Journal of Computational Physics **161,** 401–427 (2000)
doi:10.1006/jcph.2000.6500, available online at http://www.idealibrary.com on IDEAL®

A Parallel Adaptive Coupling Algorithm for Systems of Differential Equations[1]

M. Garbey* and D. Tromeur-Dervout†

Center for the Development of Parallel Scientific Computing CDCSP, University Lyon 1, Bât. 101, 43 bd du 11 Novembre 1918, 69622 Villeurbanne, France
E-mail: *mgarbey@cdcsp.univ-lyon1.fr and †dtromeur@cdcsp.univ.lyon1.fr

Received April 13, 1999; revised March 10, 2000

In this paper we address the challenge of metacomputing with two distant parallel computers linked by a slow network and running the numerical approximation of two sets of coupled PDEs. Several software tools are available for coupling codes, and large-scale computing on a network of parallel computers seems to be mature from a computer science point of view. From an algorithmic point of view, the key to obtaining parallel efficiency is the ability to overlap communication with computation: a priori, the speed of communication between the processors that run the two different codes must be of the same order as that between processors that run the same code in parallel. However, a local network of processors is still faster than a long distant network used for metacomputing by one or two orders of magnitude at least. In this paper, to overcome this limitation, we study some new adaptive time-marching schemes for coupling codes so that efficient metacomputing may be obtained. We will focus on stability and accuracy issues in order to minimize the communication processes and define under which conditions our schemes are numerically efficient. We give several examples of applications chosen as representative test cases for the numerical validation of our algorithms. Finally, efficient metacomputing with two distanced computers linked by a slow network is demonstrated for an application in combustion.

Key Words: algorithms for specific classes of architectures; complexity and performance of numerical algorithms; parallel computation; extrapolation methods; stability and convergence of numerical methods; discrete Fourier transforms; combustion; convective instability.

1. INTRODUCTION AND MOTIVATION

Today, large-scale computation of combustion problems on parallel computers can be done efficiently on dedicated large MIMD systems with hundreds of processors, but the

[1] This work was backed by Région Rhône Alpes.

0021-9991/00 $35.00

cost of these large-scale computers is prohibitive for industry and for an average academic institution. In particular, in an industry environment, one might be more interested in having a robust and efficient code that runs on a cluster of servers linked by an ordinary Ethernet network, than a high-performance code that requires a gigabit internal network. A typical affordable cluster of machines has a slow network that is shared by many users and one must deal with high latency and limited bandwidth for communication of data. An extreme situation for applications is the so-called metacomputing problem. Metacomputing refers to distributed computing with parallel computers located in different cities throughout the world. It is cheaper way to simulate very large-scale parallel computers and use all the memory and flops available than building a specific very large-scale parallel computer. Difficulties with parallel computing because of the limitations of the network have analogues at the level of a uniprocessor machine. CPU processes are much faster than access to memory [23] because memory is highly structured into several layers of cache and main memory. Access to memory is then two orders of magnitude higher for main memory than for L1 cache reference. This is an essential bottleneck for efficient computing and therefore a driving force for designing new numerical algorithms.

It is then necessary to compensate for the poor performance of the network and/or the bandwidth to access memory with a domain decomposition algorithm [14] or an operator splitting algorithm that can cope with these difficulties.

This paper is devoted to a new family of time-marching schemes for coupling codes that relax the penalty on communication in a parallel computing environment. Several software tools for coupling codes are available, and large-scale computing on a network of parallel computers seems to be mature from a computer sciences point of view [4, 7, 8]. From an algorithmic point of view, the key to obtaining parallel efficiency is the ability to overlap communication by computation. If the algorithm is not specially designed to relax the intercode communication, then, a priori, the speed of communication between the processors that run the two different codes must be of the same order as that between processors that run the same code in parallel.

In this paper, we propose some new algorithms for coupling codes that are easy to implement and useful for increasing the efficiency of metacomputing with a standard long-distance network. We have carefully designed test cases that are representative of numerical challenges such as stiffness of ODE systems, the bifurcation phenomenon, and the sharp transition front in space for PDE systems. Numerical results shown are quite encouraging and deserve some analysis. In addition, we implemented a nontrivial example of metacomputing in combustion to demonstrate that even with middle-scale problems, efficient metacomputing with two modern parallel computers linked by a slow network can be obtained.

The outline of the paper is as follows. Section 2 gives the adaptive time-marching scheme for a coupled system of ODEs. Section 3 extends this technique to a system of PDEs. Section 4 gives a numerical application to the predator–prey model. Section 5 is devoted to a numerical validation of our coupling schemes for combustion problems. Section 6 demonstrates the parallel efficiency of our coupling scheme running on two faraway parallel computers linked by a 10 Mbits/s network. Section 7 gives some conclusions and perspectives.

2. MODEL PROBLEMS AND NUMERICAL SCHEMES WITH SYSTEMS OF ODES

We consider the system of two coupled differential equations

$$\dot{X} = F(X, Y), \tag{1}$$

$$\dot{Y} = G(X, Y), \tag{2}$$

where the dot represents the time derivative. We consider second-order schemes of the form

$$\frac{3X^{n+1} - 4X^n + X^{n-1}}{2\Delta t} = F(X^{n+1}, Y^{*,n+1}) \tag{3}$$

$$\frac{3Y^{n+1} - 4Y^n + Y^{n-1}}{2\Delta t} = G(X^{*,n+1}, Y^{n+1}). \tag{4}$$

For $X^{*,n+1} = X^{n+1}$ and $Y^{*,n+1} = Y^{n+1}$, we obtain a fully implicit scheme. Our goal is to compute (3) and (4) in parallel, and therefore to use weak coupling in time marching; we therefore introduce a prediction of X^{n+1} (respectively Y^{n+1}) in (4) (respectively (3)). We assume that (3) is computed on machine I and (4) is computed on machine II. Let τ_{I} be the elapsed time needed to compute X^{n+1} when X^n, X^{n-1}, $Y^{*,n+1}$ is available in the memory of machine I. We make a similar hypothesis for machine II and further assume for simplicity that $\tau = \tau_{\mathrm{I}} = \tau_{\mathrm{II}}$.

We assume that the speed of the network that links these two machines is such that the elapsed time needed to send respectively $X^{*,n+1}$ and $Y^{*,n+1}$ from machine I and II to machine II and I is bounded by $p\tau$, p being an integer. In an ideal world p should be at most 1, but we examine a realistic situation for the so-called metacomputing for which we anticipate p as large as 10. A second-order extrapolation is written as

$$X^{*,n+1} = (p+1)X^{n-p+1} - pX^{n-p}.$$

A similar formula holds for $Y^{*,n+1}$. Because p can be large, we may want to use a third-order formula,

$$X^{*,n+1} = (p+1)\left(\frac{p}{2} + 1\right)X^{n-p+1} - (p^2 + 2p)X^{n-p} + \frac{p^2 + p}{2}X^{n-p-1}.$$

A similar formula holds for $Y^{*,n+1}$. We denote such a scheme $C(p, 1, j)$ with $j = 2$ or 3, the order of extrapolation. The drawback of this scheme from the network point of view is that machine I and machine II exchange two messages at every time step. The network will then be very busy and the buffering of the messages may affect the communication speed. In order to further relax this constraint, we therefore restrict ourselves to exchanging the messages at every q time step. The same data X^{n-p+1} and X^{n-p} will then be used to predict $X^{*,n+k}$ for q consecutive time steps: the second-order extrapolation formula used on machine II is written as

$$X^{*,n+k} = (p+k)X^{n-p+1} - (p+k-1)X^{n-p}, \qquad k = 1 \cdots q.$$

As previously mentioned, an accuracy constraint may lead us to use a third-order extrapolation,

$$X^{*,n+k} = (p+k)\left(\frac{p+k-1}{2} + 1\right)X^{n-p+1} - ((p+k-1)^2 + 2(p+k-1))X^{n-p} + \frac{(p+k-1)^2 + (p+k-1)}{2}X^{n-p-1}.$$

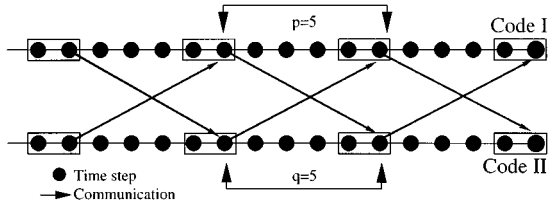

FIG. 1. Communication schedule for the $C(5, 5, j)$ scheme.

We denote such a scheme as $C(p, q, j)$ with $j = 2$ or 3 as the order of extrapolation. Figure 1 exemplifies the communication schedule for scheme $C(5, 5, j)$. It is straightforward to show that the truncation error of the scheme is of order 2. The explicit dependence on previous time steps, supposed by the predictors $X^{*,n+1}$ and by the formulae $Y^{*,n+1}$, is going to impose some stability constraint on the time step. We anticipate that this stability constraint will be weak in the case of weak coupling of the two PDEs. This will be defined more rigorously later on. Furthermore it is important to note that the scheme should be adaptive in time, in particular when the solution of the PDE system goes through oscillation relaxations.

Many techniques have been developed to control the error for the ODE solver [3]. We first notice that for our scheme, we can monitor the difference between the predicted value of $(X^{*,n+k}, Y^{*,n+k})$ used on machine (II, I) and the value actually computed on machine (I, II) with the $C(p, q, j)$ scheme later: this difference is a lower bound on the overall error of the scheme. Second, we compare second- and third-order extrapolations to get an asymptotic estimate of the error of the prediction. Therefore the delay $p + q$ should decrease when the error is larger than the tolerance number. This is a first step in adaptive control of communication processes. We notice that the monitoring of the error does not require additional communication of data fields but mainly additional memory.

A more flexible and efficient way of using the network between the two machines is to use asynchronous communication [1], i.e., to let delay p evolve in time marching in such a way that as soon as the information arrives, it is used. The adaptive criterion defined above then limits the number of time steps q where the same information can be re-used, although $p + q$ should be such that the *accuracy* of the approximation and the *stability* of the time marching are satisfied.

In order to study the stability of the scheme, we compute the stability constraint on the time step with a linear theory. This shows in which circumstances algorithm $C(p, q, j)$ is not suitable. Let us consider the linear ODE system

$$\dot{x} = a_x x + c_x y \tag{5}$$

$$\dot{y} = c_y x + a_y y. \tag{6}$$

Its solution remains bounded iff

$$(a_x + a_y) < 0, \qquad \text{and} \qquad c_x c_y < 0.$$

We notice that the characteristic polynomial of the matrix is invariant to transformations on c_x and c_y such that $c_x c_y = C^{\mathrm{t}}$ and to the permutation $a_x \longleftrightarrow a_y$. With *Maple*, we computed the amplification matrix of schemes $C(p, q, 2)$ for some given integers p and q with $p \le q$.

For the $C(p, 1, 2)$ scheme, this matrix A is written

$$\left(\begin{array}{ccccc|ccccc}
(4\alpha) & (-\alpha) & \boxed{0_1^{p-3}} & 0 & 0 & 0 & 0 & \boxed{0_1^{p-3}} & (p+1)\tilde{\alpha} & -p\tilde{\alpha} \\
1 & 0 & \boxed{0_1^{p-3}} & 0 & 0 & 0 & 0 & \boxed{0_{1,p-3}} & 0 & 0 \\
0 & 1 & \boxed{0_1^{p-3}} & 0 & 0 & 0 & 0 & \boxed{0_{1,p-3}} & 0 & 0 \\
\boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{I_{p-3}^{p-3}} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^m} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} \\
0 & 0 & \boxed{0_1^{p-3}} & 1 & 0 & 0 & 0 & \boxed{0_{1,p-3}} & 0 & 0 \\
\hline
0 & 0 & \boxed{0_1^{p-3}} & (p+1)\tilde{\beta} & -p\tilde{\beta} & 4\beta & -\beta & \boxed{0_1^{p-3}} & 0 & 0 \\
0 & 0 & \boxed{0_1^{p-3}} & 0 & 0 & 1 & 0 & \boxed{0_1^{p-3}} & 0 & 0 \\
0 & 0 & \boxed{0_1^{p-3}} & 0 & 0 & 0 & 1 & \boxed{0_1^{p-3}} & 0 & 0 \\
\boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^{p-3}} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} & \boxed{I_{p-3}^{p-3}} & \boxed{0_{p-3}^1} & \boxed{0_{p-3}^1} \\
0 & 0 & \boxed{0_1^{p-3}} & 0 & 0 & 0 & 0 & \boxed{0_1^{p-3}} & 1 & 0
\end{array}\right), \quad (7)$$

where $\alpha = (3 - 2dta_x)^{-1}$, $\beta = (3 - 2dta_y)^{-1}$, $\tilde{\alpha} = 2c_x\ dt\ \alpha$, $\tilde{\beta} = 2c_y\ dt\ \beta$, $\boxed{O_m^n}$ is a zeros matrix of m rows and n columns, and $\boxed{I_m^m}$ is the identity matrix of rank m. An analogous matrix can be written for the $C(p, 1, 3)$ case.

It is easy to compute the spectral radius of the amplification matrix for some given numerical values of the coefficients a_x, a_y, c_x, c_y and derive the stability bound on the time step; *Maple* allows the user to choose an arbitrary number of digits for the numerics and derive a reliable numerical approximation of the spectral radius of the amplification matrix. However, direct numerical simulations have been used to check the time step constraint as well. For the $C(p, p, 2)$ scheme, the construction of the amplification matrix is not straightforward. We first use the following pseudo code written in *Maple*,

```
n := 2 * (p + 1) : A := array(1..n, 1..n) :
for k from 0 to p − 1 do
        X(N + k) := expand(alpha * (4 * X(N + k − 1) − X(N + k − 2))
          + alpha1 * ((p + k + 1) * Y(N − p) − (p + k) * Y(N − p − 1))) :
        Y(N + k) := expand(alpha * (4 * Y(N + k − 1) − Y(N + k − 2))
          + alpha1 * ((p + k + 1) * X(N − p) − (p + k) * X(N − p − 1))) :
od :
```

to generate all the formulae needed in the construction of A. alpha, alpha1, ... are constant coefficients given in the $C(p, p, 2)$ scheme as a function of the time step and ODE coefficients. Then we identify each element of the matrix with the following pseudo code

```
for from 1 to p + 1 for k from 1 to p + 1 do
A[j, k] := coef f(X(N + p − j), X(N − k), 1) : od : od :
```

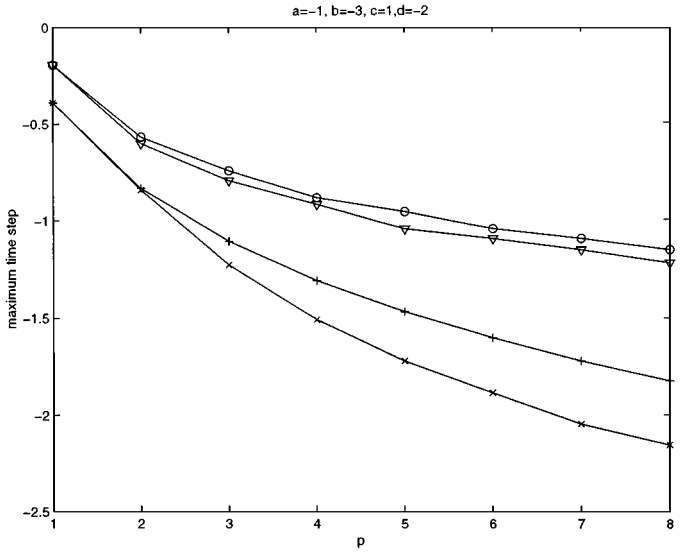

FIG. 2. Upper bound on the time step with $a_x = -1$, $c_x = -3$, $c_y = 1$, $a_y = -2$, with $C(p, 1, 2)$, ○; $C(p, 1, 3)$, +; $C(p, p, 2)$, ∇; $C(p, p, 3)$, ×.

and similar instructions for the subblocks $A[j, k+p+1]$, $A[j+p+1, k]$, $A[j+p+1, k+p+1]$, $j = 1 \ldots p+1$, $k = 1 \ldots p+1$. Then one can again compute the eigenvalues of this matrix, but the expression of the characteristic polynomial is extremely complicated. This *Maple* computation has been validated with direct numerical simulation as well. Similar computations can be done with $C(p, p, 3)$.

In Figs. 2 to 4, we have plotted the maximum time step for, respectively, $C(p, 1, 2)$ (○), $C(p, 1, 3)$ (+), $C(p, p, 2)$ (∇), and $C(p, p, 3)$ (×). The eigenvalues of the differential

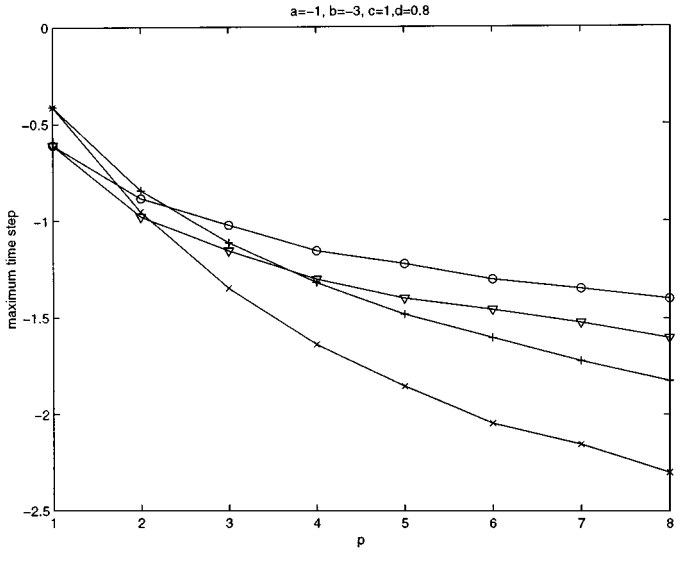

FIG. 3. Upper bound on the time step with $a_x = -1$, $c_x = -3$, $C_y = 1$, $a_y = 0.8$, with $C(p, 1, 2)$, ○; $C(p, 1, 3)$, +; $C(p, p, 2)$, ∇; $C(p, p, 3)$, ×.

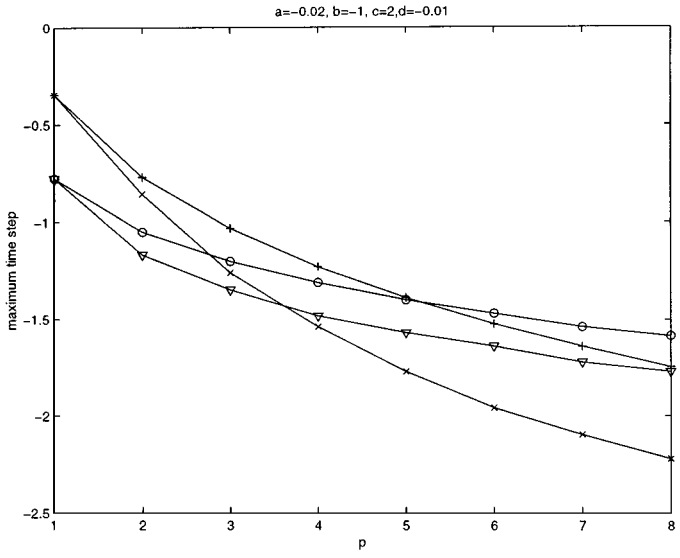

FIG. 4. Upper bound on the time step with $a_x = -0.02$, $c_x = -1$, $c_y = 2$, $a_y = -0.01$, with $C(p, 1, 2)$, $\bigcirc$; $C(p, 1, 3)$, $+$; $C(p, p, 2)$, ∇; $C(p, p, 3)$, $\times$.

system (5) and (6) corresponding to Figs. 2 to 4 are complex conjugates and the real part is respectively -1.5 for Fig. 2, -0.1 for Fig. 3, and -0.015 for Fig. 4. The strength of the coupling is then growing from Fig. 2 to Fig. 4 and the time step restriction due to the explicit treatment of the coupling terms is increasing as well. An obvious conclusion from these computations is that the time step constraint induced by the explicit treatment of the coupling terms in the $C(p, q, 2)$ schemes is unacceptable when it is a strong coupling, i.e.,

$$|c_x c_y| \succ\succ |a_x + a_y|.$$

Furthermore the time step constraint behaves roughly as the inverse of p. In addition, these results show that it is interesting to reuse the same information for some time steps since the time step limit for the $C(p, p, 2)$ scheme is larger than half the time step limit for $C(p, 1, 2)$.

The $C(p, q, 3)$, $q = 1$, $q = p$, schemes seem to be less sensitive to the nature of the coupling, but unfortunately the time step constraint is globally more severe than that for the $C(p, q, 2)$, $q = 1$, $q = p$, schemes. One may think that, after all, the $C(1, 1, 2)$ scheme is the best of all the schemes considered above because it is better to simply wait for the messages and run the code at the maximum time step than to execute the $C(p, p, j)$ scheme which requires roughly p times more computations for the same result. However, in practice, for the unsteady phenomena considered thereafter, the time step is limited independently either by the accuracy constraint or by non-linearities that have been neglected in this analysis so far. Further, we will show in the next section, devoted to PDEs, that the stability of the $C(p, q, j)$ coupling schemes depends strongly on the frequency shape of the coupling terms and may improve as the wave number increases.

3. MODEL PROBLEMS AND NUMERICAL SCHEMES WITH SYSTEMS OF PDES

We consider unsteady linear systems of PDEs in two space dimensions $(x, y) \in (0, 2\pi)^2$ with periodic boundary conditions; we first consider the system (SI)

$$\frac{\partial U}{\partial t} = \Delta U + bV,$$
$$\frac{\partial V}{\partial t} = \Delta V + cU,$$

and second the system (SII)

$$\frac{\partial U}{\partial t} = \Delta U + b\nabla V,$$
$$\frac{\partial V}{\partial t} = \Delta V + c\nabla U.$$

These systems (SI) and (SII) in Fourier space are written as

$$\dot{\hat{U}}_{k,m} = (-k^2 - m^2)\hat{U}_{k,m} + b\hat{V}_{k,m}, \tag{8}$$
$$\dot{\hat{V}}_{k,m} = (-k^2 - m^2)\hat{V}_{k,m} + c\hat{U}_{k,m}, \tag{9}$$

and

$$\dot{\hat{U}}_{k,m} = (-k^2 - m^2)\hat{U}_{k,m} + bi(k+m)\hat{V}_{k,m}, \tag{10}$$
$$\dot{\hat{V}}_{k,m} = (-k^2 - m^2)\hat{V}_{k,m} + ci(k+m)\hat{U}_{k,m}, \tag{11}$$

where k and m are the wave numbers in the x and y directions, respectively, and

$$U = \Sigma_{k,m}\hat{U}_{k,m}\exp(I(kx + my))$$

respectively

$$V = \Sigma_{k,m}\hat{V}_{k,m}\exp(I(kx + my))$$

is the Fourier expansion of U (respectively V).

It is clear that these systems of ODEs are weakly coupled for large wave numbers k or m. Let us assume that the wave number k is constant. We can make an asymptotic evaluation of the largest eigenvalue of the amplification matrix of the $C(p, 1, 2)$ scheme for a large m with *Maple* in the following way. First we start from the amplification matrix A of the scheme for the $C(p, 1, 2)$ given in (7). Then we expand with *Maple* the characteristic polynomial $\det(A - \lambda Id)$. We observe from the asymptotic order of the coefficients of the characteristic polynomial as a function of m, m large, that the largest eigenvalue is asymptotically equivalent to the ratio of the coefficient of λ with the lowest exponent divided by the coefficient of λ with the largest exponent. We obtain for (SI)

$$|\lambda_{\max}| \sim \left(p^2\frac{bc}{m^4}\right)^{\frac{1}{2(p+1)}}, \qquad \text{for } p \geq 2,$$

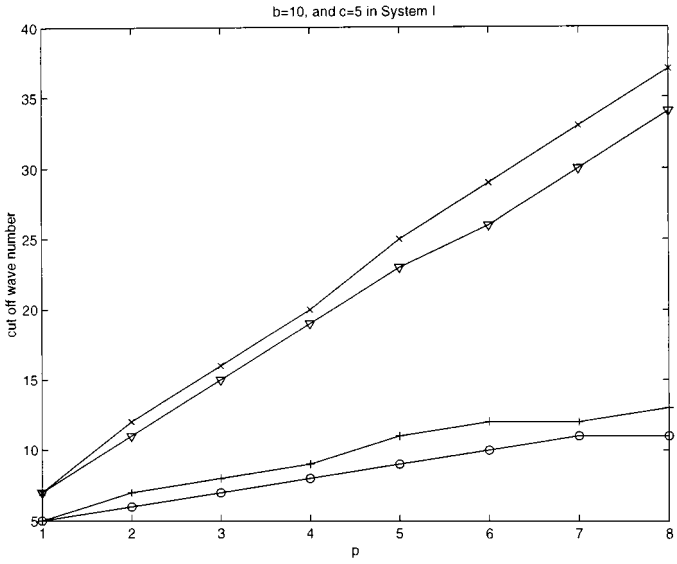

FIG. 5. m_{cut} with $b = 10$ and $c = 5$ in system I, with $C(p, 1, 2)$, $\bigcirc$; $C(p, 1, 3)$, $+$; $C(p, p, 2)$, ∇; $C(p, p, 3)$, $\times$.

respectively for (SII),

$$|\lambda_{\max}| \sim \left(p^2 \frac{bc}{m^2} \right)^{\frac{1}{2(p+1)}}, \qquad \text{for } p \geq 2.$$

In conclusion, there is no time step constraint for wave numbers with m large enough. Further we have obtained the value of the wave number m_{cut} above which the $C(p, q = \{1, p\}, j = \{2, 3\})$ schemes are stable for arbitrary time step dt in (0, 1) with direct numerical simulation in (8), (9) and (10), (11); see Figs. 5 and 6.

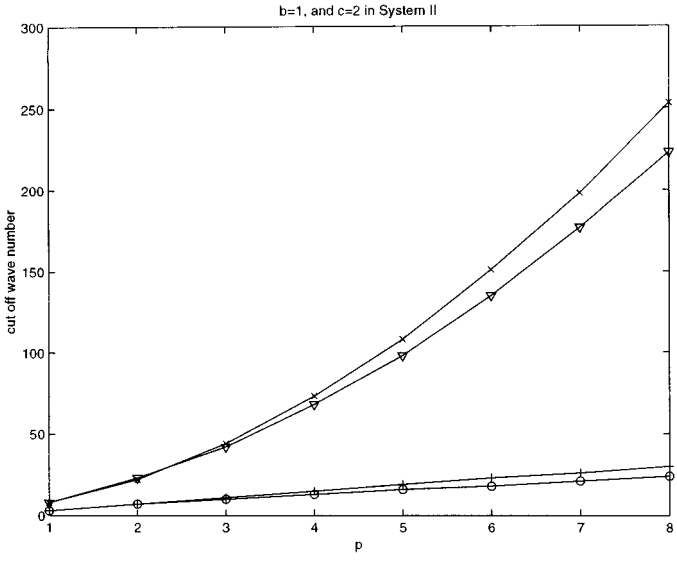

FIG. 6. m_{cut} with $b = 1$ and $c = 2$ in system II, with $C(p, 1, 2)$, $\bigcirc$; $C(p, 1, 3)$, $+$; $C(p, p, 2)$, ∇; $C(p, p, 3)$, $\times$.

The coupling for system II is significantly stronger than that in system I but for both systems it is possible to relax the constraint on communication as follows. We introduce a second type of adaptivity according to the stability constraint in our time-marching scheme based on the fact that we need to communicate the lower frequency part of the spectrum of the coupling terms more often than the higher frequency part. A practical way of implementing this adaptivity in Fourier space is the following: let $\sum_{m=-M\ldots M} \hat{X}_m$ be the Fourier expansion of X. We compute the evolution of X and Y on machine I and II, respectively, and we want to minimize the constraint on communication of X and Y to machine II and I, respectively. Let $\tilde{X}^{*,n+1}$ be the prediction used in the $C(p, q, j)$ scheme for the Fourier mode $\hat{X}_m$ and $\tilde{\tilde{X}}^{*,n+1}$ be the prediction used in the $C(2p, p, j)$ scheme; let σ be a filter of order 8 as in [16, Sect. 3, p. 654]; we use the prediction

$$\sum_{m=-M..M} \hat{X}_m^{n+1} = \sum_{m=-M..M} \sigma\left(\kappa\left|\frac{m}{M}\right|\right)\tilde{X}_m^{n+1} + \sum_{m=-M..M}\left(1-\sigma\left(\kappa\left|\frac{m}{M}\right|\right)\right)\tilde{\tilde{X}}_m^{n+1}$$

with $\kappa > 2$. This way of splitting the signal guarantees the consistency in time and the smoothness in space of the prediction. The numerical value of κ can be chosen according to the linear stability analysis. This scheme will be denoted as $\sigma_\kappa C(p/2p, q, j)$. We can decompose the signal into an arbitrary number of levels with different delays depending on the grid level. However, because the filter σ is smooth, we have redundancy between two consecutive levels on the interval of frequencies for which the filter is neither zero nor one. We notice that our scheme can be generalized in a straightforward way to systems of arbitrary numbers of PDEs and to three space dimensions. The extension of our methodology to nonperiodic boundary conditions and complex domains is not obvious because in a finite element framework, filtering a complex data structure is still an open field of research. However, for spectral methods in space with a simple domain shape, it is relatively easy to extend our methodology; one convenient possibility is to use a local Fourier basis [15, 20–22]. It is interesting to note that our use of filters for operator splitting complements, the use of filters in domain decomposition [5].

Many possible applications of our algorithms in chemical modeling with complex chemistry [18] or in ecology modeling with competition of species [19] should be investigated. In the following, we select a few test cases that are examples of oscillation relaxations for systems of ODEs, bifurcations, and/or stiffness in space for systems of PDEs.

4. AN APPLICATION TO A PREDATOR–PREY MODEL

As a test case, we applied the $C(p, q, j)$ scheme to the long time integration of oscillation relaxations with the Lokta–Volterra equations. The dynamical system is

$$\dot{u} = au - buv \tag{12}$$

$$\dot{v} = -cv + duv. \tag{13}$$

Tables I and II summarize the difference in maximum norm between the numerical solution given by the semi-implicit second-order scheme

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2dt} = au^{n+1} - bu^{n+1}(2v^n - v^{n-1}) \tag{14}$$

$$\frac{3v^{n+1} - 4v^n + v^{n-1}}{2dt} = -cv^{n+1} + d(2u^n - u^{n-1})v^{n+1} \tag{15}$$

TABLE I
Error in Maximum Norm for $t \in [0, 250]$

Time step	0.01	0.02	0.04
$p=6,\ j=2$	(0.031, 0.021)	(0.11, 0.07)	(0.28, 0.21)
$2 \le p \le 6,\ j=2$	(0.014, 0.010) with $\bar{p}=4.3$	(0.046, 0.030) with $\bar{p}=3.6$	(0.11, 0.07) with $\bar{p}=2.8$
$p=6,\ j=3$	(0.0014, 0.0007)	(0.013, 0.007)	(0.081, 0.047)
$2 \le p \le 6,\ j=3$	(0.0004, 0.0003) with $\bar{p}=4.3$	(0.0015, 0.0011) with $\bar{p}=3.6$	(0.0047, 0.0033) with $\bar{p}=2.8$

and the $C(p, p, j)$ scheme with $j=2$ or 3 for the specific set of coefficients $(a, b, c, d) = (1.2, 1.0, 0.1, 0.2)$.

In Table II the solution goes through 27 periods of the limit cycles. The time of integration is therefore long enough for this test case. We can note that $C(6, 6, 3)$ gives much better accuracy than $C(6, 6, 2)$ because the delay $p=6$ is relatively large. The adaptive time-dependent scheme with p varying according to the criterion defined in the previous section and $j=3$ gives the best results. $\bar{p}$ then denotes the average in time of the adaptive delay obtained with our procedure and p is varied in order to keep $\max(u^{*,n}-u^n,\ v^{*,n}-v^n)$ less than some tolerance number.

If predators and prey are spatially distributed, the reaction–diffusion model can be given as follows [19]:

$$\frac{\partial U}{\partial t} = \alpha \Delta U + aU - bUV \tag{16}$$

$$\frac{\partial V}{\partial t} = \beta \Delta V - cV + dUV. \tag{17}$$

For our numerical test, we consider this model in one space dimension with periodic boundary conditions in $\Omega = (0, 2\pi)$. All coefficients are constant except a, which is a step function $a = a_{\text{left}}$ on $(0, \pi)$ and $a = a_{\text{right}}$ on $(\pi, 2\pi)$. The inviscid model, i.e., $\alpha = \beta = 0$, with constant initial conditions $U = U_0$, $V = V_0$, can have two distinct limit cycles. The solution (U, V) of the model problem with nonzero diffusion terms then exhibits an interesting space–time interaction between two possible limit cycles in time. However, the diffusion terms make these limit cycles unstable and the system goes to a steady state; the steady limit solution

TABLE II
Error in Maximum Norm for $t \in [0, 500]$

Time step	0.01	0.02	0.04
$p=6,\ j=2$	(0.054, 0.036)	(0.15, 0.10)	(0.28, 0.21)
$2 \le p \le 6,\ j=2$	(0.020, 0.014) with $\bar{p}=4.3$	(0.073, 0.049) with $\bar{p}=3.7$	(0.24, 0.16) with $\bar{p}=3.0$
$p=6,\ j=3$	(0.0063, 0.0038)	(0.051, 0.031)	(0.28, 0.17)
$2 \le p \le 6,\ j=3$	(0.0004, 0.0003) with $\bar{p}=4.3$	(0.0016, 0.0011) with $\bar{p}=3.7$	(0.022, 0.012) with $\bar{p}=3.0$

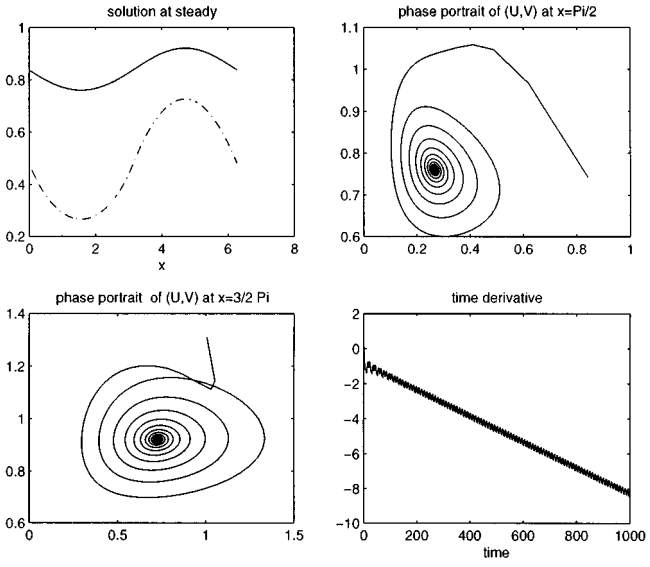

FIG. 7. Solution of predator–prey model with $(\alpha, \nu) = (1., 0.5)$; at steady state, U is in a solid line, and V is in a dashed line.

itself goes to $(U_0, V_0) = (a_{\text{left}}/b, c/d)$ on $(0, \pi)$, $(U_0, V_0) = (a_{\text{right}}/b, c/d)$ on $(\pi, 2\pi)$ as (α, β) goes to zero and we have a transition layer at $x = \pi$ and $x = 2\pi$.

We computed this solution starting from the wave $(U, V) = (1 + 0.5\sin(2x), 1 + 0.6\sin(3x))$ at time zero as well as with a constant initial condition with the Fourier method and the semi-implicit second-order scheme

$$\frac{3U^{n+1} - 4U^n + U^{n-1}}{2dt} = \alpha\Delta U^{n+1} + aU^{n+1} - b(2U^n - U^{n-1})(2V^n - V^{n-1}), \quad (18)$$

$$\frac{3V^{n+1} - 4V^n + V^{n-1}}{2dt} = \beta\Delta V^{n+1} - cV^{n+1} + d(2U^n - U^{n-1})(2V^n - V^{n-1}). \quad (19)$$

The parameter values are $a_{\text{left}} = 0.2$, $a_{\text{right}} = 1.2$, $b = 1.$, $c = 0.1$, and $d = 0.2$. We observe that the oscillations in time of the solution decrease until they reach a steady-state solution (see Fig. 7). As usual the speed of convergence to a steady state decreases as (α, β) gets smaller (see Fig. 8).

We checked for the transient solution that the time accuracy of our various schemes $\mathcal{C}(p, 1, j)$, $C(p, p, j)$ and $\sigma_\kappa C(p/2p, p, j)$ conforms to the theory. As for the stability, we observe, for example, that the scheme $\sigma_3 C(7/14, 7, 3)$ converges to the steady solution of Fig. 7 with time steps as large as $dt = 0.1$.

We shall now validate our coupling scheme methodology on combustion models that are truly challenging scientific computing problems.

5. AN APPLICATION TO A COMBUSTION MODEL

Let us consider the following model of combustion coupled to Navier–Stokes equations written in the Boussinesq approximation,

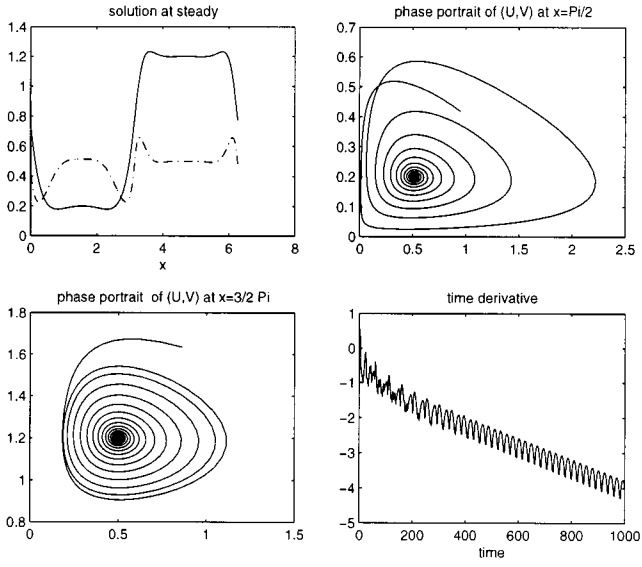

FIG. 8. Solution of predator–prey model with $(\alpha, \nu) = (10^{-2}., 0.5 \times 10^{-2})$; at steady state, U is in a solid line, and V is in a dashed line.

$$\frac{\partial \Theta}{\partial t} = \Delta \Theta - \frac{\partial \Psi}{\partial z}\frac{\partial \Theta}{\partial y} + \frac{\partial \Psi}{\partial y}\frac{\partial \Theta}{\partial z} + C \exp \frac{Z\Theta}{1 + \delta(1 - \Theta)}, \tag{20}$$

$$\frac{\partial C}{\partial t} = \epsilon \Delta C - \frac{\partial \Psi}{\partial z}\frac{\partial C}{\partial y} + \frac{\partial \Psi}{\partial y}\frac{\partial C}{\partial z} - C \exp \frac{Z\Theta}{1 + \delta(1 - \Theta)}, \tag{21}$$

$$\frac{\partial \omega}{\partial t} = \Delta \omega - \frac{\partial \Psi}{\partial z}\frac{\partial \omega}{\partial y} + \frac{\partial \Phi}{\partial y}\frac{\partial \omega}{\partial z} - RP\frac{\partial \Theta}{\partial y}, \tag{22}$$

$$\Delta \Psi = \omega, \tag{23}$$

with $(z, y) \in (-\infty, \infty) \times (0, 2\pi)$. Θ denotes the temperature, C the concentration of the reactant, ω the vorticity, and Ψ the stream function. The Zeldovich number is a scaled measure of the activation energy, R is the Rayleigh number, and P is the Prandtl number.

The boundary conditions are periodicity in the y direction and

$$\Theta(\infty, y) = 1, \Theta(-\infty, y) = 0, C(\infty, y) = 0, C(-\infty, y) = 1,$$

$$\omega(\pm\infty, y) = 0, \Psi(\pm\infty, y) = 0.$$

We refer to [10] for a precise statement of the model. This system has a traveling wave solution $(\Theta(z - vt), C(z - vt), \omega \equiv 0, \Psi \equiv 0)$ independent of y which corresponds to a flat flame front. For a given Rayleigh number R and an increasing positive Zeldovich number Z, this solution loses its stability at some critical value Z_c depending on the control parameter R. We refer to [9] for the derivation of the neutral stability curves. We shall first test our $C(p, q, j)$ scheme on a simplified quasi-linear model of (20)–(23) essential for the analysis of bifurcation phenomena.

5.1. *Numerical Study of the Influence of the C(p, q, j) Scheme on Bifurcation*

We obtain a weakly nonlinear simplified model by looking at the so-called cellular instability of a one-dimensional solution; i.e., we expand the temperature as

$$\Theta(y, z) = \Theta_0(z) + \Theta_m(z) \exp(i\, m\, y).$$

We use a similar expansion for the other unknowns C, ω, and Ψ. Integer m is the possibly unstable wave number. This model is written as

$$\frac{\partial \Theta_0}{\partial t} = \frac{\partial^2 \Theta_0}{\partial z^2} + C_0 \exp \frac{Z\Theta_0}{1+\delta(1-\Theta_0)},$$

$$\frac{\partial C_0}{\partial t} = \epsilon \frac{\partial^2 C_0}{\partial z^2} - C_0 \exp \frac{Z\Theta_0}{1+\delta(1-\Theta_0)},$$

$$\Theta_0(\infty, y) = 1, \Theta_0(-\infty, y) = 0, C_0(\infty, y) = 0, C_0(-\infty, y) = 1, \qquad \omega_0 \equiv 0, \Psi_0 \equiv 0,$$

followed by

$$(\partial \Theta_m / \partial t) = \left(\partial^2 \Theta_m / \partial z^2\right) - m^2 \Theta_m + im\boldsymbol{\Psi}_m (\partial \Theta_0 / \partial z) + W_m,$$

$$(\partial C_m / \partial t) = \epsilon((\partial^2 C_m / \partial z^2) - m^2 C_m) + im\boldsymbol{\Psi}_m (\partial C_0 / \partial z) - W_m,$$

$$\Theta_m(\pm\infty, y) = 0, \qquad C_m(\pm\infty, y) = 0,$$

where

$$W_m = Z \exp\left(\frac{Z\Theta_0}{1+\delta(1-\Theta_0)}\right)\left(C_m + \frac{C_0 Z(1+\delta)}{(1+\delta(1-\Theta_0))^2}\Theta_m\right),$$

and

$$(\partial \omega_m / \partial t) = (\partial^2 \omega_m / \partial z^2) - m^2 \omega_m - im R P \boldsymbol{\Theta}_m,$$

$$(\partial^2 \Psi_m / \partial z^2) - m^2 \Psi_m = -\omega_m,$$

$$\omega_m(\pm\infty, y) = 0, \qquad \Psi_m(\pm\infty, y) = 0.$$

We observe that the first set of PDEs is nonlinear and that Θ_0, C_0 is not necessarily a traveling wave. One can obtain oscillation relaxations of the 1D solution for a Zeldovich number large enough, typically $Z > 7.8$ when ϵ is small. This is a so-called thermal instability. Further increases in Z will lead to several bifurcations until chaos occurs. The set of PDEs for the perturbation terms depending on the wave number m is linear and gives the coupling between the combustion process and the convection process: this simplified model is relevant to the effect of convection on the linear stability of 1D nonlinear solutions. Furthermore, it is noteworthy that the coupling terms act in *the same timescale* when the main approximation Θ_0, C_0 exhibits oscillation relaxations.

The purpose of choosing this simplified model of combustion as a test case for our coupling scheme is to check that $C(p, q, j)$ gives the same bifurcation parameter value for various $p, q \leq 6$, $j = 2$ *or* 3. For simplicity, let us consider the *first bifurcation* from the basic solution, with $Z = 6$ and positive Rayleigh numbers [9]. We applied the numerical schemes $C(p, p, 2)$ for $p = 1, 3, 6$ to compute the solution of our simplified model with the

adaptive domain decomposition spectral method described in [10]. This scheme is written

$$\frac{\Theta_m^{n+1}-\Theta_m^n}{dt}=\frac{\partial^2\Theta_m^{n+1}}{\partial z^2}-m^2\Theta_m^{n+1}+im\Psi_m^{*,n+1}\frac{\partial\Theta_0^n}{\partial z}+W_m^n,$$

$$\frac{C_m^{n+1}-C_m^n}{dt}=\epsilon\left(\frac{\partial^2 C_m^{n+1}}{\partial z^2}-m^2C_m^{n+1}\right)+im\Psi_m^{*,n+1}\frac{\partial C_0^n}{\partial z}-W_m^n,$$

$$\frac{\partial\omega_m^{n+1}-\omega_m^n}{dt}=\frac{\partial^2\omega_m^{n+1}}{\partial z^2}-m^2\omega_m^{n+1}-im\,RP\Theta_m^{*,n+1},$$

$$\frac{\partial^2\Psi_m^{n+1}}{\partial z^2}-m^2\Psi_m^{n+1}=-\omega_m^{n+1}.$$

The initial condition for the perturbation is zero except for temperature: Θ_m is initialized with a hot spot at the front location of the traveling wave solution (Θ_0, C_0). We observe that the first mode $m=1$ is the most unstable mode.

Figure 9 shows the evolution of the maximum vorticity ω_1, for different values of the Rayleigh number in the neighborhood of the stability bound. Curves that end with $\circ$ (respectively $+$ and *) correspond to $p=1$ (respectively $p=3$ and $p=6$).

We see that the perturbation vanishes below $R=18.5$ and blows up above $R=19$. The stability bound is therefore between $R=18.5$ and $R=19$. This result agrees with the direct numerical simulation [10] of the complete model (20)–(23) as well as with its linear stability analysis [9]. In addition, we observe that increasing values of p in the $C(p, p, 2)$ scheme stabilize the solution, that is, introduce some possible minor delays in the bifurcation. However, the $C(p, p, 2)$ scheme for moderate values of p does not significantly affect the bifurcation parameter value.

We have computed the cellular instability of a pulsating one-dimensional solution corresponding to $Z=8$ with our model and obtained the robustness of our scheme with respect

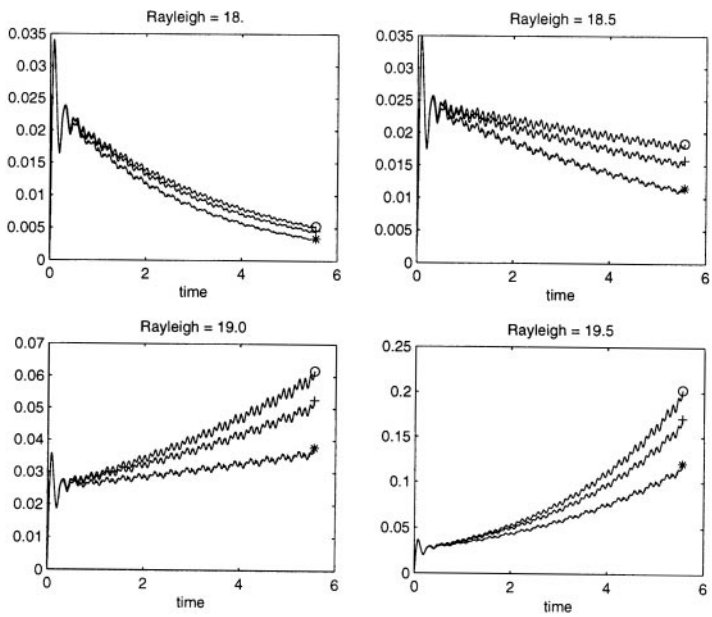

FIG. 9. Evolution of the maximum of the vorticity components in time.

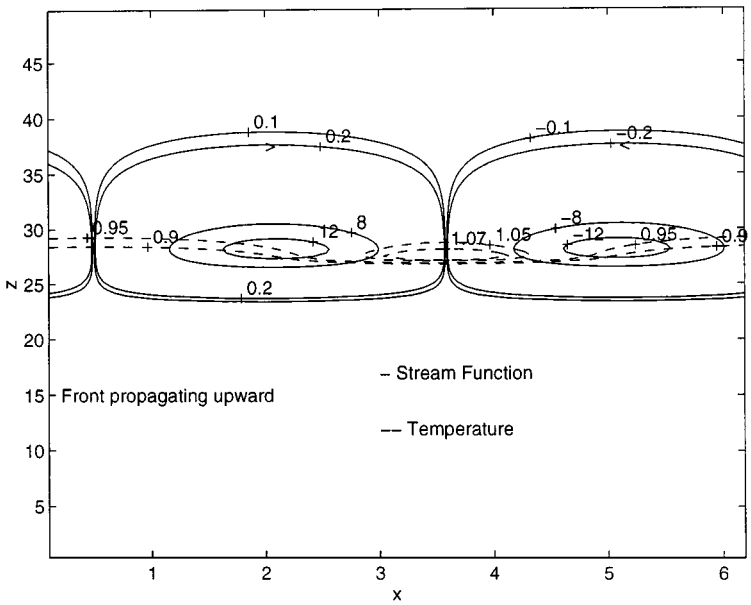

FIG. 10. Cellular instability pattern of a frontal polymerization; Zeldovich = 6.5 and Rayleigh = 20.

to p as well. Now let us proceed with the direct numerical simulation of the complete model (20)–(23).

5.2. The Periodic Case

We consider model (20)–(23) with periodic boundary conditions, gravity in the vertical direction, and the following parameter values. The Prandtl number is fixed to be one, and the dimensionless mass diffusion ϵ to be 0.005. We studied instabilities appearing for values of the Zeldovich number Z significantly less than the critical value Z_c when we let the Rayleigh number R increase or decrease. Figure 10 displays a typical cellular instability pattern of a frontal polymerization when $Z = 6$ and $R = 20$; the results show that the frontal structure is far from being an ordinary layer. We have superposed contour lines for the two-dimensional temperature and stream function fields on the same graphic. First, it is clearly seen that there exists a hot spot at the front location where the temperature is greater than the adiabatic temperature. Second, we see on the same picture the motion of the fluid with two vortices near the front which stand symmetrically on either side of the hot spot. The overall structure is time-independent and moves with constant speed in the vertical direction. The critical value of the Rayleigh number at which the stationary plane ascending front loses its stability was numerically found between the values $R = 18$ and $R = 19$. The critical value predicted by the asymptotic analysis when $\epsilon = 0$ is approximatively 20 [9]. We have numerically checked in Section 6.3 that the $C(p, p, j)$ scheme for $p = 2, \ldots, 6$, and $j = 2, 3$ reproduces this solution and that the maximum value of the hot spot does not significantly depend on the delay p.

5.3. The Nonperiodic Case

Last, we consider a more general model with nonperiodic boundary conditions and gravity with an arbitrary direction; it is worth noticing that we use the biharmonic Navier–Stokes formulation with Ψ as the stream function. We refer to [11] for the precise statement of the

problem and parameters, and we simply recall the basic equations

$$\partial T/\partial t + (\partial \Psi/\partial z)(\partial T/\partial x) - (\partial \Psi/\partial x)(\partial T/\partial z) = \Delta T + C \exp \frac{ZT}{1+\delta(1-T)}, \quad (24)$$

$$\partial C/\partial t + (\partial \Psi/\partial z)(\partial C/\partial x) - (\partial \Psi/\partial x)(\partial C/\partial z) = \epsilon \Delta C - C \exp \frac{ZT}{1+\delta(1-T)}, \quad (25)$$

$$\partial \Delta \Psi/\partial t = (\partial \Psi/\partial x)(\partial \Delta \Psi/\partial z) - (\partial \Psi/\partial z)(\partial \Delta \Psi/\partial x) + \Delta \Delta \Psi - RP\vec{g} \cdot \vec{\nabla} T, \quad (26)$$

$$T \to 0, C \to 1, \Psi \to 0, \quad \text{as } z \to -\infty, \qquad T \to 1, C \to 0, \Psi \to 0, \quad \text{as } z \to +\infty$$

$$\Psi(0,z) = 0, \partial \Psi/\partial x(0,z) = 0, \Psi(L,z) = 0, \partial \Psi/\partial x(L,z) = 0, \quad \forall z, \quad (27)$$

$$\partial T/\partial x(0,z) = 0, \partial T/\partial x(L,z) = 0, \partial C/\partial x(0,z) = 0, \partial C/\partial x(L,z) = 0, \quad \forall z.$$

The system with a zero Rayleigh number has the same well-known one-dimensional stable traveling wave solutions as before. In addition, this more general model with nonperiodic boundary conditions has new different possible non-linear regimes.

The existence and stability of these solutions depend on the specific values of the bifurcation or control parameters, such as the Zeldovich number Z or the Rayleigh number R. Above the Z_c value of the Zeldovich number, we can have pulsating fronts and/or spinning modes, thus making the problem difficult to compute.

When gravity is not parallel to the z direction, the basic solution $(T_0(z), C_0(z), \Psi \equiv 0)$ is unstable and a *nonplanar flame front* with a complex structure is obtained.

This model has been set up to describe frontal polymerization processes in a liquid phase; the model neglects many phenomena such as the change of viscosity in the combustion front, the effect of compressibility, and the possible formation of gas bubbles at the fronts. However, this model reproduces the pattern formation observed in experiments [24, 2] remarkably well.

The reaction diffusion model (24) and (25) is computed with a local Fourier basis as in [12] on the parallel computer (I). The Navier–Stokes equation (26) is computed with a local Fourier basis as well as [13] on the parallel computer (II). The method for the solution has been validated by comparison with a classical Chebyshev high-order finite difference method [10].

Figure 11 shows the effect of hydrodynamics on the structure of the flame front when the channel is *horizontal* and gravity is vertical. Thin solid lines represent the temperature isovalues, while dashed lines represent the stream function isovalues. The thick solid line represents the location of the front, centered on the level set $C = 0.5$. The computation was done with a total of $N_x = 112$ by $N_z = 256$ Fourier modes on a physical domain of size $[0, 4\pi] \times [0, 90]$. The horizontal z direction is the direction of propagation of the front. The Zeldovich number is equal to 6., the Rayleigh number is equal to 1.5, the Prandtl number is $P = 1$, the mass diffusion is given by $\epsilon = 0.02$, and the time step is set to $dt = 0.01$. This solution is a traveling wave moving toward the left with a hot spot close to the top wall of the horizontal channel. The location of the hot spot and the front curvature of the concentration profile are closely related to the circular motion of the flow. This example shows that one can compute a *nonplanar* flame front structure with local Fourier basis.

We are mainly interested in using this problem as a nontrivial test case to demonstrate and validate the feasibility of our approach with metacomputing. The metacomputing experiments described hereafter are for the generalized model only.

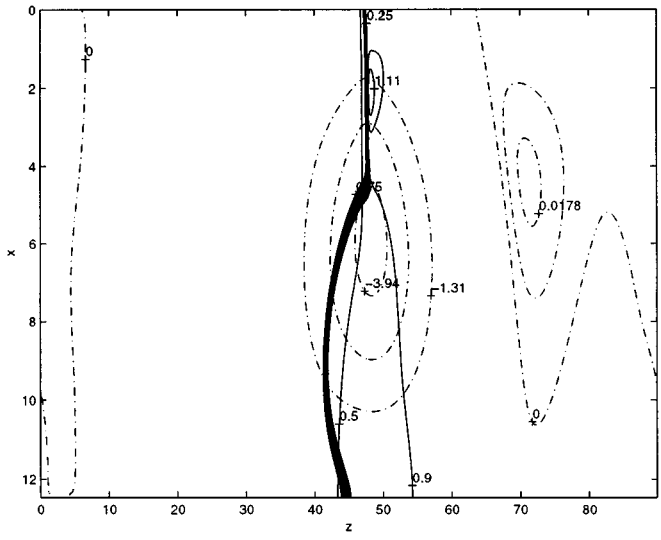

FIG. 11. Superposition of the stream function contour lines (-·-·-) to the hot spot formation; localization of the front is given by the thick line (—).

6. METACOMPUTING EXPERIMENTS

6.1. The Software Configuration

Each code taken separately has a high scalable parallel efficiency mainly because the domain decomposition involves only local communication between neighbor sub-domains for a large wave number and/or a small time step [13]. For the experiments of Table III and Table IV, each uncoupled code has a super linear parallel efficiency. The two codes have to exchange the temperature from (I) to (II) (the temperature spatial derivative is computed in (II)), and the stream function from (II) to (I) (the stream function spatial derivatives are computed in (I)).

The two parallel codes use the same global grid to represent the unknown variables. The data distribution on each node is specified by the programmer, who can use a different number of sub-domains for each application. Thus the communication of one domain's data managed by a code A to a code B requires different numbers of send or receive messages with different data size depending on each processor.

To simplify the nonblocking communication programming, which manages the coupling terms between the two physical models on each code, we developed a portable inter-code communication library (PIPCL) [6].

This communication library computes the array-based communication schedules and creates a one-to-one implicit mapping between each data representation. Programming the send or the receive communication of the data field from A to B with PIPCL requires only one instruction, as in MPI [6].

As the data distribution of a grid is not known at compilation time, the data distribution of an application is hidden from the other applications, allowing an easy parallel program interoperability on heterogeneous computers. This library, based on the MPI library [17], is compiled with the MPICH 1.1 distribution.

TABLE III

Comparison of the $C(p, q, j)$ Schemes, with Three Processors for the Navier–Stokes Code and Three Processors for the Combustion Code

	C(6, 6, 3)	C(6, 6, 2)	C(4, 4, 3)	C(4, 4, 2)	C(3, 3, 3)	C(3, 3, 2)
Coupled code with SAFIR 10Mb/s						
Combustion (CDCSP, Lyon)						
Min:	108.37	102.29	118.89	108.83	135.51	118.69
Max:	132.10	112.91	180.72	116.57	151.45	174.39
Average:	112.80	105.01	125.66	112.22	141.17	128.18
Navier–Stokes (IPGP, Paris)						
Min:	116.22	109.39	124.53	114.23	136.46	120.39
Max:	145.08	120.04	191.48	121.27	152.89	172.69
Average:	120.73	112.12	131.41	117.24	142.66	129.20
Ratio	0.8596	0.9179	0.7872	0.8742	0.7253	0.7926
Efficiency	58.44%	62.40%	52.95%	58.89%	47.96%	52.44%
Coupled code with FDDI: 100Mb/s						
Combustion (IPGP, Paris)						
Min:	96.04	94.81	99.12	98.45	104.00	102.09
Max:	105.01	109.55	105.29	104.60	109.49	106.03
Average:	93.53	98.68	101.66	101.58	106.32	103.93
Navier–Stokes (IPGP, Paris)						
Min:	102.78	101.78	103.65	102.73	106.27	103.89
Max:	105.82	114.85	106.73	107.58	110.28	107.86
Average:	103.82	103.69	105.12	104.35	107.67	105.47
Ratio	0.9996	0.9926	0.9841	0.9822	0.9610	0.9709
Efficiency	67.11%	66.95%	65.83%	65.61%	63.62%	64.46%
Noncoupled codes						
Combustion (IPGP, Paris)						
Min:	32.42	32.29	32.30	32.39	32.50	32.39
Max:	32.97	32.83	33.01	33.05	32.83	32.75
Average:	32.69	32.56	32.67	32.63	32.66	32.58
Navier–Stokes (IPGP, Paris)						
Min:	102.67	101.62	102.44	101.66	102.46	101.27
Max:	105.33	104.28	104.39	103.40	104.94	103.32
Average:	103.78	102.92	103.45	102.49	103.47	102.40
Ratio	1	1	1	1	1	1
Efficiency	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

6.2. The Hardware Configuration

For our experiment the parallel computer (I) is a Tru cluster 4100 from DEC with 400-MHz alpha chips located in Lyon, while the parallel computer (II) is a similar parallel computer located in Paris but with 440-MHz alpha chips.

The distance between Paris and Lyon is about 500 km. Each parallel computer is a cluster of alpha servers linked by an FDDI 100 Mbits/s local network. Thanks to project SAFIR, France Telecom has provided a full duplex 10 Mbits/s link between these two parallel computers through an ATM Fore interface at 155 Mbits/s. The internal speed of the network in each parallel computer is about 80 times faster with the memory channel and 10 times faster when the FDDI ring is operated. ATM is used to guarantee the quality of service of the long-distance 10 Mbits/s connection.

TABLE IV
Results with Load Balancing between the Noncoupled Codes

200 iterations	Processors	Max (s)	Min (s)	Average (s)	Efficiency	Ratio
			Nz = 64, Nx = 120			
SAFIR coupled	$P_{NS} = 4$	221.04	212.89	216.19		
	$P_{CB} = 2$	210.82	200.13	204.56	80.60%	0.9574
FFDI coupled	$P_{NS} = 4$	208.24	195.68	202.05		
	$P_{CB} = 2$	192.87	180.23	187.68	91.24%	1.0244
Noncoupled	$P_{NS} = 4$	208.45	205.13	206.98		
	$P_{CB} = 2$	109.43	108.41	108.90	100.0%	1.
			Nz = 64, Nx = 180			
SAFIR coupled	$P_{NS} = 6$	272.20	253.09	260.50		
	$P_{CB} = 3$	295.25	240.02	247.86	73.87%	0.8093
FFDI coupled	$P_{NS} = 6$	209.33	199.46	204.76		
	$P_{CB} = 3$	194.66	185.28	189.69	93.98%	1.0296
Noncoupled	$P_{NS} = 6$	214.38	205.97	210.82		
	$P_{CB} = 3$	158.92	149.24	155.64	100.0%	1.
			Nz = 64, Nx = 240[a]			
SAFIR coupled	$P_{NS} = 4$	157.45	149.81	153.28		
	$P_{CB} = 3$	141.00	133.45	136.88	63.81%	0.8030
FFDI coupled	$P_{NS} = 4$	126.11	124.53	125.67		
	$P_{CB} = 3$	107.22	105.30	106.51	77.83%	0.9794
Noncoupled	$P_{NS} = 4$	123.79	122.38	123.08		
	$P_{CB} = 3$	64.52	63.76	64.12	100.0%	1.
			Nz = 128, Nx = 120			
SAFIR coupled	$P_{NS} = 4$	253.97	241.22	247.79		
	$P_{CB} = 2$	241.50	228.89	235.38	71.33%	0.8405
FFDI coupled	$P_{NS} = 4$	208.08	199.38	205.08		
	$P_{CB} = 2$	197.17	184.73	190.78	86.76%	1.0155
Noncoupled	$P_{NS} = 4$	208.95	206.89	208.26		
	$P_{CB} = 2$	105.32	104.49	104.84	100.0%	1.
			Nz = 128, Nx = 180			
SAFIR coupled	$P_{NS} = 6$	306.76	285.09	298.23		
	$P_{CB} = 3$	294.26	271.71	284.95	65.85%	0.7932
FFDI coupled	$P_{NS} = 6$	238.60	215.00	224.62		
	$P_{CB} = 3$	222.73	199.75	209.35	88.13%	1.0532
Noncoupled	$P_{NS} = 6$	241.94	228.56	236.57		
	$P_{CB} = 3$	107.97	106.58	107.26	100%	1.

[a] 80 iterations.

Figure 12 gives the actual measured speed of the network depending on the size of the packages. It consists in blocking send and receive messages between one processor in Lyon and one processor in Paris with increasing message lengths in bits.

The times in Fig. 12 are the total average elapsed times for one execution with a sample of 100 executions: therefore they include CPU time to prepare the packages for ATM and to buffer the messages in the application sent with MPI. The time is then measured by appropriate MPI call in the application code.

The communication bandwidth depends on the size of the message. It seems linear from 64×10^3 bits to 64×10^5 bits; the communication time increases with the same amplification factor of value 10 as the message length. But for a message size of 64×10^6 bits the

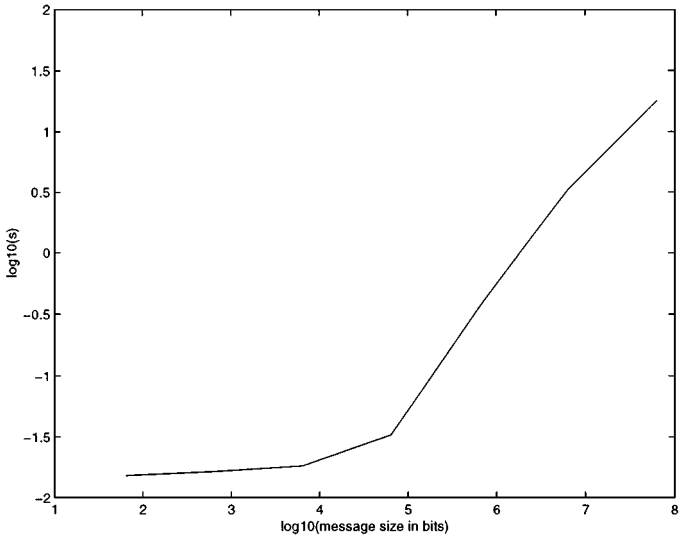

FIG. 12. Performance of SAFIR network between Paris and Lyon.

amplification factor is only 5 with a total time of 17.96 s! This figure shows that it is difficult to model the network performances a priori. In Fig. 12, we approximate the start-up of the distant processor communication to be 0.0141 s by using extrapolation of the curve toward the origin. The reported results of Fig. 12 will be used later to evidence that a trivial coupling scheme that exchanges its flow field at every time step is not a practical solution for metacomputing because overlapping communication by computation with non blocking messages cannot possibly hide communication.

To take an example, the discretization of our combustion model involves few sub-domains with a total of 128×128 Fourier modes, i.e., 4 Mbits of data per flow field in double precision arithmetic. The elapsed computation time per time step for each code is less than a second while Fig. 12 shows that communication should take more than 2 s.

6.3. The Time-Marching Schemes

We then consider the time-dependent scheme for (I)

$$\frac{3T^{n+1} - 4T^n + T^{n-1}}{2dt} + \frac{\partial \Psi^{*,n+1}}{\partial z}\left(2\frac{\partial T^n}{\partial x} - \frac{\partial T^{n-1}}{\partial x}\right) - \frac{\partial \Psi^{*,n+1}}{\partial x}\left(2\frac{\partial T^n}{\partial z} - \frac{\partial T^{n-1}}{\partial z}\right)$$
$$= \Delta T^{n+1} + (2C^n - C^{n-1}) \exp\left(\frac{Z(2T^n - T^{n-1})}{1 + \delta(1 - 2T^n + T^{n-1})}\right)$$

with a similar semi-discrete equation for the concentration, coupled to the time-dependent scheme for (II),

$$\frac{\Psi^{n+1} - \Psi^n}{dt} + \frac{\partial \Psi^n}{\partial z}\frac{\partial \Delta \Psi^n}{\partial x} - \frac{\partial \Psi^n}{\partial x}\frac{\partial \Delta \Psi^n}{\partial z} = \Delta \Delta \Psi^{n+1} - RP\mathbf{g} \cdot \boldsymbol{\nabla} T^{*,n+1}.$$

The coupling terms $\Psi^{*,n+1}$, $T^{*,n+1}$ are given by the $C(p, q, j)$ scheme. We can further adapt the delay with respect to the wave number as in Section 3.

Let us introduce the linear analog set of PDEs denoted (III),

$$\frac{\partial X}{\partial t} = \Delta X + B\frac{\partial Y}{\partial y}$$
$$\frac{\partial \Delta Y}{\partial t} = \Delta\Delta Y + R\frac{\partial X}{\partial y}.$$

This system in Fourier space is written as

$$\dot{X}_{m,k} = (-k^2 - m^2)X_{m,k} + Bim\, Y_{m,k}, \tag{28}$$
$$\dot{Y}_{m,k} = (-k^2 - m^2)Y_{m,k} - \frac{imR}{k^2+m^2}X_{m,k}, \tag{29}$$
$$\tag{30}$$

where k and m are the wave numbers in the directions z and y, respectively. It is clear that these two ODEs are weakly coupled for m large. For example, we obtain with *Maple* that the largest eigenvalue of the amplification matrix of the $C(p, 1, 2)$ schemes satisfies

$$|\lambda_{\max}| \sim \left(\frac{p^2 RB}{m^4}\right)^{1/2(p+1)}$$

for $p \geq 2$.

In conclusion, there is no time step constraint for large enough wave numbers m. With direct numerical simulation, we have obtained the wave number above which the $C(p, q, j)$ schemes are stable for all time steps dt in (0, 1). Figure 13 shows that for moderated values of B and R, the $C(p, q, j)$ scheme has good stability properties for relatively low modes.

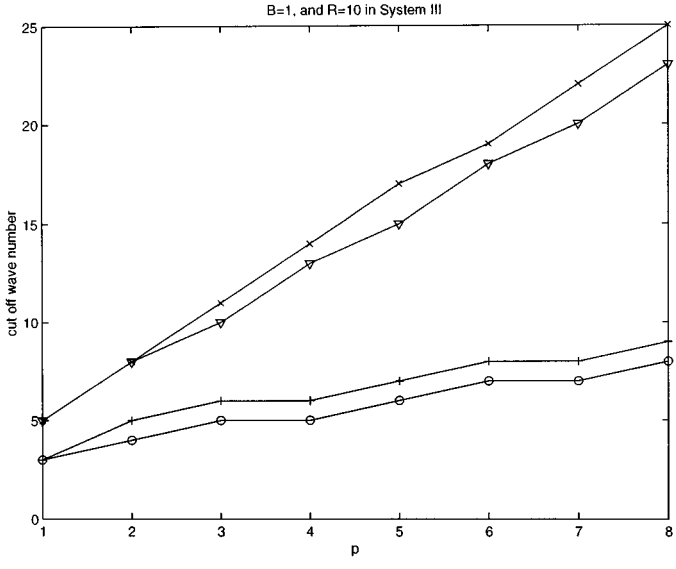

FIG. 13. m_{cut} with $B = 1$ and $R = 10$ in system II, with $C(P, 1, 2)$, ○; $C(p, 1, 3)$, +; $C(p, p, 2)$, ∇; $C(p, p, 3)$, ×.

6.4. *Parallelism Results from Metacomputing with C(p, q, j) Schemes*

Let us consider our 2D applications in combustion with three sub-domains of size 64 modes in the z direction and 40 modes per sub-domain in the x direction. The total number of data bits that must be exchanged between the two codes in one communication is then $2 \times 3{,}932{,}160$ bits.

In order to compare the impact of the network on the performances of the new scheme, the two coupled codes have been run on the SAFIR network, with one code in Lyon and the other in Paris, as well as both codes on the same location through the 100 Mbits/s FDDI local network. In this last case, we ran the two uncoupled codes on the fastest hardware at our disposal, the IPG Dec Alpha 8100 in Paris.

The ideal situation is when the execution of the coupled code with the $C(p, q, j)$ scheme takes the same time as the more time-consuming of the two uncoupled codes. In all test runs, we let the most intense computing code, i.e., the Navier–Stokes, run on the 440-MHz processors in Paris rather than on the 400-MHz processors in Lyon.

In order to estimate the efficiency of the schemes for the metacomputing, we define two measures.

First, we define a low bound efficiency of the coupling as

$$E_c = \frac{\left(T_{\mathrm{NS}}^{\text{Non-coupled}} * P_{\mathrm{NS}} + T_{\mathrm{CB}}^{\text{Non-coupled}} * P_{\mathrm{CB}}\right)}{\max\left(T_{\mathrm{CB}}^{\text{Coupled}}, T_{\mathrm{NS}}^{\text{Coupled}}\right) * (P_{\mathrm{NS}} + P_{\mathrm{CB}})}, \tag{31}$$

where P_{NS} and P_{CB} are the numbers of the processors devoted to the NS and CB codes, respectively. $T_{\mathrm{NS}}^{\text{Non-coupled}}$ and $T_{\mathrm{CB}}^{\text{Non-coupled}}$ are the elapsed times for respectively the non-coupled Navier–Stokes code and the non-coupled combustion code while $T_{\mathrm{NS}}^{\text{Coupled}}$ and $T_{\mathrm{NS}}^{\text{Coupled}}$ are the elapsed times for the coupled codes.

The $(T_{\mathrm{NS}}^{\text{Non-coupled}} * P_{\mathrm{NS}} + T_{\mathrm{CB}}^{\text{Non-coupled}} * P_{\mathrm{CB}})$ term represents an (under) estimation of the time to run the two codes with only one processor. With a real run on a single processor, the system should swap the memory on the disk, as opposed to our parallel run. In addition, we recall that for these applications, we keep the number of processors such that we have a super linear speedup for each code on its cluster, due to the memory cache effect. This definition (31) of the efficiency will be used to study the influence of the network (FDDI or SAFIR) on the performance of the scheme.

Second, we define a ratio number denoted $R_{c/u}$ which is the ratio between the maximum elapsed time of the noncoupled codes running in parallel on the fastest computer and the elapsed time of the coupled codes. This ratio makes it possible to measure the elapsed time loss in the communication between codes. A value ratio of 1 indicates that the communications between codes (I) and (II) are totally hidden by the computation of the iterative scheme.

Table III gives the elapsed time for each code with a noncoupling and with our coupling scheme for the different hardware configurations (SAFIR, FDDI, noncoupling) and different parameter values of the $C(p, q, j)$ scheme.

As the SAFIR network was not dedicated to this experiment only, some perturbation due to other traffic may have occurred. The measures for each scheme were taken considering 30 samples of 100 time steps each. Table III summarizes the minimum, maximum, and average values of the total 100 time steps for each code.

We must notice that the ATM fore interface requires some CPU resources, to pack and to unpack the communication cells to be submitted. These are totally hidden from the users. The new generations of ATM hardware will probably fix this problem in the near future.

For the time being, we restrict ourselves to using only three processors out of four for our run on a Dec 4100 server. For the sake of simplicity in our first set of experiments, we have used the same number of processors on each parallel computer.

Table III shows that:

- The best overlapping of intercode communication through the SAFIR network has been reached with the $C(6, 6, 2)$ scheme with a ratio $R_{c/u} = 91.79\%$. Obviously smaller delay p or higher order extrapolation j degrades the performance.
- In all cases the FDDI network, whose peak performance is still eight times slower than the peak performance of the memory channel, is a good enough network for our metacomputing experiments, since $R_{c/u}$ is in all cases larger than 96.1%.
- Nevertheless the two codes are strongly unbalanced in terms of computing cost and therefore it is not a good solution to take the same number of processors and domains per each code. However, we observe that in all cases the efficiency obtained with our scheme is very close to the optimal efficiency: with no delay in communication, i.e., noncoupling elapse time, this maximum efficiency for $C(6, 6, 2)$ is at most $E_c = \frac{(3\times 32.56+3\times 102.92)}{6\times 102.92} = 65.82\%$. The results obtained for the FDDI and the SAFIR networks are close to this value.
- A straightforward coupling code procedure which exchanges a field of data between parallel computers at each iteration step will give much worse results:

Considering the global data grid size of $(240 \times 128 \times 64)$ bits, we can estimate (see Fig. 12) the time needed to transfer one field from one code to the other at 1 s at least. Table III gives approximately 1 s per time step for the Navier–Stokes code and 0.33 s for the combustion code. As Fig. 14 shows, it takes nearly 3.3 s to perform two time steps with SAFIR. The ratio $R_{c/u}$ then is 60.60% at most; Table III shows that the $C(6, 6, 2)$ scheme improves the performance of the parallel computation with $R_{c/u}$ up to 92%.

6.5. *Parallelism Results on Metacomputing with $\sigma_\kappa C(p/2p, q, j)$ Schemes and Load Balancing of Codes*

In previous results the codes are greatly unbalanced, which leads to an overall mediocre efficiency of the metacomputing. We consider now the $\sigma_\kappa C(6/12, 6, 2)$ schemes and we try to achieve better load balancing between the computing cost of the two uncoupled codes. In all runs κ is such that the delay for the first 30 modes is $p = 6$, and the delay for high frequencies is $p = 12$. Different data distributions for the combustion code and the Navier–Stokes code are used. We set the number of combustion processes running in Lyon to 2 or 3 and we take between 2 and 9 processors for the Navier–Stokes code in Paris. Different grid sizes $2 \times N_z \times 2 \times N_x$ are tested. N_z is for the number of modes in the direction of the

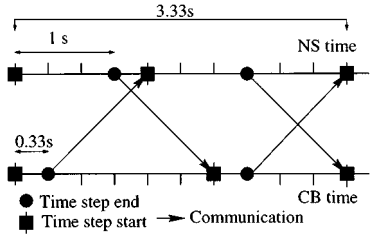

FIG. 14. Schedule for two iterations of the two codes (NS, Navier–Stokes; CB, combustion) with explicit coupling and no delay for communications.

front propagation. The number of time steps is set to 200 except for the largest problem ($N_x = 180$, $N_z = 240$) where it is set to 80 for convenience.

For each grid size, Table IV summarizes the best elapsed time obtained for the scheme using the SAFIR network, between the different data distribution configurations tested. We found that:

- A load balancing of at least 50% between the two codes (73.82% for $N_x = 180$) has been achieved.
- The efficiency of the coupling with FDDI is between 78 and 94% while the coupling with SAFIR goes from 64 to 80%.
- The performance of our scheme can deteriorate when the size of the grids is increased. This degradation is weaker with the FDDI network than with the SAFIR network.
- We notice that the elapsed time for noncoupling codes is eventually slightly larger than the elapsed time for the coupling case in the table: the sensitivity of the performance of the code with respect to memory load seems to allow such surprising results.

7. CONCLUSIONS AND PERSPECTIVES

In this paper, we studied new schemes for the metacomputing of two coupled models running on distant parallel computers connected with a low bandwidth network. These schemes are based on the explicit treatment of the coupling terms in the model. Extrapolation formulae are used to relax the number of intercode communications. The schemes' parameters have to be set adaptively in order to satisfy the accuracy and stability constraints.

First, we studied the stability and the accuracy of the schemes to solve linear and nonlinear systems of PDEs with and without oscillation relaxations.

- Our linear stability analysis for ODE systems shows the limit of the scheme due to the explicit treatment of the coupling terms. The stronger the coupling, the smaller the maximum time step allowed.
- A generalization to a system of PDEs with periodic boundary conditions shows that one can adapt the delay in the coupling scheme with respect to the wave number in order to further relax the penalty on communications.
- Several numerical examples confirm the stability and accuracy properties of our scheme. In particular, we checked on a nontrivial combustion model that the scheme $C(p, p, 2)$ does not significantly affect the bifurcation parameter values for moderate values of the delay.

Second, we tested the parallel efficiency of the scheme with two distant parallel computers running coupled codes with moderate data size, connected with a 10 Mbits/s ATM link. We have shown that:

- The $C(p, q, j)$ scheme really makes possible the use of far away parallel computers for metacomputing. Sources of inefficiency are then due to mainly unbalanced load between the two codes.
- The $\sigma_\kappa C(p/2p, q, j)$ scheme makes it possible to hide intercode communication quite well, even if the network communication bandwidth is slow. Between 63 and 80.60% (respectively 77.83 and 93.98%) efficiency has been reached with respectively 10 Mbits/s (and 100 Mbits/s) links for the test cases under consideration.

The real advantage of metacomputing with two far away parallel computers over classical parallel computing on a single large parallel computer has not been dealt with in this paper. But we have shown that improving the algorithm of coupling codes can make possible the efficient parallel metacomputing of an application with a slow network. The future of metacomputing should be the resolution of so-called "grand challenge" problems which cannot be solved on a single parallel computer. Our next step in this research is to consider large sets of coupled PDEs that are characteristic of combustion problems with realistic chemistry.

ACKNOWLEDGMENTS

We thank Professor A. Ecer for many fruitful discussions on filtering techniques for explicit schemes. This work required several different sources of technical supports and equipment, first from the Institut of Physique du Globe de Paris which allowed us access to their computing resources, and then from the Centre d'Informatique Scientifique et Médical of Lyon 1 University which helped us to deal with the network technical difficulties.

REFERENCES

1. D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation, Numerical Methods* (Prentice–Hall, Englewood Cliffs, NJ, 1989).
2. G. Bowden *et al.*, Effect of convection on a propagating front with a solid product: Comparison of theory and experiments, *J. Phy. Chem. B* **101**, No. 4, 678 (1997).
3. M. Crouzeix and A. L. Mignot, *Analyse Numérique des équations différentielles* (Masson, 1983).
4. J. Dongarra, Solving computational grand challenges using a network of heterogeneous supercomputers, in *Proceedings, Fifth SIAM Conference on Parallel Processing for Scientific Computing* (1991), pp. 596–601.
5. A. Ecer, N. Gopalaswamy, H. U. Akay, and Y. P. Chien, Digital filtering techniques for parallel computation of explicit schemes, AIAA Paper 98-0616.
6. G. Edjlali, M. Garbey, and D. Tromeur-Dervout, Interoperability parallel programs approach to simulate 3D frontal polymerization process, *J. Parallel Comput.* **25**, 1161 (1999).
7. G. E. Fagg, K. Moore, J. Dongarra, and A. Geist, Scalable networked information processing environment (SNIPE), in *Proceedings, Super Computing 97* (Comput. Soc. Press, New York, 1997).
8. I. Foster and C. Kesselman, Globus: A metacomputing infrastructure toolkit, *Int. J. Supercomput. Appl.* **11**, No. 2, 115 (1997).
9. M. Garbey, A. Taik, and V. Volpert, Linear stability analysis of reaction fronts in liquids, *Q. Appl. Math.* **54**, No. 2, 225 (1996).
10. M. Garbey and D. Tromeur-Dervout, Massively parallel computation of stiff propagating combustion front, *IOP J. Comb. Theory Model.* **3**, No. 1, 271 (1997).
11. M. Garbey and D. Tromeur-Dervout, A new parallel solver for non periodic incompressible Navier–Stokes equations with Fourier basis: Application to frontal polymerisation, *J. Comput. Phys.* **145**, No. 1, 316 (1998).
12. M. Garbey and D. Tromeur-Dervout, Application of local Fourier basis to domain decomposition in combustion problems, in *Proceedings, Inter. Conf. Parallel CFD98*, edited by C. A. Lin *et al.* (1998), pp. 199–204.
13. M. Garbey and D. Tromeur-Dervout, Domain decomposition with local Fourier bases applied to frontal polymerization problems, in *Proceedings, Inter. Conf. DD11*, edited by Ch.-H. Lai *et al.* (1998), pp. 242–250.
14. M. Garbey and D. Tromeur-Dervout, Level domain decomposition for multiclusters, in *Proceedings, Inter. Conf. DD12*, edited by T. Chan *et al.* (Chiba, 1999).
15. M. Garbey, On some applications of the superposition principle with Fourier basis, CDCSP preprint 97-03. [To appear in *SIAM Sci. Comput.*]
16. D. Gottlieb and C.W. Shu, On the Gibbs phenomenon and its resolution, *SIAM Rev.* **39**, No. 4, 644 (1998).

17. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI*, Scientific and Engineering Computation Series (MIT Press, Cambridge, MA, 1994).
18. P. Gray and S. K. Scott, *Chemical Oscillations and Instabilities*, The International Series of Monographs on Chemistry, Vol. 21 (Clarendon Press, Oxford, 1990).
19. J. D. Murray, *Mathematical Biology*, Biomathematics Texts, 2nd ed., Vol. 19 (Springer-Verlag, Berlin/New York, 1993).
20. M. Israeli, L. Vozovoi, and A. Averbuch, Spectral multidomain technique with local Fourier basis, *J. Sci. Comput.* **8**, No. 2, 135 (1993).
21. M. Israeli, L. Vozovoi, and A. Averbuch, Domain decomposition methods for solving parabolic PDEs on multiprocessors, *Appl. Numer. Math.* **12**, Nos. 1–3, 193 (1993); Domain decomposition methods with local Fourier basis for parabolic problems, *Contemp. Math.*, 157 (1994).
22. M. Israeli, L. Vozovoi, and A. Averbuch, Spectral multidomain technique with local Fourier basis, *J. Sci. Comput.* **9**, No. 3, 311 (1994).
23. D. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach* (Morgan Kaufman, San Mateo, CA, 1990).
24. J. Pojman, R. Graven, A. Khan, and W. West, Convective instabilities in traveling fronts of addition polymerization, *J. Phys. Chem.* **96**, 7466 (1992).