

Review of Tang & Riley A generic law-of-the-minimum flux limiter for simulating substrate limitation in biogeochemical models. Biogeosciences Discuss 12 13399-13425.

Reviewer: Niall Broekhuizen. Review Submitted: 16-9-2015

The authors describe a new, explicit, first-order accurate numerical integration scheme that is designed to ensure that (elemental) masses are conserved and remain non-negative regardless of the magnitude of the projection time-step.

They present a present a brief summary of prior schemes that have aimed to achieve non-negativity (but, in some cases, sacrificed conservation). They highlight the (regrettable) fact that code authors have frequently resorted to arbitrary means:

- Restricting the total quantity of substrate consumed during a time-step to be < amount of substrate available at the start of the time-step (an unnecessarily restrictive approach if substrate is also produced during the time-step)
- Restricting the total quantity of substrate consumed during a time-step without also reducing the associated quantity of product material that arises from the substrate reactions (introducing mass conservation errors)

Aside from the aforementioned disadvantages, the authors note that these 'ad hoc' schemes imply there is an undesirable coupling between the numerical integration scheme and the implementation of the model equations as computer code. As a result, the model implementation (ie the code) becomes difficult to extend and results can become sensitive to the order in which rates of change are calculated.

As the authors note, there are somewhat less arbitrary explicit-type, mass-and-positivity conserving integration schemes that better separate model implementation (specification of the model's differential equations in code) and model solution algorithms (used to project the differential equations forward in time) in existence (Sandu 2001, Broekhuizen et al 2008). Unfortunately, these too have disadvantages in terms of one or both of run-time or accuracy-limitations.

Overall, I found the paper interesting. Subject to satisfactory responses to the comments below, I believe that it should be published.

1. The authors repeatedly refer to (and compare against) the BBKS scheme (citing Broekhuizen et al 2001). In that paper, BBKS was an acronym used to refer to a numerical integration scheme that was first described in an earlier paper (namely, Bruggeman et al 2007). The central point made in Broekhuizen et al (2008) is that the original BBKS scheme was not 'scale independent' (adding more state variables into the system of equations would change the numerical results – even if the new state variables merely 'duplicated' the original ones or, indeed, are entirely uncoupled from those belonging to the original system of equations). Broekhuizen et al (2008) proposed a modified version (dubbed mBBKS) of the original BBKS scheme that overcame the scaling issue for 'duplicated equations or extended systems of coupled equations (though it would not entirely overcome the scaling-issue if a user were to endeavor to solve two or more entirely independent sets of coupled ODEs within the same code). I find it surprising that Tang & Riley would have adopted BBKS rather than mBBKS. I wonder whether all of their references to BBKS should really be to mBBKS? If they have genuinely used BBKS, they should properly attribute it to Bruggeman et al 2007 rather than to Broekhuizen et al (2008) [though it would be

appropriate to acknowledge the existence of the mBBKS scheme developed in Broekhuizen et al 2008).

2. If Tang & Riley have used BBKS rather than mBBKS, then I believe that they should adopt mBBKS in its place for these comparisons. That said, mBBKS is also prone to yielding a global flux limiter that underflows to zero (such that the predicted net changes to all state-variables are falsely said to be zero for a time-step projection). That was the explanation for the mBBKS's ultimate failure in the Robertson test-case. I suspect that it is also the explanation for the failure of the "BBKS" (Tang & Riley notation) [Methods section lines 12-13 "BBKS failed to predict any organic matter composition after the first few time-steps"].
3. (related to 2). Results section lines 12-13 I think it would be helpful if the authors explained why the (m)BBKS scheme predicted zero net change of organic matter: was it underflow of the flux limiter toward zero, or was it underflow of the product of the limiter and one or more of the raw rates of change? (I suspect, the former).
4. Writing the differential equations in reaction form (eq 4 and page 13409 line 19). Is Eq 4 conceptually equivalent to the Petersen matrix ODE-system notation used in the reports describing the IWA's anaerobic digester (Batstone et al 2002 Anaerobic Digester Model #1 IWA Scientific & Technical report) and river water quality model (Reichert et al 2001 River Water Quality Model No. 1; IWA Scientific & Technical report)? If so, I suggest citing the original Petersen work and/or the IWA reports so that people who are familiar with that approach can see the similarity. If not, I think the authors should explain how their notation differs from the Petersen matrix notation.
5. My main concerns about this manuscript relate to the pseudo-code (equation 6. Page 13405). I believe that it needs much more care and explanation:
  - a.  $p_m$  is the  $m^{\text{th}}$  element of the vector  $\mathbf{p}$  rather than being a single scalar? Similarly for  $q_n$ ? I believe that the subscripts  $m$  &  $n$  are being used to indicate vector elements and some of my comments/questions below are based upon that belief
  - b.  $M$  (first for-loop) denotes the total number of state-variables? Please clarify
  - c.  $N$  (second for-loop) denotes the total number of reactions? Please clarify
  - d. Incidentally, there are an awful lot of  $n_s$  and  $q_s$  (and, perhaps some other symbols) used to mean different things in different parts of the paper.  $N$ ,  $N_{\min}$ ,  $N_{\min,\text{sup}}$ ,  $N_{\min,\text{up}}$ ,  $q_n$ ,  $q_{N\min}$  (& the related  $q_{p\min}$ ). In some cases ( $N_{\min}$ ,  $N_{\min,\text{sup}}$ ,  $N_{\min,\text{up}}$ ) the quantities are closely related (so similar notation is justified), but in others the quantities are unrelated ( $N$  and  $N_{\min}$ ;  $q_n$  and  $q_{N\min}$ ). In these latter cases, I think it would be helpful to adopt a notation that does not imply (however weakly) that the quantities might be related.
  - e.  $k$  (summation terms for  $x_m(t+DT)$ ) denotes the  $k^{\text{th}}$  non-zero reaction influencing this state-var? Aside from the implied computational inefficiency of including zero-reaction terms, it might be clearer to replace  $k$  with  $n$  and explicitly sum over  $1, N$  (or am I completely mis-understanding things)
  - f. Rather than using unadorned `end`, I suggest using `endif` and `endfor` to aid 'bracket matching' when reading the code.
  - g. Indent the expression for  $x_m(t+DT)$  properly
  - h. Indent the expression for  $p_m$  properly in the first branch of the if-test.
  - i. Should it be `IF ( $x_m(t+DT) \leq 0$ ) THEN` rather than `IF ( $x_m(t+DT) < 0$ ) THEN`?
  - j. At entry to the `IF ( $\text{Ineg}==1$ ) THEN` block, the subscript  $m$  will have value  $M$  and it will retain that value throughout the execution of the block. As written, I believe that  $q_n$  will only even be influenced by  $p_M$  and  $v_{M,n}$ . Is that really what is intended?
  - k. I suggest inserting brackets around the terms that form the IF-condition(s)

- l. In the two “IF” conditions, be consistent in presence (or absence) of a comma following the condition. Also, a space is required between “1” and “then” in the second IF condition.
- m. Ultimately, how are the values  $q_n$  used? I think it would be useful to extend the pseudo-code such that it explicitly illustrates the manner in which the final value of  $x_m(t+DT)$  is calculated. My guess is that the pseudo code for an entire time-step projection (rather than just the calculation of the flux-limiter terms) may be something like:

```

FOR n=1,N
    qn=1
ENDFOR
Lneg=1
WHILE(1 == lneg)
    Lneg=0
    FOR (m = 1,M)
        Xm(t+ DT)=xm(t)+ [sum_over_k(S+m,krkqk) - sum_over_k(S-m,krkqk)] DT
        IF( 0 > xm(t+ DT) ){
            ...
        ELSE
            ...
        ENDIF
    ENDFOR
    IF (1== lneg)
        ... (but bearing in mind my reservations about this bit of code (see (i),
above)
    ENDIF
ENDWHILE

```

- 6. Within the discussion, the authors should emphasize that the ‘interface’ for the function which returns the rates required by the integration scheme will differ from that required by most standard explicit method numerical schemes. Those require that the function which returns the instantaneous rates of change does so by means of a vector of net rates of change (one rate per state-variable). In contrast, this numerical integration scheme will require the corresponding ‘rate calculation function’ to return two matrices [respectively, containing  $S^+_{m,k}r_k$  and  $S^-_{m,k}r_k$ ]. Since the interface will differ, this makes it a bit more difficult to switch between integration methods by means of a run-time (at model initialization) switch.
- 7. In discussion, mention that whilst an individual  $p_m$  may underflow to zero (causing the corresponding  $q_n$  values to be zero), other  $q_n$  need not underflow. Thus, in comparison with (m)BBKS, this scheme is less likely (I believe) to yield solutions in which all state-variables are falsely predicted to cease changing.
- 8. Appendix A. I am not familiar with the Parton et al model. I did obtain a copy of the paper and read it but I think it would be helpful if Tang & Riley provided a table that lists the state-variables by notation-name and gives a verbal description of what each is.