**DEVELOPMENT OF A GENERAL TRANSPORT MODULE FOR DSM2, INCLUDING SEDIMENT TRANSPORT**

Fabián A. Bombardelli

Kaveh Zamani

James Kohne

Joseph Waltz



**Work team**

Dr. Jamie Anderson (Project Manager)

Dr. Eli Ateljevich

Mr. Patrick Luzuriaga

Dr. Kevin Kuo

**Steering committee**

Dr. Francis Chung

Mrs. Tara Smith

Davis, June 21, 2011

Table of Contents

**Executive summary.................................................................**

**Chapter 1: Introduction project Goals**

1.1

1.2

...

**Chapter 2: Theoretical models**

2.1

2.2

2.3

...

**Chapter 3: Numerical implementation**

3.1

3.2

...

**Chapter 4: Framework for Code testing**

4.1

4.2

**Chapter 5: Website with data on sediment transport**

5.1

**Reference**…………………………………..………………………………………………………….XX

**Appendixes:**

A1- Advection-Lax Two Steps Method………..…….…………………………………………XX

A.2- Diffusion- Crank Nicolson Method………………………………………………………..XX

A.3- ODE Integrators………………………………………………………………………………….XX

B- Analytical Benchmark Solutions ……………………………………….…………………….XX

**list of Figures**

Figure 1:Schematic of sediment transport mechanisms in the water column .........................XX

Figure 2: Relationship between software testing components and algorithmic testing. Software testing principles................................................................................................................

Figure 3: Transport algorithm testing incremental complexity.

Figure 2: ODE (Reaction) solver test

Figure 3: 3rd order Runge-Kutta ODE solver

Figure 4: Diffusion single operator test, Dirichlet boundary condition

Figure 5: Diffusion single operator test, Neumann boundary condition

Figure 6: Advection solver, test bi-directional flow, Gaussian Plume of mass, remote boundary condition

Figure 7: Advection Decay solver test, uniform flow

Figure 8: Coupling advection diffusion and reaction with zero order implementation of intermediate boundary condition, L1 error norm convergence ratio drops from 2 to 1.02

Figure 9: Tidal flow field, advection solver with Gaussian initial mass distribution

Figure 10:Tidal flow field, advection solver with sinusoidal initial mass distribution

Figure 11: Tidal flow field, advection and reaction solvers with Gaussian initial mass distribution

Figure 12: Tidal flow field, advection and reaction solvers with sinusoidal initial mass distribution

Figure 13: Spatially varying flow field and dispersion coefficient, advection and dispersion (Zoppou)

Figure 14: Schematic of the algorithmic error in introduction of the source to advection solver

**EXECUTIVE SUMMARY**

Estuarine transport phenomena have tremendous implications on environmental processes, on the local economy, and on future planning of coastal civil works. Numerical models are usually used to increase the understanding on such phenomena, and to solve practical, crucial problems associated with tidal environments. Numerical models are relatively inexpensive and they are employed as an essential tool by operators, decision makers and regulators in the San Francisco Bay area, and the Delta of the Sacramento and San Joaquin Rivers. In 2009, the Department of Water Resources of California (DWR) and the University of California, Davis (UCD) started a project to implement new sediment transport routines into the Delta Simulation Model II (DSM2) - a one-dimensional (1D) model for flow, water quality and particle tracking in network of rivers. The ultimate goal of the project is to contribute with a tool for addressing pressing problems related to sediment in the Delta, such as levee failure, marsh restoration, dredging, transport and fate of metals, etc.

After starting the project on April 2009, it was decided by DWR personnel to extend the scope of the project. The model needed to address not only sediment transport, but also the transport of any general constituent, affected by advection, dispersion and reaction (sources/sinks) processes. This general transport solver needed to be capable of handling almost all the constituents in an estuarine environment: salinity, heat, sediment, dissolved oxygen, etc. Based on this change in scope, STM (Sediment Transport Module, the name of the new set of subroutines) was designed and coded considering certain capabilities in mind. First, STM was developed in the most flexible and general manner. Second, it was designed with relative high accuracy, in both time and space, to be capable of dealing with large domains and long simulation times. Third, it was verified in the most rigorous way, using state-of-the-art tools of the software industry. To that end, STM was coded with a meticulous attention to the Fortran data structure (to increase the computational efficiency) and the testing package FRUIT was employed. Finally, STM was developed having in mind all the intricacies of the Delta system.

The STM was developed using operator splitting, involving the sequential solution of the processes involved. For the advection, a second-order Lax, Heun method was implemented, whereas a centered in time, second-order scheme was used for dispersion. The reaction portion, in turn, was coded using diverse approaches.

The rigorous verification of STM sets these subroutines apart from existing transport models. Verification is a must in all developments of numerical models; however, some of the well-known packages for flow and transport in estuarine systems suffer from incompleteness in the verification process. The STM test package includes about 350 tests as part of a dynamic test suite. 80% of the tests are unit tests of subroutines for different inputs. Unit tests refer to checking that the different portions of the code are well designed and work properly according to plan. Scenarios of erroneous and incomplete inputs are thus included. This dynamic test suite allows re-checking the code during any future development. Oftentimes, mistakes or bugs present in every new algorithm are introduced in the process of linking those new algorithms to existing counterparts. With the dynamic test suite, STM will be safe against these kinds of errors. In addition to the unit tests, we developed a framework of tests (non-existent when we started working on the project) to verify the results. This framework consists of 17 tests which combine existing analytical solutions with new ones developed by us which allow for checking that the model results match those solutions for “canonical” cases. Overall, it can be stated that STM is one of the most (if not the most) tested A-D-R (advection-dispersion-reaction) solvers in the world.

A sediment library to address the specific problem of sediment transport was added with the assist of DWR personnel. The library automatically categorizes particles either as suspended load or bed load, and is able to handle several sediment classes. Then, it simulates the deposition, entrainment, and movement of cohesive and non-cohesive sediment in the river. For each of the main two modes of sediment transport (bed-load and suspended load), many relations have been suggested in the literature. In view of the fact that each of those relations is based on certain assumptions and is derived considering particular datasets, they are not general. Some of the most commonly used sediment source relations of the dominant processes were added to the existing sediment source library. The corresponding unit tests and system tests were coded on the contiguous module. Therefore, the data leaking errors and human-related mistakes were detected and corrected.

Finally, another important outcome of the project consisted of the development of the website with data on sediment transport, which was accomplished without funding for this project.

The STM currently operates in a single channel. Most potential difficulties for extending the module to a channel network were anticipated in the developments. Future work will include such extension to channel networks, and the development of corresponding test suite. Also, the developments will be validated with data on sediment transport for the Delta. This is ongoing already. The idea is to study the performance of different sediment entrainment functions in the Sacramento River and to fine-tune the parameters for better approximating the collected field sediment transport measurements. Further developments will introduce the processes of agglomeration and break-up of sediment particles in the STM.

**CHAPTER 1**

**INTRODUCTION. PROJECT GOALS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1.1 Generalities**

The numerical computation of water quality in river networks including (sediment transport as a particular case) has become an essential tool for the planning, design, and assessment of the feasibility of river engineering projects. Although the physical processes associated with the transport and fate of certain constituents have not yet been completely elucidated, (encompassing the entrainment of sediment into suspension in rivers and the speciation of mercury, for instance) engineers around the world are constantly called upon to provide answers regarding the impact on biota and humans of those contaminants, at diverse spatial and temporal scales. In this context, any development of modeling tools for the analysis of constituent transport in rivers should incorporate the most appropriate theoretical models to represent partially-understood, complex phenomena, and should adopt the most robust and accurate numerical techniques in order to provide optimized answers in practical cases.

Concerning the Sacramento-San Joaquin Delta (hereafter simply referred to simply as the Delta), the California Department of Water Resources (DWR) has been developing for some years now a network flow and water quality numerical model, DSM2, which provides answers to diverse problems corresponding to floods and pollutant transport in the system. This numerical model (which has been in use since 1997) has served the State notably well on many occasions. For instance, DSM2 has been used to reproduce historical flows that have been taken place in the Delta. Although there are still numerous problems in which DSM2 can serve well the State in its present form (in fact, DSM2 has been used at UC Davis to model the distribution of Striped Bass in the Delta, and at DWR to forecast scenarios of sea-level rise), the code lends itself to the increment of its capabilities via the addition of suitable sub-models.

Several proposed lines of action for the future of the Delta are currently being discussed (see Lund et al., 2007, 2009; Hanak et al., 2011), and some of those proposals include actions on the flows and sediment loads in the Delta rivers and tributaries. DSM2 is thus called to play an important role in the assessment of the technical feasibility of those proposals. In particular, with the addition of sub-models for sediment transport to DSM2, a more comprehensive assessment of the above lines of action can be performed. Specific aspects regarding sediment transport in the Delta of interest to DWR are:

1. The motion of sediment in the diverse rivers of the network coming from dredging operations undertaken in the Delta ship canals;
2. the transport and fate of sediment resulting from activities of marsh restoration;
3. the transport and fate of sediment particles resulting from levee breaches;
4. the transport and fate of metals like mercury, which are usually highly-associated with solid particles in the Delta and San Francisco Bay systems;
5. the evolution of bed levels in the Delta under historical flow conditions.

Several 1-D codes for flow and pollutant transport are customarily used around the world: CCHE1D, MIKE 11, FEQ, GSTARS, Ezeiza V, CHARIMA, FLDWAV, etc. However, only a few of those codes include sub-models to deal with the transport of sediment. In addition, some of the existing sub-models for sediment transport are *not* flexible, roboust accurate and/or general enough. Thus, the addition of general sediment-transport sub-routines to DSM2, able to deal with cohesive/non-cohesive sediment, will provide the code with capabilities that are not present in most other models.

Currently, DSM2 is structured in three main modules: HYDRO, QUAL, and DSM2-PTM. HYDRO involves a solver for surface water variables, i.e., the water levels and discharges in the rivers of the Delta. QUAL includes a Lagrangian Transport Model and the solution of advection-dispersion transport components. Constituents that can be modeled in DSM2 are: dissolved oxygen, carbonaceous BOD, phytoplankton, organic nitrogen, ammonia nitrogen, nitrate nitrogen, organic phosphorus, dissolved phosphorus, TDS and temperature. PTM in turn tracks individual particles in a pseudo-three-dimensional space. Hydrodynamic results from DSM2 for the Delta are usually employed as input for two- and three-dimensional (2-D and 3-D) models for the San Francisco Bay (see Smith, 2007).

**1.2 Original and modified goals of the project**

The original goal of the project was to:

1) Incorporate the capability of simulating sediment transport and bed-level change to DSM2;

2) validate the resulting sub-models first with measurements obtained from the literature, and second with observations of sediment concentrations in the Delta;

3) include a link to the water-quality sub-module (QUAL) in DSM2 to simulate in the future the transport of pollutants attached to particles; and

4) organize an inventory with datasets about flow and sediment transport in the Delta.

*After some work on the project and initial discussions within the work team, it was suggested by DWR collaborators and managers of the project to modify its scope.* It was realized that a general transport module should be developed, which would replace QUAL and address the original objective of the project at the same time. In addition, after our second Technical Advisory Committee (TAC) meeting (see below), it was suggested that bed-level changes in the Delta are minimal in short time scales; thus, the stage of the development involving the update of the bathymetry was left outside of the scope of the project.

These changes to the scope of the project shifted the focus from a straightforward project on sediment transport to a more general constituent transport project. Also our collaborators at DWR decided to pursue a very rigorous testing (verification of the code) using FRUIT, and automatic document generation employing doxygen. This obviously produced a delay in the validation with results of the Delta, which will be undertaken anyway but outside of the scope of the project.

This report starts with a brief review of the mechanics of sediment transport and its governing equations in Chapter 2. Then in the Chapter 3, numerical discretization of the problem is provided. Chapter 4 discusses the verification approach of the numerical solver. Finally, Chapter 5 describes the sediment transport data which was gathered by different agencies in the San Francisco Bay and Delta areas, for further validation of the sediment transport model. In the addendum, more details of the numerical discretization, code testing and sediment source terms are provided.

**CHAPTER 2**

**THEORETICAL MODELS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2.1 Key transport processes considered**

The transport processes considered in the model are (see Fig. 1 for the case of sediment transport in particular):



Figure 1: Schematic of sediment transport mechanisms in the water column (adapted from Abad et al., 2008).

a) transport of constituents in suspension, including advection due to currents and dispersion of pollutants;

b) transport of sediment as bed-load, close to the bed, including all motions accounted for through empirical formulas;

c) erosion and deposition of sediment.

The new sub-routines include the transport of non-cohesive as well as cohesive sediment, which is conveniently described by Eq. (1) above. The cohesive nature of sediment is accounted for by expressions for the settling velocity which depend on the sediment concentration and/or other convenient properties such as salinity, as reported in Appendix 1.

**2.2 General transport equations**

**2.2.1 Transport in suspension of sediment and of dissolved constituents**

The equations for the transport of constituents in suspension are as follows, including the transport of sediment as a special case:

where  is the cross-sectional wetted area (m2);  is the cross-sectional-averaged concentration of pollutant in dissolved phase, averaged over turbulence (-);  is the discharge (m3/s);  denotes the dispersion coefficient (m2/s).  indicates the entrainment rate of sediment into suspension per unit width (m2/s);  represents the deposition rate of sediment per unit width (m2/s); and  (m2/s) and  (-) refer to the lateral discharge (per unit width) and the concentration of pollutant in the lateral discharge. In turn,  and  indicate the spatial and temporal coordinates, respectively, and  denotes sources and sinks of pollutant of non-point nature. In the case of having several size classes of sediment, this equation is solved for each class including in the  terms.

The steps followed in the derivation of the above equation can be found in Rutherford (1994).

**2.2.1 Transport of sediment as bed-load**

The transport of sediment as bed-load is computed via empirical relations taken from previous studies. Such relations can be recast as special cases of the following relation:

where  is the solid discharge due to bed-load per unit width,  is the submerged specific gravity, given by , with  and  denoting the densities of sediment and water, respectively;  is the acceleration of gravity; and  is the particle diameter. These relations are said to pertain to “capacity” conditions, whereby the river can scour sediment according to the load it can actually carry. Again, this equation can be posed for each particle size class.

In the current version of the code, the bed-load formula included is the Meyer-Peter-Muller modified by Wong and Parker (2006).

**2.3 Closures (entrainment and deposition)**

Entrainment formulas for non-cohesive and cohesive sediment are usually expressed in terms of the wall-friction velocity () or the shear stress. For instance, the formula by García and Parker reads:

(1.3)

where  is a constant equal to 1.3 x 10-7; , with ( explicit particle Reynolds number);  indicates the fall velocity; and  denotes the kinematic viscosity of water. Once  (which is a non-dimensional number) is computed, it follows that: . In turn, , where  is the local sediment concentration at a distance from the bed. (This concentration can be related to the cross-sectional concentration of sediment through empirical coefficients; see Appendix 1.) The García and Parker formulation is one of the few which include a version for several classes of sediment size. Appendix 1 includes other formulations to be coded in the STM.

Numerous expressions have been presented in order to facilitate the computations of the fall velocity, . In this project, the regression proposed by Dietrich (1982) for natural particles is used, as explained in the Appendix C. We also coded the van Rijn settling expression (see Appendix C).

**CHAPTER 3**

**NUMERICAL IMPLEMENTATION**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3.1 Basic approach**

In the discretization of Sediment and Transport Model (STM), we were planned to get a literally second order accurate code in both time and space. The Operation Splitting method was adopted for combining different operators. Then each individual operator (Advection, Dispersion and Reaction) was discretized employing an appropriate second order scheme. The consistency with the nature of the problem and importance of conservation of mass require us to employ Finite Volume Method's Framework (FVM). All the values and operators are coded in FVM formulation. Here a brief presentation of our approach for choosing STM's method among the existing alternatives is given. The details and more in depth information on numerical discretization is provided in appendix XXX.

**3.2 Operator Splitting**

It the classic (Godunov) splitting or fractal step method one operator is employing on the previous step's solution (know values) and then the result is fed into the next operator. It was proven by Valocchi and Malmstead (1992) that, this method of operator splitting does not conserve mass in the presence of a source term. To overcome this drawback, it is possible to reduce the order of errors associated with the over/under estimation in the Godunov method by using a time centered method referred to Strang splitting (Strang 1968; Zysset and Stauffer 1992). This method involves centering reaction step between two transport steps. The scheme can also be used with multiple reaction and transport steps, but in each case the reaction step will be centered. Where only a single reaction step is used, the method takes the form:

Followed by reaction step:

which is in turn followed by another ½ transport step:

For the time being the different operators combined with Godunov splitting. The authors are going to also consider alternating (Strang Splitting) in future.

**3.3 Hyperbolic term (Advection)**

Because of the great interest in numerically simulating high Peclet number transport system (Advection dominant) large numbers of methods have been proposed. These include the method of characteristics or modified versions of it (Konikow and Bredehoeft, 1978; Arbogast et al., 1992; Chilakapati, 1993; Zheng and Bennett 1995; Roache, 1992) and adjoint Eulerian-Lagrangian methods (Celia et al., 1990). Both of these approaches are based on the treatment of the advection part of the transport equation using a Lagrangian scheme (a reference frame in which one follows the advective displacement of the fluid packet). An Eulerian (fixed) reference frame is then used to simulate dispersive/diffusive transport. The approach reduces numerical dispersion by reducing the effective grid Peclet number for the fixed Eulerian grid. Although there are some implementing restrictions the method of characteristics and its related approaches are still widely used when it is critical that numerical dispersion be avoided. Another method is; TVD or total variation diminishing scheme, gives more nearly oscillation-free behavior (Bobey, 1984; Yee, 1987; Gupta et al., 1991). The TVD is one of a class of methods which use limiters to ensure monotonicity of solution (Van Leer, 1977a,b; Leonard, 1984). TVD methods with flux limiter sometimes performs better than same order FCT counterpart on the reactive transport in case of oscillations (Steefel and MacQuarrie, 1996). Another class of high resolution Eulerian methods uses higher-order approximations for the first derivatives, but hybridizes these with low order schemes in an attempt to obtain monotone solutions. The solutions have the higher-order approximations in smooth regions and the low-order accuracy near discontinuities (e.g. near plume fronts). The price to be paid for these schemes is that they are non-linear, even when applied to initially linear problem such as ADR equation. In this class are the flux –corrected transport (FCT) methods (Boris and Book, 1973; Oran and Boris 1987; Zalesak 1987; Hills et al., 1994) which usually gives excellent results when applied to non-reactive solute transport (Hills et al., 1994; Yabusaki et al.,) however, as in some of the other methods discussed here, very low level oscillation still coupled into solution. Here, we chose to employ the FCT methods for discretization of Advective term. Advection was coded with modified Lax-two-step method (Leveque, 2002). For overcoming the cases of shock, the van Leer flux limiter (slope limiter) was employed in the step of flux calculation (Saltzman, 1994).

**3.4 Parabolic term (Diffusion or Dispersion)**

The conventional Crank-Nicolson discretization in the FVM frame work, was used for dispersive operator. The method is second order accurate in time and space. Crank-Nicolson method is fully implicit and unconditionally stable which allows the user to select larger time steps (Crank and Nicolson, 1947). In the solution process, the Crank-Nicolson scheme yields a tri-diagonal matrix. The tri-diagonal matrix algorithm (TDMA), also known as the Thomas algorithm is used to solve it (Press et al., 1992). TDMA is very cost effective solver and it inverts a n by n matrix with only *O(n)* number of operations. Owing to the modular nature of Operator Splitting, it would be possible to replace the TDMA solver with a more efficient tri-diagonal matrix solver library, in future.

**3.5 Ordinary Differential Equation integrator**

Heun's second order Ordinary Differential Equation (ODE) integrator was coded. the solver was combined within the predictor and corrector step. This numerical trick has been introduced by P. Colella (for example see: Chombo Design Documents, 2011) for increasing the accuracy of predictor step in the advection solver. This solver is from the family of second order explicit Runge-Kutta solvers. For the probable cases of stiff source terms, also a third order explicit Runge-Kutta solver, as an separate operator coded. a time step adaptive Runge-Kutta solver (RK 4-5) is coded for the case of stiff source term. The other solver could also utilized in the cases of need for developing a time adaptive transport solver (Press et al., 1992). Lastly, for very stiff reactive terms we considered TGV (199???) implicit method. Owing to the modular nature of Operator Splitting, it would be possible to replace the current ODE solver with a more efficient solver for especial purposes.

**3.6 Limitation**

There are regions in which each of the solvers cannot perform its desirable function. The Advection solver is coded in Eulerian frame work and Courant-Friedrichs-Lewy number (CFL) must hold less than one. CFL number, this a parameter gives the fractional distance relative to the grid spacing traveled due to advection in a single time step.

The second issue to consider for advection solver is: The Grid Peclet number. The Grid Peclet number is a non-dimensional term which compares characteristic time for diffusion (dispersion) given a length scale with the characteristic time for advection. The wiggles start at mesh Peclet number above 2 (Unger and Forsyth, 1995) and the problem becomes more severe when the Grid Peclet number increase. In the lower Grid Peclet number, the flux limiter does not work and the global order of accuracy is not reduced due to flux limiter's effect.

The diffusion solver is unconditionally stable. But spurious oscillations occurs if the ratio of time step to the square of space step is large. A similar expression to CFL may be derived for systems characterized by purely diffusion transport, giving rise to diffusion number (Fletcher, 1991):

**CHAPTER 4**

**FRAMEWORK FOR CODE TESTING**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**4.1 Generalities. Types of tests**

It is obvious the earlier a defect or error is detected, it is easier to fix it. To that end we utilized a software industry's techniques for verification of STM. In this chapter we describe the framework for software verification of the STM. The framework was crafted according to principles from both the software testing (sometimes known as Software Quality Engineering) and numerical testing fields. Herein, we describe the components and implementation of the test suite, which includes unit tests, regression tests and algorithm tests of error convergence and accuracy. We make use of some analytical solutions obtained from the literature and others developed/modified by ourselves. Applying this rigorous testing framework to our transport code provides assurance to the developers and end users that the code performs as expected.

Flow and transport codes inherently comprise both numerical algorithms and pieces of software. Well-developed testing literature exists for both. Oberkampf and Trucano (2002) describe some elements of software quality engineering (SQE) in the context of numerical verification, and note some cultural reasons why it is seldom implemented. Figure 2 is adapted from this work and depicts the relationship between software testing components and algorithmic testing such as convergence tests.

**4.2 Basic concepts in STM software testing**

* **Static versus dynamic testing**

In the testing process of legacy codes or if the final product is going to use without any further change the "static" testing approach could be employed. In the static testing the program controlled versus test cases just once. On the other hand, in the "dynamic" testing a program repeatedly controlled versus a predefined criteria. Hence, dynamic testing is utilized in cases of ongoing development of a code or modification for new reasons. Dynamic testing may begin before the code is complete and develop as a the main code is building up. Since the STM was developed from scratch, there were a good opportunity to code its accompany dynamic test suite.

* **Black box testing versus glass box testing**

In the black box testing the functionality of a code is tested without especial knowledge of internal structure of the code. Therefore, black box testing is the only option when the source code is not available. Opposed to black box testing, glass box testing is procedure in which functionality of each part and the path of data inside a code is checked. Glass box testing could be performed at the level of units or whole system. Given that in the STM project source code is under development, Glass box testing verification approach is chosen.

* **Unit tests versus system tests**

Unit testing is a procedure in which each individual unit of a code is checked if they carry out what they suppose to make. Unit is the smallest testable part. The mission of unit testing is to isolate each individual part of a code and prove it works correct. Whereas, algorithm test or system test is a test in which a larger subtask or portion or even a whole code is checked. The STM test package includes both unit tests (280 tests) and system test (70 tests).

* **Fail/pass criteria**

In the project the testing procedure is automated .A consequence of automation is that verification tests must be phrased in terms of binary assertions, true and false statements that can be tested without human intervention and that reveal whether the aspect of the code under consideration is correct. Convergence criteria are a rigorous basis for assertions, either by requiring strict convergence criteria (“the algorithm is second order accurate in time and space”) or a regression criterion (“convergence will not get any worse than last time the code was tested”).

* **Regression tests**

Regression test is a software testing approach in which a system is checked for newly introduced errors. In the regression test a system is checked versus its previous step condition by rerunning previously run tests and checking if code behavior has changed and whether previously fixed faults emerge again. In the STM project regression testing will be employed. The test suite will run automatically on a regular basis, and in case the previous results are not obtained, developers will receive a warning message.

**4.3 STM test package**

In this project a comprehensive test package was put together for the full verification of the STM solver. The test package developed having certain characters in mind: dynamic test package, automatic and continues testing, step by step increase in complexity, the test approach should foster exact specification of every unit of code, and finally testing should provide assurance of whether a set of specifications is met.

One goal of tests is that they be a continuous assessment of the code. The entire testing system is a *regression* testsuite that establishes a gauntlet through which future code changes must be passed.

****

Figure 2. Relationship between software testing components and algorithmic testing.

Software testing principles.

The software testing literature further distinguishes between *unit tests* of atomic routines and *system tests* of larger subtasks. For example, the evaluation of a gradient might be a unit of code and it would have a unit test. Convergence tests and other algorithm tests are examples of system tests.

The unit testing point of view is that code must be exercised over a range of inputs that covers every line. For instance, to test a gradient routine with a slope limiter, a developer would want to cover:

1. smooth cases in the middle of the mesh.
2. behavior near the edges of the mesh, where one-sided differences may be used instead of central differences.
3. cases that test the limiters with steep or zero gradients in both directions.

Any system test will certainly exercise the gradient code in the middle of the mesh, which in any event can seldom be wrong without being obvious. However, system-level tests might miss the more unusual cases. For example, a convergence test may miss a bug in the limiter for the case of steep decreasing slope for several reasons. First, convergence is often assessed with limiters turned off, as they are locally order reducing. Second, it is hard to fiddle with the problem in just the right way to make sure the left, right, and center cases of the gradient limiter are all triggered. This is particularly true when trying to exercise other units of code at the same time – parameter choices made to fully exercise gradient limiter the may lessen the coverage of another unit.

**4.4 Framework of algorithmic tests**

Faced with the need for verification of our Transport code, we immediately noticed a lack of a comprehensive suite of tests in the literature able to check all aspects of the ADR solver. Thus, we devised what we consider a novel framework to achieve such a step by step verification. In that framework we have three degree of freedom space to increase complexity, we vary the flow field, the boundary conditions, and number of involving operators in a three-dimensional space, as follows (see Figure 3).



Figure 3: Transport algorithm testing incremental complexity.

* Operators: The key processes tested are the operators of advection, dispersion and reaction (e.g. growth or decay), which are representative of processes essential in an estuarine environment. These are tested individually, then in combinations of two and at the end all three together. Complexity increases with including more operators in a test.
* Flow field complexity and physical setup: Our framework included the following cases:

* + No flow: the test suite started with testing of Dispersion and Reaction operators in quiescent flow condition.
  + Uniform flow: These tests involved uniform steady flow on a channel, sometimes with a reverse in direction halfway through the simulation. The mass transported is Gaussian. The suite includes advection, diffusion and reaction alone and in the combinations indicated in Figure 3.
  + Tidal flow: This test used a tidal flow field from Wang et al. (2009), adapted to be 1-D and modified to be mass conserving, to test advection and reaction. The test itself has no analytical solution, but it is periodic in a way that the initial mass distribution moves forward and return to its initial position in each tidal cycle.
  + Spatial variation in flow field and dispersion coefficient (Zoppou): This test is basically due to Zoppou and Knight (1997) and includes velocity proportional to distance and diffusion coefficients proportional to distance squared. This test had to be modified for a conservative fluid flow.
* Boundary complexity: For the uniform flow and Zoppou tests, we include cases where the boundary is far away from the transported mass and cases where the boundary is actively part of the problem. This allows us to determine the extent to which convergence rates are affected by boundaries.

In what follows, we describe with some details all tests involved in the framework of algorithmic tests.

**4.4.1 Single operator tests**

## ODE solver tests

Linear decay equation solves by second order Heun method (see Appendix B for details) of ordinary differential equation solver and results are compared with analytical solution. That is to mention the remote boundary condition is meaningless in the context of an ODE solver. Test passes the defined criteria with 2nd order convergence ratio and the results are restrained in the acceptable range of accuracy.

Also a third order accurate Runge-Kutta solver for cases of stiff reaction problem was coded. The solver was checked versus linear decay analytical solution. Test passes the defined criteria with 2nd order convergence ratio and the results are restrained in the desirable range of accuracy (Figure 3).

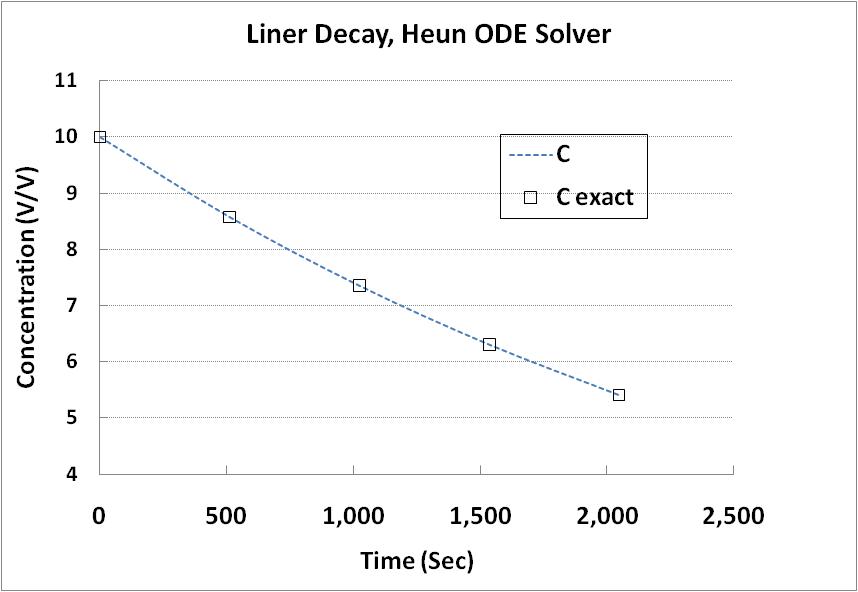


Figure : ODE (Reaction) solver test

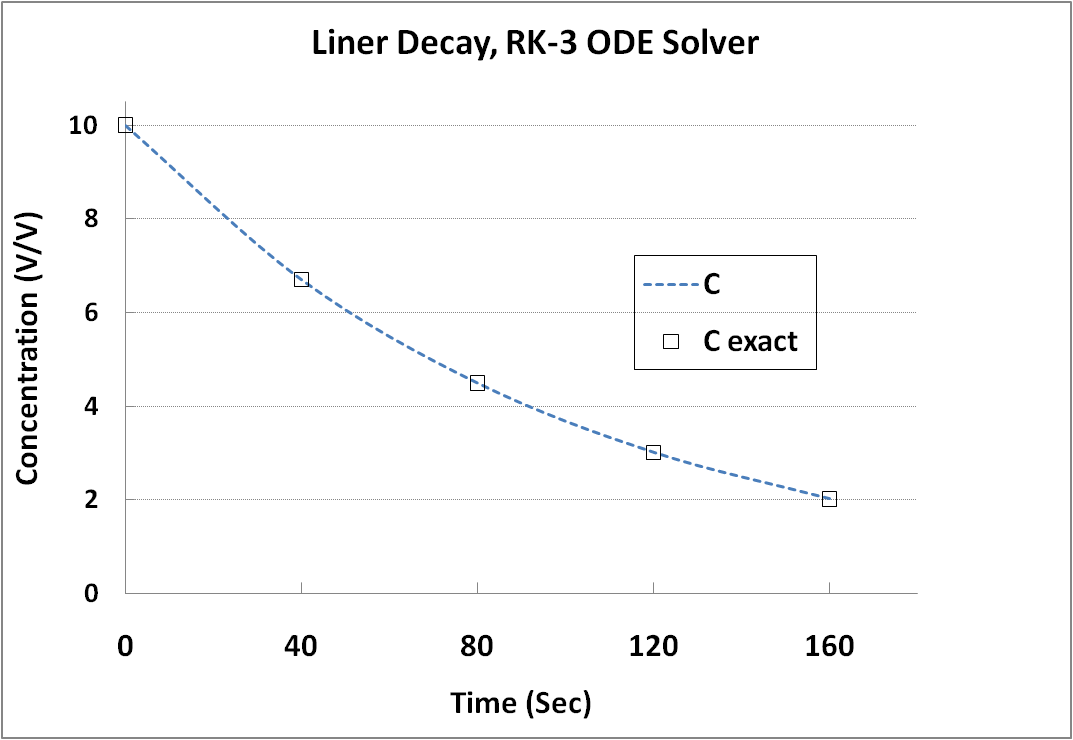


Figure : 3rd order Runge-Kutta ODE solver

## Quiescent Flow- Diffusion Test

Diffusion solver was verified versus analytical solution (second and third exact solutions, equation A2 , and A3). First, for the second analytical solution, boundaries were set far in a way that the boundary values became smaller than the machine's precision (remote boundary condition). Then, boundaries placed closer in a way the boundary values were taking a value greater than zero. Test was performed with both flux and value boundary condition. The test passes the defined criteria with 2nd order convergence ratio and the results are restrained in the acceptable range of accuracy.

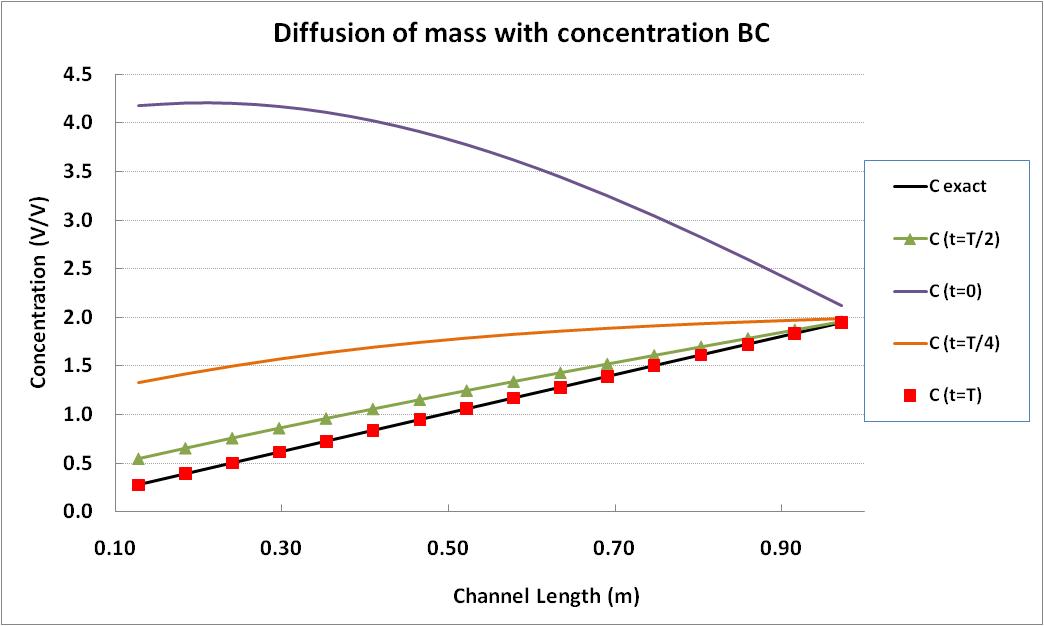


Figure 4: Diffusion single operator test, Dirichlet boundary condition

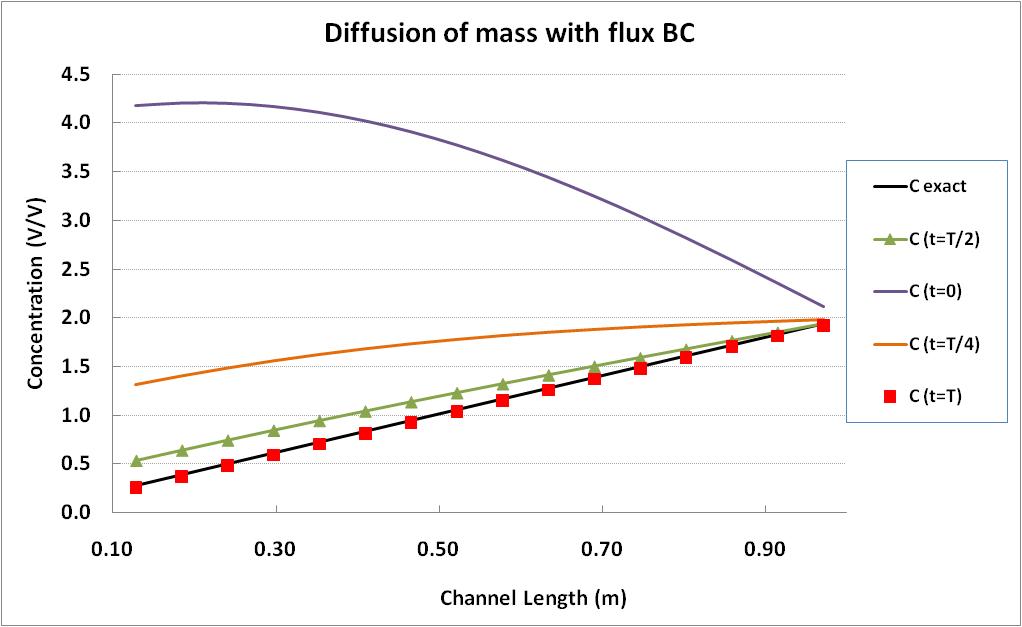


Figure 5: Diffusion single operator test, Neumann boundary condition

## 4.4.2 Uniform Flow Tests

* **Advection subjected to uniform flow:**

Advection solver was initially tested with remote boundary condition, where boundary values were defined far field where the machine precision dominates the boundary values. After passing these tests in both unidirectional and bidirectional flow setup the same tests repeated with domain boundaries set close to the Gaussian plume of mass. The detail of test where provided in the case two of analytical solutions (appendix A2). For unidirectional test, numerical results checked versus the exact solution and in the bidirectional flow field, numerical results checked versus the initial mass distribution. All of the above mentioned tests pass the predefined 2nd order accuracy in grid convergence study.

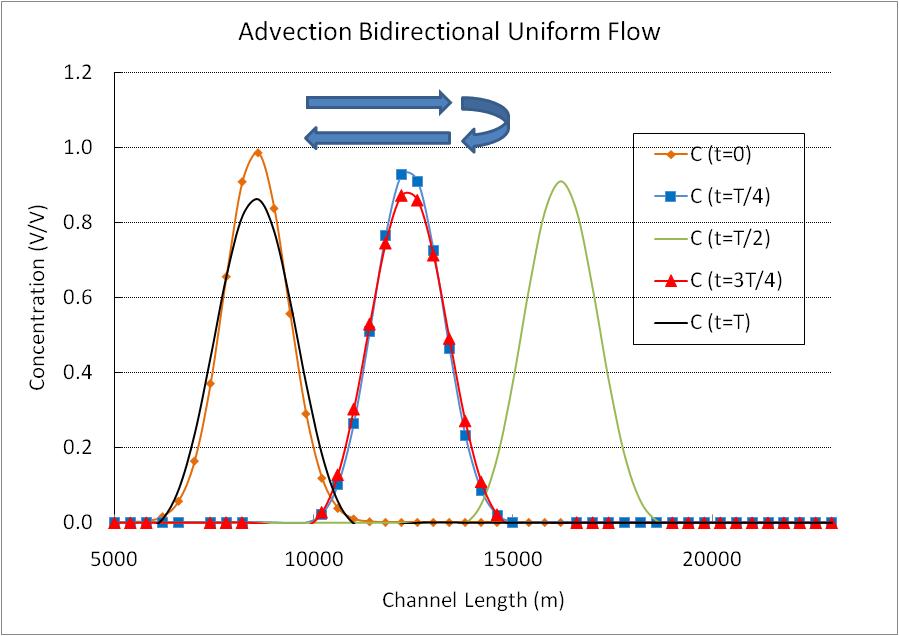


Figure 6: Advection solver, test bi-directional flow, Gaussian Plume of mass, remote boundary condition

* **Advection Reaction subjected to uniform flow:**

Advection solver and the ODE solver are integrated in one single routine. For details of numerical discritization see appendix BXXX. Using the exact solution case II, the Advection-Reaction solver was tested initially with remote boundary setup and then with active boundary. In both cases test was performed once in unidirectional flow and another time with bidirectional back and forth constant flow field. All of the above mentioned tests pass the predefined 2nd order accuracy in grid convergence study.

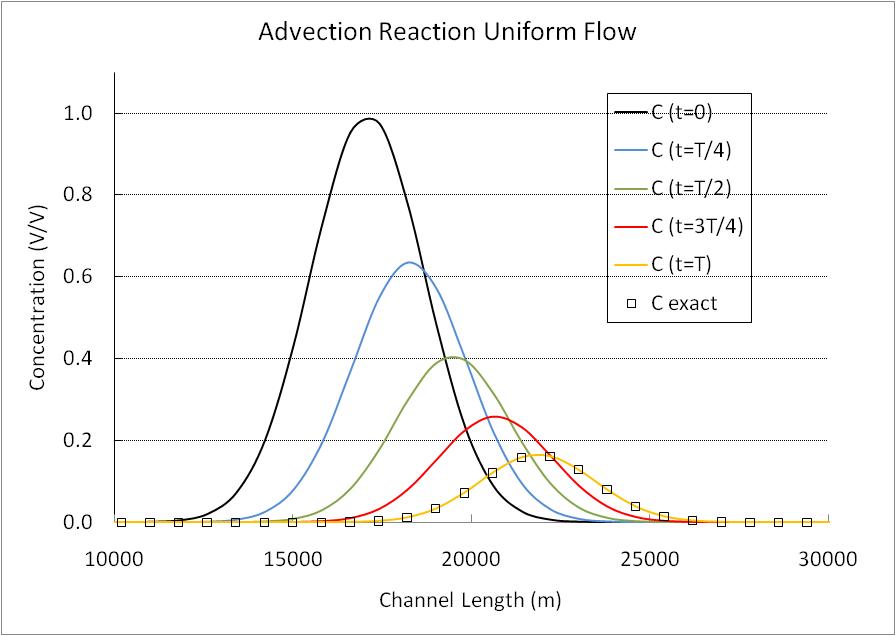


Figure 7: Advection Decay solver test, uniform flow

* **Advection Dispersion**

Advection and the diffusion solvers are combined with operator splitting to run this test. For details of numerical implementation see appendix BX. The Advection-Dispersion solver was initial tested with remote boundary setup and then with the active boundary. Test was performed with unidirectional constant flow field. Both active and remote boundary tests pass the predefined 2nd order accuracy in the grid convergence study.

* **Advection Dispersion Reaction**

Three solvers together are tested in this test. The test problem was the Gaussian mass distribution which was pushed forwarded by uniform flow and also it was subjected to linear decay (second exact solution). Similar to the previous cases test was performed in two steps. First, boundaries were far from the plume and practically their values were zero (remote boundary), and then close to the plume (active boundary). For the remote boundary set up test passed the predefined 2nd order accuracy and for the case with active boundary order of accuracy in the grid convergence study was close to 1st order.



Figure 8: Coupling advection diffusion and reaction with zero order implementation of intermediate boundary condition, L1 error norm convergence ratio drops from 2 to 1.02

## 4.4.3 Test with tidal flow field

* **Advection**

Advection solver is checked in this test. Since only the analytical solution of tidal flow field is available, the test is performed with remote boundary condition. The initial mass distribution must be at the same location after one tidal cycle (12.41 hr). Test is conducted with both sinusoidal and Gaussian initial mass distribution. The grid convergence test passes predefined second order error in both Gaussian and sinusoidal plume. Detail of the solution is given under the fifth exact solution.



Figure 9: Tidal flow field, advection solver with Gaussian initial mass distribution



Figure 10:Tidal flow field, advection solver with sinusoidal initial mass distribution

* **Advection Reaction**

Combination of Reaction and Advection solvers are checked in this test. Since only the analytical solution of tidal flow field is available, the test is performed with remote boundary condition. The initial mass distribution must be at the same location after one tidal cycle (12.41 hr). Test is conducted with both sinusoidal and Gaussian initial mass distribution. The grid convergence test passes predefined second order error in both Gaussian and sinusoidal cases. That is to mention integrating of two solvers (advection and reaction) is not extendable to the diffusion solver due to the lack of boundary conditional and also non-linearity of diffusion operator in a tidal flow field.

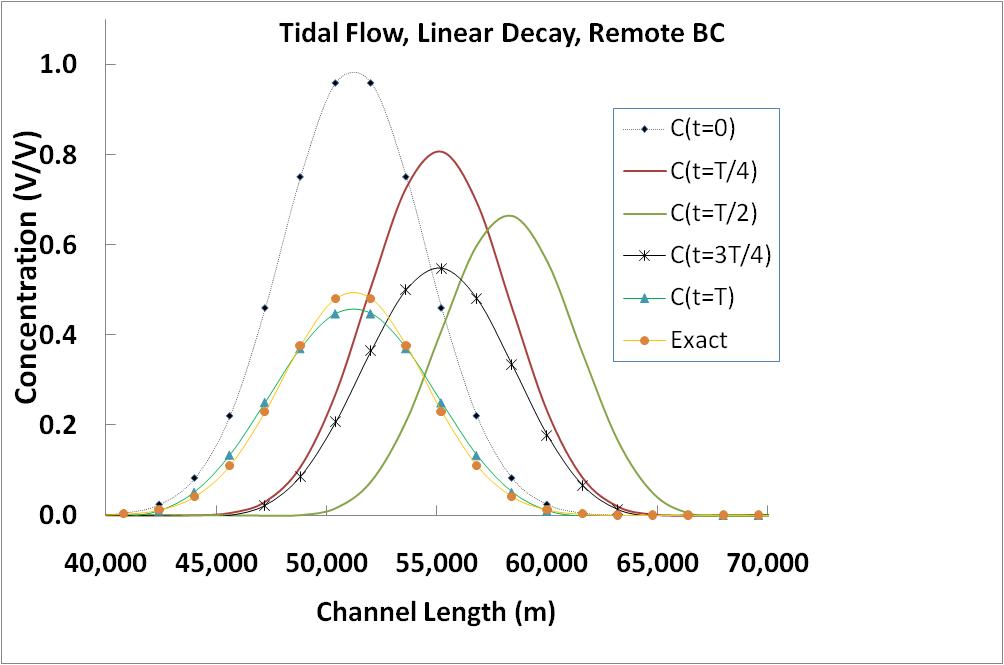


Figure 11: Tidal flow field, advection and reaction solvers with Gaussian initial mass distribution

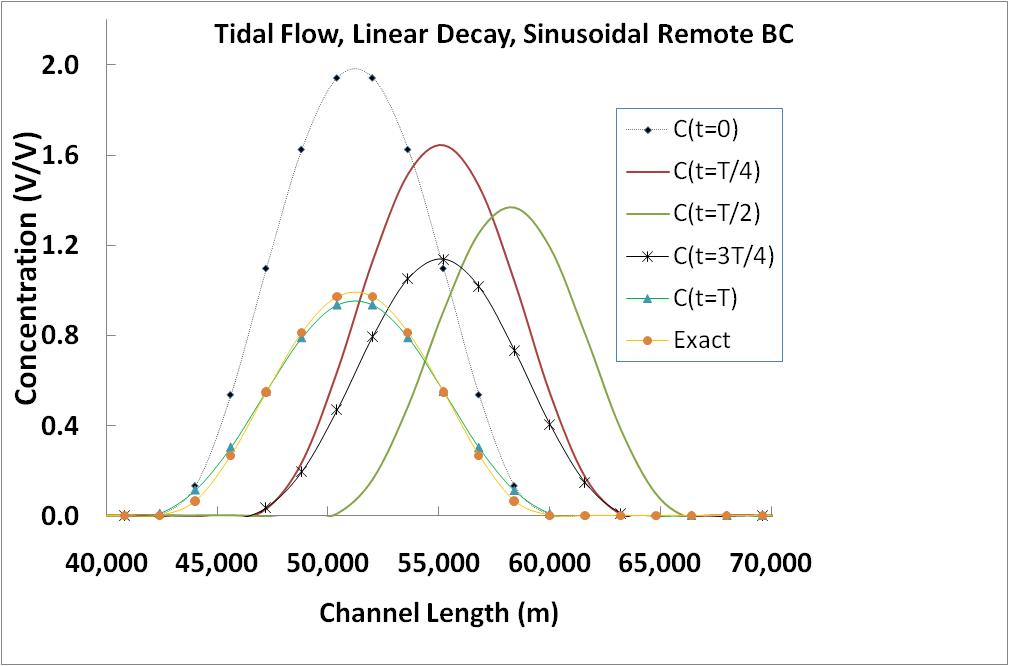


Figure 12: Tidal flow field, advection and reaction solvers with sinusoidal initial mass distribution

## 4.4.4 Test with spatially varying coefficient and flow field

Advection and Dispersion solvers are tested versus an analytical solution which was driven by Zoppou and Knight (1997). Both dispersion coefficient and velocity are function of space in that analytical solution (case four in the exact solutions). The exact solution was modified to satisfy continuity equation. The test was performed in both remote boundary and active boundary setups. Although the numerical results are in very good agreement with analytical solution, mesh convergence study for Godunov splitting of operators could not pass second order convergence rate.

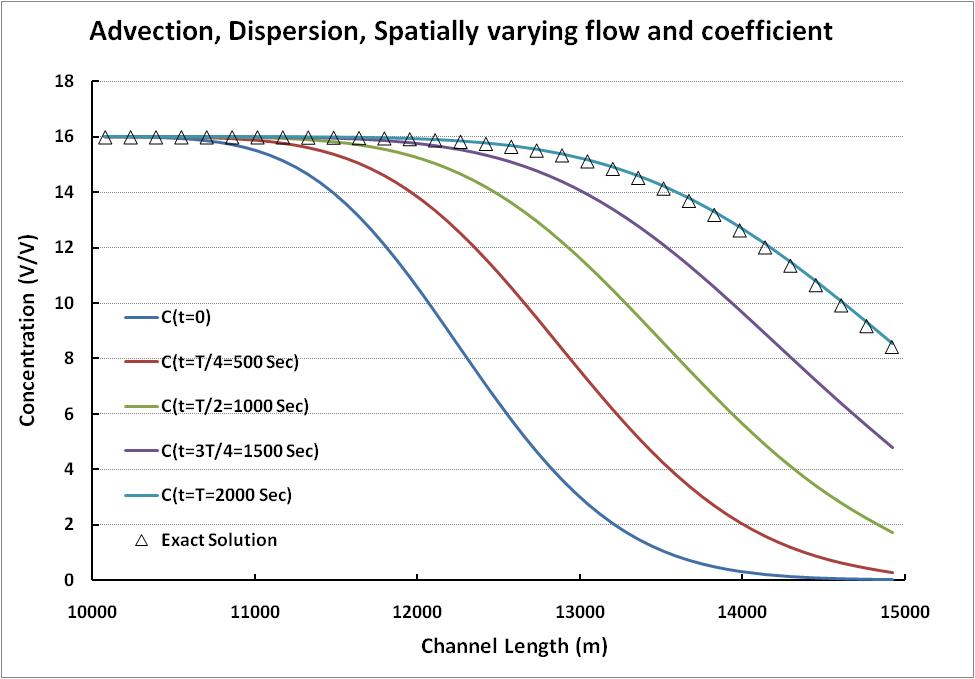


Figure 13: Spatially varying flow field and dispersion coefficient, advection and dispersion (Zoppou)

These tests were conducted for a range of parameter values that it usually occurs in an estuary. Typically the Courant number (a measure of numerical stability of the algorithm), domain length, and dispersion and decay coefficients were fixed, and the grid spacing and time steps were adjusted to maintain the same Courant number.

## 4.5 Detected errors

In the STM verification process, many initial programming errors was found and fixed in the level of unit testing. Then, in the next level of testing (algorithm and system testing) two following errors were detected utilizing the transport test suite. One of the errors was a programming error and the other was an inconsistency in the introduction of the source term.

## 4.5.1 Algorithmic error in advection

The problem was originated in the feeding of source term, which was coded in primitive value (concentration) in the extrapolate subroutine of advection routine:

conc\_lo(:,i)=conc(:,i)+half\*(-grad(:,i)-dtbydx\*grad(:,i)\*vel+dt\*source(:,i))

But latter in the update the conservative variables using divergence of fluxes the source term was introduced in the primitive values (mass):

mass(:,i)= ... +dt\*half\*source\_prev(:,i)\*area\_prev+ dt\*half\*source(:,i)\*area

At the beginning, the unidirectional advection grid convergence test was crafted for the advection routine. The test could not detect the error, because it employed unit area which is idempotent operation of multiplication. The test latter was improved to bidirectional flow to be a enhanced simulation of tidal environment and the unidirectional advection test was eliminated. While we thought every component works correct, there was a serious pitfall. The error in the foreword push of the mass plume was canceled out by the same amount of error in the backward move (figure 1). The symmetric nature of bidirectional flow concealed the algorithmic error in the advection solver. For that reason, The bug was not detectable with grid convergence study. After visual investigation of results evolution in time, the defect was noticed and then resolved.

Figure 14: Schematic of the algorithmic error in introduction of the source to advection solver

## 4.5.2 Bug in imposing the boundary condition of diffusion

Incorrect array index in implementing boundary cells:

it was: ... right\_hand\_side(ncell-1,:)- theta\_stm\*(dt/dx)\*flux\_end

and it corrected to: ... right\_hand\_side(ncell,:)- theta\_stm\*(dt/dx)\*flux\_end

The buggy line above is underlined. We did not expect the mesh convergence test to detect this bug. Although the above bug reduced the accuracy in numerical results by several order of magnitude, it did not impact the order of convergence. The bug was found through the assigning of very large number to the concentration values which are not using in the boundary value. The reason was as follows: As a general rule we intuitively think that an *O(hP)* local truncation error (LTE) leads to and *O(hP)* and an scheme with the convergence order of "p". In some cases the LTE can be lower in order at limited points without affecting the global order of error convergence. In the standard three point finite volume discretization of an elliptic differential equation, local reduction of order of accuracy in the interior cells up to 1 order hides in global error convergence ratio. What is more, the local reduction of order of error in boundary cells up to 2 order does not affect global order of convergence. Hence the classic grid convergence test could not detect the above indexing error. There are two ways to find the error: First implement point-wise error grid convergence study instead of global assessment of error. Second, perturbation of a point in domain up to the order of error buffer can cause reduction in the order of accuracy in case of existence of any bug.

if we are dealing with the elliptic system:

The relation between local error τ and global error ε is:

Since the infinity error norm is the most restrictive band the proof for automatically concludes proof for the other error norms.

so if is from order of O(hp), then on the boundary and interior nodes consider the global order of error. It is proven that is in the interior cells and on the boundary nodes (For example see Leveque, 2007) therefore with . of in interior cells and . of on the boundary the whole system still yields . of

**CHAPTER 5**

**DEVELOPMENT OF A WEBSITE**

**WITH DATA ON SEDIMENT**

**TRANSPORT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**5.1 Generalities**

**Appendix A**

# Numerical Discretization

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**A1. Advection, Lax Two Step Method**

 (A1.1)

*A,Q,S*, and *Ks* are known from other parts:

*A* and *Q* from HYDRO

*Ks* from Diffusion

*S* (Sink and source come from decay, deposition and entrainment)

1- First half step:

 (A1.2)

Where









 (A1.3)

2- Second half step:

 (A1.4-a,b)

*x*

*t*

*n*

*n+½*

*n+1*

*i-1*

*i+1*

*i*

Figure A1: schematic of lax two step method

**A1-1. Flux limiter:**

Flux limiters (slope limiters) are used in high resolution schemes, to avoid the spurious oscillations (wiggles) that would otherwise occur with high order spatial discretization schemes due to shocks, discontinuities or sharp changes in the solution domain.

van Leer (1977) flux limiter is one of the widely used limiters, which guarantees no new maximum/ minimum formed, and it could be formulated as follows for coding proposes: (Saltzman 1994).

1- Second order van Leer flux limiter:

 (A1.5)

Where:





And the limited flux will be:



2- Fourth order van Leer flux limiter:

 (A1.6)

where:



Note:  and  are the u which are already *2*nd order flux limiter applied on them

And the limited flux will be: 

**A2: Diffusion, Crank-Nicolson Method**



(A2.1)

 (A2.2)





is unknown in (A2.2) and other terms are known from measurements or previous step.



(A2.3)

In which *F* is diffusive flux re-writing (A3.2) yields:

 (A2.4)

**A2.1 Neumann Boundary condition implementation**



Just by replacing F in the first and last diffusive flux Neumann Boundary condition will be implemented

* Middle row will be:



* First row: *i=1*



* Last row*: i=m*



**A2.2 Dirichlet Boundary condition implementation**

We assume the C is known on the face of first/last cells (edges of channel)

*x*

*t*

*n*

*n+1*

*C1*

*C3*

*C2*

*C\**

Figure A2: Schematic of boundary condition implementation

**2.1- c(x) = a + bx+dx2 (quadratic)**

what we need is  (A2.8)

1-at x=0 🡪 c(0) = a+ b(0)+d(0) =a=c\*

2-at x=Δx /2 🡪 c(1/2) = c\* +bΔx /2+ dΔx /4

3-at x=3Δx /2 🡪 c(3/2) = c\*+3bΔx /2+ 9dΔx /4

where b, d, c(1/2) and c(3/2) are unknowns.

If we eliminate d between equations 2, and 3 above, regardless of time.

 (A2.9)

The only known value in the above is c\* = c at the boundary, by replacing in (A2.6-7) we may compute the changes in coefficient and right hand side matrices.

**2.2- c(x) = a + bx (linear)**

what we need is  (A2.10)

1. at x=0 🡪 c(0) = a + b(0) = c\*
2. at x=Δx/2 🡪 c(1/2) = a +bΔx /2 =c\* + bΔx /2

* b = (2c(1/2) – 2c\* )/ Δx

where b and c(1/2) are unknowns and C\* is the known value of the boundary

### Dirichlet Boundary Condition

For the left boundary,  (A2.11)

The right boundary,  (A2.12)

* Middle row: is the same as previous case
* First row : *i=1*

 (A2.13)

* Last row: *i=m*

 (A2.14)

**Appendix B**

# Analytical solutions

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## B1. Linear Decay

The (B1) first order Ordinary Differential Equation (ODE) has the analytical solution (B2)

## B2.Advection-Diffusion-Reaction with constant flow and linear decay

The equation (B3) has an analytical solution of (B4):

The initial and boundary condition could be derived from Analytical solution (Sobey 1989; Khan and Liu 1995)

## B3. Diffusion Analytical Solution

Convergence tests were performed for a case taken from Fletcher (1991), which uses both Neumann and Dirichlet boundary conditions that varies in time. In this particular case, advection and reactions are not present, meaning that only the pure dispersion term is needed from the A-D-R equation. Recalling the dispersion equation (B5) from before:

will be solved in the spatial interval *0.1 ≤ x ≤ 1.* The analytical solution (B6):

Both Neumann and Dirichlet boundary conditions could be retrieved from the above exact solution.

## B4. Analytical solution of tidal forcing in a rectangular 2D basin

Governing equation in 2D (X-Z):

1-Continuity

Assumptions:

* u, is not function of z but vertical velocity w is a function of z
* H (depth) is constant
* Ζ << H

2- Momentum

Assumptions:

* Inviscid fluid (interfacial and bottom friction are neglected)
* ρ=constant
* Non-rotating reference frame (f=0)

Analytical solution for velocity field is (B7):

Integration on *x* to get cell average value

Where

a: is amplitude (0.25-0.5 m)

L: basin length 100,000 m

Width: width (we assume unit width)

H: Depth (16 m)

A = width× =

Q =A × u

Retrieving Area from discharge for sake of mass continuity:

Area cell average

This analytical solution of tidal flow field is from the book: Principles of Physical Oceanography Neumann, G., and Pierson W. J., 1966.

## B5. Advection Diffusion solution by Zoppou and Knight (1997)

This is subjected to:

The solution is (B16):

**Appendix C**

# Sediment Sink and Source

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**