

# Implementation Considerations for the OIFS/Characteristics Approach to Convection Problems

P. F. Fischer \*

**Abstract**

---

\*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439.  
*fischer@mcs.anl.gov*

# 1 Introduction

We consider semi-implicit numerical solution of convection problems such as the incompressible Navier-Stokes equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega, \end{aligned} \quad (1)$$

and the convection-diffusion equation

$$\frac{\partial \phi}{\partial t} + \mathbf{c} \cdot \nabla \phi = \frac{1}{Pe} \nabla^2 \phi + f, \quad \text{in } \Omega. \quad (2)$$

subject to appropriate initial and boundary conditions on  $\mathbf{u}$  and  $\phi$ . As with the Navier-Stokes case, we assume a divergence free convecting velocity field,  $\nabla \cdot \mathbf{c} = 0$ .

We would like to construct a high-order scheme in time that treats the nonsymmetric convection term explicitly while treating the diffusion term implicitly. One possible approach is to use Crank-Nicolson (CN) for the diffusion term and third-order Adams-Bashforth (AB3) for the convection term. This approach amounts to computing  $\phi^n$  by approximately integrating the time-derivative of  $\phi$  over the interval  $[t^{n-1}, t^n]$ . The diffusion term is integrated using the trapezoidal rule, and the convection term is integrated by extrapolating the convection contribution over the interval  $[t^{n-1}, t^n]$  based on available values from prior timesteps,  $t^{n-q}$ ,  $q = 1, \dots, 3$ . The CN/AB3 scheme is

$$\begin{aligned} \frac{\phi^n - \phi^{n-1}}{\Delta t} &= \frac{1}{2} \frac{1}{Pe} (\nabla^2 \phi^n + \nabla^2 \phi^{n-1}) \\ &+ \frac{1}{12} (23 \mathbf{c} \cdot \nabla \phi|_{t^{n-1}} - 16 \mathbf{c} \cdot \nabla \phi|_{t^{n-2}} + 5 \mathbf{c} \cdot \nabla \phi|_{t^{n-3}}) + \frac{1}{2} (f^n + f^{n-1}). \end{aligned} \quad (3)$$

If  $f$  happens to be dependent on  $\phi^n$ , it can alternatively be lumped with the AB3 terms. The scheme is globally second-order accurate in time. We choose AB3 over AB2 because the work is essentially the same and because the stability region for AB3 encloses a significant portion of the imaginary axis and thus contains the eigenvalues of the convection-diffusion operator in the zero-diffusion limit.

An alternative approach to the CN-AB3 scheme is to use backward differentiation of order  $k$  (BDF $k$ ). Here the time-derivative of  $\phi$  is approximated to order  $k$ , using a one-sided difference formula, and the convection and diffusion terms are evaluated at time  $t^n$ . To avoid implicit treatment of the nonsymmetric convection term, we approximate  $\mathbf{c} \cdot \nabla \phi$  at  $t^n$  by extrapolation of order  $k$  (EXT $k$ ), using values from  $t^{n-q}$ ,  $q = 1, \dots, k$ . The values for  $f$  can be extrapolated, if they are dependent on  $\phi^n$ , or evaluated directly at  $t^n$ , if known. A BDF2/EXT2 formulation is given by

$$\frac{3\phi^n - 4\phi^{n-1} + \phi^{n-2}}{2\Delta t} = \frac{1}{Pe} \nabla^2 \phi^n - (2 \mathbf{c} \cdot \nabla \phi|_{t^{n-1}} - \mathbf{c} \cdot \nabla \phi|_{t^{n-2}}) + f^n. \quad (4)$$

The scheme is globally second-order accurate in time. It is readily extended to  $k$ th-order accuracy by changing the second-order approximation to the time derivative on the left-hand side of (4) and using  $k$ th-order extrapolation for the convection term on the right. As with the CN/AB3 scheme, the cost for this approach is dominated by the work associated with the implicit solve. The added cost of increasing  $k$  is thus essentially nil. Note that it's not appropriate to use AB $k$  coefficients on the right-hand side of (4) because (4) is an approximation to the *value* of the function at  $t^n$ , rather than an approximation to the *integral* of the function over the interval  $[t^{n-1}, t^n]$ .

## 2 Generalization of BDF $k$ /EXT $k$

To extend the the BDF $k$ /EXT $k$  scheme to variable timestep sizes, we first rewrite (4) in a more general form:

$$\sum_{q=0}^k \beta_q \phi^{n-q} = - \sum_{p=1}^k \gamma_p \mathbf{c} \cdot \nabla \phi|_{t^{n-p}} + \mathcal{I}^n, \quad (5)$$

where  $\mathcal{I}^n$  accounts for all the implicit contributions. The terms  $\beta_q$  are the differentiation coefficients associated with the approximation of a derivative at time  $t^n$  based on known values at  $t^{n-q}$ ,  $q = 0, \dots, k$ . The terms  $\gamma_p$  are interpolation coefficients required to approximate a function at time  $t^n$  based on known values at  $t^{n-p}$ ,  $p = 1, \dots, k$ . These coefficients are readily determined from any standard approach to polynomial interpolation at knot points  $[t^{n-k}, \dots, t^{n-j}]$  ( $j = 0$  or  $1$ ). A convenient code is the general-purpose routine for interpolation/differentiation weights, FD\_WEIGHTS, developed by Bengt Fornberg [?]. Note that, because of potential round-off errors, it is probably better for long time integrations to use just the *change* in time values,  $\delta^{n-q} := t^{n-q} - t^n$ , rather than the time values themselves, to determine the interpolation/differentiation coefficients. Of course, the values of  $\delta^{n-q}$  should be computed by summing  $\Delta t^{n-j}$ ,  $j = 0, \dots, q$ , rather than taking the difference  $t^{n-q} - t^n$ .

## 3 Stability Considerations

The explicit treatment of the convection term leads to timestep size restrictions in the CN/AB3 and BDF $k$ /EXT $k$  schemes given in (3)–(5) above. Assuming a closed domain ( $\mathbf{c} \cdot \mathbf{n} \equiv 0$ ), the linear operator will be skew-symmetric in the zero-diffusion limit, Assuming that the skew symmetry of the continuous (in space) operator is preserved in the numerical discretization, we should seek a temporal discretization that encloses a portion of the imaginary axis near the origin. Both AB3 and BDF3/EXT3 accomplish this.

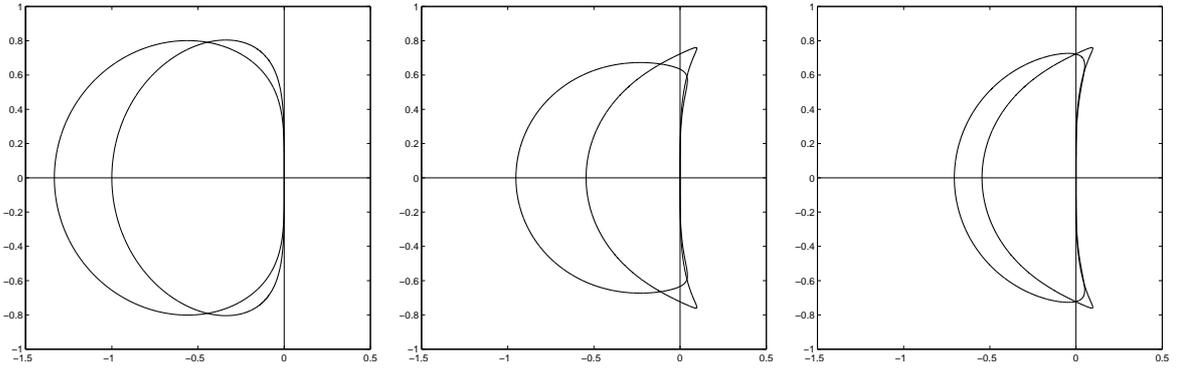


Figure 1: Stability regions for (left) AB2 and BDF2/EXT2, (center) AB3 and BDF3/EXT3, and (right) AB3 and BDF2/EXT2a.

Consider the linear convection problem

$$\frac{\partial \phi}{\partial t} + \mathbf{c} \cdot \nabla \phi = 0, \quad (6)$$

subject to appropriate initial and boundary conditions, and assume that it has been discretized in space to yield

$$\frac{d\phi}{dt} = -C\phi, \quad (7)$$

where  $\phi$  is the vector of unknown basis coefficients and  $C$  is the discrete skew-symmetric convection operator. Discretizing (7) in time using AB3 yields

$$\phi^n = \phi^{n-1} - \frac{\Delta t}{12}(23C\phi^{n-1} - 16C\phi^{n-2} + 5C\phi^{n-3}), \quad (8)$$

where we have assumed that the timestep size,  $\Delta t$ , is constant and that the velocity field  $\mathbf{c}$  and, hence,  $C$ , is time invariant. The stability region for AB3 is found by considering the model problem  $u_t = \lambda u$ , inserting this into (8) ( $\phi \leftarrow u$ ,  $-C \leftarrow \lambda$ ), and solving for the locus of points in the complex  $(\lambda\Delta t)$ -plane such that the magnitude of  $u$  is unchanging (see, e.g., [?] and Appendix A.) Figure 1 shows the stability region for AB3 and for BDF3/EXT3. We see that both schemes enclose a substantial portion of the imaginary axis near the origin. It is also possible to improve the stability of BDF2/EXT2 by using a three-term treatment of the second-order extrapolation. For example, for fixed  $\Delta t$ , the choice  $\gamma_1 = 8/3$ ,  $\gamma_2 = -7/3$ , and  $\gamma_3 = 2/3$ , will provide a stability region that is similar to AB3, as seen in the right panel of Fig. 1.

The implications of Fig. 1 are that, for stability, one must choose  $\Delta t \leq z_{\text{crit}} / \max |\lambda|$ , where  $z_{\text{crit}} = .7236$  for AB3 and  $.6339$  for BDF3/EXT3. The value of  $\max |\lambda|$  depends on the *spatial* discretization and on the computational mesh size. For classical centered finite differences (FD), one has

$$\max |\lambda| = \frac{|\mathbf{c}|}{\Delta x},$$

which leads to the well known Courant-Friedrichs-Lewy (CFL) number

$$CFL := \frac{|\mathbf{c}|\Delta t}{\Delta x} = \max |\lambda|\Delta t.$$

Thus, for FD + AB3, we must have  $CFL \leq .7236$ . For a spatial discretizations based on one-dimensional Fourier methods, the maximum eigenvalue is

$$\max |\lambda| = \pi \frac{|\mathbf{c}|}{\Delta x}$$

[?], which implies that  $CFL \leq .7236/\pi$  is required for stability. For spectral element methods, the bound is

$$\max |\lambda| = S \frac{|\mathbf{c}|}{\Delta x_{\min}},$$

where  $S$  is a value ranging between 1.52 and 1.16 as the polynomial order  $N$  is increased from 3 to  $\infty$  [?]. Note that, for the spectral element method,  $\Delta x_{\min}$  scales as  $O(N^{-2})$  as a result of the clustering of the Gauss-Lobatto-Legendre points near the ends of the reference interval. Schemes that avoid the  $CFL$  restriction are thus of greater interest in the SEM case.

## 4 Characteristics Method

The characteristics scheme due to Pironneau [?] avoids the CFL constraint in the semi-implicit formulation by rewriting the convective term in (2) as a material derivative to yield

$$\frac{D\phi}{Dt} = \frac{1}{Pe} \nabla^2 \phi + f, \quad (9)$$

where

$$\frac{D\phi}{Dt} := \frac{\partial \phi}{\partial t} + \mathbf{c} \cdot \nabla \phi \quad (10)$$

represents the change of  $\phi$  along the path line (or characteristic) associated with the convecting velocity field  $\mathbf{c}$ . The basic idea behind this method is to apply BDF $k$  to (9), to yield (e.g., for  $k=2$ ):

$$\frac{3\phi^n - 4\tilde{\phi}^{n-1} + \tilde{\phi}^{n-2}}{2\Delta t} = \frac{1}{Pe} \nabla^2 \phi^n + f^n. \quad (11)$$

(As before, if  $f^n$  is dependent upon  $\phi^n$ , it can be approximated using EXT $k$ .) In (11),  $\tilde{\phi}^{n-q}$  represents the value of  $\phi$  at an earlier point in time ( $t^{n-q}$ ) and at the point in space,  $\mathbf{X}^{n-q}$ , from which it originates prior to being convected forward by the velocity field. That is,

$$\tilde{\phi}^{n-q}(\mathbf{x}) := \phi(\mathbf{X}^{n-q}(\mathbf{x}), t^{n-q}). \quad (12)$$

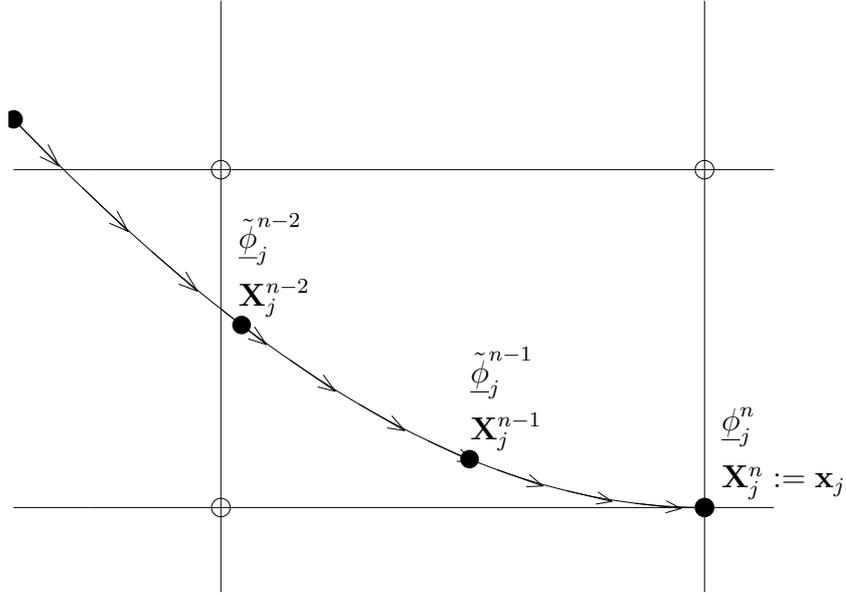


Figure 2: BDF2 approximation of material derivative  $D\phi/Dt$  along the characteristic emanating from  $\mathbf{x}_j$ .  $\mathbf{X}_j^{n-q}$  is the foot of the characteristic corresponding to  $t^n - q\Delta t$ .

Note that the domain of  $\tilde{\phi}^{n-q}$  is coincident with  $\Omega(t^n)$ , whereas the domain of  $\phi(\cdot, t^{n-q})$  must be coincident with  $\Omega(t^{n-q})$ . This is a subtle point, but the implications are that spatial discretization of (11) can proceed with the usual weighted residual formalism, that is, by multiplying (11) by a test function  $v$ , integrating over  $\Omega$ , and restricting the search (trial) and test spaces to finite dimensional subspaces of  $H_0^1(\Omega)$ .

To illustrate the principal implementation aspects of the characteristics scheme, we consider the fully discretized system (in time and space), assuming a nodal basis for  $\phi$  and introducing gridpoints  $\mathbf{x}_j$ . Assume that the spatial discretization of (11) leads to the following linear system

$$\left(-\frac{1}{Pe}A + \frac{3}{2\Delta t}B\right)\underline{\phi}^n = B\underline{g}^n, \quad (13)$$

where  $A$  is the stiffness matrix,  $B$  is the mass matrix, and the right-hand side

$$\underline{g}^n := B\left(\underline{f}^n + \frac{4}{2\Delta t}\tilde{\phi}^{n-1} - \frac{1}{2\Delta t}\tilde{\phi}^{n-2}\right), \quad (14)$$

accounts for the data and the values of  $\phi$  at the feet of the characteristics.

The central question with the characteristics scheme is how to find the values of  $\phi$  at the foot of the characteristic associated with gridpoint  $\mathbf{x}_j$ , that is,

$$\tilde{\phi}_j^{n-q} := \phi(\mathbf{X}_j^{n-q}, t^{n-q}),$$

where  $\mathbf{X}_j^{n-q}$  is the foot of the characteristic emanating from  $\mathbf{x}_j$ , as illustrated in Fig. 2. The standard semi-Lagrangian formulation, suggested by Pironneau and followed by others, is,

- starting at each point  $\mathbf{x}_j$ , march backwards along the velocity characteristic for an amount of time  $q\Delta t$ ,
- evaluate  $\mathbf{X}_j^{n-q}$  (the endpoint of the trajectory in the preceding step)
- then evaluate  $\phi(\mathbf{X}_j^{n-q}, t^{n-q}) := \phi^{n-q}(\mathbf{X}_j^{n-q})$ .

This obviously requires that one interpolate  $\phi^{n-q}(\mathbf{x})$  at points that are not coincident with the gridpoints since, in general, one will not have  $\mathbf{X}_j^{n-q} = \mathbf{x}_{j'}$  for any  $j'$ . In fact, the situation is somewhat worse. In addition to interpolating  $\phi$ , it is necessary to interpolate the vector field  $\mathbf{c}$  at each point along the characteristic, in order to accurately march back along the trajectory. In low-order unstructured methods, most of the interpolation cost arises from determination of the cell containing the point of interest. In high-order methods, one has an additional cost because off-grid interpolation for an  $N$ th-order discretization in  $d$  space dimensions has a complexity of  $O(N^d)$  *per interpolated value*. For a spectral element discretization comprising  $E$  elements of order  $N$ , there are  $\sim EN^d$  gridpoints, and the semi-Lagrangian evaluation cost is thus  $O(EN^{2d})$ . Moreover, the constant is not small, given that several interpolations are required, the Lagrangian bases need to be constructed, and, for deformed geometries, Newton's method must be applied to find the computational coordinates in the reference element as a function of  $\mathbf{X}_j^{n-q}$ . This cost must be contrasted to the cost of Eulerian operator evaluations which have operation count scaling as  $O(EN^{d+1})$  and relatively small constants (e.g., for the spectral element method, one can evaluate  $w = \mathbf{c} \cdot \nabla \phi$  in approximately  $2dEN^{d+1}$  operations).

## 5 The Operator-Integration Factor Scheme

The operator-integration factor scheme (OIFS), due to Maday, Patera, and Rønquist [?] can be viewed as a variant of the characteristics scheme (11) in which the values of  $\tilde{\phi}^{n-q}$  are obtained without using off-grid interpolation. The basic idea is to recognize that to evaluate  $\tilde{\phi}^{n-q}$ , one does not need to know  $\mathbf{X}^{n-q}$  as (12) would seem to indicate. Rather, one can find  $\tilde{\phi}^{n-q}$  by simply solving the following initial-boundary value problem:

$$\begin{aligned} \frac{\partial \tilde{\phi}}{\partial s} + \mathbf{c} \cdot \nabla \tilde{\phi} &= 0, & s \in [t^{n-q}, t^n] \\ \tilde{\phi}(\mathbf{x}, t^{n-q}) &= \phi(\mathbf{x}, t^{n-q}) & \tilde{\phi}(\mathbf{x}, t) = \phi(\mathbf{x}, t) \forall \mathbf{x} \in \partial\Omega_{\mathbf{c}}. \end{aligned} \quad (15)$$

Here,  $\partial\Omega_{\mathbf{c}}$  is defined as the subset of the boundary  $\partial\Omega$  where  $\mathbf{c} \cdot \mathbf{n} < 0$ , that is, the portion of  $\partial\Omega$  having incoming velocity characteristics. (Note that, in practice, one can assign  $\tilde{\phi}(\mathbf{x}, t) = \phi(\mathbf{x}, t^{n-q})$  on  $\partial\Omega_{\mathbf{c}}$ . PF, verify.)

Equation (15) is a pure convection problem. It has the effect of propagating the initial condition (in this case, the values of  $\phi$  at time  $t^{n-q}$ ) forward, along the characteristics of the

convecting field  $\mathbf{c}$ . After a time of  $q\Delta t$ , the values of  $\tilde{\phi}$  at the gridpoints  $\mathbf{x}_j$  are precisely the desired values  $\tilde{\phi}^{n-q}(\mathbf{x}_j) := \phi(\mathbf{X}_j^{n-q}, t^{n-q})$ . These values can be inserted directly into (14). Note that *no* interpolation is required to find  $\tilde{\phi}^{n-q}$  because (15) is solved in the Eulerian framework, using explicit time marching with a stepsize,  $\Delta s \leq \Delta t$ , that is sufficiently small to satisfy the *CFL* constraint. Because of the short duration of the time integration ( $q\Delta t$ ), it is preferable to solve (15) using a multistage integrator such as Runge-Kutta, rather than a multistep integrator such as *ABk*, to avoid the start-up problems associated with lack of information prior to the first time step. Note that if  $\mathbf{c}$  is a function of  $t$  and known only at discrete times  $t^{n-q}$ , then interpolation will need to be used to approximate it at intermediate times (e.g.,  $t^{n-q} + \Delta s$ ,  $t^{n-q} + 2\Delta s$ , etc.). Fornberg's interpolation routine is also useful in this context.

In summary, the OIF scheme involves solving (15)  $k$  times, with different initial conditions and over different time intervals; summing these solutions together with weights  $\beta_q$  to construct the right-hand side (14); and finally solving the Helmholtz problem (13) to advance the solution to the next step.

As it stands, the OIF scheme has a complexity scaling as  $O(k^2)$ , for there are  $k$  integrations to be performed over time intervals ranging from  $\Delta t$  to  $k\Delta t$  in length. For example, suppose that one wishes to use OIFS with BDF3, using a  $\Delta t$ -based CFL of 5. To compute  $\tilde{\phi}^{n-3}$ , one needs to integrate (15) from  $t^{n-3}$  to  $t^n$ , or a total of  $3\Delta t$  in time. Using RK4, whose stability curve intersects the imaginary axis at  $\lambda\Delta t \approx 2i$ , and assuming, with some margin of safety, that the maximum eigenvalue of the spectral element discretization satisfies  $\lambda_{\max}\Delta t \approx 2CFL$ , one should set the substep size  $\Delta s \approx \Delta t/5$ . That is, five RK4 steps, each requiring four function evaluations, would be required for *each*  $\Delta t$  interval covered by the integration. The total number of function evaluations would thus be  $5 \times 4 \times 3 = 60$  for the computation of  $\tilde{\phi}^{n-3}$ . For  $\tilde{\phi}^{n-2}$ , one would have 40, and for  $\tilde{\phi}^{n-1}$ , one would have 20.

This complexity can be reduced to  $O(k)$  by recognizing that (15) is *linear*. Therefore superposition can be used to compute the  $k$  solutions in a single pass. We replace (15) with the following system: For  $q = k, k-1, \dots, 1$ :

$$\begin{aligned} \frac{\partial \psi^q}{\partial s} + \mathbf{c} \cdot \nabla \psi^q &= 0, & s \in [t^{n-q}, t^{n-q+1}] \\ \psi(\mathbf{x}, t^{n-q}) &= \psi^{q+1}(t^{n-q}) + \beta_q \phi(\mathbf{x}, t^{n-q}) & \psi(\mathbf{x}, t) = \phi(\mathbf{x}, t) \quad \forall \mathbf{x} \in \partial\Omega_{\mathbf{c}}. \end{aligned} \quad (16)$$

With  $\psi^{k+1} := 0$ , the final result is

$$\psi^1(t^n) = \sum_{q=1}^k \beta_q \tilde{\phi}^{n-q}, \quad (17)$$

which is the desired contribution to the right-hand side of (14).

## 6 Extension to Navier-Stokes

The extension of the characteristics/OIF method to the Navier-Stokes equations is straightforward. Following the formalism used above for the convection-diffusion equations, we write (1) as

$$\begin{aligned}\frac{Du}{Dt} &= -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega.\end{aligned}\tag{18}$$

The (here nonlinear) convective term has again been expressed as a material derivative. As in the convection-diffusion problem, the objective is to replace the conditionally stable treatment of the convective term with an unconditionally stable BDF $k$  treatment of the material derivative. (Note that large values of  $k$  are of particular interest here since one needs to compensate for accuracy loss resulting from larger values of  $\Delta t$ .) Using this approach, the semidiscretized Navier-Stokes equations for  $k = 2$  are

$$\begin{aligned}\frac{3\mathbf{u}^n - 4\tilde{\mathbf{u}}^{n-1} + \tilde{\mathbf{u}}^{n-2}}{2\Delta t} &= -\nabla p^n + \frac{1}{Re}\nabla^2\mathbf{u}^n \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u}^n &= 0 \quad \text{in } \Omega.\end{aligned}\tag{19}$$

Again, the values  $\tilde{\mathbf{u}}^{n-q}$  represent the field  $\mathbf{u}$  at the foot of the characteristics. These may be computed using the usual semi-Lagrangian formulation involving off-grid interpolation, or by the OIF approach, in which one solves the convective subproblems

$$\begin{aligned}\frac{\partial \tilde{\mathbf{u}}}{\partial s} + \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} &= 0, \quad s \in [t^{n-q}, t^n] \\ \tilde{\mathbf{u}}(\mathbf{x}, t^{n-q}) &= \phi(\mathbf{x}, t^{n-q}) \quad \tilde{\mathbf{u}}(\mathbf{x}, t) = \phi(\mathbf{x}, t) \quad \forall \mathbf{x} \in \partial\Omega_{\mathbf{c}},\end{aligned}\tag{20}$$

where,  $\partial\Omega_{\mathbf{c}}$  is defined as the subset of the boundary  $\partial\Omega$  where  $\mathbf{u} \cdot \mathbf{n} < 0$ .

Note that there should be no confusion in (20) between the *convected* field,  $\tilde{\mathbf{u}}$ , and the *convecting* field,  $\mathbf{u}$ , which is the solution to (19). In solving (20), one treats  $\tilde{\mathbf{u}}$  as a passive advected vector field. Values of  $\mathbf{u}$  on the interval  $[t^{n-k}, t^n]$  are computed by interpolating from the known fields  $\mathbf{u}^{n-k}, \mathbf{u}^{n-k+1}, \dots, \mathbf{u}^{n-1}$ , which guarantees a divergence-free convecting field in (20). Unlike the convection-diffusion problem, however, one does not have access to  $\mathbf{u}^n$ , so  $\mathbf{u}$  is effectively *extrapolated* on  $(t^{n-1}, t]$ . One has the option of iterating between (20) and (19) to get an estimate of  $\mathbf{u}^n$ , which can then be employed in the interpolation step required when solving (20) on a second (third, etc.) pass. Formally, this should not be required, as the extrapolation-based scheme is  $O(\Delta t^k)$  accurate. However, it may be of interest in particularly challenging applications. If properly coded, (e.g., by relying on general interpolation routines such as Fornberg's), this iterative approach can be included as an option in the OIFS implementation from the outset with little overhead.

## Remark on Steady State Error

We comment that one advantage of the BDF $k$ /EXT $k$  formulation is that there is no temporal error for problems that have evolved to a steady state solution. That is, the final result (and error) is independent of  $\Delta t$ . To see this, consider (4) under steady-state conditions,  $\phi^n = \phi^{n-1} = \dots$ . The lefthand side of (4) vanishes identically and represents the only term containing  $\Delta t$ . The righthand side vanishes through a balance of the diffusion and advection terms that are independent of  $\Delta t$  and the error will (ultimately) reflect only the spatial discretization parameters.

On the other hand, the OIFS formulation *does* retain a steady state error that is influenced by  $\Delta t$ . This is evident from (11). Under steady state conditions, we again have  $\phi^n = \phi^{n-1} = \dots$ , but these are not the terms that are required to balance. The lefthand side of (11) involves weighted differences of  $\phi^n$ ,  $\tilde{\phi}^{n-1}$ , and  $\tilde{\phi}^{n-2}$  that are divided by  $\Delta t$ . In fact, there should be no expectation that the lefthand side should vanish, even as  $\Delta t \rightarrow 0$ , because the lefthand side represents the variation of  $\phi$  as one travels along the characteristic. This material derivative will generally be nonzero and the temporal finite difference scheme will retain an  $O(\Delta t)$  dependence even under steady-state conditions. The characteristics-based OIFS scheme amounts to mixing spatial and temporal discretizations and care must be exercised when implementing such methods, particularly in a high-order context.

## Appendix A: Stability Region Evaluation

The matlab code used to generate Fig. 1 is presented below. The neutral stability curves are generated by considering application of a particular timestepping scheme to the homogeneous model problem

$$\frac{du}{dt} = \lambda u.$$

Assume that a uniform stepsize,  $\Delta t$ , is employed and that the timestepping scheme is of the form:

$$\frac{1}{\Delta t} \sum_{q=0}^{k_1} \beta_q u^{n-q} = \lambda \sum_{p=1}^{k_2} \gamma_p u^{n-p}. \quad (21)$$

Eq. (21) has solutions of the form

$$u^n = (z)^n u^0, \quad (22)$$

where  $(.)$  is used to indicate that the (generally complex) argument  $z$  is raised to the  $n$ th power. In order for (22) to be neutrally stable, we require  $|z| = 1$ , that is,  $z = e^{i\theta}$  for some value of  $\theta$  between 0 and  $2\pi$ . Inserting (22) into (21) and solving for  $\lambda\Delta t$  yields

$$\lambda\Delta t = \frac{\sum_{q=0}^{k_1} \beta_q e^{-iq\theta}}{\sum_{p=1}^{k_2} \gamma_p e^{-ip\theta}}. \quad (23)$$

Evaluating (23) for  $0 \leq \theta < 2\pi$  yields the locus associated with the neutral stability curve.

The matlab code below fills a vector `ei` with complex values on the unit circle, then creates column vectors `exp(q*ei)` for `q=1,0,-1,-2`. Linear combinations of these column vectors form the numerator and denominator of (23). The matlab `plot` routine is used to plot the locus of complex pairs.

## Matlab Code

```

ii=sqrt(-1);
th=0:.001:2*pi; th=th';
ith=ii*th;
ei=exp(ith);
E = [ ei 1+0*ei 1./ei 1./(ei.*ei) ];

ep = 1.e-13

ab0 = [1 0.0 0.0 0.]';
ab1 = [0 1.0 0.0 0.]';
ab2 = [0 1.5 -.5 0.]';
ab3 = [0 23./12. -16./12. 5./12.]';
bdf1 = (([ 1. -1. 0. 0.])/1.)';
bdf2 = (([ 3. -4. 1. 0.])/2.)';
bdf3 = (([11. -18. 9. -2.])/6.)';
ex0 = [0 1 0 0]';
ex1 = [0 2 -1 0]';
ex2 = [0 3 -3 1]';
yaxis = [-1.0*ii 1.0*ii]';
xaxis = [-2.0+ep*ii 2.0+ep*ii]';
du = [1. -1. 0. 0. ]';

hold off; plot (yaxis,'k-'); axis square; axis([-1.5 0.5 -1 1]); hold on;
                                                    plot (xaxis,'k-');
ab3 = (E*du)./(E*ab3);                               plot (ab3 , 'r-');
bdf3ex2 = (E*bdf3)./(E*ex2);                         plot (bdf3ex2,'k-');
print -deps ab3bdf3.eps

hold off; plot (yaxis,'k-'); axis square; axis([-1.5 0.5 -1 1]); hold on;
                                                    plot (xaxis,'k-');

```

```
ab2      = (E*du)./(E*ab2);
bdf2ex1  = (E*bdf2)./(E*ex1);
print -deps ab2bdf2.eps

plot (ab2 , 'r-');
plot (bdf2ex1, 'k-');
```