Communications to SIMAI Congress, ISSN 1827-9015, Vol. 1 (2006)

A parallel semi-Lagrangian approximation of the effective Hamiltonian

Maurizio Falcone Dipartimento di Matematica Università di Roma "La Sapienza" falcone@mat.uniroma1.it

Marco Rorro

Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate Università di Roma "La Sapienza" rorro@dmmm.uniroma1.it

The computation of the effective Hamiltonian is a difficult and challenging problem which arises in many applications related to the analysis of Hamiltonian systems, homogenization and material science, see [7,5,4]. The characterization of the effective Hamiltonian \overline{H} in the framework of viscosity solutions has revitalized the efforts to find new methods and efficient algorithms to compute it. For reader's convenience let us recall that for every $p \in \mathbb{R}^n$ the value $\overline{H}(p)$ can be characterized as the unique value of the right-hand side for which the cell problem

(0.1)
$$H(x, Du+p) = c \quad x \in \mathbb{T}^n$$

has a viscosity solution (\mathbb{T}^n denotes the n-dimensional torus). Note that the Hamiltonian H is assumed to be convex with respect to Du.

The basic computational effort then is to compute $\overline{H}(p)$ for p in a domain D and, for p fixed, find an efficient numerical approximation for the cell problem via a sequence c_n in such a way that c_n converges to $\overline{H}(p)$ as n tends to $+\infty$. The amount of the above computations can be huge even for low dimensional problems (i.e. n = 2, 3).

We can identify two basic steps in the algorithm. The first step consists in solving for a given $p \in D$ a sequence of cell problems for different right hand sides c_n in order to obtain an approximation of $\overline{H}(p)$. At this step a crucial role is played by the accuracy of the numerical method used to solve the cell problem on a space grid G_x . The second step, though computationally expensive, is much easier since it requires to iterate the same process over a grid for p (we will denote this grid by G_p in the sequel). The final value of \overline{H} is obtained just interpolating over the values computed at the nodes of G_p .

Several methods have been proposed in the last few years to solve this problem. They rely on different characterizations of the effective Hamiltonian and on various regularizations of the cell problem. Let us examine the most significant ones. A first attempt to the numerical approximation of the effective Hamiltonian has been proposed by Gomes and Oberman in [8]. They compute the effective Hamiltonian via a discrete version of a min-max representation formula

(0.2)
$$\overline{H}(p) = \inf_{u \in C^1_{per}(\mathbb{T}^n)} \sup_{x \in \mathbb{T}^n} H(x, Du + P)$$

where the infimum is taken over the space $C^1(\mathbb{T}^n)$ of periodic function. It is important to note that they do not solve the cell problem. Our approach relies on a discretization of

M. Falcone et al.

(0.1). In this way we are able to compute both the effective Hamiltonian and a viscosity solution.

To this end it is useful to recall two regularizations of the cell problem: $small-\delta$ regularization

(0.3)
$$\delta u^{\delta}(x) + H(x, Du^{\delta} + P) = 0 \quad \text{in } \mathbb{T}^n$$

large-T regularization

(0.4)
$$u_t + H(x, Du + P) = 0 \quad \text{in } \mathbb{T}^n \times (0, +\infty)$$

The approximation of the effective Hamiltonian relies on the following properties. Denote by u^{δ} the solution of (0.3). Under suitable assumptions (see [5]) $-\delta u^{\delta}$ converges to $\overline{H}(P)$ as δ goes to 0. Moreover, for any fixed $x_0 \in \mathbb{T}$, $u^{\delta}(x) - u^{\delta}(x_0)$ converges, up to a subsequence, to a viscosity solution of (0.1). Now let u(x,t) be a solution of (0.4). Under suitable assumptions, (see [4,2,1]) $-\frac{u(x,t)}{t}$ converges to $\overline{H}(P)$ as t goes to $+\infty$. Moreover, for any fixed $x_0 \in \mathbb{T}$, $u(x,t) - u(x_0,t)$ converges, up to a subsequence, to a viscosity solution of (0.1). The interested reader can find in [3] some estimates on the rate of convergence. In the following section we will adopt the large-T regularization to construct our scheme. A similar approach has been adopted by Qian in [9] using finite difference schemes for the cell problem.

The goal of this paper is to propose a parallel version of the algorithm in [10,11] in order to overcome the computational difficulties and open the way to applications in higher dimensions.

1 The semi-Lagrangian approximation

A serial version of the semi-Lagrangian approximation has been proposed in [10] for a general Hamiltonian (convex and coercive). For simplicity, we write it for the eikonal equation, H(x, Du + p) = |Du + p| - f(x),

(1.1)
$$u_i^{n+1} = \min_{a \in B(0,1)} \left\{ u^n (x_i + a\Delta t) + a \cdot p\Delta t \right\} + [f(x_i) + c]\Delta t.$$

where x_i , $i = 1, ..., N_x$ are the nodes of G_x and we adopt the standard notation $u_i^n = u(x_i, t_n)$. Note that the value $u^n(x_i + a\Delta t)$ is computed by interpolation as usual for SL schemes (see [6]). In particular, in our numerical tests we use a linear interpolation reconstruction.

We fix the starting value $c_0 = -\min f$ and update it by the iteration $c_{n+1} = c_n - \overline{v}$ where $\overline{v} = \sum_i u_i^n / \Delta x$. This is possible because the method produces a sequence u^n which converges to a solution u, see [10,11] for details.

The parallel SL scheme

As we said in the introduction the computational effort to solve even a two dimensional problem is huge. For example, for a cell problem in $\mathbb{T}^2 = [0, 1]^2$ with a grid G_x of 32×32 nodes and G_p of 16×16 we have to solve 256 cell problems over a grid of 1024 space nodes.

These remarks motivate the introduction of a parallel algorithm. The nature of the problem suggests to adopt a nested parallelism. A low-level parallelism for the domain

decomposition of G_x , and a high-level parallelism for the domain decomposition of G_p . The domain decomposition of G_x is obtained via OpenMP directives for shared memory architecture. This avoids the communications bottleneck between the processors due to the overlapping regions of the domain decomposition. For the domain decomposition of G_p we use MPI, distributing the columns of the grid G_p between the processors. So every processor solves the problem in a subdomain of G_p , and at every p the cell problem is solved in parallel. In the following section we will show the advantages of this technique. Note that the parallelization of the code is rather simple since only few directives are needed to parallelize the loop over G_x and easy collective communications allow to distribute and collect the data to parallelize the loop over G_p .

2 Numerical tests

Let us examine some results on the classical problem:

(2.1)
$$|Du + p| - \sin(2\pi x_1)\sin(2\pi x_2) = \overline{H}(p)$$

that is solved for $p \in [0, 4]^2$ for different values of Δp , and on the unitary torus \mathbb{T}^2 for different values of Δx .

In the following tables, the rows show how the CPU time (in seconds) varies by increasing the number of processors created by MPI (high-level loop); on the columns the same information is given for the OpenMP loop (low-level). The reverse diagonals show the behaviour for a fixed global number of processors (the product of the number of processors on the row times that on the columns). The tests are performed on an IBM Power5 (8 CPU at 1.9 GHz for each node) at CASPUR. The CPU time decreases along the rows and the columns and sometimes also on the reverse diagonals. For example, Table 2.1 and 2.2 show the advantage of passing from an MPI parallelization with 32 processors to an MPI parallelization with 16 processors, each one of them uses 2 processors managed by OpenMP. This is more evident in Figure 2.1 and 2.2, where the speed-up and efficiency for $\Delta x = 0.0625$ and $\Delta p = 0.1$ are represented. On the other hand this advantage disappears for $\Delta p = 0.05$ (see Table 2.3 and 2.4). Although in all the tests presented the numbers of nodes of G_p are much greater than those of G_x , it is evident that when increasing the number of nodes of G_x it is more convenient to increase the processors managed by OpenMP rather than those by MPI.

# procs	MPI	1	2	4	8	16	32
OpenMP							
1		3154	1640	872	483	244	166
2		1603	831	442	245	123	
4		900	468	247	137		
8		552	286	151			

Table 2.1: CPU time in seconds for different values of the number of CPU for $\Delta x = 0.0625$, $\Delta p = 0.1$

# procs	MPI	8	16	32
OpenMP				
1		6528	3301	2244
2		3358	1702	
4		1793		

Table 2.2: CPU time in seconds for different values of the number of CPU for $\Delta x = 0.03125$, $\Delta p = 0.1$

# procs	MPI	8	16	32
OpenMP				
1		1650	908	459
2		880	486	
4		493		

Table 2.3: CPU time in seconds for different values of the number of CPU for $\Delta x = 0.0625$, $\Delta p = 0.05$

# procs	MPI	16	32
OpenMP			
1		12958	6594
2		6698	

Table 2.4: CPU time in seconds for different values of the number of CPU for $\Delta x = 0.03125$, $\Delta p = 0.05$



Figure 2.1: Speed-up for Table 2.1

Figure 2.2: Efficiency for Table 2.1

REFERENCES

- 1. G. Barles and P.E. Souganidis, On the large time behavior of solutions of Hamilton– Jacobi equations, *SIAM J. Mat. Anal.*, vol. 31, 2000, 925–939.
- 2. G. Barles and J.M. Roquejoffre, Large time behaviour of fronts governed by eikonal equations, *Interfaces and Free Boundaries*, vol. 5, 2003, 83–102.
- F. Camilli, I. Capuzzo Dolcetta, An error estimate for the approximation of the effective Hamiltonian, Dipartimento di Matematica, Università di Roma "La Sapienza", prepint, 2005.
- M.C. Concordel, Periodic homogenization of Hamilton–Jacobi equation: Additive eigenvalues and variational formula, *Indiana Univ. Math. J.*, vol. 45, 1996, 1095– 1117.
- 5. L.C. Evans, Periodic homogenization of certain nonlinear partial differential equation, *Proc. Roy. Soc. Edinburgh*, vol. 120 A, 1992, 245–265.
- M. Falcone and R. Ferretti, Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Goudonov methods, J. Comput. Phys., vol. 175, 2002, 559-575.
- 7. A. Fathi, Weak KAM Theorem in Lagrangian Dynamics, Cambridge University Press, in press.
- 8. D.A. Gomes and A. Oberman, Computing the effective Hamiltonian using a variational approach, *SIAM J. Cont. Opt.*, vol. 43, 2004, 798–812.
- 9. J. Qian, Two approximations for effective Hamiltonians arising from homogenization of Hamilton-Jacobi equations, UCLA CAM report, 03-39, 2003.
- 10. M. Rorro, Numerical approximation of the effective Hamiltonian and of the Aubry set for first order Hamilton–Jacobi equations, *PoS Sissa* cstna2005-016, 2005.
- 11. M. Rorro, An approximation scheme for the effective Hamiltonian and applications, *Applied Numerical Mathematics*, Elsevier, to appear.