

Preserving Positivity when Advecting Interacting Biological Scalar Fields

By Jorge Willemsen and Alan Wallcraft, 12 December 2004

Overview

Over the last couple of years, Alan Wallcraft has introduced several biological models into the `trcupd.f` code in HYCOM. However, problems with positivity can arise due to the nature of the nonlinear interactions, for example zooplankton consume phytoplankton, etc. In particular, although the MPDATA approach, for example, preserves sign in the advection/diffusion of the scalar fields, it is not guaranteed that sign is preserved if the nonlinear interactions are computed using a *separate* method under the framework of operator splitting.

Fortunately, Smolarkiewicz and Margolin (1998) have devised a scheme that preserves the sign in the presence of “source” terms. An abbreviated version of their Eq. (19) is

$$\partial\psi/\partial t + \mathbf{div}(\mathbf{v}\psi) = R,$$

where ψ is any biological entity, e.g. N, Z, P or D, $\mathbf{v}(\mathbf{x},t)$ is the fluid velocity, and $R(\mathbf{x},t)$ is an appropriate source function. For biological modeling, it would represent the nonlinear interactions among the distinct constituents (R is a function of NZPD). However for present purposes the precise form is irrelevant.

The aim of this note is to incorporate the Smolarkiewicz - Margolin formulation of the inclusion of R into HYCOM.

Implementation

The basic ideas are contained in the following statements. As originally formulated, A is the complete MPDATA advection-diffusion operation, although it can represent any alternative advection scheme that preserves positivity. The expressions $\Psi(n)$ and $R(n)$ refer to the values at time indexed by n , while dt is the discretized timestep. An asterisk refers to an intermediate updated value through the operation A or evaluation of the source term R .

- 1) $\Psi^* = A[\Psi(n) + dt R(n)]$
- 2) $R^* = R(\Psi^*)$
- 3) $\Psi(n+1) = A[\Psi(n) + dt/2 R(n)] + dt/2 R^*$
- 4) $R(n+1) = R(\Psi(n+1))$

To carry out these operations, Willemsen initially introduced an array “stern” to represent the source term R with the same dimensions as “tracer” (ψ), together with storage arrays stern0 and tracer0. Later Wallcraft showed that there is no need for stern0.

We start with line 466 of tsadvec, do ktr = 1, ntracr. The specific line number referred to the version of src being used at the time, but it has been verified that after revisions since then this line number is still correct.

```
c STERM ADDED FOR BIO line 466 in tsadvc.f
  do ktr= 1,ntracr
    call xctilr(tracer( 1-nbdy,1-nbdy,1,1,ktr),1,2*kk, 6,6,
halo_ps)
    call xctilr(stern( 1-nbdy,1-nbdy,1,1,ktr),1,2*kk, 6,6,
halo_ps)
  enddo
```

c store values at time n

```
  do ktr= 1,ntracr
    trolld(i,j,ktr)=tracer(i,j,k,n,ktr)
    if (trcflg(ktr).eq.2) then
      tracer(i,j,k,n,ktr)=tracer(i,j,k,n,ktr)+pdtemp
    endif
```

c construct the quantity to be advected (operator A in 1))

```
c ADDED FOR BIO line 551
  tracer0(i,j,k,n,ktr)=tracer(i,j,k,n,ktr)
  & +0.25*delt1*stern(i,j,k,m,ktr)
  tracer(i,j,k,n,ktr)=tracer(i,j,k,n,ktr)
  & +0.5*delt1*stern(i,j,k,m,ktr)
c END ADDED

  enddo
```

c advect it with the understanding that tracer is both input and output from
c advem -this is unchanged from previous location, but note there are 2 calls

```
  call advem(2,tracer...,...)
  call advem(2,tracer0...,...)
```

c to implement step 2, the R operation refers to a call to trcupd which has stern
c in common block just like tracer – the modifications to trcupd will be seen
c below then construct new quantity to be advected (step 3)

```
c ADDED FOR BIO line 858
  call trcupd(n,n)
  tracer(:, :, :, n, :)=tracer0(:, :, :, n, :)+
  & +0.5*delt1*stern(:, :, :, n, :)
```

c step 4, since trcupd is now a calculation using tracer(n+1), update the sources outside of last ijk loop.

```
call trcupd(n,n)
```

Note that while the calls to advem invoke MPDATA in the current version of HYCOM, the method should be robust to changes in the advection method since the treatment of the source term is a standard predictor-corrector method.

Modifications to trcupd

Roughly speaking the dynamics was initially evaluated using, with m and n the leapfrogging steps,

```
bm=tracer(m)
bn=tracer(n)
tracer(n)=bn+S(m)dt ! where the S(m) are the explicit
nonlinear
                                ! interactions evaluated with
tracer(m)'s.
```

A different version appropriate for use within the present scheme involves

- 1) Removing the bm fields;
- 2) computing S(n) rather than S(m) and these become the stern outputs;
- 3) do not compute any update to tracer here, it's done in tsadvec as above.

In detail,

```
c  NEW VERSION line 683 in trcupd.f
!      tracer(i,j,k,n,ibio)   = bn_n + deltt1/86400.0 * bu_n
!      tracer(i,j,k,n,ibio+1) = bn_p + deltt1/86400.0 * bu_p
!      tracer(i,j,k,n,ibio+2) = bn_z + deltt1/86400.0 * bu_z
!      tracer(i,j,k,n,ibio+3) = bn_d + deltt1/86400.0 * bu_d

      stern(i,j,k,n,ibio)   = 1.0/86400.0 *bu_n
      stern(i,j,k,n,ibio+1) = 1.0/86400.0 *bu_p
      stern(i,j,k,n,ibio+2) = 1.0/86400.0 *bu_z
      stern(i,j,k,n,ibio+3) = 1.0/86400.0 *bu_d

      enddo
    enddo !i
  enddo !l
enddo !j
return
end subroutine trcupd_904
```

Note the 1.0 rather than deltt1 because the deltt1 are introduced in the calls.

Reference

P.K. Smolarkiewicz & L.G. Margolin, "MPDATA: A Finite-Difference Solver for Geophysical Flows", J. Comp. Physics **140**, 459-480 (1998)