MULTISCALE MODEL. SIMUL.
Vol. 7, No. 3, pp. 1101–1129

DIRECT NUMERICAL SIMULATION OF TURBULENCE USING DIVERGENCE-FREE WAVELETS*

ERWAN DERIAZ† AND VALÉRIE PERRIER‡

Abstract. We present a numerical method based on divergence-free wavelets to solve the incompressible Navier–Stokes equations. We introduce a new scheme which uses anisotropic (or generalized) divergence-free wavelets and which needs only fast wavelet transform algorithms. We prove its stability and show convincing numerical experiments.

Key words. Navier–Stokes, wavelets, divergence-free, incompressible fluid, stability, numerical scheme

AMS subject classifications. Primary, 65M05; Secondary, 65M12

DOI. 10.1137/070701017

Introduction. The numerical simulation of turbulent flows has many applications in various engineering and environmental problems. Direct numerical simulation (DNS) of turbulence requires the integration in time of the full nonlinear Navier–Stokes equations. However, at high Reynolds number, turbulent flows create a wide range of scales, which induces a huge degree of complexity. Hence, in order to accurately compute all the scales of a turbulent flow, the discretizations in space and in time must be of very small size, leading to a huge number of degrees of freedom, impossible to handle in the DNS of industrial problems.

DNS of homogeneous turbulent flows has been performed extensively to increase the understanding of small-scale structure mechanisms. Among the computational methods commonly used for these simulations, one can cite spectral methods, well localized in frequency [5], or finite element methods, well localized in physical space [21]. In between these two approaches, wavelet bases offer alternative decompositions, more suitable to represent the intermittent spatial structure of the flows.

The wavelet decomposition was first introduced in fluid mechanics to analyze turbulent flows [32, 17, 29, 23]. The first wavelet-based schemes for the computation of homogeneous turbulent flows looked promising [6, 20, 26, 22], especially concerning adaptivity issues. These wavelet schemes are based on Galerkin, Petrov–Galerkin, and collocation methods but also on wavelet/vaguelette decompositions. Wavelets may also be used for turbulence modeling [19]. Most of the cited works use wavelets as a decomposition basis of the *vorticity* field, with periodic boundary conditions, which makes the generalization to nonperiodic boundary conditions very difficult.

Another serious difficulty in the numerical simulation of the Navier–Stokes equations lies in the determination of the pressure field and in the fulfillment of the incompressibility condition. In the primitive variable $(\mathbf{u}, p)$-formulation of the Navier–Stokes equations, physical boundary conditions (on the *velocity*) can be easily incorporated,

*Received by the editors August 23, 2007; accepted for publication (in revised form) August 18, 2008; published electronically December 17, 2008. Part of this work has been supported by European Union project TODEQ (Operator Theory Methods for Differential Equations) contract MTKD-CT-2005-030042.

http://www.siam.org/journals/mms/7-3/70101.html

†Institute of Mathematics, Polish Academy of Sciences. ul. Śniadeckich 8, 00-956 Warszawa, Poland (E.Deriaz@impan.gov.pl).

‡Laboratoire Jean Kuntzmann, Grenoble Institute of Technology, BP 53, 38 041 Grenoble cedex 9, France (Valerie.Perrier@imag.fr).

but, from the numerical point of view, the computation of the velocity and the pressure presents some difficulties: in the setting of classical discretizations, like spectral methods (in the nonperiodic case) [3], finite element methods [21], or wavelets [10, 18, 4], the discretization spaces for the velocity and for the pressure have to fulfill an inf-sup condition (also called an LBB condition); otherwise spurious modes appear in the computation of the pressure which break the incompressibility condition. Moreover, numerical difficulties arise in the computation of the pressure, since it asks one to solve an ill-conditioned linear system (in the variational formulation) or a Poisson equation.

Provided that divergence-free trial functions are available for the computation of the *velocity field*, these difficulties will be totally avoided: first, the incompressibility condition is directly taken into account by the approximation space of the velocity. Moreover, the pressure disappears after projecting the velocity equation onto the space of divergence-free vector functions (leading to the Leray formulation of the Navier–Stokes equations), and it will be computed through the Helmholtz decomposition of the nonlinear term.

Therefore in this article we propose using *divergence-free wavelets* as a decomposition basis of the *velocity* field, the solution of the incompressible Navier–Stokes equations. Such wavelets were originally defined by Lemarié-Rieusset [28], and first used to analyze two-dimensional (2D) turbulent flows [1, 25], as well as to compute the Stokes solution for the driven-cavity problem [36, 38]. We will consider here the anisotropic divergence-free wavelets constructed in [14]. The key point of such a numerical scheme based on divergence-free trial bases lies on the orthogonal projection of the nonlinear term onto the space of divergence-free functions, which is explicit in the Fourier spectral case. Since divergence-free wavelets are not orthogonal bases, we propose in [15] an iterative algorithm to provide the Helmholtz decomposition of any flow in the wavelet domain. Then we are in a position to define a new numerical scheme for solving incompressible Navier–Stokes equations: first, we project the equations onto the space of divergence-free vector fields, which eliminates the pressure. Second, we introduce a semi-implicit time scheme and propose an algorithm which requires only *fast wavelet transforms* for the computation of the *velocity* (contrary to finite element methods which need linear systems solvers). Finally, the *pressure* is directly recovered through the Helmholtz decomposition of the nonlinear term, without any further computation.

Hence, by construction, this scheme takes benefit from the localizations both in the space and frequency allowed by wavelets and should be used in a completely adaptive context. It should also be available in arbitrary dimension and extend readily to nonperiodic boundary conditions.

From a numerical point of view, the scheme we propose will be proved to be stable under a Courant–Friedrich–Levy (CFL) condition, provided that a sufficiently smooth solution exists. The stability of this wavelet scheme is mainly due to the divergence-free condition which is automatically and exactly satisfied by the divergence-free wavelet decomposition of the solution.

The paper is organized as follows. Section 1 briefly recalls the definitions of anisotropic divergence-free and curl-free wavelets. Section 2 presents the wavelet Helmholtz decomposition, as well as a wavelet numerical scheme for solving the heat equation. Section 3 introduces a new numerical scheme for the incompressible Navier–Stokes equations and studies its stability. Finally, section 4 shows numerical tests on the example of the merging of three vortices in dimension 2. The last part of this

section gives insight into how to make the method adaptive, with the support of some experiments.

1. Divergence-free and curl-free wavelets. One-dimensional wavelet bases are (typically orthogonal or biorthogonal [8]) bases of the space $L^2(\mathbb{R})$ which have the form $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$ $(j,k \in \mathbb{Z})$, where the *mother wavelet* ψ is a zero mean function [31, 24]. They are known to provide optimal approximations for a large class of functions [9]. From the numerical point of view, wavelet coefficients of a given function are computed by fast wavelet transforms (FWTs). In this setting, compactly supported divergence-free wavelet bases have been constructed by Lemarié-Rieusset [28], and Urban has extended the principle of their construction to derive curl-free wavelets [37].

In this section we will recall the definitions of anisotropic divergence-free and gradient (i.e., curl-free) wavelets. These wavelets are constructed thanks to two one-dimensional (1D) wavelets ψ_0 and ψ_1 related by differentiation: $\psi_1'(x) = 4\ \psi_0(x)$ [28].

For the sake of simplicity, below we give the expressions of the basis functions in dimension 2, but these constructions have been extended in arbitrary dimension d [14, 15, 11].

1.1. Divergence-free wavelets. The 2D anisotropic divergence-free wavelets are generated from the vector function (plotted in Figure 1, left)

$$\Psi^{\mathrm{div}}(x_1,x_2) = \left| \begin{array}{l} \psi_1(x_1)\psi_0(x_2), \\ -\psi_0(x_1)\psi_1(x_2) \end{array} \right.$$

by anisotropic dilations and translations. Hence, the 2D anisotropic divergence-free wavelets are given by

$$\Psi^{\mathrm{div}}_{\mathbf{j},\mathbf{k}}(x_1,x_2) = \left| \begin{array}{l} 2^{j_2}\psi_1(2^{j_1}x_1 - k_1)\psi_0(2^{j_2}x_2 - k_2), \\ -2^{j_1}\psi_0(2^{j_1}x_1 - k_1)\psi_1(2^{j_2}x_2 - k_2), \end{array} \right.$$

where $\mathbf{j} = (j_1, j_2) \in \mathbb{Z}^2$ is the scale parameter, and $\mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2$ is the position parameter. For $\mathbf{j}, \mathbf{k} \in \mathbb{Z}^2$, the family $\{\Psi^{\mathrm{div}}_{\mathbf{j},\mathbf{k}}\}$ forms a basis of

$$\mathbf{H}_{\mathrm{div},0}(\mathbb{R}^2) = \{\mathbf{f} \in (L^2(\mathbb{R}^2))^2\ ;\ \mathrm{div}\ \mathbf{f} \in L^2(\mathbb{R}^2), \quad \mathrm{div}\ \mathbf{f} = 0\}.$$

We introduce

$$\Psi^{\mathrm{n}}_{\mathbf{j},\mathbf{k}}(x_1,x_2) = \left| \begin{array}{l} 2^{j_1}\psi_1(2^{j_1}x_1 - k_1)\psi_0(2^{j_2}x_2 - k_2), \\ 2^{j_2}\psi_0(2^{j_1}x_1 - k_1)\psi_1(2^{j_2}x_2 - k_2) \end{array} \right.$$

as complement functions since $\Psi^{\mathrm{n}}_{\mathbf{j},\mathbf{k}}$ is orthogonal to $\Psi^{\mathrm{div}}_{\mathbf{j},\mathbf{k}}$ ($\mathbf{j},\mathbf{k}$ being fixed). Thus we have

$$(L^2(\mathbb{R}^2))^2 = \mathbf{H}_{\mathrm{div},0} \oplus \mathbf{H}_{\mathrm{n}} \tag{1.1}$$

with $\mathbf{H}_{\mathrm{div},0} = \mathrm{span}\{\Psi^{\mathrm{div}}_{\mathbf{j},\mathbf{k}}\ ;\ \mathbf{j},\mathbf{k} \in \mathbb{Z}^2\}$ and $\mathbf{H}_{\mathrm{n}} = \mathrm{span}\{\Psi^{\mathrm{n}}_{\mathbf{j},\mathbf{k}}\ ;\ \mathbf{j},\mathbf{k} \in \mathbb{Z}^2\}$.

1.2. Curl-free wavelets. Let $\mathbf{H}_{\mathrm{curl},0}(\mathbb{R}^2)$ be the space of gradient functions in $L^2(\mathbb{R}^2)$. We construct gradient wavelets by taking the gradient of a 2D wavelet basis. If we neglect the L^2-normalization, then the anisotropic gradient wavelets will be

defined by

$$\Psi^{\mathrm{curl}}_{\mathbf{j},\mathbf{k}}(x_1,x_2) = \frac{1}{4}\,\nabla\left(\psi_1(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2)\right) = \left|\begin{array}{l} 2^{j_1}\psi_0(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2),\\ \\ 2^{j_2}\psi_1(2^{j_1}x_1-k_1)\psi_0(2^{j_2}x_2-k_2).\end{array}\right.$$

For $\mathbf{j}=(0,0)$ the function is plotted in Figure 1, right. For $\mathbf{j}=(j_1,j_2),\mathbf{k}=(k_1,k_2)\in\mathbb{Z}^2$, the family $\{\Psi^{\mathrm{curl}}_{\mathbf{j},\mathbf{k}}\}$ forms a wavelet basis of $\mathbf{H}_{\mathrm{curl},0}(\mathbb{R}^2)$, where $\mathbf{H}_{\mathrm{curl},0}$ will be defined in section 2.1.1. We complete this basis to a $\left(L^2(\mathbb{R}^2)\right)^2$-basis with the following complement wavelets:

$$\Psi^{\mathrm{N}}_{\mathbf{j},\mathbf{k}}(x_1,x_2) = \left|\begin{array}{l} 2^{j_2}\psi_0(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2),\\ \\ -2^{j_1}\psi_1(2^{j_1}x_1-k_1)\psi_0(2^{j_2}x_2-k_2).\end{array}\right.$$

We have the following space decomposition:

$$(L^2(\mathbb{R}^2))^2 = \mathbf{H}_{\mathrm{N}}\oplus\mathbf{H}_{\mathrm{curl},0}$$

with $\mathbf{H}_{\mathrm{N}}=\mathrm{span}\{\Psi^{\mathrm{N}}_{\mathbf{j},\mathbf{k}}\ ;\ \mathbf{j},\mathbf{k}\in\mathbb{Z}^2\}$ and $\mathbf{H}_{\mathrm{curl},0}=\mathrm{span}\{\Psi^{\mathrm{curl}}_{\mathbf{j},\mathbf{k}}\ ;\ \mathbf{j},\mathbf{k}\in\mathbb{Z}^2\}$.

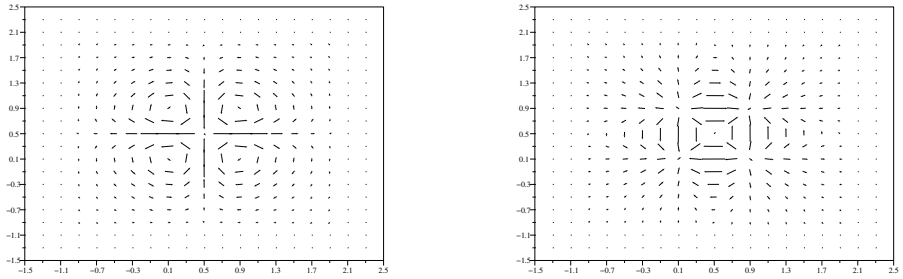

FIG. 1. *Examples of divergence-free (on the left) and curl-free (on the right) vector wavelets in dimension* 2.

2. Wavelet numerical algorithms.

2.1. Wavelet Helmholtz decomposition.

2.1.1. Principle of the Helmholtz decomposition. The Helmholtz decomposition [21, 7] consists in splitting a vector function $\mathbf{u}\in(L^2(\mathbb{R}^d))^d$, $d=2$ or 3, into its divergence-free component $\mathbf{u}_{\mathrm{div}}$ and a gradient vector. More precisely, there exist a potential-function p and a stream-function ψ such that

$$\mathbf{u}=\mathbf{u}_{\mathrm{div}}+\nabla p \quad\text{and}\quad \mathbf{u}_{\mathrm{div}}=\mathbf{curl}\ \psi. \tag{2.1}$$

Moreover, the functions $\mathbf{curl}\ \psi$ and ∇p are orthogonal in $(L^2(\mathbb{R}^d))^d$. The stream-function ψ—which we assume to be divergence-free for $d=3$—and the potential-function p are unique up to an additive constant.

In $\mathbb{R}^2$, the stream-function is a scalar-valued function, whereas in $\mathbb{R}^3$ it is a three-dimensional (3D) vector function. This decomposition may be viewed as the following orthogonal space splitting:

$$(L^2(\mathbb{R}^d))^d = \mathbf{H}_{\mathrm{div},0}(\mathbb{R}^d) \oplus^{\perp} \mathbf{H}_{\mathrm{curl},0}(\mathbb{R}^d), \tag{2.2}$$

where

$$\mathbf{H}_{\mathrm{div},0}(\mathbb{R}^d) = \{\mathbf{v} \in (L^2(\mathbb{R}^d))^d \ ; \ \mathrm{div}\ \mathbf{v} \in L^2(\mathbb{R}^d), \quad \mathrm{div}\ \mathbf{v} = 0\}$$

is the space of divergence-free vector functions, and

$$\mathbf{H}_{\mathrm{curl},0}(\mathbb{R}^d) = \{\mathbf{v} \in (L^2(\mathbb{R}^d))^d \ ; \ \mathbf{curl}\ \mathbf{v} \in (L^2(\mathbb{R}^d))^d, \quad \mathbf{curl}\ \mathbf{v} = 0\}$$

is the space of curl-free vector functions (if $d = 2$, then we have to replace $\mathbf{curl}\ \mathbf{v} \in (L^2(\mathbb{R}^d))^d$ by $\mathrm{curl}\ \mathbf{v} \in L^2(\mathbb{R}^2)$ in the definition). Let $\mathbb{P}$ be the orthogonal projector onto the space $\mathbf{H}_{\mathrm{div},0}(\mathbb{R}^d)$, also called the Leray projector. From the Helmholtz decomposition (2.1) of $\mathbf{u}$ we have $\mathbb{P}\mathbf{u} = \mathbf{u}_{\mathrm{div}}$.

For the whole space $\mathbb{R}^d$, the proof of the above decomposition can be derived easily by means of the Fourier transform. In more general domains, we refer the reader to [21, 7]. Notice also that $\mathbf{H}_{\mathrm{div},0}(\mathbb{R}^d)$ is the space of **curl** functions, whereas $\mathbf{H}_{\mathrm{curl},0}(\mathbb{R}^d)$ is the space of gradient functions.

The objective now is to make explicit the Helmholtz decomposition of any vector field in the wavelet domain.

2.1.2. Iterative wavelet Helmholtz decomposition algorithm. Instead of the previous orthogonal sum (2.2), the divergence-free and gradient wavelet decompositions provide the following (nonorthogonal) direct sums of vector spaces:

$$(L^2(\mathbb{R}^d))^d = \mathbf{H}_{\mathrm{div},0} \oplus \mathbf{H}_{\mathrm{n}}, \qquad (L^2(\mathbb{R}^d))^d = \mathbf{H}_{\mathrm{N}} \oplus \mathbf{H}_{\mathrm{curl},0},$$

where the spaces $\mathbf{H}_{\mathrm{n}} = \mathrm{span}\{\Psi^{\mathrm{n}}_{\mathbf{j},\mathbf{k}}\}$ and $\mathbf{H}_{\mathrm{N}} = \mathrm{span}\{\Psi^{\mathrm{N}}_{\mathbf{j},\mathbf{k}}\}$ are generated by the complementary wavelets, as introduced in section 1. These direct sums may be rewritten by introducing the associated nonorthogonal projectors (see [14] for practical computations):

$$\mathbf{v} = \mathrm{P}_{\mathrm{div}}\,\mathbf{v} + \mathrm{Q}_{\mathrm{n}}\,\mathbf{v}, \qquad \mathbf{v} = \mathrm{P}_{\mathrm{N}}\,\mathbf{v} + \mathrm{Q}_{\mathrm{curl}}\,\mathbf{v}.$$

Then, applying alternatively the divergence-free and the curl-free wavelet decompositions, we define a sequence $(\mathbf{v}^p)_{p\in\mathbb{N}} \in \left(L^2(\mathbb{R}^d)\right)^d$:

$$\begin{aligned}
&\mathbf{v}^0 = \mathbf{v},\\
&\mathbf{v}^p = (\mathrm{Q}_{\mathrm{curl}} + \mathrm{P}_{\mathrm{N}})\ (\mathrm{P}_{\mathrm{div}} + \mathrm{Q}_{\mathrm{n}})\ \mathbf{v}^p \ = \ \underbrace{\mathrm{P}_{\mathrm{div}}\,\mathbf{v}^p}_{\mathbf{v}^p_{\mathrm{div}}} + \underbrace{\mathrm{Q}_{\mathrm{curl}}\,\mathrm{Q}_{\mathrm{n}}\,\mathbf{v}^p}_{\mathbf{v}^p_{\mathrm{curl}}} + \underbrace{\mathrm{P}_{\mathrm{N}}\,\mathrm{Q}_{\mathrm{n}}\,\mathbf{v}^p}_{\mathbf{v}^{p+1}}.
\end{aligned}$$

Finally, if this sequence converges to 0 in L^2, then we obtain

$$\mathbf{v}_{\mathrm{div}} = \sum_{p=0}^{+\infty} \mathbf{v}^p_{\mathrm{div}}, \quad \mathbf{v}_{\mathrm{curl}} = \sum_{p=0}^{+\infty} \mathbf{v}^p_{\mathrm{curl}}.$$

This algorithm has been proved to converge in two dimensions using any kind of wavelet and in arbitrary dimension using Shannon wavelets [15]. In the case of Shannon wavelets, we also have the following convergence theorem.

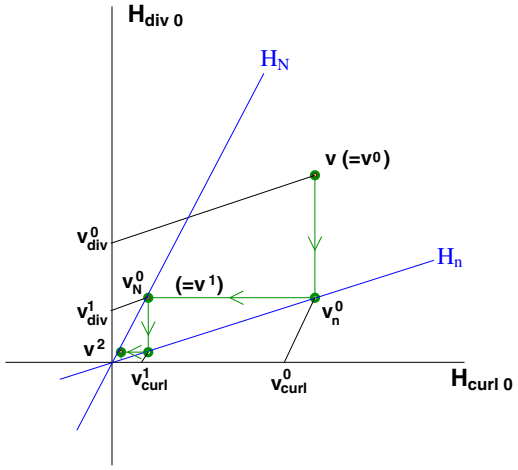

FIG. 2. *Construction of the sequences* $\mathbf{v}^p_{\mathrm{div}}$ *and* $\mathbf{v}^p_{\mathrm{curl}}$*, and schematization of the convergence process of the algorithm with* $\mathbf{H}_{\mathrm{n}} = \mathrm{span}\{\Psi^{\mathrm{n}}_{\mathbf{j},\mathbf{k}}\}$ *and* $\mathbf{H}_{\mathrm{N}} = \mathrm{span}\{\Psi^{\mathrm{N}}_{\mathbf{j},\mathbf{k}}\}$.

THEOREM 2.1. *Let* $\mathbf{v} \in \left(L^2(\mathbb{R}^d)\right)^d$*, and let the sequence* $(\mathbf{v}^p)_{p\geq 0}$ *be defined by*

$$\mathbf{v}^0 = \mathbf{v} \qquad \text{and} \qquad \mathbf{v}^{p+1} = \mathrm{P}_N \mathrm{Q}_n \ \mathbf{v}^p, \quad p \geq 0, \tag{2.3}$$

where Q_n *and* P_N *are the complementary projectors associated, respectively, with divergence-free wavelets and curl-free wavelets. If the wavelet* ψ_1 *used in section* 1 *for the construction of divergence-free and curl-free wavelets is the Shannon wavelet,*[1] *then the sequence* $(\mathbf{v}^p)$ *satisfies, in the* L^2*-norm,*

$$\|\mathbf{v}^p\| \leq \left(\frac{9}{16}\right)^p \|\mathbf{v}\|.$$

Experimentally, we also observe the convergence for many kinds of 2D and 3D wavelets [14] as schematized in Figure 2. This algorithm will be used in section 3, in the numerical scheme for incompressible Navier–Stokes equations.

2.2. Wavelet solution of the discrete heat equation. As we would like the numerical scheme to be sufficiently stable in time, we will adopt an implicit scheme for the diffusive part of the Navier–Stokes equation. Therefore we focus on the scalar equation

$$(Id - \alpha\Delta)u = f, \quad \alpha = \nu\delta t, \quad f : \mathbb{R}^d \to \mathbb{R} \tag{2.4}$$

which arises in the resolution of the heat equation, after introducing an implicit time scheme. We will now present an algorithm for solving such problem, based on wavelet preconditioners of elliptic operators and related to the works of Liandrat and Cohen (see [30, 9]). In the vector case, the following algorithm could be applied to each component of the vector field, but we will prefer to project (2.4) onto the divergence-free space as indicated later in formula (2.8).

[1] $\psi_1(x) = \frac{\sin 2\pi(x-1/2)}{\pi(x-1/2)} - \frac{\sin \pi(x-1/2)}{\pi(x-1/2)}$, $\widehat{\psi_1}(\xi) = e^{-i\xi/2}\ \chi_{[-2\pi,-\pi]\cup[\pi,2\pi]}(\xi)$.

Let $(\psi_{\mathbf{j},\mathbf{k}})_{\mathbf{j},\mathbf{k}\in\mathbb{Z}^d}$ be a multivariate wavelet basis in $\mathbb{R}^d$, constructed by tensor-products of d 1D wavelets. Then we expand the scalar function f into the basis $(\psi_{\mathbf{j},\mathbf{k}})$ up to a scale J:

$$f = \sum_{\mathbf{j},\mathbf{k}\in\mathbb{Z}^d,\, |j|<J} d_{\mathbf{j},\mathbf{k}}\ \psi_{\mathbf{j},\mathbf{k}}. \tag{2.5}$$

We gather the wavelets of a same level $\mathbf{j}$ in $f_{\mathbf{j}} = \sum_{\mathbf{k}\in\mathbb{Z}^d} d_{\mathbf{j},\mathbf{k}}\ \psi_{\mathbf{j},\mathbf{k}}$; then

$$f = \sum_{\mathbf{j}\in\mathbb{Z}^d,\, |j|<J} f_{\mathbf{j}}.$$

For $\mathbf{j} = (j_1, \ldots, j_d)$, $f_{\mathbf{j}}$ is localized in the Fourier domain around a wavenumber equal to $\rho(2^{j_1},\ 2^{j_2}, \ldots, 2^{j_d})$ with ρ approximating the average spectrum location of the wavelet ψ (for instance, optimally $\rho = \frac{\sqrt{5}\pi}{\sqrt{2}}$ for the Shannon wavelet [12]). Then we introduce

$$\omega_{\mathbf{j}}^2\ = \rho^2 \sum_{i=1}^{d} 2^{2j_i}.$$

In order to solve $(Id - \alpha\Delta)u = f$, we start with

$$u_0 = \sum_{\mathbf{j}\in\mathbb{Z}^d,\, |j|<J} u_{\mathbf{j},0}, \qquad \text{where} \qquad u_{\mathbf{j},0} = \frac{1}{1+\alpha\omega_{\mathbf{j}}^2} f_{\mathbf{j}}, \tag{2.6}$$

which is a first approximation of the solution.

Then, for $k \geq 0$, let

$$u_{k+1} = u_k + \sum_{\mathbf{j}\in\mathbb{Z}^d,\, |j|<J} \frac{1}{1+\alpha\omega_{\mathbf{j}}^2}\ (f_{\mathbf{j}} - (Id - \alpha\Delta)u_{k\,\mathbf{j}}). \tag{2.7}$$

This algorithm has been presented in detail in [12], where it has been proved that the sequence (u_k) converges to the exact solution u of (2.4). The convergence rate depends on α: the smaller α is, the faster the algorithm converges. More precisely, we have the following convergence theorem.

THEOREM 2.2. *Let f be in $L^2(\mathbb{R}^d)$, and let the sequence $(u_k)_{k\geq 0}$ be defined by* (2.6) *and* (2.7). *Using Shannon wavelets in the decomposition* (2.5), *the sequence (u_k) satisfies, in the L^2-norm,*

$$\|u_k - u\| \leq \left(\frac{3\alpha}{2\delta x^2 + 5\alpha}\right)^k \|u_0 - u\|,$$

where u is the solution of (2.4) *and $\delta x = 2^{-J}$ the mesh size of the smallest computed scale.*

In particular, this implies that

$$\|u_k - u\| \leq \left(\frac{3}{5}\right)^k \|u_0 - u\|,$$

which proves the convergence.

But if $\frac{\nu\delta t}{\delta x^2} \ll 1$ (recall that $\alpha = \nu\delta t$), then it could be more interesting to consider

$$\|u_k - u\| \leq \left(\frac{3\nu\delta t}{2\delta x^2}\right)^k \|u_0 - u\|.$$

In practice, since we are working with spline wavelets, the Laplacian $\Delta u_{k\,\mathbf{j}}$ in (2.7) is analytically computed and then reprojected onto the finite-dimensional wavelet basis by a spline approximation. In the following, we will apply the above method in a divergence-free wavelet basis: in this case, $\mathbf{f}$ and $\mathbf{u}$ are vector functions, expanded into the divergence-free basis ($\psi_{\mathbf{j},\mathbf{k}} = \Psi^{\text{div}}_{\mathbf{j},\mathbf{k}}$ in (2.5)). Since the divergence-free condition is preserved under the application of $(Id - \alpha\Delta)$, we solve the discrete heat equation $(Id - \alpha\Delta)\mathbf{u} = \mathbf{f}$ with div $\mathbf{u} =$ div $\mathbf{f} = 0$ by projecting (2.7) onto a finite-dimensional divergence-free multiresolution analysis (MRA):

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \sum_{|\mathbf{j}|<J} \frac{1}{1+\alpha\omega_{\mathbf{j}}^2}\ (\mathbf{f}_{\mathbf{j}} - [P_{\text{div,J}}(Id - \alpha\Delta)\mathbf{u}_k]_{\mathbf{j}}), \tag{2.8}$$

where $P_{\text{div,J}}$ is the truncated version of P_{div} introduced in section 2.1.2, meaning the oblique projector onto the space spanned by divergence-free wavelets of scale index $\mathbf{j}$ such that $|\mathbf{j}| < J$. Since the term $(Id - \alpha\Delta)\mathbf{u}_k$ is already divergence-free, the action of $P_{\text{div,J}}$ reduces to a truncation in scale of the divergence-free wavelet decomposition. Hence, the resulting error is the usual approximation error in an MRA. We remark that a similar algorithm could also be applied to the scaling low-pass subband [12] and that alternatively the lowest scale does not diffuse and does not need to be treated.

3. A numerical scheme for Navier–Stokes equations. Below we present a divergence-free wavelet numerical method, for the solution of the incompressible Navier–Stokes equations, written in velocity-pressure formulation (without a forcing term) in the whole space:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \nu\Delta\mathbf{u} = 0, \quad t \in [0,T],\ \mathbf{x} \in \mathbb{R}^d, \\ \nabla\cdot\mathbf{u} = 0, \end{cases} \tag{3.1}$$

with initial data $\mathbf{u}_0$.

Computation of the velocity. First, the velocity $\mathbf{u}$ is searched in $\mathbf{H}_{\text{div},0}(\mathbb{R}^d)$ as a linear combination of divergence-free wavelets. If we denote by $\mathbb{P}$ the orthogonal Leray projector onto $\mathbf{H}_{\text{div},0}(\mathbb{R}^d)$ (introduced in section 2.1.1, and here expressed in the wavelet domain), then (3.1) becomes, after projection onto the divergence-free vector functions,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbb{P}\left[(\mathbf{u}\cdot\nabla)\mathbf{u}\right] - \nu\Delta\mathbf{u} = 0. \tag{3.2}$$

The pressure is eliminated, but it will simply be recovered through the wavelet Helmholtz decomposition of the nonlinear term.

Computation of the pressure. The wavelet Helmholtz decomposition of the nonlinear term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ reads

$$(\mathbf{u}\cdot\nabla)\mathbf{u} = \mathbb{P}\left[(\mathbf{u}\cdot\nabla)\mathbf{u}\right] + \left[(\mathbf{u}\cdot\nabla)\mathbf{u}\right]_{\text{curl}} = \mathbb{P}\left[(\mathbf{u}\cdot\nabla)\mathbf{u}\right] - \nabla p.$$

The divergence-free term $\mathbb{P}\left[(\mathbf{u}\cdot\nabla)\mathbf{u}\right]$ is used for the computation of the velocity in (3.2), whereas the curl-free term is used for the computation of the pressure as follows: remembering the definition of gradient (curl-free) wavelets given in section 1.2 (2D case), $-\nabla p$ in the above decomposition reads

$$-\nabla p(t,\mathbf{x}) = \sum_{\mathbf{j},\mathbf{k}} d_{\text{curl}\,\mathbf{j},\mathbf{k}}(t)\ \frac{1}{4}\ \nabla\left(\psi_1(2^{j_1}x_1 - k_1)\cdots\psi_1(2^{j_d}x_d - k_d)\right),$$

which simply gives the pressure imposing $\int_{\mathbf{x}} p(t,\mathbf{x})\,d\mathbf{x} = 0$:

$$p(t,\mathbf{x}) = -\frac{1}{4}\sum_{\mathbf{j},\mathbf{k}} d_{\mathrm{curl}\,\mathbf{j},\mathbf{k}}(t)\ \psi_1(2^{j_1}x_1 - k_1)\cdots\psi_1(2^{j_d}x_d - k_d),$$

where $d_{\mathrm{curl}\,\mathbf{j},\mathbf{k}}$ are the (curl-free) wavelet coefficients of the curl-free part of the nonlinear term. We remark that the computation of the pressure in physical space requires only an inverse FWT.

We will now focus on (3.2). We will introduce a finite difference time scheme, for which we will prove the stability, provided a CFL condition.

3.1. Time discretizations and stability conditions. Let δt be a time step. At each time $t_n = n\delta t$, the exact solution $\mathbf{u}(t_n,\mathbf{x})$ is approximated by $\mathbf{u}_n$ such that

$$\mathbf{u}_n(\mathbf{x}) = \sum_{\mathbf{j},\max_i |j_i|<J}\ \sum_{\mathbf{k}\in\mathbb{Z}^2} u^n_{\mathbf{j},\mathbf{k}}\ \Psi^{\mathrm{div}}_{\mathbf{j},\mathbf{k}}(\mathbf{x}). \tag{3.3}$$

Here the divergence-free wavelet expansion is truncated in scale, which corresponds to a finest mesh size $\delta x = 2^{-J}$. In practice, the sums in (3.3) are finite.

For stability reasons the diffusive term of the Navier–Stokes equation will be implicitly treated, whereas for simplicity the nonlinear term will be explicitly treated. We begin with the study of the Euler scheme.

3.1.1. Semi-implicit Euler scheme. The semi-implicit Euler scheme applied to (3.2) is given by

$$(Id - \nu\delta t\Delta)\,\mathbf{u}_{n+1} = \mathbf{u}_n - \delta t\ \mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbf{u}_n\right], \tag{3.4}$$

where $\mathbf{u}_n$ is expanded into a divergence-free wavelet basis, up to a scale $J-1$ (3.3). In the above expression, the gradient operator is exactly computed, whereas the other operators are approximated. Actually, the operators Δ and $\mathbb{P}$ should be denoted by Δ_J and $\mathbb{P}_J$, as they stand for the operators into the discretized wavelet space: $\Delta_J = P_{\mathrm{div},J}\,\Delta$ and $\mathbb{P}_J = P_J\,\mathbb{P}$ with $P_{\mathrm{div},J}$ introduced in section 2.2 after formula (2.8). And P_J is the L^2-orthogonal projector onto the discretization space. Nevertheless, we will keep the notation Δ and $\mathbb{P}$ for readability reasons. Then each time step n is divided into two parts:

- First, compute the divergence-free wavelet coefficients of $\mathbb{P}\left[(\mathbf{u_n}\cdot\nabla)\mathbf{u_n}\right]$: this computation is achieved through the reconstruction of $\mathbf{u}_n$ in physical space from its wavelet coefficients $u^n_{\mathbf{j},\mathbf{k}}$; then the nonlinear term is evaluated by a collocation technique in physical space. Finally, the wavelet Helmholtz decomposition algorithm of section 2.1.2 provides the expected coefficients.
- Second, compute the divergence-free wavelet coefficients $u^{n+1}_{\mathbf{j},\mathbf{k}}$ of $\mathbf{u}_{n+1}$ by using the algorithm described in section 2.2 for the discrete heat equation solution. As the heat kernel is directly computed in the divergence-free wavelet basis, there is no need for further projection onto the divergence-free function space as indicated in section 2.2, (2.8).

Both computations involve an iterative method, but it requires only FWTs. We will see in numerical tests that in practice these numbers of iterations could be very low.

L^2-stability of the semi-implicit Euler scheme. In this paragraph we proceed to study the stability of the semi-implicit Euler scheme. We will provide a CFL stability condition: $\delta t \le C\delta x^2$. This study is inspired by the approach of Temam in his book

[35] to derive the L^2-stability of one-step schemes, including the explicit Euler scheme, within the framework of finite elements. Here we will limit ourselves to simplified proofs which will be sufficient for recovering the effective CFL conditions encountered in the numerical experiments of section 4.

For studying the stability, we assume that, at time step n, we observe a small error ε_n. Hence, instead of having exactly $\mathbf{u}_n$—a discrete solution of (3.4), as close to $\mathbf{u}(n\delta t, \cdot)$ as possible—we have $\mathbf{u}_n + \varepsilon_n$. Then, at step $n+1$, the error with respect to $\mathbf{u}((n+1)\delta t, \cdot)$ is due, on one hand, to the newly introduced error in the algorithm (which corresponds to the consistency error) and, on the other hand, to the increase of the error coming from the previous steps (and which is concerned with stability).

On the other hand, let P_J—already introduced at the beginning of this section 3.1.1—be the L^2-orthogonal projector onto the discretization space; we introduce the consistency error as follows. If at time $t_n = n\delta t$ the discrete solution $\mathbf{u}_n$ coincides with the projection of the exact solution $P_J(\mathbf{u}(t_n, .))$, then we obtain

$$
\begin{aligned}
e_{n+1} &= \frac{1}{\delta t}\left[P_J\left(\mathbf{u}((n+1)\delta t, .\right) - P_J(\mathbf{u}(n\delta t, .)) - (\mathbf{u}_{n+1} - \mathbf{u}_n)\right] \\
&= \frac{1}{\delta t}\left[P_J\left(\int_{n\delta t}^{(n+1)\delta t} \partial_t \mathbf{u}\, dt\right)\right] - (\nu\Delta\mathbf{u}_{n+1} - \mathbb{P}\left[(\mathbf{u}_n \cdot \nabla)\mathbf{u}_n\right]),
\end{aligned} \tag{3.5}
$$

which leads to

$$
\begin{aligned}
e_{n+1} = \frac{1}{\delta t}&\left[P_J\left(\mathbf{u}(t_{n+1}, .\right) - \mathbf{u}(t_n, .))\right] \\
&- \nu\Delta(Id - \nu\delta t\Delta)^{-1}\left(P_J\mathbf{u}(t_n, .) - \ \delta t\mathbb{P}\left[(P_J\mathbf{u}(t_n, .) \cdot \nabla)\, P_J\mathbf{u}(t_n, .)\right]\right) \\
&+ \mathbb{P}\left[(P_J\mathbf{u}(t_n, .) \cdot \nabla)\, P_J\mathbf{u}(t_n, .)\right].
\end{aligned} \tag{3.6}
$$

We will assume that the above consistency error is small and behaves at each time step as

$$\|e_{n+1}\|_{L^2} = O(\delta t) + O(\delta x^2),$$

which means that it is supposed to be of order one in time and order two in space.

Assumptions. Let us assume that the solution $\mathbf{u}(t, \mathbf{x})$ of the Navier–Stokes equations (3.1) is continuously differentiable in space and that $\mathbf{u}$ and its derivatives go to zero at infinity ($\|\mathbf{x}\| \to +\infty$). And let us assume that $\mathbf{u}_n$ and its derivatives have similar bounds as $\mathbf{u}(t, \mathbf{x})$ and its derivatives. Hence,

$$\sup_{t\in[0,T],\ \mathbf{x}\in\mathbb{R}^d} |\mathbf{u}(t, \mathbf{x})| < +\infty \quad \text{and} \quad \|\mathbf{u}_n\|_{L^\infty} \le A_0, \tag{3.7}$$

$$\sup_{t\in[0,T],\ \mathbf{x}\in\mathbb{R}^d} |\nabla\mathbf{u}(t, \mathbf{x})| < +\infty \quad \text{and} \quad \|\nabla\mathbf{u}_n\|_{L^\infty} \le A_1 \tag{3.8}$$

with A_0 a constant close to $\|\mathbf{u}\|_{L^\infty}$ and A_1 a constant close to $\|\nabla\mathbf{u}\|_{L^\infty}$.

A priori estimate. For the stability, the error ε_n lives in our *finite-dimensional* discretization space. We assume, and later verify, the following a priori estimate on ε_n (cf. inequality (3.18)):

$$\|\varepsilon_n\|_{L^2} = O(\delta x^2). \tag{3.9}$$

We also assume that $\delta t = o(\delta x)$, which means that $\frac{\delta t}{\delta x} \to 0$ when $\delta x \to 0$. This will be satisfied since we will have $\delta t \leq C\delta x^2$ as a CFL condition.

Proof of stability. If we prove that there exists a constant C (which will depend on A_0 and A_1) such that $\|\varepsilon_{n+1}\| \leq (1 + C\delta t)\|\varepsilon_n\|$, then the scheme is stable, in the sense that an initial error propagates in time within a fixed bounded domain. In our case, we will recover the L^2-stability thanks to the following well-known result ([21, Chapter IV, Lemma 2.1]).

LEMMA 3.1. *Let* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in H^1(\mathbb{R}^d)^d$, H^1 *denoting the Sobolev space, be such that* $(\mathbf{u}\cdot\nabla)\mathbf{v}, (\mathbf{u}\cdot\nabla)\mathbf{w} \in L^2$. *If* $\mathbf{u} \in \boldsymbol{H}_{\mathrm{div},0}(\mathbb{R}^d)$, *then*

$$\langle \mathbf{v}, (\mathbf{u}\cdot\nabla)\mathbf{w}\rangle_{L^2} = -\langle (\mathbf{u}\cdot\nabla)\mathbf{v}, \mathbf{w}\rangle_{L^2}.$$

Proof.

$$\begin{aligned}
\langle \mathbf{v}, (\mathbf{u}\cdot\nabla)\mathbf{w}\rangle &= \int_{\mathbf{x}\in\mathbb{R}^d} \mathbf{v}\cdot(\mathbf{u}\cdot\nabla)\mathbf{w}\, d\mathbf{x} \\
&= \int_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^{d} v_i(\mathbf{x}) \sum_{k=1}^{d} u_k(\mathbf{x})\partial_k w_i(\mathbf{x})\, d\mathbf{x} \\
&= \int_{\mathbf{x}\in\mathbb{R}^d} \sum_{k=1}^{d} u_k(\mathbf{x}) \left(\sum_{i=1}^{d} v_i(\mathbf{x})\partial_k w_i(\mathbf{x}) \right) d\mathbf{x} \\
&= \int_{\mathbf{x}\in\mathbb{R}^d} \sum_{k=1}^{d} u_k(\mathbf{x}) \left(\partial_k \left(\sum_{i=1}^{d} v_i(\mathbf{x}) w_i(x) \right) \right. \\
&\quad \left. - \left(\sum_{i=1}^{d} w_i(\mathbf{x})\partial_k v_i(\mathbf{x}) \right) \right) d\mathbf{x} \\
&= -\int_{\mathbf{x}\in\mathbb{R}^d} \left(\sum_{k=1}^{d} \partial_k u_k(\mathbf{x}) \right) \left(\sum_{i=1}^{d} v_i(\mathbf{x}) w_i(x) \right) d\mathbf{x} \\
&\quad -\langle \mathbf{w}, (\mathbf{u}\cdot\nabla)\mathbf{v}\rangle \\
&= -\langle \mathbf{w}, (\mathbf{u}\cdot\nabla)\mathbf{v}\rangle.
\end{aligned}$$

Here we first used integration by parts and then the fact that $\operatorname{div}\mathbf{u} = \sum_{k=1}^{d} \partial_k u_k = 0$.

Remark 3.1. This result is still valid on an open set Ω with slipping conditions $\mathbf{u}\cdot\mathbf{n} = 0$ on the boundary $\partial\Omega$, with $\mathbf{n}$ the normal to $\partial\Omega$.

COROLLARY 3.2. *Let* $\mathbf{u}, \mathbf{v} \in H^1(\mathbb{R}^d)^d$ *be such that* $(\mathbf{u}\cdot\nabla)\mathbf{v} \in L^2$. *If* $\mathbf{u} \in \boldsymbol{H}_{\mathrm{div},0}(\mathbb{R}^d)$, *then* $\mathbf{v} \perp (\mathbf{u}\cdot\nabla)\mathbf{v}$ *for the* L^2 *scalar product, i.e.,*

$$\langle \mathbf{v}, (\mathbf{u}\cdot\nabla)\mathbf{v}\rangle_{L^2} = \int_{\mathbf{x}\in\mathbb{R}^d} \mathbf{v}\cdot(\mathbf{u}\cdot\nabla)\mathbf{v}\, d\mathbf{x} = 0.$$

Proof. We apply Lemma 3.1 with $\mathbf{w} = \mathbf{v}$ and get $\langle \mathbf{v}, (\mathbf{u}\cdot\nabla)\mathbf{v}\rangle = -\langle (\mathbf{u}\cdot\nabla)\mathbf{v}, \mathbf{v}\rangle$.

Remark 3.2. This result also yields orthogonality between the vectors $\mathbf{v}$ and $(\mathbf{v}\cdot\nabla)\mathbf{v}$ for $\mathbf{v} \in \mathbf{H}_{\mathrm{div},0}(\mathbb{R}^d)$.

THEOREM 3.3. *An error ε_t equal to $\varepsilon_0 = O(\delta x^2)$ at time $t = 0$ propagates in the wavelet numerical scheme* (3.4) *bounded by*

$$\|\varepsilon_t\|_{L^2} \leq e^{\lambda(\frac{A_0^2}{2}\frac{\delta t}{\delta x^2}+A_1)t}\|\varepsilon_0\|_{L^2},$$

where A_0 and A_1 are the constants depending on $\mathbf{u}$ introduced in formulas (3.7) *and* (3.8)*, δt the time step, δx the mesh of the smallest computed scale, and λ a constant close to* 1.

Then the scheme (3.4) *is stable under the CFL condition $\frac{\delta t}{\delta x^2} \leq C$ for a fixed constant $C > 0$ which can be chosen equal to $\frac{2A_1}{A_0^2}$, for instance.*

Proof. If we want to estimate the propagation of the error ε_n, then from the scheme (3.4) we obtain

$$(Id - \nu\delta t\Delta)(\mathbf{u}_{n+1} + \varepsilon_{n+1}) = \mathbf{u}_n + \varepsilon_n - \delta t\mathbb{P}\left[((\mathbf{u}_n + \varepsilon_n)\cdot\nabla)(\mathbf{u}_n + \varepsilon_n)\right]. \tag{3.10}$$

Then the sequence ε_n satisfies

$$(Id - \nu\delta t\Delta)\,\varepsilon_{n+1} = \varepsilon_n - \delta t\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n + (\mathbf{u}_n\cdot\nabla)\varepsilon_n + (\varepsilon_n\cdot\nabla)\varepsilon_n\right]. \tag{3.11}$$

We apply Corollary 3.2 to $\varepsilon_n \in \mathbf{H}_{\mathrm{div},0}(\mathbb{R}^d)$, and by orthogonality of ε_n to all gradient functions we have

$$\varepsilon_n \perp (\mathbf{u}_n\cdot\nabla)\varepsilon_n \quad \Longrightarrow \quad \varepsilon_n \perp \mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right].$$

and

$$\varepsilon_n \perp (\varepsilon_n\cdot\nabla)\varepsilon_n \quad \Longrightarrow \quad \varepsilon_n \perp \mathbb{P}\left[(\varepsilon_n\cdot\nabla)\varepsilon_n\right].$$

Then, denoting $\eta_n = \mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n + (\varepsilon_n\cdot\nabla)\varepsilon_n\right]$, we get

$$\|\varepsilon_n - \eta_n\delta t\|_{L^2}^2 = \|\varepsilon_n\|_{L^2}^2 + \|\eta_n\|_{L^2}^2\delta t^2$$

and

$$\|\varepsilon_n - \eta_n\delta t\|_{L^2} = \|\varepsilon_n\|_{L^2}\left(1 + \frac{\|\eta_n\|_{L^2}^2\delta t^2}{\|\varepsilon_n\|_{L^2}^2}\right)^{1/2} \leq \|\varepsilon_n\|_{L^2} + \frac{\|\eta_n\|_{L^2}^2}{2\|\varepsilon_n\|_{L^2}}\delta t^2$$

since $(1+x)^{1/2} \leq 1 + \frac{x}{2}\ \forall x \geq -1$.

The discrete solution $\mathbf{u}_n = P_J(\mathbf{u}_n)$ is a truncated wavelet expansion as indicated in formula (3.3), and the stability error ε_n lives in the same discretization space. Then the differentiation operator will multiply ε_n by only $(\delta x)^{-1}$ at most. Hence, we have the following bound:

$$\begin{aligned}\|\eta_n\|_{L^2} = \|\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n + (\varepsilon_n\cdot\nabla)\varepsilon_n\right]\|_{L^2} &\leq \|(\mathbf{u}_n\cdot\nabla)\varepsilon_n + (\varepsilon_n\cdot\nabla)\varepsilon_n\|_{L^2}\\ &\leq \|\mathbf{u}_n\|_{L^\infty}|\varepsilon_n|_{H^1} + \|\varepsilon_n\|_{L^\infty}|\varepsilon_n|_{H^1}.\end{aligned}$$

On the other hand, let $\varepsilon_n = \sum_{\mathbf{j},\mathbf{k}} d_{\mathbf{j},\mathbf{k}}\Psi_{\mathbf{j},\mathbf{k}}$ be the wavelet expansion of ε_n. Then each partial derivative writes $\partial_i\varepsilon_n = \sum_{\mathbf{j},\mathbf{k}} 2^{j_i+2}d_{\mathbf{j},\mathbf{k}}\Psi_{\mathbf{j},\mathbf{k}}^{\partial_i}$ (where $\partial_i\Psi = 4\Psi^{\partial_i}$). The norm equivalence provided by the wavelet decomposition in Sobolev spaces leads to

$$|\varepsilon_n|_{H^1} \sim \sqrt{\sum_{i=1}^{d}\|2^{j_i+2}d_{\mathbf{j},\mathbf{k}}\|_{\ell^2}^2} \ \leq\ 2^{J+1}\ \sqrt{d}\ \|d_{\mathbf{j},\mathbf{k}}\|_{\ell^2} \sim \frac{\|\varepsilon_n\|_{L^2}}{\delta x} \tag{3.12}$$

since $\delta x = 2^{-J}$.

Now let $\varepsilon_n = \sum_{\mathbf{k}} c_{J,\mathbf{k}} \Phi_{J,\mathbf{k}}$ be the scaling decomposition of ε_n, the scaling functions $\Phi_{J,\mathbf{k}} = \Phi(2^J \mathbf{x} - \mathbf{k})$ being L^∞-normalized $\|\Phi_{J,\mathbf{k}}\|_{L^\infty} = 1$. Then

$$\|\varepsilon_n\|_{L^2} \sim \|2^{-\frac{Jd}{2}}\ c_{J,\mathbf{k}}\|_{\ell^2}\ \geq\ 2^{-\frac{Jd}{2}} \|c_{J,\mathbf{k}}\|_{\ell^\infty} = \delta x^{\frac{d}{2}} \|c_{J,\mathbf{k}}\|_{\ell^\infty}.$$

Since

$$\|\varepsilon_n\|_{L^\infty} \sim \|c_{J,\mathbf{k}}\|_{\ell^\infty} \|\Phi_{J,\mathbf{k}}\|_{L^\infty} \sim \|c_{J,\mathbf{k}}\|_{\ell^\infty}$$

we obtain the following result:

$$\|\varepsilon_n\|_{L^\infty}\ \lesssim\ \frac{\|\varepsilon_n\|_{L^2}}{\delta x^{d/2}}. \tag{3.13}$$

This leads to

$$\frac{\|\eta_n\|_{L^2}^2}{2\|\varepsilon_n\|_{L^2}} \lesssim \frac{A_0^2 \|\varepsilon_n\|_{L^2}}{2\delta x^2} + A_0 \frac{\|\varepsilon_n\|_{L^2}^2}{\delta x^{2+d/2}} + \frac{\|\varepsilon_n\|_{L^2}^3}{2\delta x^{2+d}}. \tag{3.14}$$

Moreover,

$$\|\delta t \mathbb{P}\left[(\varepsilon_n \cdot \nabla)\mathbf{u}_n\right]\|_{L^2} \leq \delta t \|\varepsilon_n\|_{L^2} \|\nabla \mathbf{u}_n\|_{L^\infty} \leq \delta t A_1 \|\varepsilon_n\|_{L^2}. \tag{3.15}$$

Finally,

$$\|\left(Id - \nu\delta t\Delta\right)\varepsilon_{n+1}\|_{L^2} \leq \left(1 + \lambda \frac{A_0^2}{2}\frac{\delta t^2}{\delta x^2} + \lambda A_0 \frac{\|\varepsilon_n\|_{L^2}\delta t^2}{\delta x^{2+d/2}} + \lambda \frac{\|\varepsilon_n\|_{L^2}^2}{2\delta x^{2+d}}\delta t^2 + A_1 \delta t\right) \|\varepsilon_n\|_{L^2}$$

for some λ constant (close to 1). Thanks to a priori assumptions (3.9), $\|\varepsilon_n\|_{L^2} = o(\delta x^{d/2})$, for $d = 2, 3$, since $\frac{d}{2} < 2$. This leads to

$$\|\left(Id - \nu\delta t\Delta\right)\varepsilon_{n+1}\|_{L^2} \leq \left(1 + \left(\lambda \frac{A_0^2}{2}\frac{\delta t}{\delta x^2} + A_1 + o(1)\right)\delta t\right) \|\varepsilon_n\|_{L^2}.$$

Any role played by the discrete heat kernel would be favorable for the L^2-stability since $\|\varepsilon_{n+1}\|_{L^2} \leq \|\left(Id - \nu\delta t\Delta\right)\varepsilon_{n+1}\|_{L^2}$. For a first stability condition we use this inequality (neglecting the role of $(Id - \nu\delta t\Delta)$); then we have the following CFL condition:

$$\frac{\delta t}{\delta x^2} \leq C \tag{3.16}$$

for some $C > 0$ constant. Or, if we want something more *precise*, a good choice should be

$$\lambda \frac{A_0^2}{2}\frac{\delta t}{\delta x^2} = A_1. \tag{3.17}$$

The stability is then guaranteed since, for an initial error $\varepsilon_0 \in L^2$ with $\|\varepsilon_0\|_{L^2} = O(\delta x^2)$ at time 0 (this assumption on $\|\varepsilon_0\|_{L^2}$ comes from the consistency assumptions and from $\delta t \leq C\delta x^2$),

$$\|\varepsilon_{T/\delta t}\|_{L^2} \leq \left(1 + \lambda\left(\frac{A_0^2}{2}\frac{\delta t}{\delta x^2} + A_1\right)\delta t\right)^{T/\delta t} \|\varepsilon_0\|_{L^2} \leq e^{\lambda(\frac{A_0^2}{2}\frac{\delta t}{\delta x^2} + A_1)T} \|\varepsilon_0\|_{L^2} \tag{3.18}$$

with λ a new constant close to 1 which includes the term $o()$.

Under the CFL condition, $e^{\lambda(\frac{A_0^2}{2}\frac{\delta t}{\delta x^2} + A_1)T}$ is bounded by a constant. Then

$$\|\varepsilon_{T/\delta t}\|_{L^2} = O(\delta x^2).$$

Hence, we obtain the a priori estimate (3.9).

Remark 3.3. The stability condition $\delta t \leq C\delta x^2$ should be compared with the usual CFL condition $\delta t \leq C\delta x$ which is weaker and more interesting. In practice, we observe a stability condition $\delta t \leq C\delta x^2$ for the Euler scheme (3.4) which confirms the theoretical CFL condition (3.16).

Remark 3.4. If we just had

$$\varepsilon_{n+1} = \varepsilon_n + \delta t \eta_n$$

with no orthogonality between ε_n and η_n, then the stability would not hold since

$$\|\varepsilon_{n+1}\|_{L^2} \sim \|\varepsilon_n\|_{L^2} + \|\eta_n\|_{L^2}\delta t \sim \left(1 + \frac{A_0}{\delta x}\delta t\right)\|\varepsilon_n\|_{L^2}$$

and then

$$\|\varepsilon_{T/\delta t}\|_{L^2} \sim e^{(\frac{A_0}{\delta x})T}\|\varepsilon_0\|,$$

which goes exponentially to infinity for $\delta x \to 0$.

Now we take the implicit Laplacian into account. Assume that we first apply the implicit Laplacian and then the convection term; then

$$\mathbf{u}_{n+1} = (Id - \nu\delta t\Delta)^{-1}\mathbf{u}_n - \delta t\mathbb{P}\left[((Id - \nu\delta t\Delta)^{-1}\mathbf{u}_n \cdot \nabla)(Id - \nu\delta t\Delta)^{-1}\mathbf{u}_n\right]$$

and so

$$\begin{aligned}\varepsilon_{n+1} = (Id - \nu\delta t\Delta)^{-1}\varepsilon_n - \delta t\mathbb{P}\Big[&((Id - \nu\delta t\Delta)^{-1}\varepsilon_n \cdot \nabla)(Id - \nu\delta t\Delta)^{-1}\mathbf{u}_n \\ &+ ((Id - \nu\delta t\Delta)^{-1}\mathbf{u}_n \cdot \nabla)(Id - \nu\delta t\Delta)^{-1}\varepsilon_n\Big].\end{aligned}$$

We neglect the last term $(\varepsilon_n \cdot \nabla)\varepsilon_n$ in comparison with (3.11) since it does not play any role. As $(Id - \nu\delta t\Delta)^{-1}\varepsilon_n$ is divergence-free, we have the same orthogonalities as in (3.11). We replace the bound for $|\varepsilon_n|_{H^1}$ by

$$\|\nabla(Id - \nu\delta t\Delta)^{-1}\varepsilon_n\|_{L^2} \leq \sup_{\delta x \leq \alpha < +\infty}\left(1 + \frac{\nu\delta t}{\alpha^2}\right)^{-1}\frac{\|\varepsilon_n\|_{L^2}}{\alpha} \leq \frac{\|\varepsilon_n\|_{L^2}}{2\sqrt{\nu\delta t}},$$

where α represents the different computed scales, and with $\frac{1}{2\sqrt{\nu\delta t}}$ the maximal value of $\alpha \mapsto \frac{1}{\alpha}\left(1 + \frac{\nu\delta t}{\alpha^2}\right)^{-1}$. Hence, we have

$$\|\varepsilon_{n+1}\|_{L^2} \leq \left(1 + \left(\frac{A_0^2}{8\nu} + A_1\right)\delta t\right)\|\varepsilon_n\|_{L^2}. \tag{3.19}$$

Hence, we have unconditional stability. But one has to remark that this is interesting only if $\frac{A_0^2}{8\nu} \leq A_1$, that is, for low Reynolds numbers.

Conclusion. The stability conditions of the semi-implicit Euler scheme are summarized:

If

$$A_0^2 \leq 8\nu A_1,$$

then the scheme is unconditionally stable; otherwise the scheme is conditionally stable under the CFL condition

$$\delta t \leq C\delta x^2$$

with $C > 0$ a constant.

3.1.2. An order two time scheme. In the numerical experiments of section 4, we will use a second-order central difference scheme for incompressible flows based on velocity variables. This scheme was presented in [27], where the authors noticed its robustness with no further justifications. In the following, we will prove that a CFL condition can be derived from the divergence-free condition, like for the Euler semi-implicit scheme.

This order two central scheme proceeds with an intermediate step $\mathbf{u}_{n+1/2}$ as follows:

$$\begin{aligned}\left(Id-\nu\frac{\delta t}{2}\Delta\right)\mathbf{u}_{n+1/2}&=\mathbf{u}_n-\frac{\delta t}{2}\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbf{u}_n\right],\\ \left(Id-\nu\frac{\delta t}{2}\Delta\right)\mathbf{u}_{n+1}&=\mathbf{u}_n+\delta t\left(\frac{\nu}{2}\Delta\mathbf{u}_n-\mathbb{P}\left[(\mathbf{u}_{n+1/2}\cdot\nabla)\mathbf{u}_{n+1/2}\right]\right).\end{aligned}\tag{3.20}$$

Like for the Euler semi-implicit scheme, the operators Δ and $\mathbb{P}$ should be denoted by Δ_J and $\mathbb{P}_J$, as they stand for the operators into the discretized wavelet space: $\Delta_J = P_{\mathrm{div},J}\,\Delta$ and $\mathbb{P}_J = P_J\,\mathbb{P}$. But we will keep the notation Δ and $\mathbb{P}$.

In this scheme, the nonlinear term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ and the heat equation are handled in two different manners, both second order in time:

- The central point $\mathbf{u}_{n+1/2}$ (computed by an Euler explicit method for the nonlinear term and an Euler implicit method for the heat equation) is used only for the computation of the nonlinear term contribution $\mathbf{u}_{n+1} = \mathbf{u}_n - \delta t\mathbb{P}\left[(\mathbf{u}_{n+1/2}\cdot\nabla)\mathbf{u}_{n+1/2}\right]$.
- The heat equation is solved by the order two scheme $\left(Id-\nu\frac{\delta t}{2}\Delta\right)\mathbf{u}_{n+1} = \left(Id+\nu\frac{\delta t}{2}\Delta\right)\mathbf{u}_n$.

This treatment of the nonlinear term improves the stability condition. The L^2-stability is guaranteed by the CFL-like condition $\delta t \le C\delta x^{4/3}$.

L^2-stability of the order two central scheme. The problem of L^2-stability of this scheme is addressed in the book by Wesseling [40]. Nevertheless, the techniques used in this book rely on a linearization followed by a Fourier transform. Here we will use a different technique which is slightly simpler: we show that, under some condition on the time step, a small perturbation is not amplified by the numerical scheme.

In the following computations, we do not take into account the diffusion term.

LEMMA 3.4. *Let us assume that* (3.1) *has a C^2 solution $\mathbf{u}(t,\mathbf{x})$. A small error ε_n, with $\|\varepsilon_n\|_{L^2} = O(\delta x^2)$, on $\mathbf{u}_n$—a discrete solution close to $\mathbf{u}(n\delta t,\cdot)$—in the wavelet numerical scheme*

$$\begin{aligned}\mathbf{u}_{n+1/2}&=\mathbf{u}_n-\frac{\delta t}{2}\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbf{u}_n\right],\\ \mathbf{u}_{n+1}&=\mathbf{u}_n-\delta t\mathbb{P}\left[(\mathbf{u}_{n+1/2}\cdot\nabla)\mathbf{u}_{n+1/2}\right]\end{aligned}\tag{3.21}$$

becomes, at step $n+1$, ε_{n+1} whose L^2-norm is bounded as follows:

$$\|\varepsilon_{n+1}\|_{L^2}\le\left(1+\lambda\frac{\delta t^4}{8\delta x^4}A_0^4+\delta t A_1+o()\delta t\right)\|\varepsilon_n\|_{L^2}\tag{3.22}$$

with λ a constant close to 1 and $o()$ a function going to 0 under the condition $\delta t \le C\delta x$.

Proof. We replace $\mathbf{u}_n$ by $\mathbf{u}_n+\varepsilon_n$ in the equations (3.21):

$$\begin{aligned}\mathbf{u}_{n+1/2}+\varepsilon_{n+1/2}=\mathbf{u}_n+\varepsilon_n-\frac{\delta t}{2}\big(&\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbf{u}_n\right]+\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n\right]+\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\\ &+\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\varepsilon_n\right]\big).\end{aligned}$$

Then

$$\mathbf{u}_{n+1}+\varepsilon_{n+1}=\mathbf{u}_n+\varepsilon_n-\delta t\mathbb{P}\left[((\mathbf{u}_{n+1/2}+\varepsilon_{n+1/2})\cdot\nabla)(\mathbf{u}_{n+1/2}+\varepsilon_{n+1/2})\right].$$

Then we gather all of the most important terms and obtain

$$\varepsilon_{n+1}=\varepsilon_n-\delta t\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]+\frac{\delta t^2}{2}\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right]-\delta t\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n\right]+o().$$

Then, on one hand, thanks to Corollary 3.2, we have the following:

- $\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\perp\varepsilon_n$.
- $(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\perp\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]$
  $\implies\ \mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right]\perp\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]$.

And, on the other hand, thanks to Lemma 3.1, as $\operatorname{div}\varepsilon_n=0$,

$$\begin{aligned}\left\langle\varepsilon_n,\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right]\right\rangle&=\left\langle\varepsilon_n,(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right\rangle\\&=-\left\langle(\mathbf{u}_n\cdot\nabla)\varepsilon_n,\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right\rangle\\&=-\left\langle\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right],\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right\rangle\\&=-\|\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\|_{L^2}^2.\end{aligned}\tag{3.23}$$

Then the computation of $\|\varepsilon_{n+1}\|_{L^2}^2$, taking apart $\delta t\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n\right]$, yields

$$\begin{aligned}\|\varepsilon_{n+1}+\delta t\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n\right]\|_{L^2}^2&=\|\varepsilon_n\|_{L^2}^2+\delta t^2\|\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\|_{L^2}^2\\&\quad+\delta t^2\left\langle\varepsilon_n,\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right]\right\rangle\\&\quad+\frac{\delta t^4}{4}\|\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right]\|_{L^2}^2+o()\delta t\\&=\|\varepsilon_n\|_{L^2}^2+\frac{\delta t^4}{4}\|\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\mathbb{P}\left[(\mathbf{u}_n\cdot\nabla)\varepsilon_n\right]\right]\|_{L^2}^2+o()\delta t\\&\le\|\varepsilon_n\|_{L^2}^2+\lambda\frac{\delta t^4}{4}\|\mathbf{u}_n\|_{L^\infty}^4\frac{\|\varepsilon_n\|_{L^2}^2}{\delta x^4}+o()\delta t,\end{aligned}\tag{3.24}$$

where we used $|\varepsilon_n|_{H^2}\le\lambda\frac{\|\varepsilon_n\|_{L^2}}{\delta x^2}$ with λ a constant close to 1 similarly to inequality (3.12).

Taking the square root of the latter expression and isolating ε_{n+1}, we obtain

$$\|\varepsilon_{n+1}\|_{L^2}\le\delta t\|\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n\right]\|_{L^2}+\left(1+\lambda\frac{\delta t^4}{8\delta x^4}A_0^4+o()\delta t\right)\|\varepsilon_n\|_{L^2},$$

which allows us to conclude by applying inequality (3.15) to the term $\|\mathbb{P}\left[(\varepsilon_n\cdot\nabla)\mathbf{u}_n\right]\|_{L^2}$.

THEOREM 3.5. *The central numerical scheme* (3.20) *is stable under the CFL-like condition*

$$\delta t\le C\delta x^{4/3}\tag{3.25}$$

with C a constant.

Proof. A stability condition for the scheme (3.20) is

$$\|\varepsilon_{n+1}\|_{L^2}\le(1+C\delta t)\|\varepsilon_n\|_{L^2}$$

with C a constant. According to (3.22), this means that

$$1+\lambda\frac{\delta t^4}{8\delta x^4}A_0^4+\delta tA_1+o()\delta t\le1+C\delta t$$

for some constant C. This is equivalent to the existence of a constant C' such that

$$\delta t \leq C' \delta x^{4/3}.$$

Remark 3.5. For the order two central scheme with divergence-free wavelets (3.20), we observe numerically the stability condition $\delta t \leq C \delta x^{4/3}$.

Remark 3.6. In a similar fashion, explicit numerical schemes of order $2p$ in time show a CFL-like stability condition $\delta t \leq C \delta x^{(2p+2)/(2p+1)}$, at worst, for the incompressible Euler equation [13].

3.2. Wavelet adaptivity in space. At each step n of the time discretization (semi-implicit Euler scheme (3.4) or order two scheme (3.20)), the solution $\mathbf{u}_n$ and the nonlinear term $\mathbb{P}\left[(\mathbf{u}_n \cdot \nabla)\mathbf{u}_n\right]$ are expressed in terms of divergence-free wavelet coefficients. Moreover, computing the divergence-free wavelet coefficients of the solution at the next time step $\mathbf{u}_{n+1}$ requires only FWTs (Appendix C).

Therefore we will sketch in this section an adaptive method for the computation of the solution $\mathbf{u}$. The adaptivity is based on the wavelet expansion of $\mathbf{u}_n$ (3.3) applied to the numerical scheme (3.20).

At each time step n, we first find a set Λ_n of active wavelet coefficients. The other coefficients are ignored and cancelled. Suppose that the set Λ_n has N elements ($\#(\Lambda_n) = N$); the computed solution $\mathbf{u}_n$ is then replaced by $\mathbf{u}_n^N$:

$$\mathbf{u}_n^N(x) = \sum_{\lambda \in \Lambda_n} u_\lambda^n \ \Psi_\lambda^{\mathrm{div}}(x).$$

To define the set Λ_n, we consider the two wavelet decompositions of the velocity

$$\mathbf{u}_n(x) = \sum_{\lambda} u_\lambda^n \ \Psi_\lambda^{\mathrm{div}}(x)$$

and of the nonlinear term

$$\mathbb{P}\left[(\mathbf{u}_n \cdot \nabla)\mathbf{u}_n\right] = \sum_{\lambda} d_\lambda^n \ \Psi_\lambda^{\mathrm{div}},$$

where the wavelets $\Psi_\lambda^{\mathrm{div}}$ are supposed to be L^2-normalized. Let $\sigma_{0\,n} > 0$ and $\sigma_{1\,n} > 0$ be two threshold parameters. Then the index λ is activated if $|u_\lambda^n| \geq \sigma_{0\,n}$ or $|d_\lambda^n| \geq \sigma_{1\,n}$ in the previous expansions, which leads to

$$\Lambda_n = \{\lambda \in \Lambda \ ; \ |u_\lambda^n| \geq \sigma_{0\,n} \ \text{ or } \ |d_\lambda^n| \geq \sigma_{1\,n}\},$$

where the set Λ of indexes depends on the type of decomposition we use. For anisotropic wavelets, $\lambda = (i, \mathbf{j}, \mathbf{k}) \in [1, \ldots, d-1] \times (\mathbb{Z}^d)^2$, while for isotropic wavelets, $\lambda = (i, \varepsilon, j, \mathbf{k}) \in [1, \ldots, d-1] \times \{0,1\}^{d*} \times \mathbb{Z} \times \mathbb{Z}^d$ [16].

The first threshold $\sigma_{0\,n}$ corresponds to the nonlinear approximation of $\mathbf{u}_n$ provided by the N-best term approximation of the wavelet transform [9]: it allows us to store valuable information on $\mathbf{u}_n$. The second threshold $\sigma_{1\,n}$ takes into account the future changes in the wavelet decomposition of the velocity.

This thresholding procedure is applied at each time step of the divergence-free wavelet scheme, and corresponding numerical tests are presented in section 4.2.

Unfortunately, we will see in numerical experiments that anisotropic wavelets are not well suited for adaptivity, because of their supports, which can be highly elongated, mixing large scales in one direction with small scales in the other one.

We will remedy this problem by using "generalized" divergence-free wavelets, which are a mix between isotropic divergence-free wavelets and anisotropic divergence-free wavelets. These wavelets were first introduced in [11] and extensively detailed in [16].

Below we explain their construction in dimension 2.

Let $m \in \mathbb{N}$. The "generalized" divergence-free wavelets are composed of the vector wavelets

- $\Psi^{\mathrm{div}\,(1,1)}_{\mathbf{j},\mathbf{k}} = \left| \begin{array}{l} 2^{j_2}\psi_1(2^{j_1}x_1-k_1)\psi_0(2^{j_2}x_2-k_2) \\ -2^{j_1}\psi_0(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2) \end{array} \right.$ with $|j_1-j_2| \le m$,
- $\Psi^{\mathrm{div}\,(1,0)}_{\mathbf{j},\mathbf{k}} = \left| \begin{array}{l} -2^{j_2-2}\psi_1(2^{j_1}x_1-k_1)(\varphi_0(2^{j_2}x_2-k_2)-\varphi_0(2^{j_2}x_2-k_2-1)) \\ 2^{j_1}\psi_0(2^{j_1}x_1-k_1)\varphi_1(2^{j_2}x_2-k_2) \end{array} \right.$ with $j_2 = j_1 - m$,
- $\Psi^{\mathrm{div}\,(0,1)}_{\mathbf{j},\mathbf{k}} = \left| \begin{array}{l} 2^{j_2}\varphi_1(2^{j_1}x_1-k_1)\psi_0(2^{j_2}x_2-k_2) \\ -2^{j_1-2}(\varphi_0(2^{j_1}x_1-k_1)-\varphi_0(2^{j_1}x_1-k_1-1))\psi_1(2^{j_2}x_2-k_2) \end{array} \right.$ with $j_1 = j_2 - m$,

where $\psi_1' = 4\psi_0$ and $\varphi_1' = \varphi_0(\,.\,) - \varphi_0(\,.\,-1)$.

The corresponding complement functions used for the divergence-free wavelet transform are

- $\Psi^{\mathrm{n}\,(1,1)}_{\mathbf{j},\mathbf{k}} = \left| \begin{array}{l} 2^{j_1}\psi_1(2^{j_1}x_1-k_1)\psi_0(2^{j_2}x_2-k_2) \\ 2^{j_2}\psi_0(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2) \end{array} \right.$ with $|j_1-j_2| \le m$,
- $\Psi^{\mathrm{n}\,(1,0)}_{\mathbf{j},\mathbf{k}} = \left| \begin{array}{l} \psi_1(2^{j_1}x_1-k_1)\varphi_0(2^{j_2}x_2-k_2) \\ 0 \end{array} \right.$ with $j_2 = j_1 - m$,
- $\Psi^{\mathrm{n}\,(0,1)}_{\mathbf{j},\mathbf{k}} = \left| \begin{array}{l} 0 \\ \varphi_0(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2) \end{array} \right.$ with $j_1 = j_2 - m$.

For $m = 0$, these wavelets are the usual isotropic divergence-free wavelets introduced by Lemarié [28].

Contrary to anisotropic wavelets, the supports of these wavelets do not lengthen and allow one to refine the grid locally. Notice also that with these "generalized" divergence-free wavelets the algorithms of sections 2.1.2 and 2.2 converge as well, with satisfactory convergence rates (except for the isotropic wavelets $m = 0$). The particular case $m = 1$ gives quasi-isotropic functions, with a good behavior for the convergence of the wavelet Helmholtz decomposition.

4. Numerical experiments. The experiment on which we apply our method is the three vortex interaction. This experiment was originally designed by Farge and Kevlahan and is often used to test new numerical methods [34, 6, 22]. In order to provide a reference solution, the experiment of [6] was first reproduced by using a pseudospectral method, solving the Navier–Stokes equations in *velocity-pressure* formulation.

The initial state is displayed in Figure 3, left. In the periodic box $[0,1]^2$, three vortices with a Gaussian vorticity profile $\omega_i(\mathbf{x}) = A_i\pi e^{-4\pi^4((x_1-\alpha_i)^2+(x_2-\beta_i)^2)}$ are present:

– one is centered at $(\alpha_1,\beta_1) = (3/8, 1/2)$ with amplitude $A_1 = 1$;

– one is centered at $(\alpha_2,\beta_2) = (5/8, 1/2)$ also with amplitude $A_2 = 1$;

– and one is centered at $(\alpha_3,\beta_3) = (5/8, 1/2+\sqrt{2}/8)$ with amplitude $A_3 = -1/2$.

The negative vortex is here to force the merging of the two positive ones. The time step is $\delta t = 10^{-2}$ and the viscosity $\nu = 5.10^{-5}$. The solution is computed on a 512×512 grid.

The vorticity fields at times $t = 0, 10, 20$, and 40 are displayed in Figure 3. The second row of Figure 3 displays the absolute values of the isotropic divergence-

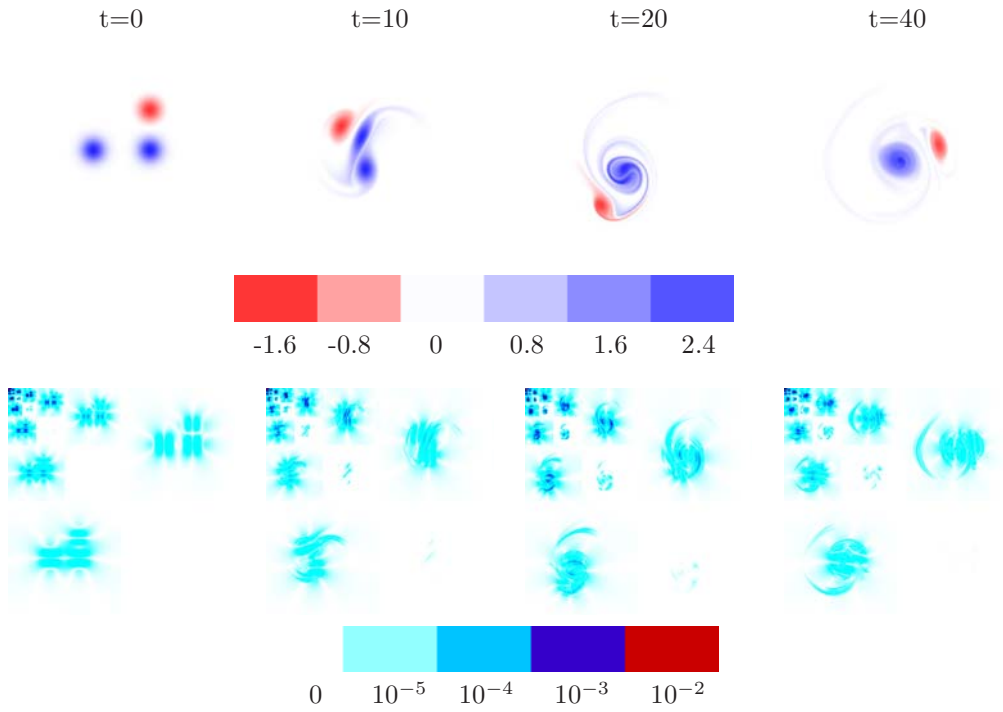

FIG. 3. *Vorticity fields at times* $t = 0, 10, 20,$ *and* 40*, and corresponding isotropic divergence-free wavelet coefficients of the velocity fields, for the reference solution, given by a pseudospectral method on a* 512^2 *grid.*

free wavelet coefficients of the *velocity field* at corresponding times, with a L^∞-normalization. This time evolution of the wavelet modes can be compared to the time evolution of the wavelet coefficients of the *vorticity field* presented in [34], with isotropic orthogonal scalar wavelets: divergence-free wavelet coefficients are concentrated on zones with high energy (around structures and in strain zones), whereas wavelet coefficients of the vorticity fields are concentrated on zones with high enstrophy (inside vortices and in strain zones). In both cases, the wavelet maps give a good illustration of the sparsity provided by the wavelet decomposition of flows containing coherent structures.

4.1. Full divergence-free wavelet code. We show in this part the results of the three vortex interaction provided by the anisotropic divergence-free wavelet method described in section 3.1.2. To construct divergence-free wavelets, we have used the spline wavelets ψ_0 of order 2 and ψ_1 of order 3, represented in Figure 4.

The velocity field was computed on a 256^2 grid with $\delta t = 0.02$ and $\nu = 5.10^{-5}$. For this simulation, we do not use any threshold on the wavelet coefficients. At each time step, we compute the wavelet Helmholtz decomposition of $(\mathbf{u}_n \cdot \nabla)\mathbf{u}_n$ with 7 iterations of the algorithm of section 2.1.2, taking the result of the previous step as the initial guess. The nonlinear term is evaluated by collocation in physical space. We also solve the discrete heat equation with 3 iterations of the method described in section 2.2. These (fixed in time) numbers of iterations have been chosen to provide a sufficient accuracy.

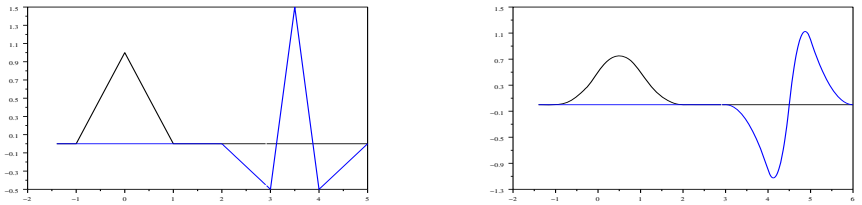

FIG. 4. *Scaling spline functions and associated spline wavelets of order* 2 *and* 3 *related by differentiation:* $\psi_1' = 4\psi_0$.

The time evolution of the vorticity field (reconstructed from the computed velocity field) is very similar to Figure 3. Hence, the results are close to the reference solution. Notice that this code uses only wavelet transforms.

We also observe the time evolution of active divergence-free wavelet modes. Figure 5 represents the evolution in time of the ratio of wavelet coefficients (in L^2-normalization) above some fixed threshold. Let $\varepsilon_0 = \sup(|u_\lambda^0|)$ $(= 0.066062)$ be the maximal value of the divergence-free wavelet coefficients of the initial velocity field. We introduce 7 values for the threshold parameter: $\varepsilon_i = \varepsilon_0/4^i$ $(1 \le i \le 7)$, and we plot the ration of wavelet coefficients of $\mathbf{u}_n$ verifying $|u_\lambda^n| > \varepsilon_i$. The lowest threshold represented is $\varepsilon_0/16,384$.

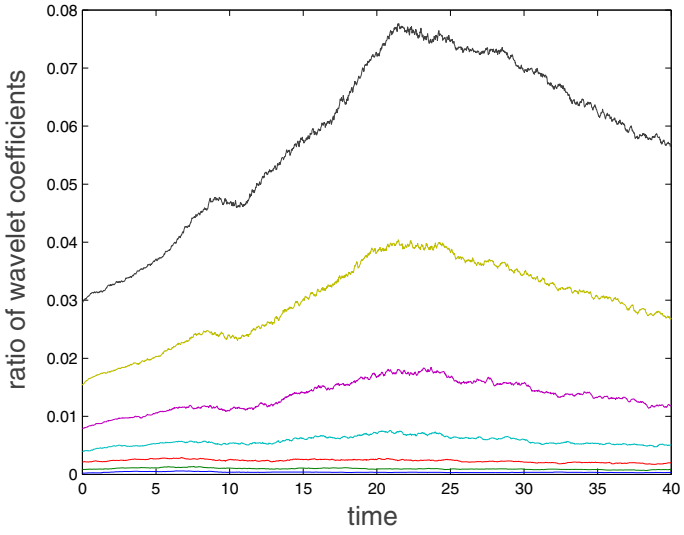

FIG. 5. *Time evolution of the ratio—there are* 256^2 *of them in total—of anisotropic divergence-free wavelet coefficients of the solution* $\mathbf{u}_n$ *above thresholds equal to* $\varepsilon_i = \varepsilon_0/4^i$ *for* $1 \le i \le 7$ *(each curve corresponds to a different threshold* $\varepsilon_i \in \{16.5E-3,\ 4.13E-3,\ 1.03E-3,\ 258E-6,\ 64.5E-6,\ 16.1E-6,\ 4.03E-6\}$*).*

One can observe that, for this experiment, the complexity of the flow structure increases until reaching a maximum at time $t = 22$ and then decreases slowly.

4.2. Towards an adaptive divergence-free wavelet code, using anisotropic wavelets. We investigate in this section how the numerical wavelet scheme behaves with thresholding the wavelet coefficients. As the implementation of a fully adaptive wavelet scheme is heavy (and still in progress), we just compute the evolution of the solution by filtering, at each time step, the wavelet coefficients of the velocity and of the nonlinear term. In practice, at each time step of the full anisotropic divergence-free wavelet code, we eliminate the wavelet coefficients below the thresholds σ_0 (for the velocity coefficients) and σ_1 for the divergence-free part of the nonlinear term, as explained in section 3.2. The thresholds σ_0 and σ_1 are chosen according to empirical criteria.

Figure 6 displays the time evolution of the wavelet coefficients above the thresholds for the velocity, for the nonlinear term, for the intersection of these two sets, and for their union, this last one representing the number of active modes.

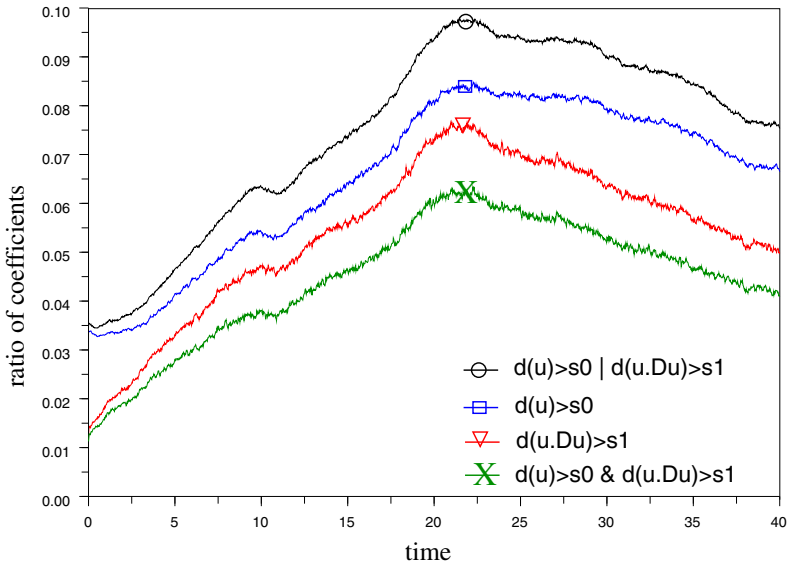

FIG. 6. *Time evolution of the ratio of anisotropic wavelet coefficients above the thresholds: Starting from the bottom, the second lowest curve represents coefficients of the nonlinear term above* $\sigma_1 = 32E - 6$*, the third lowest curve those of the velocity* $\mathbf{u}_n$ *above* $\sigma_0 = 6E - 6$*, and the lowest curve the intersection of these two sets, and the uppermost curve is the union of these two sets.*

The solution obtained presents no perceptible differences when compared with Figures 3.

Figure 7 displays the L^2 relative error between this numerical solution and the reference solution of Figure 3. This error is compared with the error obtained for a pseudospectral code with the same number of grid points (256^2). One can notice that neither the use of wavelets nor the thresholding destroys the accuracy of the solution.

For higher thresholdings, the *effects of the anisotropy* begin to be visible. These effects are already clearly present in Figure 8, where both thresholds σ_0 and σ_1 were multiplied by 3, but without destroying the solution. And the evolution of the thresholding gives satisfactory results: the maximum ratio of active coefficients goes to 6% instead of 10%, and the evolution of the curve follows the complexity of the

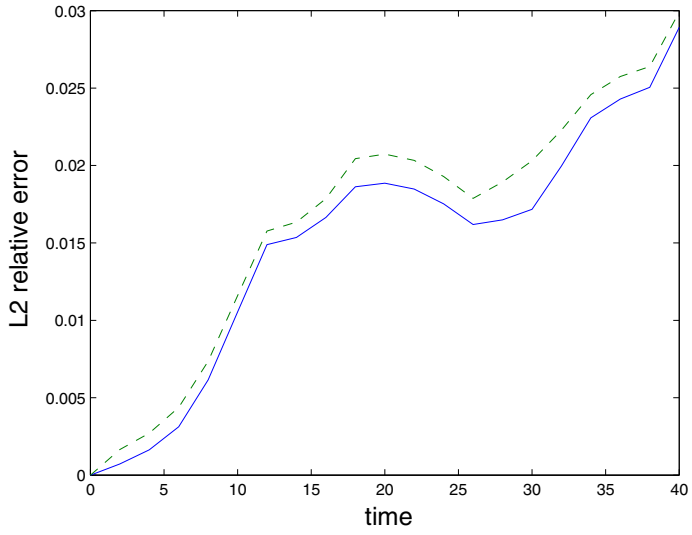

FIG. 7. *L^2 relative errors for the pseudospectral code (lower curve) and the filtered wavelet code (upper curve), corresponding to Figure* 6*, on a* 256^2 *grid, compared with the reference pseudospectral solution.*

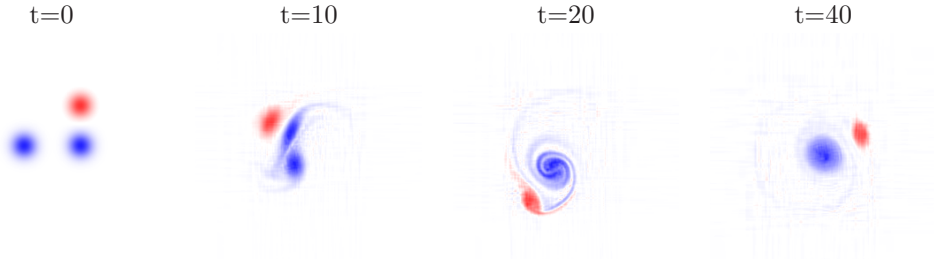

FIG. 8. *Time evolution of the vorticity, reconstructed from the velocity provided by the filtered wavelet code (*256^2 *grid, anisotropic divergence-free wavelets).*

flow.

On the contrary, we see in Figure 9 that if we take a higher threshold with a maximum of 4.5% of the coefficients, then it leads to a nonadmissible final result.

4.3. Filtered divergence-free wavelet code, using "generalized" wavelets. We have remedied the problem of anisotropy induced by thresholding by using "generalized" divergence-free wavelets which are a mix between isotropic divergence-free wavelets and anisotropic divergence-free wavelets. These wavelets have been defined in section 3.2. We choose to use quasi-isotropic wavelets, corresponding to the parameter $m = 1$, which are a good compromise between isotropy and convergence of the Helmholtz algorithm of section 2.1.2.

We have tested these wavelets with the experiment of the three vortex interaction. We display the results in Figures 10 and 11.

Even with a rather high threshold (a maximum of only 5% of the wavelet coefficients is needed), the quality of the solution remains reasonably good. And the

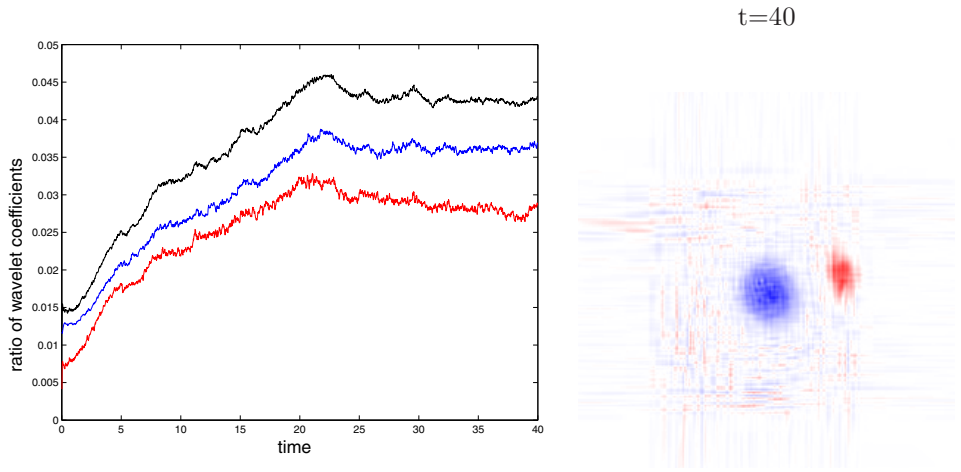

FIG. 9. *Ratio of active coefficients for a high thresholding (on the left): The lowest curve represents coefficients of the nonlinear term above* $\sigma_1 = 160E-6$*, the middle curve represents those of the velocity* $\mathbf{u}_n$ *above* $\sigma_0 = 30E-6$*, and the upper curve is the union of these two. The final result at* $t = 40$ *(on the right), on a* 256^2 *grid, uses anisotropic divergence-free wavelets.*

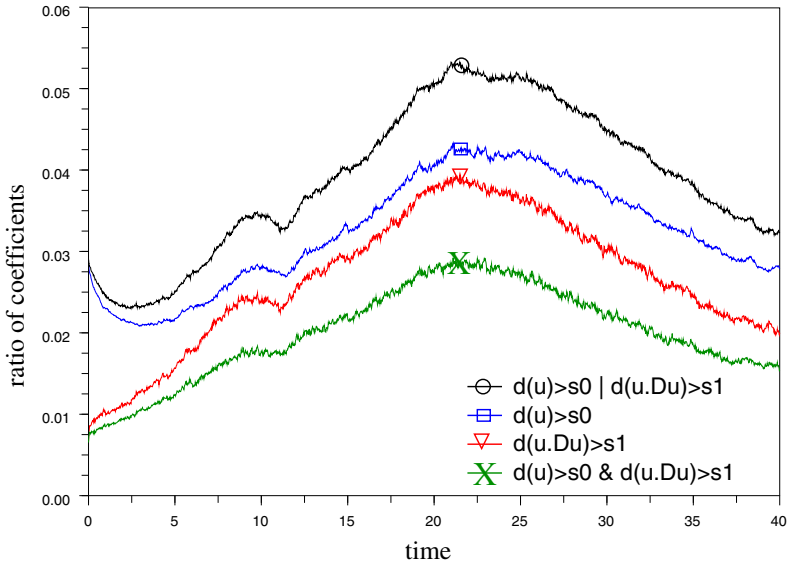

FIG. 10. *Ratio of activated wavelet coefficients in the case of "generalized" divergence-free wavelets on a* 256^2 *grid with thresholds* $\sigma_1 = 20E-6$ *and* $\sigma_0 = 5E-6$*.*

thresholding evolves in a satisfying way. We do not observe the appearance of lines on the whole domain as in the case for anisotropic wavelets in Figure 9. Even when we compare these results with those in Figure 8, we notice an improvement.

Conclusion. We derived in this paper a new numerical divergence-free wavelet scheme for solving Navier–Stokes equations. The proposed method relies only on

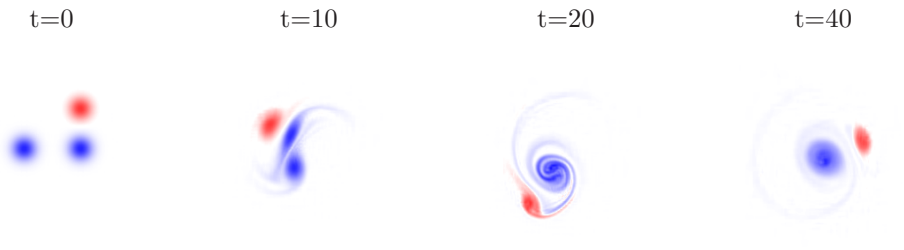

FIG. 11. *Time evolution of the vorticity, reconstructed from the velocity coefficients on "generalized" divergence-free wavelets, with thresholding corresponding to Figure* 10*, on a* 256^2 *grid.*

FWTs and is perfectly fitted for adaptivity. Nevertheless, work should be done to optimize this in practice and take more advantage of wavelets.

Until now, the use of divergence-free wavelets was limited to linear problems such as the Stokes problem [36] or the equations of electromagnetism [38]. This limitation was due to the nonexistence of divergence-free wavelet algorithms dealing with nonlinear terms, like $(\mathbf{u}\cdot\nabla)\mathbf{u}$ for the Navier–Stokes equations. The investigation of anisotropic divergence-free wavelets and, more specifically, the invention of "generalized" divergence-free wavelets [11] enable such an algorithm (see section 2.1.2 or [16]).

The numerical stability of this numerical scheme is fulfilled under a CFL condition and is mainly due to the use of divergence-free wavelets that allow one to verify exactly the divergence-free condition ($\operatorname{div}\mathbf{u}=0$).

Extensive numerical tests on the experiment of three vortex interaction were presented. The results provided by the new divergence-free wavelet method can be compared to those obtained in [22]. In [22], Griebel and Koster use anisotropic interpolating wavelets and a Poisson solver for the same experiment (three vortex interaction), the equations being written in velocity-vorticity formulation. With 10,000 degrees of freedom, their results are of lesser quality (ibid. "overestimation of the rotation of the cores of the vortices") than the ones we obtain with 3,500 degrees of freedom. The computational time for the full wavelet code is about four times the Fourier code in the periodic case for which the Fourier spectral method is known to be nearly optimal. But the interest of such wavelet method is that it can be extended to other boundary conditions, such as Dirichlet or Neumann boundary conditions, using wavelets on the interval satisfying homogeneous boundary conditions (see, for instance, [33]). To deal with complex geometries, wavelet methods can be incorporated into a fictitious domain approach, and, in this context, adaptive codes are also available [2]. Last, while only results in dimension 2 are presented in this paper, our divergence-free wavelet method extends directly to dimension 3.

Appendix A. The "generalized" wavelet transform in two dimensions. The "generalized" wavelets make a compromise between isotropic wavelets and anisotropic wavelets as indicated in section 3.2. In the periodic case ($\mathbb{T}^2$), for $m\geq 0$, the "generalized" wavelet transform in the MRA $(\psi_1\otimes\psi_2)$ is given by the following operations:

$$f(x_1,x_2)=\sum_{k_1,k_2=0}^{2^J-1} c_{J\,\mathbf{k}}\varphi_1(2^Jx_1-k_1)\varphi_2(2^Jx_2-k_2).$$

The first step consists in performing an isotropic transform:

$$
\begin{aligned}
f(x_1,x_2) = &\sum_{j=0}^{J-1}\sum_{k_1,k_2=0}^{2^j-1} d_{j\,\mathbf{k}}^{(11)}\psi_1(2^jx_1-k_1)\psi_2(2^jx_2-k_2)\\
&+ d_{j\,\mathbf{k}}^{(10)}\psi_1(2^jx_1-k_1)\varphi_2(2^jx_2-k_2)\\
&+ d_{j\,\mathbf{k}}^{(01)}\varphi_1(2^jx_1-k_1)\psi_2(2^jx_2-k_2). \qquad \text{(A.1)}
\end{aligned}
$$

Some additional wavelet transforms for $d_{j\,\mathbf{k}}^{(10)}$ in the x_2-direction and for $d_{j\,\mathbf{k}}^{(01)}$ in the x_1-direction yield the "generalized" wavelet transform:

$$
\begin{aligned}
f(x_1,x_2) \;=\; &\sum_{j=0}^{J-1}\Bigg(\sum_{k_1,k_2=0}^{2^j-1} d_{(j,j)\,\mathbf{k}}^{(11)}\psi_1(2^jx_1-k_1)\psi_2(2^jx_2-k_2)\\
&+\sum_{\ell_2=1}^{m}\sum_{k_1=0}^{2^j-1}\sum_{k_2=0}^{2^{j-\ell_2}-1} d_{(j,j-\ell_2)\,\mathbf{k}}^{(11)}\psi_1(2^jx_1-k_1)\psi_2(2^{j-\ell_2}x_2-k_2)\\
&+\sum_{\ell_1=1}^{m}\sum_{k_1=0}^{2^{j-\ell_1}-1}\sum_{k_2=0}^{2^j-1} d_{(j-\ell_1,j)\,\mathbf{k}}^{(11)}\psi_1(2^{j-\ell_1}x_1-k_1)\psi_2(2^jx_2-k_2)\\
&+\sum_{k_1=0}^{2^j-1}\sum_{k_2=0}^{2^{j-m}-1} d_{(j,j-m)\,\mathbf{k}}^{(10)}\psi_1(2^jx_1-k_1)\varphi_2(2^{j-m}x_2-k_2)\\
&+\sum_{k_1=0}^{2^{j-m}-1}\sum_{k_2=0}^{2^j-1} d_{(j-m,j)\,\mathbf{k}}^{(01)}\varphi_1(2^{j-m}x_1-k_1)\psi_2(2^jx_2-k_2)\Bigg).
\end{aligned}
$$

If $m=+\infty$, we have the anisotropic transform:

$$
f(x_1,x_2)=\sum_{j_1,j_2=0}^{J-1}\sum_{k_1=0}^{2^{j_1}-1}\sum_{k_2=0}^{2^{j_2}-1} d_{\mathbf{j},\mathbf{k}}\psi_1(2^{j_1}x_1-k_1)\psi_2(2^{j_2}x_2-k_2).
$$

In all of the previous equations we noted that $\psi_{i\,0\,0}=\varphi_{i\,0\,0}=1$ on $\mathbb{T}^2$. These transforms are schematized in Figure 12.

Appendix B. Divergence-free wavelet transform in the "generalized" case in two dimensions. In the following the wavelets ψ_1 and ψ_0 are linked by derivation: $\psi_1' = 4\,\psi_0$. First, we apply the "generalized" wavelet transform as defined in Appendix A to a vector function $\mathbf{u}$ on $\mathbb{T}^2$:

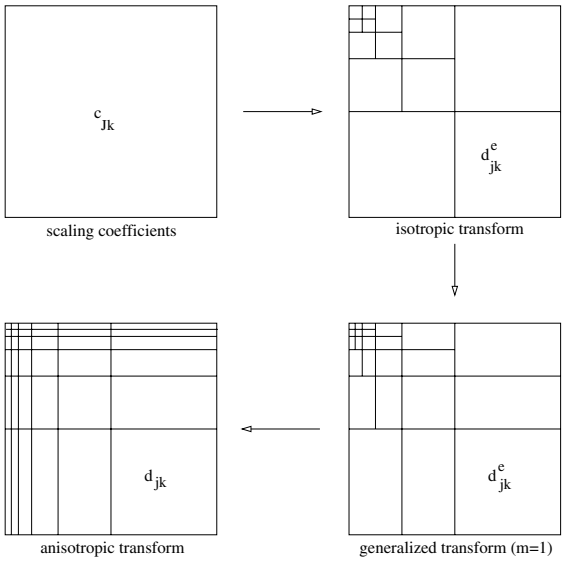

FIG. 12. *Coefficient repartition for the isotropic, anisotropic, and "generalized" 2D wavelet transforms.*

$$
\mathbf{u} = \left| \begin{array}{rcl}
u_1 & = & \displaystyle\sum_{0\le j_1,j_2\le J-1,\,|j_1-j_2|\le m} \sum_{k_1=0}^{2^{j_1}-1} \sum_{k_2=0}^{2^{j_2}-1} d^{(11)}_{1\,\mathbf{j},\mathbf{k}} \psi_1(2^{j_1}x_1-k_1)\psi_0(2^{j_2}x_2-k_2) \\
& & + \displaystyle\sum_{j_1=m}^{J-1} \sum_{k_1=0}^{2^{j_1}-1} \sum_{k_2=0}^{2^{j_1-m}-1} d^{(10)}_{1\,(j_1,j_1-m)\mathbf{k}} \psi_1(2^{j_1}x_1-k_1)\varphi_0(2^{j_1-m}x_2-k_2) \\
& & + \displaystyle\sum_{j_2=m}^{J-1} \sum_{k_1=0}^{2^{j_2-m}-1} \sum_{k_2=0}^{2^{j_2}-1} d^{(01)}_{1\,(j_2-m,j_2)\mathbf{k}} \varphi_1(2^{j_2-m}x_1-k_1)\psi_0(2^{j_2}x_2-k_2) \\
u_2 & = & \displaystyle\sum_{0\le j_1,j_2\le J-1,\,|j_1-j_2|\le m} \sum_{k_1=0}^{2^{j_1}-1} \sum_{k_2=0}^{2^{j_2}-1} d^{(11)}_{2\,\mathbf{j},\mathbf{k}} \psi_0(2^{j_1}x_1-k_1)\psi_1(2^{j_2}x_2-k_2) \\
& & + \displaystyle\sum_{j_1=m}^{J-1} \sum_{k_1=0}^{2^{j_1}-1} \sum_{k_2=0}^{2^{j_1-m}-1} d^{(10)}_{2\,(j_1,j_1-m)\mathbf{k}} \psi_0(2^{j_1}x_1-k_1)\varphi_1(2^{j_1-m}x_2-k_2) \\
& & + \displaystyle\sum_{j_2=m}^{J-1} \sum_{k_1=0}^{2^{j_2-m}-1} \sum_{k_2=0}^{2^{j_2}-1} d^{(01)}_{2\,(j_2-m,j_2)\mathbf{k}} \varphi_0(2^{j_2-m}x_1-k_1)\psi_1(2^{j_2}x_2-k_2).
\end{array} \right.
$$

Then we obtain the decomposition

$$
\mathbf{u} = \sum_{\varepsilon,\mathbf{j},\mathbf{k}} d^{\mathrm{div}\,\varepsilon}_{\mathbf{j},\mathbf{k}} \Psi^{\mathrm{div}\,\varepsilon}_{\mathbf{j},\mathbf{k}} + d^{\mathrm{n}\,\varepsilon}_{\mathbf{j},\mathbf{k}} \Psi^{\mathrm{n}\,\varepsilon}_{\mathbf{j},\mathbf{k}}
$$

with the wavelets $\Psi^{\mathrm{div}\,\varepsilon}$ and $\Psi^{\mathrm{n}\,\varepsilon}$ introduced in section 3.2. The wavelet coefficients $d^{\mathrm{div}\,\varepsilon}_{\mathbf{j},\mathbf{k}}$ and $d^{\mathrm{n}\,\varepsilon}_{\mathbf{j},\mathbf{k}}$ are given by

$$\begin{cases} d^{\mathrm{div}\,(11)}_{\mathbf{j},\mathbf{k}} = \frac{2^{j_2} d^{(11)}_{1\,\mathbf{j},\mathbf{k}} - 2^{j_1} d^{(11)}_{2\,\mathbf{j},\mathbf{k}}}{2^{2j_1}+2^{2j_2}}, \\ d^{\mathrm{n}\,(11)}_{\mathbf{j},\mathbf{k}} = \frac{2^{j_1} d^{(11)}_{1\,\mathbf{j},\mathbf{k}} + 2^{j_2} d^{(11)}_{2\,\mathbf{j},\mathbf{k}}}{2^{2j_1}+2^{2j_2}}, \end{cases}$$

$$\begin{cases} d^{\mathrm{div}\,(10)}_{\mathbf{j},\mathbf{k}} = 2^{-j_1} d^{(10)}_{2\,\mathbf{j},\mathbf{k}}, \\ d^{\mathrm{n}\,(11)}_{\mathbf{j},\mathbf{k}} = d^{(10)}_{1\,\mathbf{j},\mathbf{k}} + 2^{-m-2} d^{(10)}_{2\,\mathbf{j},\mathbf{k}} - 2^{-m-2} d^{(10)}_{2\,\mathbf{j},(k_1,k_2-1)}, \end{cases}$$

and

$$\begin{cases} d^{\mathrm{div}\,(01)}_{\mathbf{j},\mathbf{k}} = 2^{-j_2} d^{(01)}_{1\,\mathbf{j},\mathbf{k}}, \\ d^{\mathrm{n}\,(01)}_{\mathbf{j},\mathbf{k}} = d^{(01)}_{2\,\mathbf{j},\mathbf{k}} + 2^{-m-2} d^{(01)}_{1\,\mathbf{j},\mathbf{k}} - 2^{-m-2} d^{(01)}_{1\,\mathbf{j},(k_1-1,k_2)}. \end{cases}$$

Appendix C. Order two Navier–Stokes numerical scheme. We present the pseudocode corresponding to section 3.1.2 in dimension 2.

Subroutines.

- fast divergence-free wavelet transform: $d^{\mathrm{div}} = \mathrm{FWT}_{\mathrm{div}}(c)$, where c = coefficients of the discretization of a vector function $\mathbf{u}$ on the scaling functions $(\varphi_1\varphi_0, \varphi_0\varphi_1)$;
- inverse fast divergence-free wavelet transform: $c = \mathrm{IFWT}_{\mathrm{div}}(d^{\mathrm{div}})$;
- wavelet Helmholtz decomposition: $d^{\mathrm{div}} = \mathrm{WHD}(\mathbf{u}, d^{\mathrm{div}}_0, It)$, where $\mathbf{u}$ = vector function to decompose: $\mathbf{u} = \mathbf{u}_{\mathrm{div}} + \mathbf{u}_{\mathrm{curl}}$, $\mathbf{u}_{\mathrm{div}} = \sum d^{\mathrm{div}} \Psi^{\mathrm{div}}$, d^{div}_0 = initial guess for d^{div}, It = number of iterations;
- implicit heat kernel integrator: $d^{\mathrm{div}}_1 = \mathrm{IHKI}(d^{\mathrm{div}}_0, \alpha, It)$, which solves $(Id - \alpha\Delta)\mathbf{u} = \mathbf{v}$ with given $\mathbf{v} = \sum d^{\mathrm{div}}_0 \Psi^{\mathrm{div}}$ and unknown $\mathbf{u} = \sum d^{\mathrm{div}}_1 \Psi^{\mathrm{div}}$.

Navier–Stokes solver order two.

Loop on time $t_n = n\delta t$

1. $(\mathtt{c_n}) = \mathrm{IFWT}_{\mathrm{div}}(\mathtt{d^{div}_n})$ (computation of $\mathbf{u}_\mathtt{n}$)
2. $\mathtt{ugu_n} = (\mathbf{u}_\mathtt{n} \cdot \nabla)\mathbf{u}_\mathtt{n}$
3. $\mathtt{d^{div}_{ugu_n}} = \mathrm{WHD}(\mathtt{ugu_n}, \mathtt{d^{div}_{ugu_{n-1/2}}}, \mathtt{It_1})$
4. $\mathtt{d^{div}_{n+1/2}} = \mathrm{IHKI}(\mathtt{d^{div}_n} - \frac{\delta \mathtt{t}}{2}\mathtt{d^{div}_{ugu_n}}, \frac{\nu\delta\mathtt{t}}{2}, \mathtt{It_2})$
5. $(\mathtt{c_{n+1/2}}) = \mathrm{IFWT}_{\mathrm{div}}(\mathtt{d^{div}_{n+1/2}})$ (computation of $\mathbf{u}_{\mathtt{n}+1/2}$)
6. $\mathtt{ugu_{n+1/2}} = (\mathbf{u}_{\mathtt{n}+1/2} \cdot \nabla)\mathbf{u}_{\mathtt{n}+1/2}$
7. $\mathtt{d^{div}_{ugu_{n+1/2}}} = \mathrm{WHD}(\mathtt{ugu_{n+1/2}}, \mathtt{d^{div}_{ugu_n}}, \mathtt{It_1})$
8. $(\mathtt{c_{\Delta u_n}}) =$ approximation of $\mathtt{\Delta u_n}$ in the MRA $(\varphi_1\varphi_0, \varphi_0\varphi_1)$
9. $(\mathtt{d^{div}_{\Delta u_n}}) = \mathrm{FWT}_{\mathrm{div}}(\mathtt{c_{\Delta u_n}})$
10. $\mathtt{d^{div}_{n+1}} = \mathrm{IHKI}(\mathtt{d^{div}_n} + \delta\mathtt{t}(-\mathtt{d^{div}_{ugu_{n+1/2}}} + \frac{\nu}{2}\mathtt{d^{div}_{\Delta u_n}}), \frac{\nu\delta\mathtt{t}}{2}, \mathtt{It_2})$

End of the loop

Acknowledgments. The authors gratefully acknowledge the University of Ulm, and especially the Numerical Analysis Department, where most of the numerical

experiments presented in this paper were conducted. The authors also wish to express their gratitude to Kai Bittner, Nicholas Kevlahan, and Karsten Urban for fruitful discussions.

REFERENCES

[1] C.-M. Albukrek, K. Urban, D. Rempfer, and J.-L. Lumley, *Divergence-free wavelet analysis of turbulent flows*, J. Sci. Comput., 17 (2002), pp. 49–66.
[2] J. Baccou and J. Liandrat, *Definition and analysis of a wavelet/fictitious domain solver for the 2D-heat equation*, Math. Models Methods Appl. Sci., 16 (2006), pp. 819–845.
[3] C. Bernardi and Y. Maday, *Approximations spectrales de problèmes aux limites elliptiques*, Math. Appl. (Berlin) 10, Springer-Verlag, Paris, 1992.
[4] S. Bertoluzza and R. Masson, *Velocity-pressure adaptative wavelet spaces satisfying the Inf-Sup condition*, C.R. Acad. Sci. Paris Sér. I Math., 323 (1996), pp. 407–412.
[5] C. Canuto, M. T. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.
[6] P. Charton and V. Perrier, *A pseudo-wavelet scheme for the two-dimensional Navier-Stokes equations*, Comput. Appl. Math., 15 (1996), pp. 139–160.
[7] A. J. Chorin and J. E. Marsden, *A Mathematical Introduction to Fluid Mechanics*, 3rd ed., Springer-Verlag, New York, 1993.
[8] A. Cohen, I. Daubechies, and J. C. Feauveau, *Biorthogonal bases of compactly supported wavelets*, Comm. Pure Appl. Math., 45 (1992), pp. 485–560.
[9] A. Cohen, *Wavelet methods in numerical analysis*, in Handbook of Numerical Analysis, Vol. VII, P. G. Ciarlet and J. L. Lions, eds., Elsevier, Amsterdam, 2000, pp. 417–711.
[10] W. Dahmen, A. Kunoth, and K. Urban, *A wavelet-Galerkin method for the Stokes problem*, Computing, 56 (1996), pp. 259–302.
[11] E. Deriaz, *Ondelettes pour la Simulation des Écoulements Fluides Incompressibles en Turbulence*, Ph.D. thesis, INP Grenoble, Grenoble, France, 2006.
[12] E. Deriaz, *Shannon Wavelet Approximations of Linear Differential Operators*, Preprint IMPAN 676, http://www.impan.gov.pl/EN/Preprints/index.html (2007).
[13] E. Deriaz, *L^2-Stability of Explicit Schemes for Incompressible Euler Equations*, preprint, http://arxiv.org/abs/0712.2328 (2007).
[14] E. Deriaz and V. Perrier, *Divergence-free wavelets in 2D and 3D, application to the Navier-Stokes equations*, J. Turbulence, 7 (2006), pp. 1–37.
[15] E. Deriaz, K. Bittner, and V. Perrier, *Décomposition de Helmholtz par ondelettes: Convergence d'un algorithme itératif*, ESAIM Proc., 18 (2007), pp. 23–37.
[16] E. Deriaz and V. Perrier, *Orthogonal Helmholtz decomposition in arbitrary dimension using divergence-free and curl-free wavelets*, Appl. Comput. Harmon. Anal., submitted.
[17] M. Farge, *Wavelet transforms and their applications to turbulence*, in Annual Review of Fluid Mechanics, Annu. Rev. Fluid Mech. 24, Annual Reviews, Palo Alto, CA, 1992, pp. 395–457.
[18] M. Farge, N. Kevlahan, V. Perrier, and E. Goirand, *Wavelets and turbulence*, Proc. IEEE, 84 (1996), pp. 639–669.
[19] M. Farge and K. Schneider, *Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets*, Flow Turbul. Combust., 66 (2001), pp. 393–426.
[20] J. Fröhlich and K. Schneider, *Numerical simulation of decaying turbulence in an adaptive wavelet basis*, Appl. Comput. Harmon. Anal., 3 (1996), pp. 393–397.
[21] V. Girault and P. A. Raviart, *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, Berlin, 1986.
[22] M. Griebel and F. Koster, *Adaptive wavelet solvers for the unsteady incompressible Navier-Stokes equations*, in Advances in Mathematical Fluid Mechanics, J. Malek, J. Necas, and M. Rokyta, eds., Springer-Verlag, Berlin, 2000, pp. 67–118.
[23] S. Grossmann and M. Löhden, *Wavelet analysis of Navier-Stokes flow*, Z. Phys. B, 100 (1996), pp. 137–147.
[24] J.-P. Kahane and P. G. Lemarié-Rieusset, *Fourier Series and Wavelets*, Gordon and Breach, London, 1995.
[25] J. Ko, A. J. Kurdila, and O. K. Rediniotis, *Divergence-free bases and multiresolution methods for reduced-order flow modeling*, AIAA J., 38 (2000), pp. 2219–2232.
[26] F. Koster, M. Griebel, N. Kevlahan, M. Farge, and K. Schneider, *Towards an adaptive wavelet-based 3D Navier-Stokes solver*, in Numerical Flow Simulation I, Notes Numer. Fluid Mech. 66, E. H. Hirschel ed., Vieweg-Verlag, Braunschweig, 1998, pp. 339–364.

[27] R. Kupferman and E. Tadmor, *A fast, high resolution, second-order central scheme for incompressible flows*, Proc. Nat. Acad. Sci. U.S.A., 94 (1997), pp. 4848–4852.
[28] P. G. Lemarié-Rieusset, *Analyses multi-résolutions non orthogonales, commutation entre projecteurs et dérivation et ondelettes vecteurs à divergence nulle*, Rev. Mat. Iberoamericana, 8 (1992), pp. 221–237.
[29] J. Lewalle, *Wavelet transform of the Navier-Stokes equations and the generalized dimensions of turbulence*, Appl. Sci. Res., 51 (1993), pp. 109–113.
[30] J. Liandrat and P. Tchamitchian, *On the fast approximation of some nonlinear operators in non regular wavelet space*, Adv. Comput. Math., 8 (1998), pp. 179–192.
[31] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1998.
[32] C. Meneveau, *Analysis of turbulence in the orthonormal wavelet representation*, J. Fluid Mech., 232 (1991), pp. 469–520.
[33] P. Monasse and V. Perrier, *Orthonormal wavelet bases adapted for partial differential equations with boundary conditions*, SIAM J. Math. Anal., 29 (1998), pp. 1040–1065.
[34] K. Schneider, N. Kevlahan, and M. Farge, *Comparison of an adaptive wavelet method and nonlinearly filtered pseudo-spectral methods for two-dimensional turbulence*, Theor. Comput. Fluid Dyn., 9 (1997), pp. 191–206.
[35] R. Temam, *The Navier-Stokes Equations*, North–Holland, Amsterdam, 1984.
[36] K. Urban, *Using divergence-free wavelets for the numerical solution of the Stokes problem*, in AMLI'96: Proceedings of the Conference on Algebraic Multilevel Iteration Methods with Applications, Vol. 2, University of Nijmegen, Nijmegen, The Netherlands, 1996, pp. 261–277.
[37] K. Urban, *Wavelet bases in H(div) and H(curl)*, Math. Comp., 70 (2000), pp. 739–766.
[38] K. Urban, *Wavelets in Numerical Simulation*, Springer-Verlag, Berlin, 2002.
[39] O. V. Vasilyev and N. K.-R. Kevlahan, *An adaptive multilevel wavelet collocation method for elliptic problems*, J. Comput. Phys., 206 (2005), pp. 412–431.
[40] P. Wesseling, *Principles of Computational Fluid Dynamics*, Springer-Verlag, Berlin, 2001.