HOW TO PREDICT ACCURATE WAVELET GRIDS IN ADAPTIVE SEMI-LAGRANGIAN SCHEMES? *

Martin Campos Pinto[1]

Résumé. Je présente dans cet article un nouveau schéma semi-Lagrangien à base d'ondelettes pour l'approximation de problèmes de transport associés à des flots réguliers. Cette méthode s'inspire de celle proposée par Besse, Filbet, Gutnic, Paun et Sonnendrücker [1], mais s'en distingue notamment par le fait qu'elle s'articule autour de la notion de bonne adaptation d'un arbre d'ondelettes à une fonction donnée. De plus, elle s'accompagne d'une estimation d'erreur a priori.

Dans un travail précédent effectué en collaboration avec Mehrenberger [3], nous avions conçu un premier schéma adaptatif semi-Lagrangien basé sur des maillages multi-échelles et hiérarchiques. Notre méthode consistait à prédire un nouveau maillage à chaque pas de temps par une stratégie peu coûteuse, puis à le réadapter suivant la régularité de la solution numérique transportée. Par une analyse rigoureuse, nous avions établi qu'une simple application de notre algorithme de prédiction et de correction suffisait à garantir une estimation d'erreur a priori.

Le schéma présenté ici met en œuvre des idées semblables, mais dans un cadre différent. La propriété d'être (fortement) bien adapté à une fonction donnée est donc redéfinie pour des arbres d'ondelettes, et l'on peut montrer qu'elle est (faiblement) préservée par une stratégie de prédiction, toujours peu coûteuse, qui transporte les grilles d'ondelettes le long de flots réguliers. En conséquence, il est possible d'établir une estimation d'erreur a priori pour ce nouveau schéma de "prédiction et réadaptation", sous l'hypothèse principale que le flot sous-jacent, ainsi que son approximation numérique, est un difféomorphisme stable. Un résultat partiel concernant la complexité des grilles obtenues est également énoncé. Les preuves correspondantes sont disponibles dans les notes de cours [2].

44

Abstract. In this article I present a new adaptive semi-Lagrangian scheme based on wavelet approximations for solving transport equations with underlying smooth flow. Inspired by the method of Besse, Filbet, Gutnic, Paun and Sonnendrücker [1], this new approach differs in the fact that it is mostly driven by the notion of good adaptation of a wavelet tree to a given function. Moreover it comes with guaranteed error estimates.

In a previous joint work with Mehrenberger [3], we had designed a first adaptive semi-Lagrangian scheme based on multilevel, hierarchical meshes. The method consisted in predicting a new adaptive mesh for every time step by using a low-cost strategy, and next readapt it once according to the smoothness of the transported numerical solutions. By a rigorous analysis we could prove that our scheme had a prescribed accuracy, achieved by applying the prediction and correction algorithms only once per time step.

The present scheme implements similar ideas, but now in the framework of interpolatory wavelets. For this purpose I translate the property of being (strongly) well-adapted to a given function in the context of wavelet trees, and show that it is (weakly) preserved by a low-cost prediction algorithm which transports wavelet grids along any smooth flow. As a consequence, error estimates can be established for the resulting "predict and readapt" scheme under the essential assumption that the flow underlying the transport equation, as well as its numerical approximation, is a stable diffeomorphism. One complexity result is stated in addition. The proofs can be found in the lecture notes [2].

1. Introduction

1.1. The model problem: an abstract smooth transport equation

We consider a transport equation in several dimensions, of the form

$$\partial_t f(t, \boldsymbol{x}) + \boldsymbol{V}(t, \boldsymbol{x}) \cdot \nabla f(t, \boldsymbol{x}) = 0, \qquad t \in [0, T], \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{1}$$

associated with a given initial condition $f_0 : \mathbb{R}^d \mapsto \mathbb{R}$ and a final time $T > 0$. In order to keep some level of generality we allow the velocity field $\boldsymbol{V} : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ to depend on f, but assume that it is *smooth* in the following sense.

Assumption 1.1. For any $s \in [0, T]$, $\boldsymbol{x} \in \mathbb{R}^d$, the ordinary differential equation

$$\partial_t \boldsymbol{X}(t) = \boldsymbol{V}(t, \boldsymbol{X}(t)) \quad \text{for} \quad t \in [0, T], \quad \text{given} \quad \boldsymbol{X}(s) = \boldsymbol{x}, \tag{2}$$

possess a unique solution $\boldsymbol{X} = \boldsymbol{X}(\cdot; s, \boldsymbol{x})$ on $[0, T]$. Moreover, for any instant t in $[0, T]$, the associated flow $\mathcal{F}_{s,t} : \boldsymbol{x} \to \boldsymbol{X}(t; s, \boldsymbol{x})$ is a diffeomorphism from $\mathbb{R}^d$ into itself, with inverse $\mathcal{B}_{s,t} := (\mathcal{F}_{s,t})^{-1} : \boldsymbol{x} \mapsto \boldsymbol{X}(s; t, \boldsymbol{x})$. In particular,

$$\max\left\{ \|\mathcal{B}_{s,t}(\boldsymbol{x} + \boldsymbol{h}) - \mathcal{B}_{s,t}(\boldsymbol{x})\|, \|(\mathcal{B}_{s,t})^{-1}(\boldsymbol{x} + \boldsymbol{h}) - (\mathcal{B}_{s,t})^{-1}(\boldsymbol{x})\| \right\} \leq L\|\boldsymbol{h}\|, \qquad \boldsymbol{x}, \, \boldsymbol{h} \in \mathbb{R}^d, \tag{3}$$

with a constant L that is independent of s and t (here and below, $\|\cdot\|$ denotes the Euclidean norm of $\mathbb{R}^d$).

Définition 1.2. The mapping $\mathcal{B}_{s,t}$ is usually referred to as the (backward, when $s \leq t$) *characteristic flow* of equation (1), as we have formally

$$f(t, \boldsymbol{x}) = f\big(s, \mathcal{B}_{s,t}(\boldsymbol{x})\big), \qquad \boldsymbol{x} \in \mathbb{R}^d, \quad s, t \in [0, T]. \tag{4}$$

From now on, we shall work for simplicity in $d = 2$ dimensions, and assume that every solution is supported in $\Omega := (0, 1)^2$. Moreover we shall consider a uniform time discretization involving N time steps and denote $\Delta t := T/N$ and $t_n := n\Delta t$ for $n = 0, \ldots, N$.

1.2. **The backward semi-Lagrangian method**

Based on the pointwise transport property (4), the semi-Lagrangian approach (see e.g. [5] or [13]) consists of combining a transport and a projection operator within every time step, as in

$$f_{n+1} := P\mathcal{T}f_n, \qquad n = 0, \dots, N-1,$$

where $f_n \approx f(t_n)$ denotes the numerical solution. More precisely, the schemes that we consider in this article decompose as follows:

(i) given f_n, approach the exact backward flow $\mathcal{B}_{t_n,t_{n+1}}$ by a computable diffeomorphism $\mathcal{B}[f_n]$,
(ii) define an intermediate solution by (mentally) transporting f_n along this approximate flow:

$$\mathcal{T}f_n := f_n \circ \mathcal{B}[f_n], \tag{5}$$

(iii) obtain f_{n+1} by interpolating the intermediate $\mathcal{T}f_n$ on a given grid Λ^{n+1} of Ω, as sketched in Figure 1.
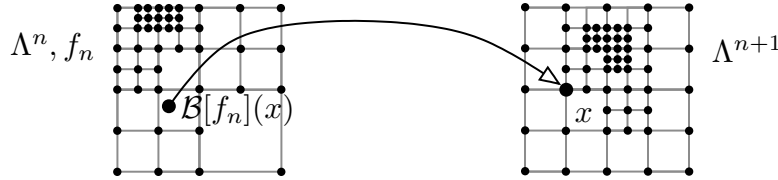

FIGURE 1. Pictorial depiction of the backward semi-Lagrangian method.

Following this principle, the *adaptive* semi-Lagrangian method consists of interpolating $\mathcal{T}f_n$ on a grid of the phase space that is – and remains – well adapted to the numerical solution, in a sense that is to be specified.

In this article we shall see the operator $\mathcal{B}[\cdot] : (f_n, \boldsymbol{x}) \mapsto \mathcal{B}[f_n](\boldsymbol{x}) \in \Omega$ as a black box (in practice, it will be an ode numerical solver) satisfying the following smoothness and stability properties.

Assumption 1.3. The numerical scheme $\mathcal{B}[\cdot]$ maps any Lipschitz function $g \in W^{1,\infty}(\Omega)$ to a mapping $\mathcal{B}[g]: \Omega \mapsto \Omega$ such that:

- for any numerical solution f_n, $n = 0, \cdots, N-1$, the approximate backward flow $\mathcal{B} = \mathcal{B}[f_n]$ is a diffeomorphism, and (3), with $\mathcal{B}_{s,t}$ replaced by $\mathcal{B}$, holds with a constant L independent of n,
- the mapping $\mathcal{B}[\cdot]$ is stable in the sense that there exists a constant independent of Δt such that

$$\left\| \mathcal{B}[g] - \mathcal{B}[\tilde{g}] \right\|_{L^\infty(\Omega)} \lesssim \Delta t \|g - \tilde{g}\|_{L^\infty(\Omega)} \tag{6}$$

  holds for any pair g, $\tilde{g}$ of Lipschitz functions, and
- the approximation is locally r-th order accurate with $r > 1$, in the sense that

$$\left\| \mathcal{B}_{t_n,t_{n+1}} - \mathcal{B}[f(t_n)] \right\|_{L^\infty(\Omega)} \lesssim (\Delta t)^r \tag{7}$$

  holds for all $n = 0, \dots N-1$ with a constant depending on f_0 and $T = N\Delta t$ only.

As for the cost of the overall semi-Lagrangian method, we note that computing one pointwise value of $\mathcal{T}f_n$ amounts in one application of the scheme $\mathcal{B}[\cdot]$, plus one pointwise evaluation of the numerical solution f_n. Therefore the computational cost of f_{n+1} is strongly dependent on the cardinality of the grid where it is defined. The main purpose of this article is to study a low-cost strategy for transporting a wavelet grid along the flow in a such way that both its *sparsity* and its *accuracy* are guaranteed.

The outline is as follows. Following a multilevel approach, I first draw a coarse sketch of the wavelet scheme in Section 2, and postpone the fine details and results to Section 3. Thus, the general features of the wavelet representation is recalled in the next section, for the readers who are not familiar with the subject. Also, the main algorithms that compose the adaptive multilevel scheme are briefly presented, as well as their motivation. Section 3 is then devoted to the precise exposition of these tools. The complete construction of interpolatory wavelets is first given, together with the multilevel tree structure of the associated grids, and some useful approximation properties. The algorithms for dynamically handling the adaptive wavelet grids are then specified in full details, and their main properties, in terms of accuracy and complexity, are rigorously stated. Eventually, an error estimate is given for the resulting "predict and readapt" semi-Lagrangian scheme.

2. THE ADAPTIVE WAVELET SCHEME, SKETCHED

2.1. Structure of the hierarchical wavelet basis

We briefly describe the main features of the wavelet representation, for the reader who is not familiar with such tools. A detailed exposition will be given in Section 3.1 (for comprehensive introductions on wavelets, we shall refer to the excellent books of Daubechies [9] and Cohen [7]).

As wavelet approximations, our numerical solutions f_n, $n = 0, ..., N$, are given as a hierarchy of multilevel coefficients representing finer and finer details. The corresponding wavelet basis relies indeed on a nested hierarchy of uniform grids with increasing resolutions, $\Gamma_{j_0} \subset \Gamma_{j_0+1} \subset \cdots \subset \Gamma_J$, with sets of finer nodes denoted $\nabla_j := \Gamma_j \setminus \Gamma_{j-1}$. Thus, a node $\lambda \in \Gamma_J$ is said of level $|\lambda| := j$ iff $\lambda \in \nabla_j$. The coefficients of f_n consist then of two groups, as follows:

$$f_n = \sum_{\lambda \in \Gamma_{j_0}} c_\lambda(f_n)\varphi_{j_0,\lambda} + \sum_{j=j_0+1}^{J} \sum_{\lambda \in \nabla_j} d_\lambda(f_n)\varphi_\lambda.$$

The first sum is a coarse approximation of f_n. It involves a layer of nodal basis functions $\varphi_{j_0,\lambda}$ that is associated with the coarsest uniform grid Γ_{j_0}. As we are to use interpolatory nodal functions, the corresponding approximation is an interpolation and the coefficients are simply point values, i.e., $c_\lambda(f_n) = f_n(\lambda)$. The second sum consists of several layers of finer and finer details that are recursively added to the coarser approximations. An interesting feature is that the associated basis functions (the wavelets) φ_λ are defined as contracted copies of the coarsest $\varphi_{j_0,\mu}$, $\mu \in \Gamma_{j_0}$. More important to us, the wavelet coefficients can be seen as differences between two interpolations of successive resolutions. Namely, on a node of level $|\lambda| = j$ we have

$$d_\lambda(f_n) = (P_j f_n - P_{j-1} f_n)(\lambda),$$

where P_j, P_{j-1} are local interpolation operators associated with the grids Γ_j, Γ_{j-1}. In the regions where f_n is smooth, it is therefore no surprise that the details $d_\lambda(f_n)$ are small, and the higher the level of λ, the smaller these details. Thus fine coefficients are only needed in the regions where f_n has poor regularity, which naturally leads to considering wavelet grids that are no longer uniform but instead adapted to the numerical solutions. Conversely, it is possible to estimate the smoothness of f_n from the amplitude of its wavelet coefficients, and thus predict where fine details will be needed on the next time step, in order to achieve some prescribed accuracy. These facts, combined with a proper definition of a "well-adapted" wavelet grid, will be the main ingredient of our dynamic adaptive strategy.

2.2. The algorithms for adapting and transporting the wavelet grids

In order to build multilevel wavelet grids that are and remain well-adapted to the solutions, up to any prescribed accuracy, we shall essentially need two algorithms, and a definition.

The definition, first, will help us specify the notion of being well-adapted to a given function (see Definition 3.11 for a detailed statement). It is twofold, as it will meet two requirements: at first, we shall ask that an accuracy of the order of ε is achieved when interpolating a function g on an grid that is ε-adapted to g. That property alone will be guaranteed by what we call a "weakly" adapted grid. The second requirement, fulfilled by "strongly" adapted grids, is that the foregoing property is robust enough to be valid even after a small displacement of both the function and the grid. We shall give a more precise description of this robustness at the end of this section.

The first algorithm $\mathbb{A}_\varepsilon$ is an adaptation algorithm. Its purpose is to build multilevel grids that are ε-adapted to any given (continuous) function. It essentially consists in computing the wavelet coefficients of g on the existing nodes, and refining in the regions where these coefficients are above a certain tolerance. See Algorithm 3.13 for a complete statement.

The second algorithm $\mathbb{T}_\mathcal{B}$ is sometimes referred to as a grid "prediction" algorithm, but we may also see it as a grid "transport" algorithm (as this is what $\mathbb{T}$ stands for). Indeed, it basically transports wavelet grids along any given smooth flow $\mathcal{B}$, in such a way that the resolutions of any two regions connected by the flow are comparable. This is done by recursive refinements, relying on both the tree structure of the wavelet grids, and multilevel adaptive meshes that can be associated to them. More specifically, fine nodes are added in the new grid if their parent (coarser) node λ is mapped by $\mathcal{B}$ in an adaptive cell of the original grid with relative high level (compared to $|\lambda|$). Note that such a strategy essentially requires computing $\mathcal{B}(\lambda)$ and localizing the resulting point in the original grid, which needs to be done anyway in the semi-Lagrangian scheme. Therefore it is not an expensive strategy for transporting adaptive, multilevel grids. See Algorithm 3.17 for a complete statement.

The aimed property of the above tools is that, starting from a wavelet grid Λ that is well-adapted to a function g, the prediction algorithm succeeds in building a new grid $\mathbb{T}_\mathcal{B}(\Lambda)$ that remains well-adapted to the transported $\mathcal{T}_\mathcal{B} g = g \circ \mathcal{B}$ (which is not known, thus the name "prediction"). Thanks to a grading property that is usually satisfied by wavelet grids, and which can be required without significant additional cost, this property can be proven true. In other terms, our "low-cost" grid prediction algorithm is accurate. Moreover it is shown to preserve the order of complexity of the grids, see Theorem 3.18.

2.3. Structure of the adaptive semi-Lagrangian scheme

With these tools at hand, the overall "predict and readapt" semi-Lagrangian scheme takes the following form. For simplicity we omit to mention here the initial step, as well as an additional algorithm $\mathbb{G}$ that guarantees that the wavelet grids are indeed graded. For the complete scheme, see Section 3.4.

- Starting from a numerical solution f_{n-1} that is given by its wavelet coefficients on the multilevel grid Λ^{n-1}, we first predict a new grid

$$\tilde{\Lambda}^n := \mathbb{T}_{\mathcal{B}[f_{n-1}]}(\Lambda^{n-1})$$

  by using the numerical flow $\mathcal{B}[f_{n-1}]$ (provided to us by our ode numerical solver).
- Then we perform a standard semi-Lagrangian step on this grid. That is, we interpolate

$$\tilde{f}_n := P_{\tilde{\Lambda}^n} \mathcal{T} f_{n-1}.$$

  Note that the positions $\{\mathcal{B}[f_{n-1}](\lambda) : \lambda \in \tilde{\Lambda}^n\}$ that are needed for this interpolation have already been computed on the first step.
- Finally we correct the grid $\tilde{\Lambda}^n$ by readapting it, i.e. we set

$$\Lambda^n := \mathbb{A}_\varepsilon(\tilde{f}_n), \quad \text{and compute accordingly} \quad f_n := P_{\Lambda^n} \tilde{f}_{n-1}.$$

What motivates this last step is that, although the prediction algorithm preserves the order of adaptation of the grids, multiplicative constants can appear, that would eventually ruin the accuracy of the scheme if no readaptation were performed once in a while.

3. The adaptive wavelet scheme, in detail

3.1. **The hierarchical (wavelet) basis**

In order to later describe the precise algorithms for dynamically adapting the wavelet grids within the semi-Lagrangian scheme, we now recall in detail the construction of the interpolatory wavelets, and the structure of the associated hierarchical basis. To our knowledge, this construction has first been studied by Deslauriers and Dubuc, see [10].

Thus, we first denote the two-dimensional uniform dyadic grids at every level $j \in \mathbb{N}$ by

$$\Gamma_j := \left\{ (2^{-j}k, 2^{-j}k') : k, k' \in \mathbb{Z} \right\} \ \subset \ \Gamma_{j+1} \ \subset \ \cdots \ \subset \ \mathbb{R}^2, \qquad \text{and let} \qquad \nabla_{j+1} := \Gamma_{j+1} \setminus \Gamma_j$$

be the set of nodes of level $j + 1$, i.e., appearing in the refinement of Γ_j into Γ_{j+1}. In the sequel, the level of a dyadic node γ will be denoted by the short notation $|\gamma|$, and the set of all dyadic nodes will be denoted by

$$\Gamma_\infty := \cup_{j \geq 0} \Gamma_j.$$

Note that if we let $\nabla_0 := \Gamma_0$, the sets ∇_j, $j \geq 0$, form a partition of Γ_∞.

The next ingredients are inter-grid operators P_j^{j+1}, P_{j+1}^j acting on sequences and standing for restriction and reconstruction, respectively: the general idea being that if the sequences $\mathbf{g}^{[j]} := \{g(\gamma) : \gamma \in \Gamma_j\} \in \ell^\infty(\Gamma_j)$, $j \in \mathbb{N}$, correspond to samples of a given $g \in \mathcal{C}(\mathbb{R}^2)$, the restricted sequences always satisfy

$$P_j^{j+1} \mathbf{g}^{[j+1]} = \mathbf{g}^{[j]},$$

whereas the predicted sequences $P_{j+1}^j \mathbf{g}^{[j]}$ generally differ from $\mathbf{g}^{[j+1]}$ on the finer nodes $\gamma \in \nabla_{j+1}$. On the other hand, the discrepancy should be small in the regions where g is smooth.

In order to be more specific we introduce stencils $S_\gamma \subset \Gamma_j$ associated with nodes of level $|\gamma| = j + 1$, as follows (see Figure 2 for an illustration).
- If $\gamma = (2^{-(j+1)}(2k+1), 2^{-j}k')$ corresponds to a refinement of Γ_j in the first dimension, we set

$$S_\gamma := \{(2^{-j}(k+l), 2^{-j}k') : -R+1 \leq l \leq R\}. \tag{8}$$

- Similarly if $\gamma = (2^{-j}k, 2^{-(j+1)}(2k'+1))$ corresponds to a refinement in the second dimension, we set

$$S_\gamma := \{(2^{-j}k, 2^{-j}(k'+l)) : -R+1 \leq l \leq R\}. \tag{9}$$

- Finally if $\gamma = (2^{-(j+1)}(2k+1), 2^{-(j+1)}(2k'+1))$ corresponds to a refinement of Γ_j in both dimensions, we set

$$S_\gamma := \{(2^{-j}(2k+l), 2^{-j}(k'+l')) : -R+1 \leq l, l' \leq R\}. \tag{10}$$

Remarque 3.1. Nodes of the latter type will play a particular role in the design of adaptive grids, and will be called **-nodes* in the sequel, see in particular Section 3.2.

The two inter-grid operators are then defined as follows:

- the restriction $P_j^{j+1} : \ell^\infty(\Gamma_{j+1}) \to \ell^\infty(\Gamma_j)$ is a simple decimation, i.e.,

$$\left(P_j^{j+1}\mathbf{g}^{[j+1]}\right)(\gamma) := \mathbf{g}^{[j+1]}(\gamma) \qquad \forall \gamma \in \Gamma_j,$$

- the prediction $P_{j+1}^j : \ell^\infty(\Gamma_j) \to \ell^\infty(\Gamma_{j+1})$ is a reconstruction given by

$$\left(P_{j+1}^j\mathbf{g}^{[j]}\right)(\gamma) := \begin{cases} \mathbf{g}^{[j]}(\gamma) & \text{if } \gamma \in \Gamma_j \\ \sum_{\mu \in S_\gamma} \pi(\gamma,\mu)\mathbf{g}^{[j]}(\mu) & \text{if } \gamma \in \nabla_{j+1} := \Gamma_{j+1} \setminus \Gamma_j \end{cases}$$

for any $\gamma \in \Gamma_{j+1}$. Here the coefficients $\pi(\gamma,\mu)$, $\mu \in S_\gamma$, are defined in such a way that P_{j+1}^j corresponds to Lagrangian interpolation of maximal coordinate degree $2R-1$. More precisely, for any $\mu \in S_\gamma$ we let $L_{\gamma,\mu}$ denote the unique polynomial of $\mathbb{Q}_{2R-1} := \text{span}\{x^a y^b : a,b \in \{0,1,\ldots,2R-1\}\}$ that satisfies

$$L_{\gamma,\mu}(\lambda) = \delta_{\mu,\lambda} \qquad \forall \lambda \in S_\gamma$$

(where δ stands for the Kronecker symbol), and is constant with respect to x or y in the case where S_γ is given by (8) or (9), respectively. Finally we set

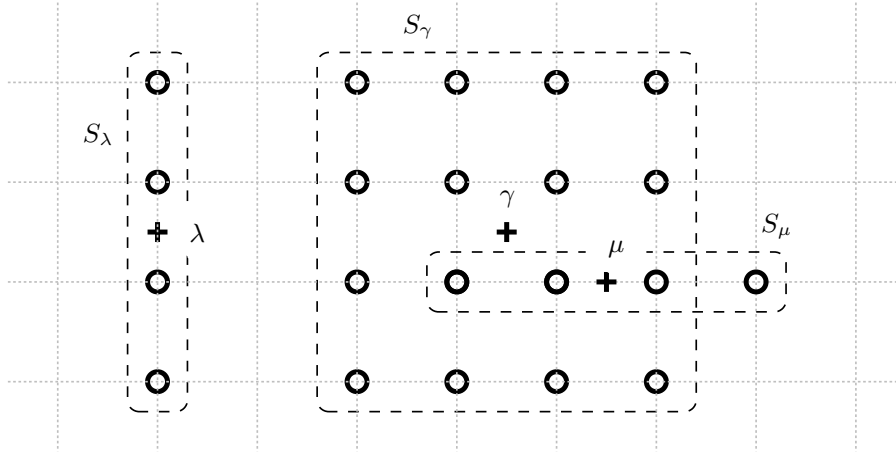$$\pi(\gamma,\mu) := L_{\gamma,\mu}(\gamma).$$

FIGURE 2. The three kinds of stencils corresponding to (8), (9) and (10), for $R = 2$.

Remarque 3.2. By using the shift invariance and self-similarity of the dyadic grids, we can check that the value of $\pi(\gamma,\mu)$ only depends on the relative positions of γ and μ. More precisely, for any R there exists a sequence $(h_{n,n'})_{n,n' \in \mathbb{Z}}$ such that

$$\pi(\gamma,\mu) = h_{k-2i,k'-2i'} \quad \text{for} \quad \gamma = (2^{-(j+1)}k, 2^{-(j+1)}k'), \ \mu = (2^{-j}i, 2^{-j}i').$$

Moreover we note that $h_{n,n'} = 0$ for $\max(|n|,|n'|) \geq 2R$, and it follows that the prediction coefficients $\pi(\gamma,\mu)$ are bounded by a constant that only depends on R.

Now, as previously announced we see that by restricting and further reconstructing samples at a given level one looses information, i.e. $P^j_{j+1}P^{j+1}_j \mathbf{g}^{[j+1]} = P^j_{j+1}\mathbf{g}^{[j]}$ generally differs from $\mathbf{g}^{[j+1]}$ on the finest nodes $\gamma \in \nabla_{j+1}$. The prediction error is then stored in a sequence

$$\mathbf{d}^{[j+1]} := \mathbf{g}^{[j+1]} - P^j_{j+1}\mathbf{g}^{[j]} \equiv \left\{ d_\gamma(g) := \left(\mathbf{g}^{[j+1]} - P^j_{j+1}\mathbf{g}^{[j]}\right)(\gamma) : \gamma \in \nabla_{j+1} \right\},$$

and introducing the (Radon) measures $\tilde{\varphi}_\gamma := \delta_\gamma - \sum_{\mu \in S_\gamma} \pi(\gamma, \mu)\delta_\mu$, every $d_\gamma(g)$ rewrites as

$$d_\gamma(g) = g(\gamma) - \sum_{\mu \in S_\gamma} \pi(\gamma, \mu)g(\mu) = \langle \tilde{\varphi}_\gamma, g \rangle. \tag{11}$$

In the wavelet terminology these coefficients are called *details*, as they are seen as the additional information needed to recover the exact values of g from a coarse sampling. Intuitively, one would expect these coefficients to be small in the regions were g is smooth, and indeed it is easy to write a rigorous estimate: By observing that the prediction is exact for polynomials of order $2R$, i.e., we have $\langle \tilde{\varphi}_\gamma, p \rangle = 0$ for any $p \in \mathbb{Q}_{2R-1}$, we can exploit the bound on the coefficients $\pi(\gamma, \mu)$ (see Remark 3.2) together with the fact that every $\tilde{\varphi}_\gamma$ vanishes outside

$$\Sigma_\gamma := \overline{B}_{\ell^\infty}(\gamma, 2^{-|\gamma|}(2R-1)). \tag{12}$$

Indeed this yields (with constants that only depend on R)

$$|d_\gamma(g)| \le \inf_{p \in \mathbb{Q}_{2R-1}} |\langle \tilde{\varphi}_\gamma, g - p \rangle| \lesssim \inf_{p \in \mathbb{Q}_{2R-1}} \|g - p\|_{L^\infty(\Sigma_\gamma)} \lesssim 2^{-\nu|\gamma|} |g|_{W^{\nu,\infty}(\Sigma_\gamma)} \tag{13}$$

for any integer $\nu \le 2R$, where the third inequality follows from the Deny-Lions theorem, see [6, Th. 14.1].

From the above iterative interpolation scheme, it is possible to define a hierarchical wavelet basis for the full space $\mathcal{C}(\mathbb{R}^2)$ in which the details $d_\gamma(g)$ will play the role of the coefficients of g. In order to make this statement more precise, we introduce for any j and $\gamma \in \Gamma_j$ a sequence (of sequences) $\phi^{[j']}_{j,\gamma} \in \ell^\infty(\Gamma_{j'})$, $j' \ge j$, defined by

$$\phi^{[j]}_{j,\gamma}(\lambda) := \delta_{\gamma,\lambda} \quad \forall \lambda \in \Gamma_j \qquad \text{and} \qquad \phi^{[j'+1]}_{j,\gamma} := P^{j'}_{j'+1}\phi^{[j']}_{j,\gamma} \quad \forall j' \ge j. \tag{14}$$

Note that by definition of the prediction operator, we have

$$\phi^{[j'']}_{j,\gamma}(\lambda) = \phi^{[j']}_{j,\gamma}(\lambda) \qquad \forall \lambda \in \Gamma_{j'} \text{ and } j'' \ge j' \ge j,$$

therefore the above process essentially consists in refining growing sets of values. Now, as is well known (see e.g. [4,7,9]), for any $j \in \mathbb{N}$ and $\gamma \in \Gamma_j$ this process converges towards a continuous function $\varphi_{j,\gamma} : \mathbb{R}^2 \to \mathbb{R}$ in the sense that

$$\phi^{[j']}_{j,\gamma}(\lambda) = \varphi_{j,\gamma}(\lambda) \qquad \forall j' \ge j, \ \lambda \in \Gamma_{j'}, \tag{15}$$

and as a consequence of the shift invariance and the self-similarity of the dyadic grids, every limit function is a shrinked and/or translated copy of a *mother scaling function* φ. Indeed, for all $x \in \mathbb{R}^2$ it holds

$$\varphi_{j,\gamma}(x,y) = \varphi_{j,0}(x - \gamma) = \varphi_{j-1,0}\left(2(x - \gamma)\right) = \cdots = \varphi\left(2^j(x - \gamma)\right), \qquad \varphi \equiv \varphi_{0,0}. \tag{16}$$

The reason why φ is called a "scaling" function comes from the fact that it solves a two-scale relation which plays a central role in the construction of the hierarchical system, see e.g. [7,9,11,12]. One particular consequence of such a relation is that the spaces $V_j := \text{span}\{\varphi_{j,\gamma} : \gamma \in \Gamma_j\}$ spanned by the scaling functions are nested, i.e., we have $V_j \subset V_{j+1} \subset \cdots$. Moreover, by using the polynomial reproduction of the iterative scheme one easily

shows that their union is dense in the space $\mathcal{C}(\mathbb{R}^2)$. In order to build a hierarchical basis of the latter, it is then possible to eliminate the redundancy by keeping only the functions of the type

$$\varphi_\gamma := \varphi_{|\gamma|,\gamma}, \qquad \gamma \in \Gamma_\infty = \cup_{j \geq 0} \Gamma_j.$$

Indeed, for every $g \in \mathcal{C}(\mathbb{R}^2)$ we have (the limit holding in a pointwise sense)

$$g = \sum_{\gamma \in \Gamma_{j_0}} g(\gamma)\varphi_{j_0,\gamma} + \sum_{j \geq j_0+1} \sum_{\gamma \in \nabla_j} d_\gamma(g)\varphi_\gamma, \tag{17}$$

for any choice of the coarse level $j_0 \in \mathbb{N}$ (in the sequel we shall sometimes adopt the convenient notations $\varphi_\gamma = \varphi_{j_0,\gamma}$ and $d_\gamma(g) = g(\gamma)$, for the coarse nodes $\gamma \in \Gamma_{j_0}$).

In order to further study our adaptive scheme, let us now review some additional properties of the hierarchical decomposition (17): First, it is easily seen that every layer of scaling functions $\varphi_{j,\gamma}$, $\gamma \in \Gamma_j$, are interpolatory in the sense that

$$\varphi_{j,\gamma}(\lambda) = \delta_{\gamma,\lambda}, \qquad \forall \gamma, \lambda \in \Gamma_j. \tag{18}$$

In particular they form a nodal basis for the space V_j, i.e., every $g \in V_j$ reads $g = \sum_{\gamma \in \Gamma_j} g(\gamma)\varphi_{j,\gamma}$. Second, we can infer from the locality of the prediction stencils that the scaling functions are supported in balls of radius $\sim 2^{-j}$. More precisely, we have $\operatorname{supp}(\varphi_{j,\gamma}) \subset \overline{B}_{\ell^\infty}(\gamma, 2^{-j}(2R-1))$ for all $\gamma \in \Gamma_j$, which in turn implies a bounded overlapping property:

$$\#\big(\{\lambda \in \Gamma_j : \operatorname{supp}(\varphi_{j,\gamma}) \cap \operatorname{supp}(\varphi_{j,\lambda}) \neq \emptyset\}\big) \leq C_s \qquad \forall \gamma \in \Gamma_j \tag{19}$$

with a constant $C_s = C_s(R)$ independent of j. One important consequence of this fact is that the basis $\{\varphi_{j,\gamma} : \gamma \in \Gamma_j\}$ is *stable* in the sense that

$$c\|g\|_{L^\infty(\mathbb{R}^2)} \leq \sup_{\gamma \in \Gamma_j} |g(\gamma)| \leq \|g\|_{L^\infty(\mathbb{R}^2)} \qquad \forall g = \sum_{\gamma \in \Gamma_j} g(\gamma)\varphi_{j,\gamma} \in V_j \tag{20}$$

holds with $c = (C_s\|\varphi\|_{L^\infty(\mathbb{R}^2)})^{-1}$, see (16).

3.2. Wavelet interpolations, admissible trees and well-adapted meshes

We now have a representation of any continuous function g in terms of the multilevel nodal functions φ_γ, $\gamma \in \Gamma_\infty$, with small coefficients in the regions where g is smooth, see (13). Adaptivity will be achieved by discarding small coefficients in this expansion. In order to study such approximation schemes, we introduce the following definition.

Définition 3.3. Let $j_0 \in \mathbb{N}$. A grid $\Lambda \subset \Gamma_\infty$ is said to be *admissible* if it contains the coarse grid Γ_{j_0} and if it satisfies

$$\gamma \in \Lambda \implies S_\gamma \subset \Lambda.$$

Next we define a mapping $P_\Lambda : \mathcal{C}(\mathbb{R}^2) \to V_\Lambda := \operatorname{span}\{\varphi_\lambda : \lambda \in \Lambda\}$ by

$$P_\Lambda g := \sum_{\lambda \in \Lambda} d_\lambda(g)\varphi_\lambda. \tag{21}$$

Clearly this makes sense for any grid $\Lambda \subset \Gamma_\infty$, but we note that if Λ is admissible, then P_Λ is an interpolation.

Lemme 3.4. *If the grid Λ is admissible then $P_\Lambda g(\gamma) = g(\gamma)$ for all $\gamma \in \Lambda$.*

Proof. First, since (21) is a wavelet expansion of $P_\Lambda g$, we clearly have $d_\gamma(P_\Lambda g) = d_\gamma(g) \quad \forall \gamma \in \Lambda$. Observe next that any φ_μ, $|\mu| \geq j_0 + 1$, vanishes on any $\gamma \in \Gamma_{j_0}$, and calculate

$$P_\Lambda g(\gamma) = \sum_{|\lambda| \leq j_0} d_\lambda(g)\varphi_\lambda(\gamma) = \sum_{|\lambda| \geq 0} d_\lambda(g)\varphi_\lambda(\gamma) = g(\gamma), \quad \gamma \in \Gamma_{j_0}.$$

Now assume that $P_\Lambda g$ and g coincide on $\Lambda \cap \Gamma_{j-1}$, and consider $\gamma \in \Lambda \cap \nabla_j$: since Λ is admissible, we have by definition of the details d_γ

$$P_\Lambda g(\gamma) = \sum_{\mu \in S_\gamma} \pi(\gamma, \mu) P_\Lambda g(\mu) + d_\gamma(P_\Lambda g) = \sum_{\mu \in S_\gamma} \pi(\gamma, \mu) g(\mu) + d_\gamma(g) = g(\gamma).$$

$\square$

As we said before, the error resulting from "discarding the small details" should be controlled by the amplitude of these details. Here is the precise statement that we shall use for an approximation in the supremum norm.

Lemme 3.5. *The approximation of any $g \in \mathcal{C}(\mathbb{R}^2)$, associated with the wavelet grid Λ, is bounded by*

$$\|g - P_\Lambda g\|_{L^\infty(\mathbb{R}^2)} \leq C \sum_{j \geq 0} \sup_{\gamma \in \nabla_j \setminus \Lambda} |d_\gamma(g)|$$

with $C = C_s \|\varphi\|_{L^\infty(\mathbb{R}^2)}$,

Proof. In views of (17), g can be written as an infinite wavelet expansion. The approximation error thus satisfies

$$\|g - P_\Lambda g\|_{L^\infty(\mathbb{R}^2)} = \Big\| \sum_{j \geq 0} \sum_{\gamma \in \nabla_j \setminus \Lambda} d_\gamma(g)\varphi_\gamma \Big\|_{L^\infty(\mathbb{R}^2)} \leq \sum_{j \geq 0} \Big\| \sum_{\gamma \in \nabla_j \setminus \Lambda} d_\gamma(g)\varphi_\gamma \Big\|_{L^\infty(\mathbb{R}^2)} \leq C \sum_{j \geq 0} \sup_{\gamma \in \nabla_j \setminus \Lambda} |d_\gamma(g)|,$$

with $C = C_s \|\varphi\|_{L^\infty(\mathbb{R}^2)}$, by using (20) in the last inequality. $\square$

Remarque 3.6. The foregoing estimate is sharp. Indeed, consider the one-dimensional case (for the sake of simplicity) where for $R = 1$, the reference scaling function is given by $\varphi(x) = \max(1 - |x|, 0)$. Now let $\gamma_i := 2^{-2i}(1 + 4 + \cdots + 4^{(i-1)}) \in \nabla_{2i}$, and check that

$$\varphi_{\gamma_i} = \varphi(2^{2i}(x - \gamma_i)) \geq \frac{1}{2} \quad \text{on} \quad \Big[\frac{4^i - 1}{3 \cdot 4^i}, \frac{4^i - 1}{3 \cdot 4^i} + \frac{1}{2 \cdot 4^i}\Big].$$

In particular, $\varphi_{\gamma_i}(1/3) \geq 1/2$ for all i, hence $\|\sum_{i \leq J} \varphi_{\gamma_i}\|_{L^\infty(\mathbb{R}^2)} \geq J/2$ for all J.

In order to transport the numerical solutions along the flow, it will be convenient to associate every (admissible) adaptive grid with a partition of the phase space. Moreover, our scheme will be based on tree algorithms, hence we shall equip the dyadic grids with a tree structure. In order to do so, we first introduce the set ∇_j^* of *-nodes of level $j \geq 1$ that correspond to a refinement of Γ_{j-1} in both directions,

$$\nabla_j^* := \big\{(2^{-j}(2k + 1), 2^{-j}(2k' + 1)) : k, k' \in \{0, 2^{j-1} - 1\}\big\} \subset \nabla_j,$$

and associate an (open) square cell to every *-node by setting

$$\Omega_\gamma := B_{\ell^\infty}(\gamma, 2^{-|\gamma|}), \qquad \gamma \in \Gamma_\infty^* := \cup_{j \geq 1} \nabla_j^*. \tag{22}$$

Next, we equip the set of dyadic nodes with a tree structure by defining for every $\gamma \in \nabla_j$ one set of *children* in ∇_{j+1} as follows: If γ is a *-node of level j, we define its children as

$$\mathcal{C}(\gamma) := \{\gamma + 2^{-(j+1)}(l, l') : (l, l') \in \{-1, 0, 1\} \times \{-1, 0, 1\} \setminus (0, 0)\} \subset \nabla_{j+1};$$

if $\gamma = (2^{-j}k, 2^{-(j-1)}k')$ (hence with k odd), we define its children as

$$\mathcal{C}(\gamma) := \{\gamma + 2^{-(j+1)}(l,0) : l \in \{-1,1\}\} \subset \nabla_{j+1};$$

and if $\gamma = (2^{-(j-1)}k, 2^{-j}k')$ (hence with k' odd), we define its children as

$$\mathcal{C}(\gamma) := \{\gamma + 2^{-(j+1)}(0,l') : l' \in \{-1,1\}\} \subset \nabla_{j+1}.$$

Finally, we say that λ is a *parent* of γ if $\gamma \in \mathcal{C}(\lambda)$. Note that this process partitions the levels in the sense that every dyadic node γ of positive level has one (and only one) parent $\mathcal{P}(\gamma)$, moreover $|\mathcal{P}(\gamma)| = |\gamma| - 1$. Now, as it can be checked, every parent of a *-node is also a *-node but the converse is not true, i.e., not every children of a *-node is a *-node itself. Hence we can define *-children* by setting

$$\mathcal{C}^*(\gamma) := \{\gamma + 2^{-(j+1)}(m,m') : (m,m') \in \{-1,1\}^2\} = \mathcal{C}(\gamma) \cap \nabla_{j+1}^*, \qquad \gamma \in \nabla_j^*.$$

Let us adopt the following definition for *wavelet-trees*.

Définition 3.7 (W-trees)**.** A grid $\Lambda \subset \Gamma_\infty$ is said to be a *W-tree* if it contains the coarse grid Γ_{j_0} and if it satisfies

$$\gamma \in \Lambda \quad \Longrightarrow \quad \mathcal{P}(\gamma) \subset \Lambda.$$

Clearly, the simplest way to build a tree Λ consists in starting from the coarsest grid Γ_{j_0} and adding recursively children, according to some criterion. Observe that by doing so, one also builds a non-uniform partition of $\mathbb{R}^2$, given by

$$M(\Lambda) := \{\Omega_\gamma : \gamma \in \mathcal{L}_{\text{out}}(\Lambda) \cap \Gamma_\infty^*\}, \qquad \text{where} \qquad \mathcal{L}_{\text{out}}(\Lambda) := \{\gamma \notin \Lambda : \mathcal{P}(\lambda) \in \Lambda\} \qquad (23)$$

denotes the set of *outer leaves* of the tree Λ. See Remark 3.10 below. For accuracy purposes, we will need that the trees satisfy an additional grading property.

Définition 3.8 (graded W-trees)**.** A W-tree Λ is said to be *graded* if

$$\gamma \in \Lambda \cap \Gamma_\infty^* \quad \Longrightarrow \quad \{\mu \in \Gamma_{|\gamma|-1} : \Sigma_\mu \cap \Omega_\gamma \neq \emptyset\} \subset \Lambda,$$

where $\Sigma_\mu := \overline{B}_{\ell^\infty}(\mu, 2^{-|\mu|}(2R-1))$ and $\Omega_\gamma := B_{\ell^\infty}(\gamma, 2^{-|\gamma|})$, see (12) and (22).

As we shall see, this definition is mostly motivated by the accuracy of the transported meshes. But it turns out that the use of graded trees is already imposed by the admissibility of the dyadic wavelet grids.

Lemme 3.9 (see [2])**.** *Every admissible wavelet grid is a graded W-tree.*

Remarque 3.10 (associating W-trees and meshes)**.** We have just seen that with any W-tree Λ, it is possible to associate an adaptive mesh $M(\Lambda)$ that essentially consists of its outer *-leaves. Conversely, one can associate several W-trees Λ with any such mesh M, that satisfy $M(\Lambda) = M$: The smallest such one consists of all the ancestors (i.e., parents, parents of parents, etc., of level $j \geq j_0$) of the nodes in M. Since this tree contains only *-nodes, let us call it $\Lambda^*(M)$. In order to ensure the accuracy of the interpolations in the semi-Lagrangian scheme, we will prefer to work with admissible trees. Therefore associate with M an admissible tree that *strictly* contains $\Lambda^*(M)$. In practice this will be done by applying a grading algorithm to $\Lambda^*(M)$. See Algorithm 3.14 and Lemma 3.15 below.

In order to accurately transport the grids with a low-cost algorithm, we now introduce a crucial property which involve the *neighbors* of a *-node, i.e.,

$$\mathcal{N}(\gamma) := \{\mu \in \Gamma_\infty : \Sigma_\mu \cap \Omega_\gamma \neq \emptyset\}, \qquad \gamma \in \Gamma_\infty^*.$$

For the remainder, let us fix one positive constant $\kappa < 1$, and remind that $\varepsilon > 0$ is an arbitrary tolerance.

Définition 3.11 (well-adapted W-trees)**.** We shall say that the W-tree Λ is *weakly ε-adapted* to $g \in \mathcal{C}(\mathbb{R}^2)$ if

$$\sup_{\gamma \in M(\Lambda)} \sup_{\mu \in \mathcal{N}(\gamma) \setminus \Lambda} 2^{\kappa(|\mu| - |\gamma|)} |d_\mu(g)| \leq \varepsilon \tag{24}$$

see (23). We say that Λ is *strongly ε-adapted* to g if the second supremum can be taken over all $\mu \in \mathcal{N}(\gamma)$.

We note that the foregoing criteria (24) is somehow related to the prediction strategy suggested by Cohen, Kaber, Müller and Postel [8] in the context of wavelet-based finite volume schemes with guaranteed error estimates. A first property associated with these definitions is the following.

Lemme 3.12. *If Λ is graded (or admissible) and weakly ε-adapted to g, then the associated interpolation error satisfies*

$$\|g - P_\Lambda g\|_{L^\infty(\mathbb{R}^2)} \lesssim \varepsilon, \qquad g \in \mathcal{C}(\mathbb{R}^2)$$

with a constant independent of g.

Proof. Let us begin with the following observation: If Λ is graded, then for any $\gamma \in \mathcal{L}_{\text{out}}(\Lambda) \cap \Gamma_\infty^*$ and $\mu \in \Gamma_{|\gamma|-2} = \Gamma_{|\mathcal{P}(\gamma)|-1}$ with $\mu \notin \Lambda$, we have $\Sigma_\mu \cap \Omega_\gamma \subset \Sigma_\mu \cap \Omega_{\mathcal{P}(\gamma)} = \emptyset$ since $\mathcal{P}(\gamma) \in \Lambda$. In other terms, we see that

$$|\mu| \geq |\gamma| - 1, \qquad \forall \mu \in \mathcal{N}(\gamma) \setminus \Lambda. \tag{25}$$

Now using that $\text{supp}(\varphi_\mu) \subset \Sigma_\mu$, we calculate

$$\|g - P_\Lambda g\|_{L^\infty(\Omega_\gamma)} = \Big\| \sum_{j \geq j_0+1} \sum_{\substack{\mu \in \nabla_j \setminus \Lambda \\ \Sigma_\mu \cap \Omega_\gamma \neq \emptyset}} d_\mu(g) \varphi_\mu \Big\|_{L^\infty(\Omega_\gamma)} \lesssim \sum_{j \geq j_0+1} \sup_{\substack{\mu \in \nabla_j \setminus \Lambda \\ \Sigma_\mu \cap \Omega_\gamma \neq \emptyset}} |d_\mu(g)| \lesssim \sum_{j \geq |\gamma|-1} \sup_{\substack{\mu \in \nabla_j \setminus \Lambda \\ \Sigma_\mu \cap \Omega_\gamma \neq \emptyset}} |d_\mu(g)|$$

where the first inequality follows from Lemma 3.5 and the second one from the above observation (which uses the gradedness of Λ). By using the ε-adaptivity, we then have $\|g - P_\Lambda g\|_{L^\infty(\Omega_\gamma)} \lesssim \sum_{j \geq |\gamma|-1} 2^{\kappa(|\gamma|-j)} \varepsilon \lesssim \varepsilon$. The fact that $M(\Lambda) = \{\Omega_\gamma : \gamma \in \mathcal{L}_{\text{out}}(\Lambda) \cap \Gamma_\infty^*\}$ is a partition of $\mathbb{R}^2$ completes the proof. $\qquad\square$

3.3. Dynamic adaptivity

We now describe the algorithms that construct an ε-adapted wavelet tree to any given function g, and that transport it along a smooth flow, while conserving the property of being adapted to the transported solution.

The following algorithm builds a W-tree (only composed of *-nodes) that is strongly ε-adapted to $g \in C(\mathbb{R}^2)$.

Algorithme 3.13 ($\mathbb{A}_\varepsilon(g)$: strong ε-adaption of W-trees)**.** *Starting from $\Lambda_0^* := \Gamma_{j_0}^*$, set*

$$\Lambda_{\ell+1}^* := \Lambda_\ell^* \cup \big\{ \gamma \in M(\Lambda_\ell^*) : \max\{2^{\kappa|\mu|} |d_\mu(g)| : \mu \in \mathcal{N}(\gamma)\} > 2^{\kappa|\gamma|} \varepsilon \big\}$$

for $\ell = 0, 1, \dots$ until $\Lambda_{L+1}^* = \Lambda_L^*$, and finally set $\mathbb{A}_\varepsilon(g) := \Lambda_L^*$.

As the resulting trees are only made of *-nodes, they are unlikely to be graded. Therefore we need an additional algorithm for building admissible (hence graded) refinements.

Algorithme 3.14 ($\mathbb{G}(\Lambda)$: graded refinement of W-trees)**.** *Given any tree Λ, set*

$$\mathbb{G}(\Lambda) := \bigcup_{\gamma \in \Lambda} \big\{ \gamma + 2^{-|\gamma|}(m_1, m_2) : m_1, m_2 \in \{-(2R-1), \dots, (2R-1)\} \big\}.$$

The foregoing process indeed builds graded W-trees.

Lemme 3.15 (see [2])**.** *For any W-tree Λ, $\mathbb{G}(\Lambda)$ is an admissible (hence graded) W-tree.*

Remarque 3.16. We could also give a variant of Algorithm 3.13 that builds *weakly ε-adapted* W-trees to some given g. According to Lemma 3.12, we know that interpolating g on the resulting grid (once graded) would be $C\varepsilon$ accurate, but this is not enough to ensure the accuracy of the adaptive semi-Lagrangian scheme (indeed, think of the case where g consists of one isolated basis function φ_γ).

Let us now turn to the central issue, which is the grid prediction. As previously said the algorithm is based on recursive splittings of adaptive meshes, and essentially amounts in transporting the local space resolution along the (approximate) flow $\mathcal{B}$. In order to establish the accuracy of the predicted grids, we now introduce a fixed parameter $\delta \in \mathbb{N}$ (the value of which can be chosen in the proof of Theorem 3.18 below), that corresponds to a number of additional refinement levels.

Algorithme 3.17 ($\mathbb{T}_\mathcal{B}(\Lambda)$: prediction of W-trees). *Starting from $\Lambda_0^* := \Gamma_{j_0}^*$, build*

$$\Lambda_{\ell+1}^* := \Lambda_\ell^* \cup \left\{ \gamma \in \mathcal{L}_{\text{out}}(\Lambda_\ell^*) \cap \Gamma_\infty^* : \min\{|\lambda| : \lambda \in \mathcal{L}_{\text{out}}(\Lambda) \cap \Gamma_\infty^*, \mathcal{B}(\gamma) \in \overline{\Omega}_\lambda\} > |\gamma| - \delta \right\}$$

until $\Lambda_{L+1}^* = \Lambda_L^*$, and set $\mathbb{T}_\mathcal{B}(\Lambda) := \Lambda_L^*$.

In the following theorem we state two important properties of this strategy.

Théorème 3.18 (see [2]). *Let $\mathcal{B}$ be a diffeomorphism of Ω into itself, i.e., an invertible Lipschitz-continuous mapping with Lipschitz-continuous inverse. Then Algorithm 3.17 guarantees the accuracy of the predicted W-trees in the sense that*

> *if Λ is strongly ε-adapted to g, then $\mathbb{T}_\mathcal{B}(\Lambda)$ is weakly $C\varepsilon$-adapted to $\mathcal{T}_\mathcal{B}g$.*

with a constant C that only depends on $\mathcal{B}$. Moreover the cardinalities of the predicted grids are stable; i.e.,

$$\#\big(\mathbb{T}_\mathcal{B}(\Lambda)\big) \lesssim \#(\Lambda). \tag{26}$$

Remarque 3.19. A similar result has been established in [3], in the context of adaptive meshes.

3.4. The "predict and readapt" semi-Lagrangian scheme

Let us summarize what we have seen so far: In Section 1.2 we have recalled the principle of the (adaptive) semi-Lagrangian method, assuming for simplicity the knowledge of a black box numerical scheme for computing a numerical flow $\mathcal{B}[f_{n-1}]$ on each time step, from the numerical solution $f_{n-1} \approx f(t_{n-1})$. Then in Sections 3.1 and 3.2 we have described a tree-structured discretization of wavelet type that is suitable for adaptive interpolations. Finally, we have introduced in Section 3.3 algorithms for (i) building admissible W-trees which are ε-adapted to a given function g, and (ii) given a computable flow $\mathcal{B}$, predicting W-trees that stay well-adapted to the transported $\mathcal{T}_\mathcal{B}g := g \circ \mathcal{B}$.

Note that in views of Theorem 3.18, the quality of being well-adapted to the transported solution is only preserved by the predicted trees up to a multiplicative constant C that might be larger than 1. Hence it is necessary to readapt the trees once in a while in order to guarantee that *all* the interpolation errors stay within a bound of the order ε.

The resulting semi-Lagrangian scheme is as follows: given an initial datum f^0, compute first

$$f_0 := P_{\Lambda^0} f^0, \tag{27}$$

where P_{Λ^0} denotes the adaptive wavelet interpolation associated with

$$\Lambda^0 := \mathbb{G}(\mathbb{A}_\varepsilon(f^0)). \tag{28}$$

Then for $n = 1, \ldots, N = T/\Delta t$, compute (with $\mathcal{T} f_{n-1} := f_{n-1} \circ \mathcal{B}[f_{n-1}]$)

$$f_n := P_{\Lambda^n} P_{\tilde{\Lambda}^n} \mathcal{T} f_{n-1}, \tag{29}$$

where $\tilde{\Lambda}^n$ and Λ^n respectively denote the predicted and readapted W-trees, defined by

$$\tilde{\Lambda}^n := \mathbb{G}\big(\mathbb{T}_{\mathcal{B}[f_{n-1}]}(\Lambda^{n-1})\big) \qquad \text{and} \qquad \Lambda^n := \mathbb{G}\big(\mathbb{A}_\varepsilon(\tilde{f}_n)\big) \quad \text{with} \quad \tilde{f}_n := P_{\tilde{\Lambda}^n} \mathcal{T} f_{n-1}. \tag{30}$$

3.5. **A guaranteed error estimate**

Our main result is that the above adaptive semi-Lagrangian schemes satisfies an a priori error estimate.

Théorème 3.20. *Under Assumptions 1.1 and 1.3, the numerical solution given by the adaptive semi-Lagrangian scheme* (27)-(30) *satisfies*

$$\|f(t_n) - f_n\|_{L^\infty(\Omega)} \lesssim (\Delta t)^{r-1} + \varepsilon/\Delta t \tag{31}$$

for $n = 0, \ldots, N = T/\Delta t$, as long as the initial datum f^0 is in $W^{1,\infty}(\Omega)$.

Proof. We decompose the error $e_{n+1} := \|f(t_{n+1}) - f_{n+1}\|_{L^\infty(\Omega)}$ into three parts as follows. A first term $e_{n+1,1} := \|f(t_{n+1}) - \mathcal{T} f(t_n)\|_{L^\infty(\Omega)}$ which corresponds to the approximation of the characteristics by the numerical transport operator $\mathcal{T}$, a second term $e_{n+1,2} := \|(I - P_{\Lambda^{n+1}} P_{\tilde{\Lambda}^{n+1}}) \mathcal{T} f_n\|_{L^\infty(\Omega)}$ which corresponds to the interpolation error, and a third term $e_{n+1,3} := \|\mathcal{T} f(t_n) - \mathcal{T} f_n\|_{L^\infty(\Omega)}$ which represents the (nonlinear) transport of the numerical error at time step n.

In order to estimate the first term we can rely on the accuracy (7). Indeed, since the approximate and exact transport operators are characterized by the corresponding flows, according to $\mathcal{T} f(t_n) = f(t_n) \circ \mathcal{B}[f(t_n)]$ and $f(t_{n+1}) = f(t_n) \circ \mathcal{B}_{t_n, t_{n+1}}$, we have

$$e_{n+1,1} \leq |f(t_n)|_{W^{1,\infty}(\Omega)} \sup_{(x,v) \in \Omega} \big\|\mathcal{B}_{t_n, t_{n+1}}(x,v) - \mathcal{B}[f(t_n)](x,v)\big\| \lesssim (\Delta t)^r$$

as long as $f^0 \in W^{1,\infty}(\Omega)$.

As for the second term, we note that it involves the numerical solution instead of the exact one. This is in order to exploit the main properties of the predicted grids, i.e., the (weak) $C\varepsilon$-adaptivity to $\mathcal{T} f_n$. Indeed we can write

$$e_{n+1,2} \lesssim \|(I - P_{\tilde{\Lambda}^{n+1}}) \mathcal{T} f_n\|_{L^\infty(\Omega)} + \|(I - P_{\Lambda^{n+1}}) P_{\tilde{\Lambda}^{n+1}} \mathcal{T} f_n\|_{L^\infty(\Omega)} \lesssim \varepsilon,$$

by using the fact that according to Theorem 3.18 and by construction, respectively, the trees $\tilde{\Lambda}^{n+1}$ and Λ^{n+1} are weakly $C\varepsilon$ and strongly ε-adapted to $\mathcal{T} f_n$ and $P_{\tilde{\Lambda}^{n+1}} \mathcal{T} f_n$, respectively. The above inequality follows then from Lemma 3.5.

An interesting by-product of this error decomposition is that it allows the third term, which propagates the error from the previous time steps, *not* to involve the interpolation operator. Hence for any interpolation order, we have

$$e_{n+1,3} = \big\|f(t_n) \circ \mathcal{B}[f(t_n)] - f(t_n) \circ \mathcal{B}[f_n]\big\|_{L^\infty(\Omega)} + \big\|(f(t_n) - f_n) \circ \mathcal{B}[f_n]\big\|_{L^\infty(\Omega)}$$
$$\leq |f(t_n)|_{W^{1,\infty}(\Omega)} \big\|\mathcal{B}[f(t_n)] - \mathcal{B}[f(t_n)]\big\|_{L^\infty(\Omega)} + e_n \leq (1 + C\Delta t) e_n$$

by only using the stability (6) of the mapping $\mathcal{B}[\cdot]$.

The error estimate follows then by gathering the above bounds, and by applying a discrete Gronwall lemma.

$\square$

Remarque 3.21. In contrast to what happens with uniform schemes, an a priori L^∞ bound is available for high-order adaptive semi-Lagrangian schemes where the interpolations may, in general, increase the supremum

norm. Indeed, by denoting $P_n := P_{\Lambda^n} P_{\tilde{\Lambda}^n}$ we can always bound $\bar{e}_n := \|\mathcal{T}^n f_0 - f_n\|_{L^\infty(\Omega)}$ by

$$\bar{e}_n \le \|\mathcal{T}^n f_0 - \mathcal{T} f_{n-1}\|_{L^\infty(\Omega)} + \|(I - P_n)\mathcal{T} f_{n-1}\|_{L^\infty(\Omega)} \le (1 + C\Delta t)\bar{e}_{n-1} + C'\varepsilon$$

by using arguments from the above proof (and in particular, with $C = 0$ in the case of a linear transport). This yields, again by employing a discrete Gronwall lemma,

$$\|f_n\|_{L^\infty(\Omega)} \le \|\mathcal{T}^n f_0\|_{L^\infty(\Omega)} + \bar{e}_n \le \|f_0\|_{L^\infty(\Omega)} + C\varepsilon/\Delta t$$

with a constant independent of ε and Δt.

References

[1] N. Besse, F. Filbet, M. Gutnic, I. Paun, and E. Sonnendrücker. An adaptive numerical method for the Vlasov equation based on a multiresolution analysis. In F. Brezzi, A. Buffa, S. Escorsaro, and A. Murli, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2001*, pages 437–446. Springer, 2001.

[2] M. Campos Pinto. Adaptive semi-Lagrangian schemes for Vlasov equations. In E. Emmrich and P. Wittbold, editors, *Analytical and Numerical Aspects of Partial Differential Equations*. de Gruyter, Berlin, to appear.

[3] M. Campos Pinto and M. Mehrenberger. Convergence of an adaptive semi-Lagrangian scheme for the Vlasov-Poisson system. *Numer. Math.*, 108(3):407–444, 2008.

[4] A.S. Cavaretta, W. Dahmen, and C.A. Micchelli. Stationary subdivision. *Mem. Amer. Math. Soc.*, 93(453), 1991.

[5] C.Z. Cheng and G. Knorr. The integration of the Vlasov equation in configuration space. *J. Comput. Phys.*, 22:330–351, 1976.

[6] P.G. Ciarlet. Basic error estimates for elliptic problems. In *Handbook of numerical analysis, Vol. II*, pages 17–351. North-Holland, Amsterdam, 1991.

[7] A. Cohen. *Numerical analysis of wavelet methods*, volume 32. North-Holland, Amsterdam, 2003.

[8] A. Cohen, S.M. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comp.*, 72(241):183–225, 2003.

[9] I. Daubechies. *Ten lectures on wavelets*, volume 61 of CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1992.

[10] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constr. Approx.*, 5(1):49–68, 1989. Fractal approximation.

[11] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

[12] Y. Meyer. *Ondelettes et opérateurs*. Hermann, Paris, 1990.

[13] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo. The semi-Lagrangian method for the numerical resolution of the Vlasov equation. *J. Comput. Phys.*, 149(2):201–220, 1999.