Available online at www.sciencedirect.com
ScienceDirect
Ocean Modelling xxx (2007) xxx–xxx
Ocean Modelling
www.elsevier.com/locate/ocemod

Semi-Lagrangian methods for a finite element coastal ocean model

Roy A. Walters [a,*], E.M. Lane [b], R.F. Henry [c]

[a] National Institute for Water and Atmospheric Research, P.O. Box 8602, Christchurch, New Zealand
[b] National Institute for Water and Atmospheric Research, Christchurch, New Zealand
[c] Triton Consultants Ltd., Vancouver, BC, Canada

Received 8 March 2007; received in revised form 15 May 2007; accepted 25 June 2007

Abstract

Coastal ocean hydrodynamic models are subject to a number of stability constraints. The most important of these are the Courant–Friedrichs–Levy (CFL) constraint on gravity waves, a Courant (*Cr*) number constraint on advection, and a time step constraint on the vertical component of viscous stresses. The model described here removes these constraints using a semi-implicit approximation in time and a semi-Lagrangian approximation for advection. The accuracy and efficiency of semi-Lagrangian methods depends crucially on the methods used to calculate trajectories and interpolate at the foot of the trajectory. The focus of this paper is on evaluation of several new and old semi-Langrangian methods. In particular, we compare 3 methods to calculate trajectories (Runge–Kutta (RK2), analytical integration (AN), power-series expansion (PS)) and 3 methods to interpolate (local linear (LL), global linear (GL), global quadratic (GQ)) on unstructured grids. The AN and PS methods are both efficient and accurate, and the latter can be expanded in a straightforward manner to treat time-dependent velocity. The GQ interpolation method provides a major step in introducing efficient and accurate semi-Lagrangian methods to unstructured grids.
© 2007 Published by Elsevier Ltd.

Keywords: Numerical model; Finite element; Semi-Lagrangian; Coastal ocean

1. Introduction

Coastal ocean models operate over a wide range of time and space scales, some that can be resolved and some that cannot. Typical time scales are of the order of the period of long waves (tens of minutes) and longer. Typical flow features range from tsunami propagation to seasonal baroclinic circulation. Embedded in these flows are short gravity waves which may or may not be of interest.

These models are also subject to several stability constraints. The most important of these are the Courant–Friedrichs–Levy (CFL) constraint on gravity waves, a Courant (*Cr*) number constraint on advection, and a time step constraint on the vertical component of horizontal viscous stresses.

* Corresponding author. Tel.: +64 3 348 8987; fax: +64 3 348 5548.
E-mail addresses: r.walters@niwa.co.nz (R.A. Walters), e.lane@niwa.co.nz (E.M. Lane), rfhenry@shaw.ca (R.F. Henry).

1463-5003/$ - see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.ocemod.2007.06.008

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

UNCORRECTED PROOF

In the end, it is desirable to formulate the discrete equations so that there are no major stability constraints. Thus there would be no need for mode splitting to accommodate fast gravity waves as an example. The traditional way to eliminate time step constraints on gravity wave and viscous terms is to use a semi-implicit approach in time. Treating the advection terms is more problematic.

The stability of many explicit advection schemes is limited by Cr. Ideally, calculations that are stable at higher Cr but do not incur significant overhead are desired and Leonard (2002) shows how such schemes can be formulated. On the other hand, implicit schemes will involve the solution of a large matrix equation for the 3-dimensional velocity and thus have a large computational cost. One of the few stable methods available that does not require a matrix solution is a semi-Lagrangian approximation. This approximation is essentially explicit, removes the stability constraint, and does not necessarily involve a high computational cost.

Semi-Lagrangian methods take advantage of their simple formulation on a fixed grid and the inherent accuracy of integrating along streamlines. The equations are integrated in such a way that the trajectory ends at a fixed node at the end of each time step. Robert (1981, 1982) developed semi-implicit, semi-Lagrangian methods in a set of seminal papers and an extensive review of these methods for application to atmospheric problems is found in Staniforth and Côté (1991). Semi-Lagrangian schemes for the shallow water equations have been analysed in several papers (Casulli, 1987, 1990) where tracking methods, stability, artificial viscosity, and interpolation methods are considered. In an oceanic context, these methods have been used by Walters and Casulli (1998), Casulli and Walters (2000), Hanert et al. (2005), and Ham et al. (2005), among others.

Semi-Lagrangian methods seem ideal; however, the proper choice of tracking and interpolation methods is crucial for maintaining accuracy. Although high-order methods are well established for structured grids with regular quadrilaterals, they are difficult to implement on an unstructured grid (Staniforth and Côté, 1991; Le Roux et al., 1997). In a recent paper dealing with unstructured grids, Hanert et al. (2005) compared an explicit upwind scheme with a semi-Lagrangian scheme that uses a kriging interpolator. As an indication of efficiency, the semi-Lagrangian calculations were about ten times more expensive than the Eulerian calculations and both gave acceptable results for the large-scale test problems (Hanert et al., 2005). Hence, there is a need to increase the efficiency of these methods while maintaining accuracy.

The general problem is to solve $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{u}$ for a time step Δt to define the trajectory, then evaluate a function G at the foot of the trajectory. Because of the influence of coastal geometry, the streamlines tend to be curved. Hence, the methods of choice track the streamline rather than make a straight-line approximation such as used by Staniforth and Côté (1991) and Hanert et al. (2005). We consider three such methods in this paper.

Interpolating on an unstructured grid in an accurate and efficient manner is somewhat of a greater problem. Low-order methods tend to introduce considerable damping on the solution (McCalpin, 1988; Malcherek, 2001). On the other hand, high-order methods are difficult to implement and can be very inefficient (Hanert et al., 2005). One high-order method that requires high-order elements is presented in Xiu and Karniadakis (2001). We present here two low-order (linear) methods and compare these with a new high-order method.

In the following section, an overview of the numerical model is set forth with the details contained in the cited references. Next, the semi-Lagrangian methods are described in more detail including both tracking and interpolation. In Section 4, we present some comparative results for the different methods employed here. In Section 5, we summarise the conclusions.

2. Governing equations

The numerical model is formulated from the Reynolds-averaged Navier-Stokes equations (RANS) with a free surface. For the simulation of weakly dispersive surface waves, these equations can be averaged over the water depth to derive a set of equations similar to the standard shallow water equations but containing additional terms that describe non-hydrostatic forces (Walters, 2005). This procedure can be extended to solve the full Navier-Stokes equations.

However, for the purposes of this study a simplified set of equations are used to investigate different semi-Lagrangian advection approximations. In particular, the general equations are depth-averaged and the hydrostatic and Boussinesq approximations are used. The resulting equations are then a simple form of the shallow water equations.

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

UNCORRECTED PROOF

The equation for the free surface η is derived by an integration of the continuity equation over water depth and application of the kinematic free surface and bottom boundary conditions:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H\mathbf{u}) = 0, \tag{1}$$

where $\eta(x, y, t)$ is the water surface elevation measured from the vertical datum, ∇ is the horizontal gradient operator, $h(x, y)$ is the land elevation, and $H = \eta(x, y, t) - h(x, y)$ is the water depth. The variable t is time and the coordinate system (x, y, z) is directed eastward, northward, and up, respectively.

After depth averaging, the horizontal momentum equation becomes

$$\frac{D\mathbf{u}}{Dt} + \mathbf{f} \times \mathbf{u} = -g\nabla\eta - \frac{\boldsymbol{\tau}_\mathrm{b}}{\rho H}, \tag{2}$$

where D/Dt is the material derivative given by $D/Dt = \partial/\partial t + \mathbf{u} \cdot \nabla$, $\mathbf{u}$ is the depth-averaged velocity, $\mathbf{f} = 2\Omega_\mathrm{E} \sin\theta \hat{\mathbf{z}}$ is the Coriolis parameter, Ω_E is the rotation rate of the Earth, θ is the latitude, g is gravitational acceleration, ρ is density, and $\boldsymbol{\tau}_\mathrm{b}$ is the bottom friction. Surface stress and atmospheric pressure have been neglected, along with the horizontal friction. Bottom friction is written as

$$\frac{\tau_b}{\rho H} = \frac{C_\mathrm{D}|\mathbf{u}|\mathbf{u}}{H} = \gamma\mathbf{u}, \tag{3}$$

where C_D is a drag coefficient and γ is defined by (3).

At open (sea) boundaries, either a specified surface elevation or a radiation condition is enforced so that the outgoing wave will not reflect back into the modelled area.

2.1. Spatial discretisation

Unstructured grids are chosen because of their ability to resolve the complicated and multi-scale geometry of coastal environments. Finite elements with unstructured triangular and quadrilateral elements of varying-size and shape are used for the spatial approximation. The specific element used is the Raviart–Thomas element of lowest order (RT0) (Raviart and Thomas, 1977; Abrogast, 1995). The basic computational grid is a tessellation using arbitrary quadrilaterals or triangles in the horizontal, and the projection of the nodes (corners) in the vertical to derive brick or pie shaped elements for the 3-dimensional form of the model.

For this element, the continuous variables η and $\mathbf{u}$ are approximated with the discrete variables η^h and $\mathbf{u}^h$ where $\eta^h(\mathbf{x}, t) = \sum_{j=1}^{ne} \varphi_j(\mathbf{x})\eta_j(t)$ and $\mathbf{u}^h(\mathbf{x}, t) = \sum_{i=1}^{ns} \boldsymbol{\Phi}_i(\mathbf{x})u_{ni}(t)$, where j is the index for an element in the global vector, the basis function $\varphi_j = 1$ on element j and 0 otherwise, ne is the number of elements, ns is the number of element edges (sides), u_{ni} is the normal velocity u_n on edge i, and $\boldsymbol{\Phi}_i$ is the linear vector basis function for velocity. $\boldsymbol{\Phi}_i$ is given by $\boldsymbol{\Phi}_i(\mathbf{x})|_{\Omega_e} = \frac{\mathbf{x}-\mathbf{x}_i}{L_e}$ on element Ω_e where $\mathbf{x}_\mathrm{i}$ is the vertex opposite to edge i and L_e is the normal distance from edge i to $\boldsymbol{x}_i$.

Using this element, the free surface equation is expressed in a standard Galerkin weighted residual form (e.g. Hanert et al., 2003). Because the basis function for is piecewise constant, the equation reduces to a finite volume form that conserves mass both locally and globally (Walters and Casulli, 1998).

$$A_e \frac{\partial \eta_e}{\partial t} + \oint_{\Gamma_e} (Hu_n)\mathrm{d}\Gamma_e = 0 \tag{4}$$

where subscript e denotes the value for a specific element; A_e is the element area; u_n is the normal velocity on an edge, positive outwards; and Γ_e is the boundary of the element. The last term has been converted from a divergence form to a line integral using the Gauss Divergence Theorem.

The approximation for the momentum equation is:

$$\int_\Omega \boldsymbol{\Phi} \cdot \left(\frac{D\mathbf{u}^h}{Dt} + \gamma\mathbf{u}^h + \mathbf{f} \times \mathbf{u}^h \right) \mathrm{d}\Omega = \int_\Omega g\eta^h \nabla_h \cdot \boldsymbol{\Phi}\mathrm{d}\Omega - \oint_\Gamma [g\eta^h \boldsymbol{\Phi} \cdot \hat{\mathbf{n}}]\mathrm{d}\Gamma \tag{5}$$

where Γ is the boundary of the computational domain Ω, and the pressure gradient term has been integrated by parts. The line integral in these equations provides a convenient means to specify the boundary conditions on η. This equation is developed further below.

UNCORRECTED PROOF

2.2. Time discretisation

Both explicit and implicit methods can be used in the solution of Eqs. (4) and (5). By definition, semi-implicit means that some of the terms in the equations are treated implicitly and some explicitly. Implicit will mean that a term is approximated in the time interval [N, N+1] by the weight θ where $F^{N+\theta} = \theta F^{N+1} + (1-\theta)F^N$ such that $\theta = 0$ is an explicit approximation and $\theta = 1$ is a fully-implicit approximation.

The objective is to choose methods in such a way as to enhance model accuracy, remove restrictive stability constraints, and enhance model efficiency by removing computational overhead. All of these cannot be optimized simultaneously; nonetheless, certain choices are very advantageous (Casulli, 1990; Casulli and Cattani, 1994) and are described below.

For explicit methods, the two most restrictive stability constraints are the CFL condition for gravity waves and the viscosity constraint on the vertical component of the viscous terms. Hence the divergence term in (4) and the gravity and vertical viscosity terms in (5) are treated implicitly thereby removing the stability constraints (Casulli and Cattani, 1994). Note however that there are still accuracy constraints that limit the magnitude of the CFL number ($c\Delta t/\Delta x$, where c is the phase speed, Δt is the time step, and Δx is an edge length). In general, the semi-implicit approach is useful for large-scale, long period simulations such as wind-driven and tidal motions and eliminates the need for mode splitting in baroclinic calculations. On the other hand, the approach is less useful for short period gravity waves such as tsunami where the time step must be reduced to resolve the motions. Even so, for complicated field-scale domains there are frequently parts of the grid that limit stability of explicit methods so that the semi-implicit method is more robust and usually more efficient.

Using a semi-implicit approach for the equations, (4) becomes

$$A_e \frac{\eta_e^{N+1} - \eta_e^N}{\Delta t} + \theta \oint_{\Gamma_e} H^{N+1} u_n^{N+1} \mathrm{d}\Gamma = -(1-\theta) \oint_{\Gamma_e} H^N u_n^N \mathrm{d}\Gamma \tag{6}$$

which expresses the change in surface elevation as a function of the fluxes through the element sides. In most cases, H^{N+1} is approximated as H^N because η is usually small compared to depth. For cases where this is not true, the time step can be reduced if necessary to give more accurate results, or a simple explicit predictor can be used for H^{N+1}. Other options include the logarithmic form of Le Roux et al. (2000). The value for η on an edge is computed several ways, depending on the local flow. Some form of upwind weighting is normally used, such as the methods described in Walters and Casulli (1998), Casulli and Walters (2000) and Stelling and Duinmeijer (2003).

Next consider the weighted residual form of the momentum equation. Integrating (5) by a midside quadrature rule on each element with equal weighting on each edge, and using a discrete time operator for the material derivative, the momentum equation becomes (Walters and Casulli, 1998)

$$A u_n^{N+1} = G_n - \theta \Delta t N_n^{N+1} \tag{7}$$

where

$$A = M(1 + \gamma^N \Delta t)$$
$$G_n = M u_n^* - F_n^* - (1-\theta)\Delta t N_n^*$$
$$N_n = -\int_\Omega g\eta^h \nabla_n \cdot \mathbf{\Phi} \mathrm{d}\Omega + \oint_\Gamma g\eta^h \mathbf{\Phi} \cdot \hat{\mathbf{n}} \mathrm{d}\Gamma$$

and * denotes a value evaluated at the foot of the Lagrangian trajectory, F_n contains the explicit and Coriolis terms, and M is the mass matrix given by $M = \int_\Omega \mathbf{\Phi} \cdot \mathbf{\Phi}^{\mathrm{T}} \mathrm{d}\Omega$. Note that the mass matrix is diagonal when using numerical integration at the midpoints of the edges. The Coriolis term is integrated by a two-step procedure described in Casulli and Walters (2000).

In order to make the model more efficient, Eq. (7) is used to eliminate u_n^{N+1} from the free surface Eq. (6). The resulting equation is in the form of a wave equation at the discrete level and contains only η at the $N+1$ time level.

$$A_e \eta_e^{N+1} - \theta^2 \Delta t^2 \oint_{\Gamma_e} H^N A^{-1} N_n^{N+1} \mathrm{d}\Gamma = A_e \eta_e^N - (1-\theta)\Delta t \oint_{\Gamma_e} H^N u_n^N \mathrm{d}\Gamma - \theta \Delta t \oint_{\Gamma_e} H^N A^{-1} G_n \mathrm{d}\Gamma \tag{8}$$

UNCORRECTED PROOF

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

In practice, Eq. (8) is assembled and solved for η^{N+1}. Using these results, Eq. (7) is solved for u_n^{N+1}. The full velocity is recovered by calculating the velocity at the vertices of each element, then interpolating the tangential component of velocity at the midsides (Walters and Casulli, 1998).

Note that in the discrete wave Eq. (8), water depth H is a factor in all the side flux terms. When $H = 0$ (i.e. the side is dry), there is automatically no water flux through that side. When all sides of an element are dry, the water level is stationary in time. Hence, wetting and drying are implemented without any special treatment.

3. Semi-Lagrangian approximation

The semi-Lagrangian method has two crucial steps: calculation of the trajectory of a fluid particle from a node in the Eulerian grid backwards to its starting location at the beginning of the time step (referred to as the foot of the trajectory), and interpolation of a function at this starting location. The purpose here is to present and evaluate some old and new methods for calculating trajectories and a new high-order interpolation method.

3.1. Trajectory calculations

Four methods were originally considered for calculating the trajectory location: a simple explicit method originally used in the numerical model, a second-order Runge–Kutta method, an analytical integration of the equation for the trajectory, and a power-series approximation method. After some initial tests, the explicit method was dropped from testing as it was too inaccurate and led to noisy solutions.

3.1.1. Runge–Kutta method (RK2)

The second-order Runge–Kutta algorithm chosen here is:

$$x_{j+1} = x_j + \Delta t \mathbf{u}(x_j + (\Delta t/2)\mathbf{u}(x_j))$$

The basic procedure is to integrate backwards to find the departure point at the foot of the trajectory (x_1, y_1) which arrives at the grid node (x_0, y_0) after a time Δt, which is the model time step. If the point (x_1, y_1) is within the element, then immediately interpolate within the element. If the point (x_1, y_1) is outside the element, then reduce Δt to substeps $\Delta t/m$ and find the exit point from the element and the time required to traverse the element. Then repeat the procedure for the next element until the model time step is used up and interpolate at the endpoint. The number of substep intervals m varies in the range of 10–50. The intersection of the trajectory and the element edge is found by an explicit step from the last substep that lies within the element.

The major overhead with this method is a single interpolation of the velocity for each iteration and determining the location of the exit point on each element. The accuracy is highly dependent on the number of substeps and the variability of the velocity on an element.

3.1.2. Analytical integration (AN)

Since velocity varies linearly over an element it can be represented as

$$\mathbf{u} = \mathbf{A}\mathbf{x} + \mathbf{b}$$

where $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$. The form of the solution then depends on the form of matrix $\mathbf{A}$. Matrix $\mathbf{A}$ can be classified into seven different types according to its eigenvalues and eigenvectors:

1. Two real distinct nonzero eigenvalues.
2. One real nonzero eigenvalue with two separate eigenvectors. i.e. $\mathbf{A} = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}$.
3. One real nonzero eigenvalue with one root (defective matrix).
4. Two real distinct eigenvalues, one nonzero, one zero. (Singular matrix).
5. Zero eigenvalue, two separate eigenvectors. i.e. $\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ (zero matrix).
6. Zero eigenvalue, one eigenvector (singular, defective matrix).
7. Complex conjugate pair of eigenvectors.

UNCORRECTED PROOF

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

In the cases where there are real eigenvalues and the full complement of eigenvectors the matrix can be decomposed into the Jordan form, viz.

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \text{ where } \mathbf{V} = [\,\mathbf{v}_1 \quad \mathbf{v}_2\,] \text{ and } \mathbf{D} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix},$$

whereas for defective matrices the Schur decomposition is used, viz.

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^{H}, \text{ where } \mathbf{U} \text{ is unitary and } \mathbf{D} = \begin{pmatrix} \lambda_1 & \alpha \\ 0 & \lambda_1 \end{pmatrix},$$

and H represents the Hermitian transpose (Stoer and Bulirsch, 1993).

These decompositions aid in the calculation of $e^{-\mathbf{A}dt}$ and $\mathbf{A}^{-1}$, which are used below, i.e.

$$e^{-\mathbf{A}\Delta t} = \mathbf{V}\begin{pmatrix} e^{-\lambda_1\Delta t} & 0 \\ 0 & e^{-\lambda_2\Delta t} \end{pmatrix}\mathbf{V}^{-1} \text{ and } \mathbf{A}^{-1} = \mathbf{V}\begin{pmatrix} 1/\lambda_1 & 0 \\ 0 & 1/\lambda_2 \end{pmatrix}\mathbf{V}^{-1},$$

for the Jordan form and

$$e^{-\mathbf{A}\Delta t} = \mathbf{U}\begin{pmatrix} e^{-\lambda_1\Delta t} & -\alpha\Delta t e^{-\lambda_1\Delta t} \\ 0 & e^{-\lambda_1\Delta t} \end{pmatrix}\mathbf{U}^{H} \text{ and } \mathbf{A}^{-1} = \mathbf{U}\begin{pmatrix} 1/\lambda_1 & -\alpha/\lambda_1 \\ 0 & 1/\lambda_1 \end{pmatrix}\mathbf{U}^{H},$$

for the Schur decomposition.

Assuming the arrival point is $\mathbf{x}_0$, we wish to find the departure point $\mathbf{x}_1$ where it left from a time Δt ago. For non-singular $\mathbf{A}$ with real eigenvalues (i.e. classes 1, 2, or 3),

$$\mathbf{x}_1 = e^{-\mathbf{A}\Delta t}\mathbf{x}_0 + \mathbf{A}^{-1}(e^{-\mathbf{A}\Delta t} - \mathbf{I})\mathbf{b}.$$

If $\mathbf{A}$ is singular but has one nonzero eigenvalue, λ, then

$$\mathbf{x}_1 = \mathbf{V}e^{-\mathbf{D}\Delta t}\mathbf{V}^{-1}\mathbf{x}_0 + \mathbf{V}\begin{pmatrix} \frac{1}{\lambda}(e^{-\lambda\Delta t} - 1) & 0 \\ 0 & -\Delta t \end{pmatrix}\mathbf{V}^{-1}\mathbf{b}.$$

For $\mathbf{A} = \mathbf{0}$, $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{b}\Delta t$. For a singular, defective matrix

$$\mathbf{x}_1 = \mathbf{U}e^{-\mathbf{D}\Delta t}\mathbf{U}^{H}\mathbf{x}_0 + \mathbf{U}\begin{pmatrix} -\Delta t & \alpha\Delta t^2/2 \\ 0 & -\Delta t \end{pmatrix}\mathbf{U}^{H}\mathbf{b}.$$

For complex conjugate eigenvalues, $\lambda_R \pm i\lambda_I$, with eigenvectors, $\mathbf{v}_R \pm i\mathbf{v}_I$, respectively $\mathbf{x}_1$ can be calculated as

$$\mathbf{x}_1 = \mathbf{S}\mathbf{G}^{-1}(\Delta t)\mathbf{S}^{-1}\mathbf{x}_0 - \mathbf{S}\mathbf{H}(\Delta t)\mathbf{S}^{-1}\mathbf{b},$$

where

$$\mathbf{S} = (\,\mathbf{v}_R \quad \mathbf{v}_I\,),$$

$$\mathbf{G}(t) = e^{\lambda_R t}\begin{pmatrix} \cos(\lambda_I t) & \sin(\lambda_I t) \\ -\sin(\lambda_I t) & \cos(\lambda_I t) \end{pmatrix},$$

$$\mathbf{G}^{-1}(t) = \mathbf{G}(-t) \text{ and}$$

$$\mathbf{H}(t) = \int_0^t \mathbf{G}^{-1}(\tau)\mathrm{d}\tau.$$

The basic procedure is to find (x_1, y_1) for the case Δt, the model time step. If the point (x_1, y_1) is within the element, then interpolate within the element. If the point (x_1, y_1) is outside the element, then iterate, reducing Δt, to find the entry point and determine Δt_t which is the time taken to traverse the element. Then repeat the procedure for the next element until the model time step is used up and interpolate at the departure point $\mathbf{x}_1$. A Newton–Raphson method was also tried to find where the trajectory entered the element and the corresponding value for Δt, but this suffered from the difficulty of determining which side of the element the trajectory entered through. The first method was found to be more robust.

The major overheads with this method are in classifying and decomposing matrix $\mathbf{A}$, evaluating the transcendental functions and iterating the entry point when multiple elements were traversed.

UNCORRECTED PROOF

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

Ham et al. (2006) have also developed an analytical integration approach which seems to differ mostly in the treatment of inter-element crossings.

3.1.3. Power-series approximation (PS)

A power-series approximation is given by expanding the coordinate variables as a power-series in time about the arrival point (x_0, y_0):

$$x(t) = x_0 + p_1 t^1 + p_2 t^2 + \cdots + p_n t^n$$
$$y(t) = y_0 + q_1 t^1 + q_2 t^2 + \cdots + q_n t^n$$

The velocity is given as a linear function on each element:

$$u(x,t) = a_{11}x(t) + a_{12}y(t) + b_1$$
$$v(x,t) = a_{21}x(t) + a_{22}y(t) + b_2$$

Then expand $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{u}$ in terms of these expressions. Matching powers of t, it follows that $p_1 = u_0 = b_1$ and $q_1 = v_0 = b_2$, and the following recursion relation holds:

$$p_j = \frac{a_{11}}{j}p_{j-1} + \frac{a_{12}}{j}q_{j-1}$$
$$q_j = \frac{a_{21}}{j}p_{j-1} + \frac{a_{22}}{j}q_{j-1}$$

The series converges for all finite t (Pearson, 1974, p. 84).

This approach is applied by first evaluating the location of the departure point for the model time step Δt. Similar to the case of analytical integration, if the location is outside the element, the time step is repeatedly reduced by a factor of 2 until the point is within the element. Then half of the last time step change is added or subtracted from the estimated time depending whether the point is inside or outside the element. Eventually this converges to the entry point on an edge.

The major overhead with this method is calculating the velocity gradients on each element in order to determine the coefficients (a_{ij}, b_i, $i, j = 1, 2$), and finding the intersection of the trajectory and the element edge when the trajectory passes out of the element. The determination of the intersection can compromise the inherent speed of this method as it involves repeatedly reassembling the time series as part of an iteration loop. Obviously, using an iterative procedure that converges faster is desirable.

3.2. Interpolation

Three methods were considered for interpolating within the elements: a local linear approximation based on the finite element approximation within each element, a global linear approximation based on a global reconstruction of the fields, and a global quadratic approximation.

3.2.1. Local linear approximation (LL)

The normal component of the momentum equation is defined at the midpoint of the element edges. From the normal values of a function and the direction of the normal on the edges, a local linear vector field can be uniquely determined for that function. In this manner, the function G_n in (7) is defined at the vertices and the standard area coordinates are used to interpolate the function at any point within the element.

However, the normal component is constant along each edge as a consequence of the definition of the bases for the RT0 element. Thus the tangential variation of the function is not included in the approximation which in a sense is then incomplete. Although not explored here, this limitation appears to cause jets and small scale eddies in field-scale simulations.

3.2.2. Global linear approximation (GL)

As an alternative, the vector field for G_n can be approximated at the element vertices and a standard linear interpolation used on each element. Although there are a variety of approximations that can be used, we use a

UNCORRECTED PROOF

standard weighted residual statement to project the variables with RT0 bases to linear bases at the vertices. This can be written

$$\int_\Omega \Psi\Psi^T\hat{\mathbf{G}}\mathrm{d}\Omega = \int_\Omega \mathbf{\Phi}\cdot\mathbf{\Phi}^T\mathbf{G}\mathrm{d}\Omega$$

where Ψ are the linear bases (hat functions) for the vertices, $\hat{\mathbf{G}}$ is the vector of values of $\mathbf{G}$ at the vertices, Φ are the bases for the RT0 elements, and G is the vector of values for G_n.

Linear interpolation has been found to be highly dissipative (Staniforth and Côté, 1991; McCalpin, 1988; Malcherek, 2001). Moreover, this approximation does not reproduce the exact values for $\mathbf{G}$ at the midpoints on the element edges. These errors cause unwanted accelerations in the momentum equations.

3.2.3. Global quadratic approximation (GQ)

This approximation was developed to satisfy two criteria: to reproduce the values of a function $\mathbf{G}$ accurately at the midside of the edges where the momentum equations are discretised, and to provide a higher order approximation that has minimal dissipation. The procedure is to first calculate the values for $\mathbf{G}$ at the vertices in the same manner as for the global linear approximation. Then quadratic bases are used for interpolation on the element with the midside values specified by G_n.

In retrospect, this seems like a natural way to make this approximation and takes advantage of the discretisation on the RT0 element. A similar method could also be applied to the $P_1^{nc} - P_1$ element, which also has midside velocity nodes. On the other hand, it is not obvious how to extend this to yet higher-order without extensive multi-element approximations.

4. Results

The objective of the first set of test problems is to evaluate the accuracy and efficiency of the three different methods for calculating the trajectory location. Following this, two examples illustrate the accuracy and efficiency of these methods for a gravity wave (tsunami) problem at low *Cr*, and a flood problem with a large *Cr* and Froude number (*Fr*).

4.1. Trajectory tests-accuracy and efficiency

The first set of test cases is designed to evaluate the accuracy and relative efficiency of the different methods for calculating the trajectory location. A square basin is constructed using square elements and a rotating flow is specified where $u = -y$ and $v = x$. The basin extends from -2 to 2 in both x and y. In the first case, one element is used for the entire domain so the trajectories are contained entirely in one element (no inter-element crossings). The time step is adjusted to give a rotation of $\pi/2$, π, and 2π from the initial point. This result gives a measure of the relative execution times and accuracy for the different methods. In the second test, four elements are used and the time step was adjusted so the trajectory crossed 0, 1, 2, or 3 elements (rotation of $\pi/4$, $3\pi/4$, $5\pi/4$, and $7\pi/4$). These results give a measure of the relative performance when intersection of the trajectories with the element edges is included.

For the first test case, which has no element crossings, the execution time for AN was independent of the time step size. On the other hand, the execution time for PS increased with time step size because the number of terms in the time series increased (11 for a rotation of $\pi/2$, 16 for a rotation of π, and 25 for a rotation of 2π). For 16 terms in the series, PS and AN had the same computational overhead. For less terms, PS was faster (timePS/timeAN = 0.63 for a rotation of $\pi/2$) and for more terms AN was faster (timePS/timeAN = 1.45 for a rotation of 2π). Both methods had errors of the order of machine accuracy. For RK2, the execution times are at least an order of magnitude slower and the accuracy depends on the number of iterations used. For a time step of $\pi/2$ (quarter of a circle), the errors were of the order 6×10^{-3} with 10 iterations and 10^{-3} with 20 iterations. As the time step increased, the required iterations increased so that the relative performance as compared to PS and AN worsened.

In summary for the first test case, AN and PS were roughly equivalent in accuracy and run times and RK2 was slower by an order of magnitude with decreased accuracy (Fig. 1, case 1).

UNCORRECTED PROOF

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

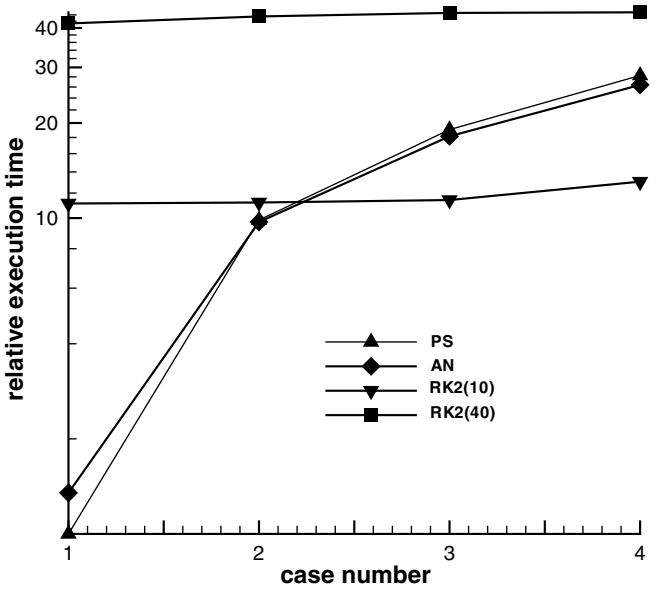

Fig. 1. Comparison of execution times for the three trajectory calculation methods: PS, power-series approximation; AN, analytical integration; RK2(10), second-order Runge–Kutta with 10 substeps; RK2(40), second-order Runge–Kutta with 40 substeps. All the times are normalised with respect to the time for PS with no element crossings (case 1, single element). Errors for PS and AN were of the order 10^{-7}. Errors for RK2(10) cases 2, 3, and 4 and for RK2(40) cases 3 and 4 were greater than 1% and unacceptable. Case 1 has no element crossings, case 2 has 1 element crossing, case 3 has 2 element crossings, and case 4 has 3 element crossings.

For the second case, the execution times for PS and AN increase significantly when element edge crossings are included although the errors remain at machine accuracy. Both PS and AN use an iterative procedure with about 10 iterations to find the element crossings so the execution time increases by an order of magnitude in these cases. On the other hand, RK2 was generally less efficient and became less accurate unless the number of substeps was increased. With RK2, there are significant trade offs between accuracy and execution speed. To be competitive with PS and AN, the number of substeps must be decreased to where the error is unacceptable (10–20%). For better accuracy, the execution times for RK2 were at least an order of magnitude larger than PS or AN.

In summary for the second test case, the execution times for AN and PS increased significantly with the number of element crossings but a high accuracy was retained. An obvious improvement is to use methods that converge more rapidly at the element crossings. The execution times for RK2 were always larger than PS and AN with poor accuracy (Fig. 1).

4.2. Tsunami test problem-wave height and stability

This example is a field-scale problem simulating the propagation of a tsunami generated by a submarine fault rupture on the New Zealand continental shelf (Walters et al., 2006). The location of the fault is near the town of Kaikoura on the northeast coast of the South Island of New Zealand (Walters et al., 2006; Fig. 1). The area is characterized as the convergence zone between the Pacific and Australian plates and the fault is a thrust fault on the continental slope at a depth of about 1000 m. The fault is about 100 km long and 10 km wide, and has a maximum vertical displacement of 4 m (Fig. 2). The fault displacement shown in Fig. 2 was used as the initial condition for water surface elevation.

Initially, the water elevation is specified and the velocity is zero. This initial condition gives rise to two waves propagating in opposite directions. As the shoreward propagating wave shoals, its speed decreases and its height increases. Initially the trailing side of the wave is steeper because of the initial wave shape. Phase speed is only dependent on the height of the wave so the crest travels faster than the trailing slope and the entire wave steepens on its forward face as it propagates shoreward (from 0 to 6 in Fig. 2). The waves finally steepen into a bore directly inshore from the source area.

UNCORRECTED

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

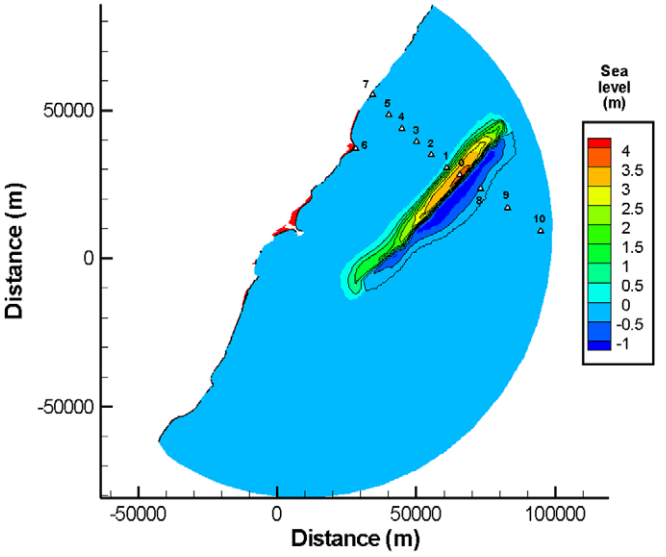

Fig. 2. Surface deformation (initial condition) for the fault generated tsunami near Kaikoura.

The time series of sea level for three different interpolation schemes (LL, GL, and GQ) are shown in Fig. 3. NA is the case where advection is neglected. The large, excessive damping from the low order scheme (GL) is obvious. In reality, advection should have a minor effect on sea level for these long waves until they approach shore. The damping is exacerbated by the small time step (2 s) required to resolve the short period of the wave (60 s). On the other hand, LL and GQ compare realistically with the NA case. The case LL eventually generates large unreasonable oscillations along the shoreline. The simulation does not become unstable, but a rapid run off at the shoreline causes water level oscillations. We believe this is related to the poor approximation of the tangential variations in velocity with the LL approximation.

The computational overhead associated with the different schemes was highly variable. As a rough guide, a single iteration of the PS or AN scheme has about the same overhead of a single step of the RK2 scheme

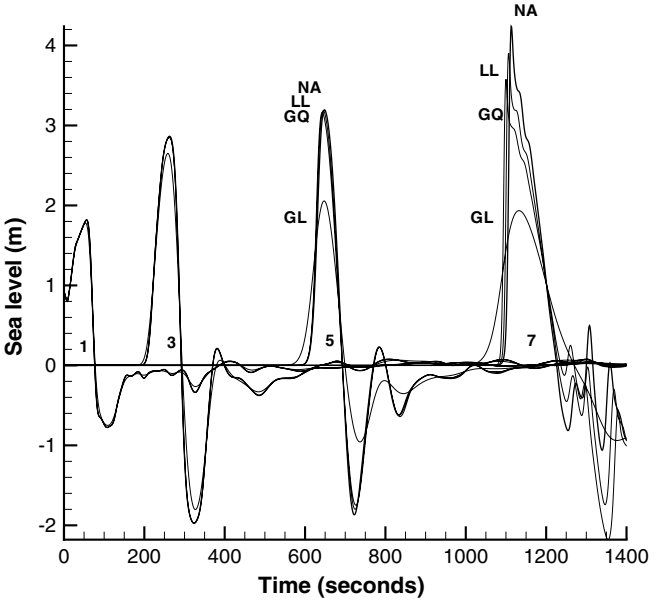

Fig. 3. Time series of sea level at sites shown in Fig. 2. The sites are 1, 3, 5, and 7, located shoreward of the source. NA, no advection included; LL, local linear interpolation; GL, global linear interpolation; GQ, global quadratic interpolation.

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

(assembling a power-series as opposed to interpolating the velocity). For 20 steps of the RK2 scheme per model time step, The RK2 scheme takes 6.2 times the run time of the PS scheme for later times in the simulation when the solution is more dynamic. The run time of the RK2 scheme scales linearly with the number of steps so reducing the number of steps makes the scheme more competitive, although at reduced accuracy. The AN scheme takes 1.68 times the run time of the PS scheme because the power-series required a relatively small maximum of 5 terms to converge. In the end, the PS scheme provided high accuracy with runtime always less than RK2, and was more robust in determining the inter-element crossings. In addition, the PS scheme was more robust than the AN scheme because of the difficulty in trapping all the pathological solution cases.

When compared with the total runtime, RK2 took about 60%, AN took about 22%, and PS took about 15% on average. The percentage was low during the initial period of propagation, and higher when the wave ran up on land and velocity became large. The overhead of RK2 can be reduced with fewer substeps, but with reduced accuracy.

4.3. Flood test problem-relative efficiency and comparison for transcritical flows

During January and February of 1993, a flood on the Verde River in Arizona achieved water levels not seen in the previous 1100 years (House et al., 2002). The channel is bedrock and sand, the bedrock forming constrictions along the river that provides rigid boundaries to flow. One such constriction, 15 km above the confluence with Horseshoe Reservoir, is at Sheep Bridge (Fig. 4). The reach above and below Sheep Bridge was measured with a combination of real-time kinematic GPS and aerial photogrammetry when the river was at a low flow of a few tens of cubic meters per second. High water indicators, as described in Denlinger et al. (2002) were also measured. In most cases these marks were grasses caught in bushes or flotsam deposited on the bank. In all cases the high water indicators are a lower bound of the high water level, as rising water would have removed and re-deposited the material. The accuracy of the measurements was estimated at 0.1 m.

Topographic data were used to construct a detailed grid of the topography surrounding the Verde River (Fig. 4). Steady flows of discharges between 300 cms and 4300 cms were modeled on the grid, and the corresponding model high water compared to the observed high water at various grid locations along both sides of the channel.

The different semi-Lagrangian schemes were compared for a discharge of 3105 m^3/s, the measured maximum flow during the flood. The discharge was introduced at the upstream end of the grid in Fig. 4 (lower right). A major control point on the flow was at Sheep Bridge where the flow turned sharply to the right, passed through a constriction, and dropped. The flow was supercritical in parts of this reach, with large eddies spun off on both sides. From an oceanographic perspective, this flow is similar to the flow through tidal rapids, which are a common occurrence on the coast of British Columbia, Canada, among other places.

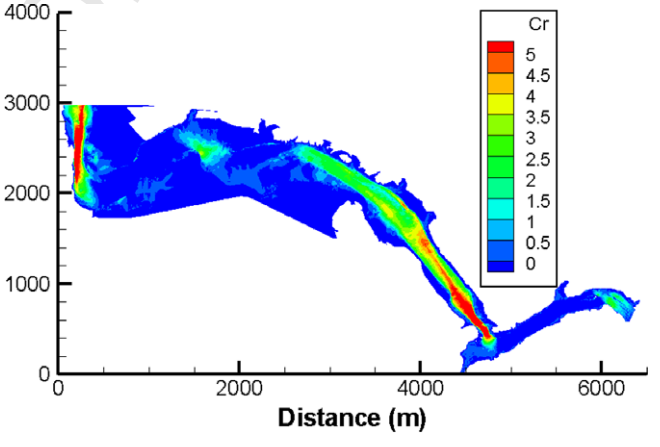

Fig. 4. Reach of the Verde River used for the flood tests. Maximum *Cr* is 10.4. The flow enters from the lower right, passes through the flow constriction and exits into a reservoir at the upper left.

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

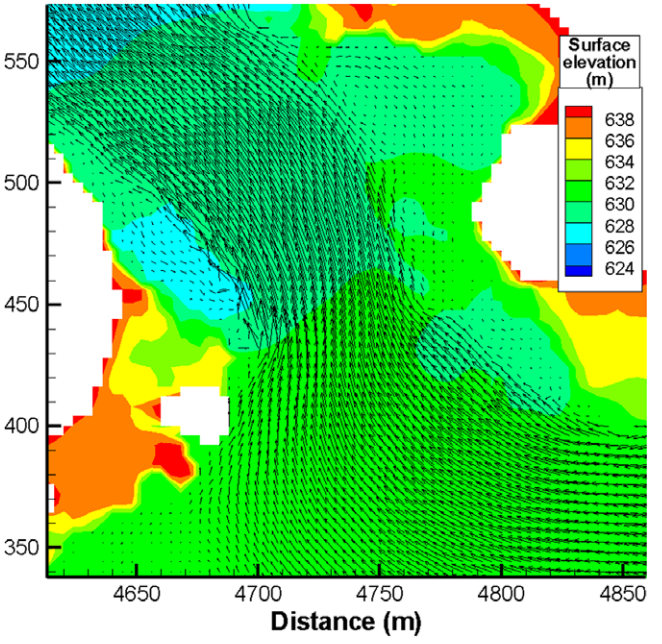

Fig. 5. Detail of the flow in the vicinity of the major flow constriction.

In general, the trajectory calculation is more time consuming than the interpolation at the foot of the trajectory. Moreover, *Cr* was relatively high (up to 10.4 for this case) over large parts of the flow. Hence, inter-element crossings of the trajectory are a significant part of the calculations. RK2 with 20 substeps has an execution time about 3 times that of PS. This can be reduced by a half with judicious choice of substeps, but with reduced accuracy. Even with 20 substeps, the method has poor accuracy for the high *Cr* cases.

Model water surface elevations gave excellent agreement when compared to the measured high water marks. A detail of the flow through the constriction is shown in Fig. 5. The maximum velocity was 8.3 m/s with a strong flow core and counter-rotating eddies along the edges.

5. Conclusions

The choices described here result in a model that is applicable to a wide range of time and space scales and flow regimes that are encountered in coastal oceanography. For the semi-Lagrangian approximation, the crucial elements are the accurate calculation of the trajectory and the interpolation at the foot of the trajectory. Of the methods evaluated here, the power-series expansion method (PS) and analytical solution (AN) appear to be the most accurate and efficient as compared to the second-order Runge–Kutta method. Of these, PS is straightforward to implement and was found to be more robust. All the interpolation methods tested have similar efficiency. However, the global quadratic (GL) method has higher-order accuracy and does not suffer from excessive damping that is common with linear interpolation.

UNCORRECTED

Acknowledgement

The Kaikoura tsunami study was partly funded by Environment Canterbury.

References

Abrogast, T., 1995. Mixed methods for flow and transport problems on general geometry. In: Carey, G.F. (Ed.), Finite Element Modeling of Environmental Problems. Wiley and Sons, New York, pp. 226–275.

Casulli, V., 1987. Eulerian-Lagrangian methods for hyperbolic and convection dominated parabolic problems. In: Taylor, C., Owen, D.R.J., Hinton, E. (Eds.), Computational Methods for Non-linear Problems. Pineridge Press, Swansea, pp. 39–268.

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008

Casulli, V., 1990. Semi-implicit finite difference methods for the two-dimensional shallow water equations. J. Comp. Phys. 86, 56–74.
Casulli, V., Cattani, E., 1994. Stability, accuracy, and efficiency of a semi-implicit method for three-dimensional shallow water flow. Comput. Math. Appl. 27 (4), 99–112.
Casulli, V., Walters, R.A., 2000. An unstructured grid, three-dimensional model based on the shallow water equations. Int. J. Numer. Meth. Fluids 32, 331–348.
Denlinger, R.P., O'Connell, D.R.H., House, P.K., 2002. Robust determination of stage and discharge: an example from an extreme flood on the Verde River, Arizona. In: House, P.Kyle et al. (Eds.), Ancient Floods, Modern Hazards: Principles and Applications of Paleoflood Hydrology. American Geophysical Union, Washington.
Ham, D.A., Pietrzak, J.D., Stelling, G.S., 2005. A scalable unstructured grid 3-dimensional finite volume model for the shallow water equations. Ocean Modell. 10, 153–169.
Ham, D.A., Pietrzak, J.D., Stelling, G.S., 2006. A streamline tracking algorithm for semi-Lagrangian advection schemes based on the analytic integration of the velocity field. J. Comput. Appl. Math. 192, 168–174.
Hanert, E., Le Roux, D.Y., Legat, V., Deleersnijder, E., 2005. An efficient Eulerian finite element method for the shallow water equations. Ocean Modell. 10, 115–136.
Hanert, E., Legat, V., Deleersnijder, E., 2003. A comparison of three finite elements to solve the linear shallow water equations. Ocean Modell. 5, 17–35.
House, P.Kyle et al. (Eds.), 2002. Ancient Floods, Modern Hazards: Principles and Applications of Paleoflood Hydrology. American Geophysical Union, Washington.
Le Roux, D.Y., Lin, C.A., Staniforth, A., 1997. An accurate interpolating scheme for semi-Lagrangian advection on an unstructured mesh for ocean modelling. Tellus A 49, 119–138.
Leonard, B.P., 2002. Stability of explicit advection schemes. The Balance point location rule. Int. J. Numer. Meth. Fluids 38, 471–514.
McCalpin, J.D., 1988. A quantitative analysis of the dissipation inherent in semi-Lagrangian advection. Monthly Weather Rev. 116, 2330–2336.
Malcherek, A., 2001. A second order Lagrangean scheme without numerical viscosity for the propagation of tidal energy. In: Spaulding, M.L. (Ed.), Estuarine and Coastal Modeling: Proceedings of the 7th International Conference. ASCE, pp. 491–500.
Pearson, H.L., 1974. Elements of Analysis. In: Pearson, C.E. (Ed.), Handbook of Applied Mathematics. Van Nostrand Reinhold Co, London, p. 1265.
Raviart, P.A., Thomas, J.M., 1977. A mixed finite element method for 2nd order elliptic problems. Mathematical aspects of the finite element method. Lecture Notes in Math. Springer-Verlag, Berlin.
Robert, A., 1981. A stable numerical integration scheme for the primitive meteorological equations. Atmosphere-Ocean 19, 35–46.
Robert, A., 1982. A semi-Lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations. Jpn. Meteorol. Soc. 60, 319–325.
Staniforth, A., Côté, J., 1991. Semi-Lagrangian integration schemes for atmospheric models – A review. Monthly Weather Rev. 119, 2206–2223.
Stelling, G.S., Duinmeijer, S.P.A., 2003. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. Int. J. Numer. Meth. Fluids 43, 1329–1354.
Stoer, J., Bulirsch, R., 1993. Introduction to Numerical Analysis, second ed. Springer-Verlag, Berlin.
Walters, R.A., 2005. A semi-implicit finite element model for non-hydrostatic (dispersive) surface waves. Int. J. Numer. Meth. Fluids 49, 721–737.
Walters, R.A., Barnes, P., Goff, J., 2006. Locally generated tsunami along the Kaikoura coastal margin: Part 1. Fault ruptures. NZ J. Mar. Freshw. Res. 40, 1–17.
Walters, R.A., Casulli, V., 1998. A robust, finite element model for hydrostatic surface water flows. Comm. Numer. Meth. Eng. 14, 931–940.
Xiu, D., Karniadakis, G.E., 2001. A semi-Lagrangian high-order method for Navier-Stokes equations. J. Comput. Phys. 172, 658–684.

UNCORRECTED PROOF

Please cite this article in press as: Walters, R.A. et al., Semi-Lagrangian methods for a finite element ..., Ocean Modell. (2007), doi:10.1016/j.ocemod.2007.06.008