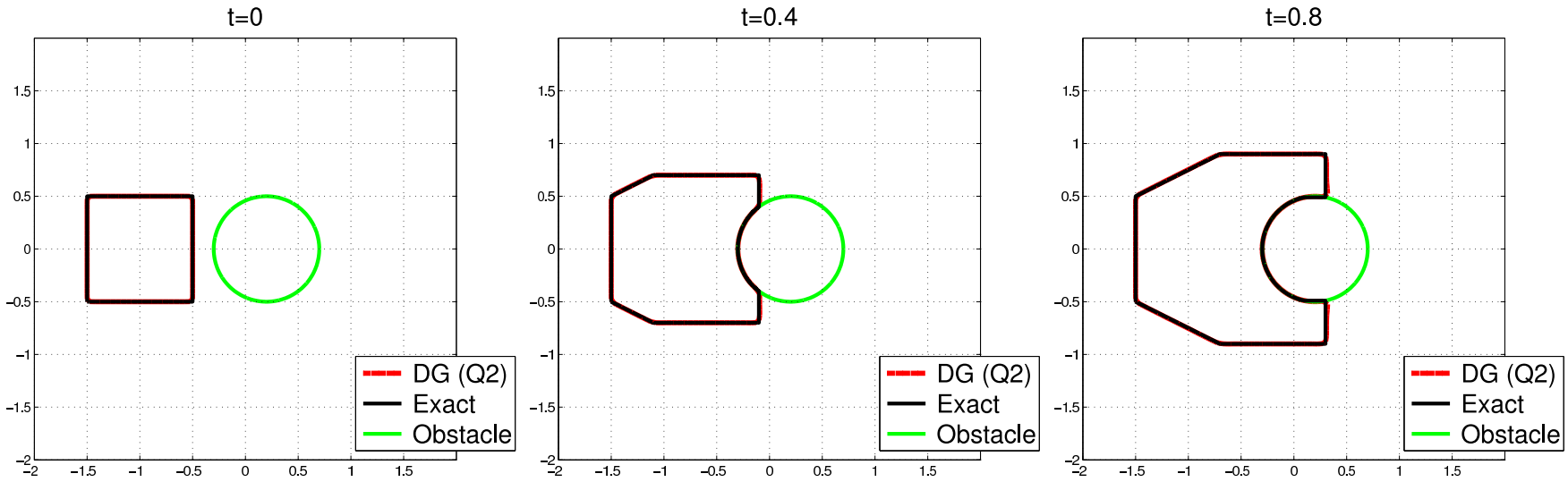

Figure 10: Example 7, plots at times $t \in \{0,\ 0.4,\ 0.8,\ 1.2,\ 1.6,\ 2.0\}$, with Q_2 and 80×80 mesh cells.

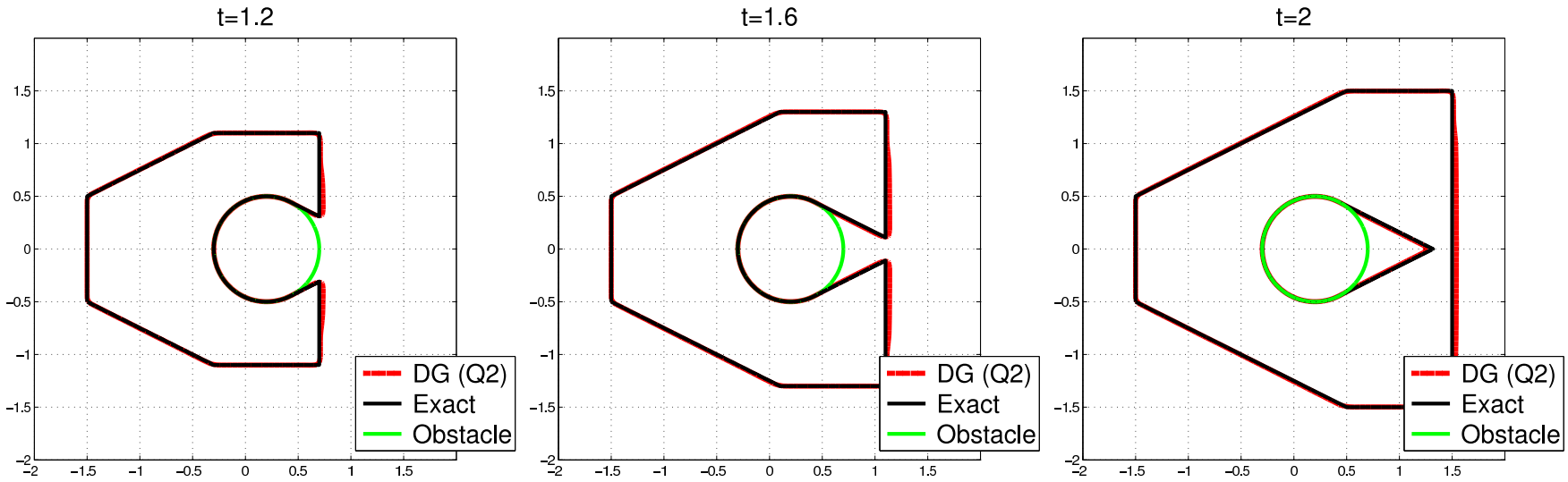

Figure 11: Example 14 (continued)

Example 8.

In this example we propose a simple narrow band approach adapted to front propagation problems using the DG schemes of the present paper.

In Figure 12 we apply the narrow band approach and have plotted with dots the narrow band cells which are used at different times.

In Table 2 we show some CPU times for a simplified advection problem (rotation of a circle)

$$u_t + 2\pi(-y, x) \cdot \nabla u = 0, \quad t > 0, \ (x, y) \in \Omega, \tag{32}$$

$$u(0, x, y) = u_0(x, y), \quad (x, y) \in \Omega, \tag{33}$$

The results for the complete example are similar except for a scaling factor on the CPU times.

Here the "order" is computed as the ratio of CPU times $time(N_x)/time(N_x/2)$. For large N_x values, we observe an order of 8 (approximately) for the full approach, and of 4 (approximately) for the narrow band approach. This is justified for the narrow band approach because it will use a number of cells proportional to the length of the front (which accounts for a factor of 2) and there is another factor of 2 coming from the CFL condition $\tau \leq const.\Delta x$.

Note that in Table 3 we have made use of a parallel version of the code (Fortran OpenMP on a 8 core processor) to reduce the computational cost, since the DG schemes have the advantage to be easily parallelizable.

Table 2: (Example 8) comparison of CPU times (in sec.) for full and narrow band approaches, $t = 0.5$

N_x	full	"order"	narrow band	"order"	Gain (full / band)
20	8.1 s	-	6.9 s	-	1.17
40	45.2 s	5.58	17.1 s	2.47	2.64
80	347.4 s	7.68	83.4 s	4.87	4.16
160	2705.3 s	7.78	386.0 s	4.62	7.00

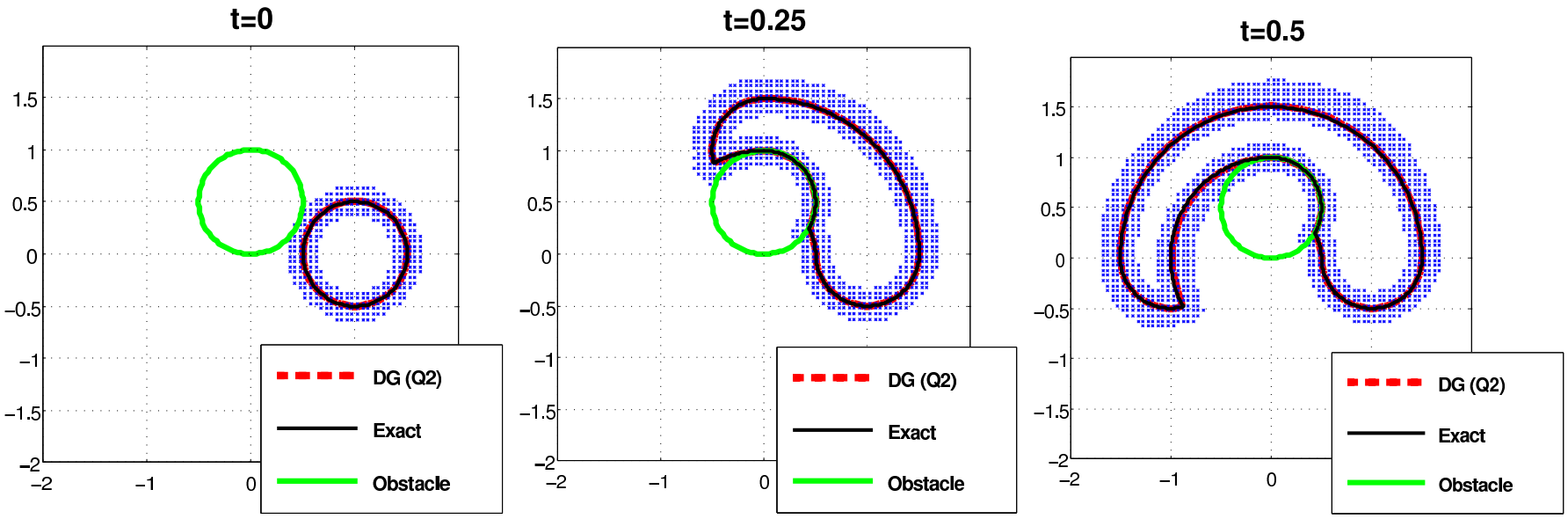

Figure 12: Example 8. Narrow band approach for same problem as in Example 6, with $t \in \{0, 0.25, 0.5\}$, with Q_2 and 80×80 mesh cells.

The End

THANK YOU!