DSM2 Sediment Transport Module

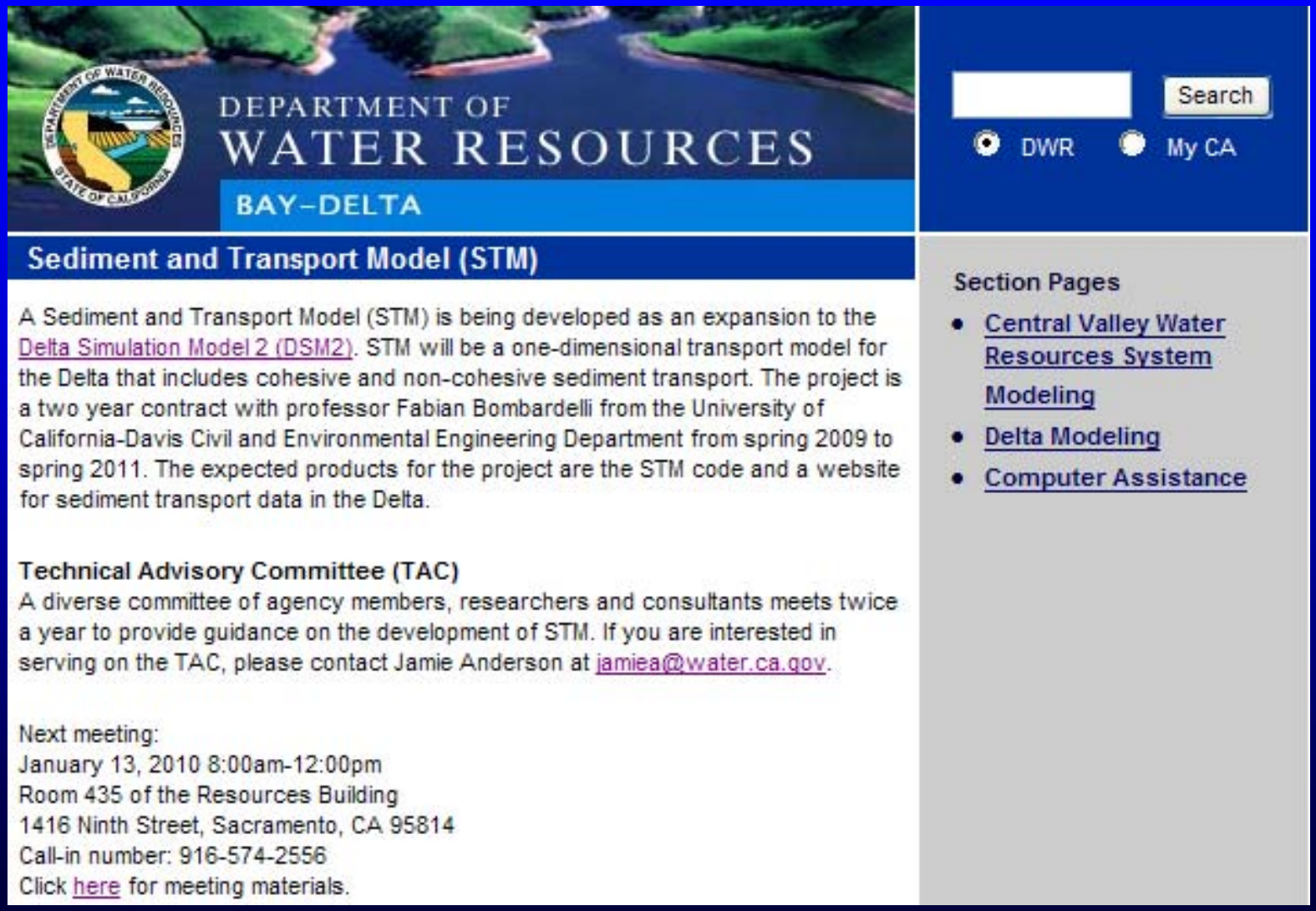
DSM2 User Group Update
January 27, 2010

Jamie Anderson, Ph.D., P.E.

Department of Water Resources
Modeling Support Branch
Bay-Delta Office

STM Website

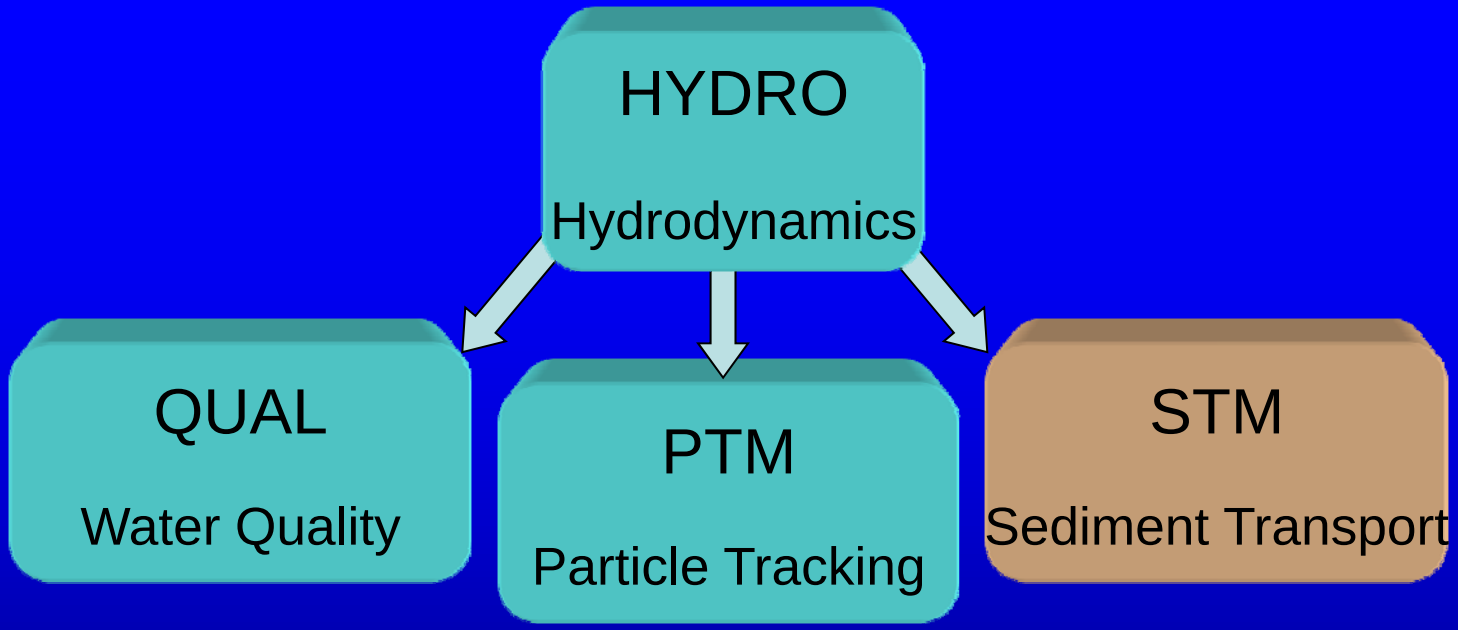
http://baydeltaoffice.water.ca.gov/modeling/deltamodeling/models/stm/stm.cfm



California Water and Environmental Modeling Forum Conference

- Feb 22-24, 2010
- Asilomar near Monterey
- STM update in DSM2 session
- http://cwemf.org

DSM2 Sediment Transport Module (STM)



2 year MACHRO contract with UC Davis

PI Fabian Bombardelli Civil & Env. Engineering

- Kaveh Zamani (modeling)
- Jamie Kohne & Joseph Waltz (data)*

Project products

- Web site with available sediment data in the Delta
- DSM2 Sediment Transport Module
  - Suspended sediment and bed load
  - Multiple sediment class sizes
  - One-dimensional model

*supported by another funding source

STM Code Development Plan

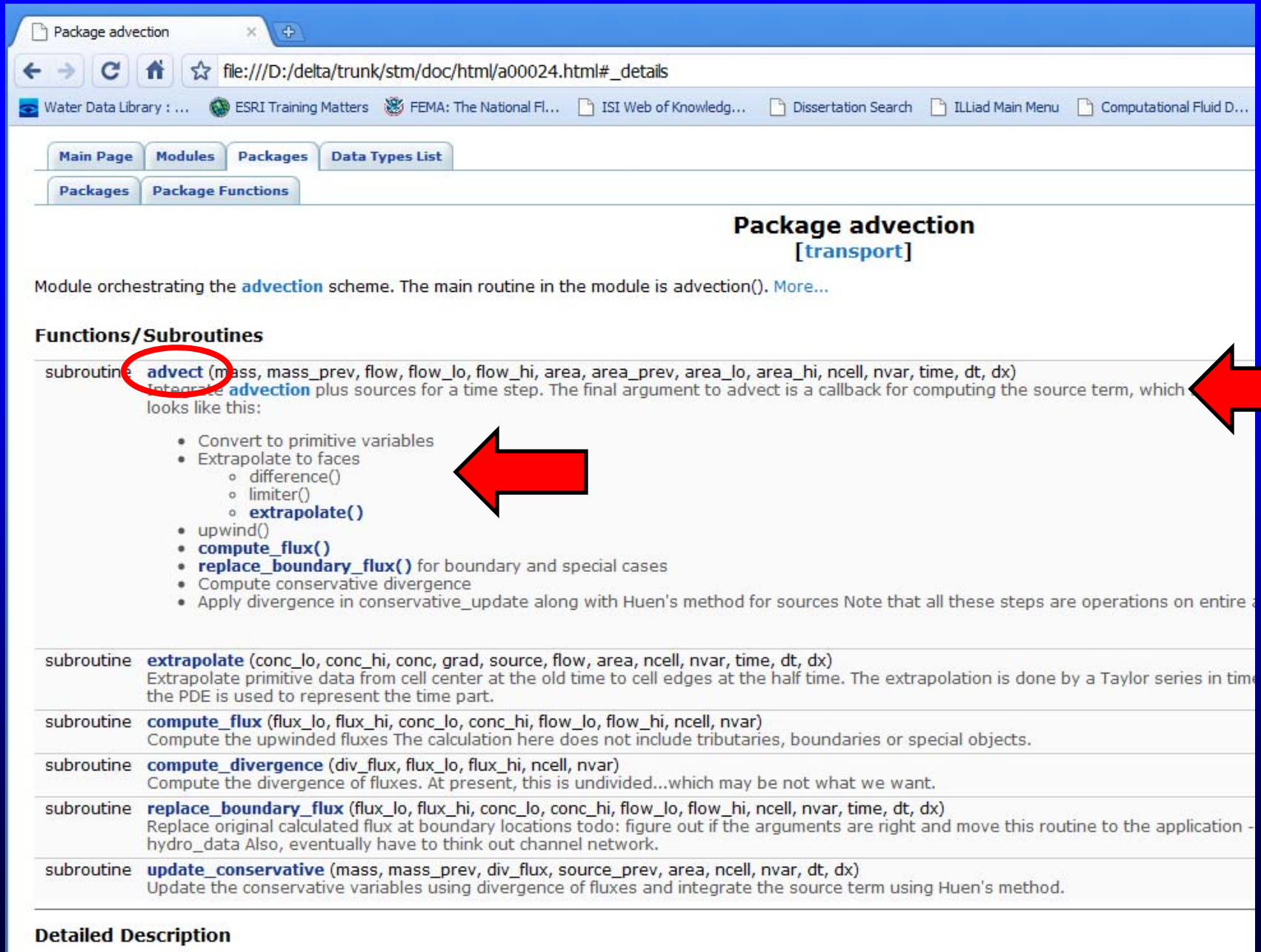
- Flexible, modular design
- Generalize Eulerian transport that could be adapted to other constituents
  - New fixed frame of reference transport model
  - Clean slate, develop testable code
  - Combine resources with data assimilation project
  - Future benefit: baroclinic term in HYDRO, salinity based op rules
- Self documenting with Doxygen

STM=Sediment Transport Module Eulerian=fixed frame of reference

Self-Documenting Code using Doxygen

```
!> Integrate advection plus sources for a time step.
!> The final argument to advect is a callback for computing the source term,
!> which should conform to the source_if interface
!> The algoritm looks like this:
!>    - Convert to primitive variables
!>    - Extrapolate to faces
!>        - difference()
!>        - limiter()
!>        - extrapolate()
!>    - upwind()
!>    - compute_flux()
!>    - replace_boundary_flux()   for boundary and special cases
!>    - Compute conservative divergence
!>    - Apply divergence in conservative_update along with Huen's method for sources
!>   Note that all these steps are operations on entire arrays of values -- this ke
subroutine advect(mass,      &
                  mass_prev,&
                  flow,      &
                  flow_lo,   &
                  flow_hi,   &
                  area,      &
                  area_prev,&
                  area_lo,   &
                  area_hi,   &
                  ncell,     &
                  nvar,      &
                  time,      &
                  dt,        &
                  dx)
```

Self-Documenting Code using Doxygen



Package advection

file:///D:/delta/trunk/stm/doc/html/a00024.html#_details

Water Data Library : ... | ESRI Training Matters | FEMA: The National Fl... | ISI Web of Knowledg... | Dissertation Search | ILLiad Main Menu | Computational Fluid D...

Main Page | Modules | Packages | Data Types List

Packages | Package Functions

Package advection
[transport]

Module orchestrating the advection scheme. The main routine in the module is advection(). More...

Functions/Subroutines

subroutine **advect** (mass, mass_prev, flow, flow_lo, flow_hi, area, area_prev, area_lo, area_hi, ncell, nvar, time, dt, dx)
Integrate **advection** plus sources for a time step. The final argument to advect is a callback for computing the source term, which looks like this:

- Convert to primitive variables
- Extrapolate to faces
  - difference()
  - limiter()
  - **extrapolate()**
- upwind()
- **compute_flux()**
- **replace_boundary_flux()** for boundary and special cases
- Compute conservative divergence
- Apply divergence in conservative_update along with Huen's method for sources Note that all these steps are operations on entire

subroutine **extrapolate** (conc_lo, conc_hi, conc, grad, source, flow, area, ncell, nvar, time, dt, dx)
Extrapolate primitive data from cell center at the old time to cell edges at the half time. The extrapolation is done by a Taylor series in time the PDE is used to represent the time part.

subroutine **compute_flux** (flux_lo, flux_hi, conc_lo, conc_hi, flow_lo, flow_hi, ncell, nvar)
Compute the upwinded fluxes The calculation here does not include tributaries, boundaries or special objects.

subroutine **compute_divergence** (div_flux, flux_lo, flux_hi, ncell, nvar)
Compute the divergence of fluxes. At present, this is undivided...which may be not what we want.

subroutine **replace_boundary_flux** (flux_lo, flux_hi, conc_lo, conc_hi, flow_lo, flow_hi, ncell, nvar, time, dt, dx)
Replace original calculated flux at boundary locations todo: figure out if the arguments are right and move this routine to the application - hydro_data Also, eventually have to think out channel network.

subroutine **update_conservative** (mass, mass_prev, div_flux, source_prev, area, ncell, nvar, dt, dx)
Update the conservative variables using divergence of fluxes and integrate the source term using Huen's method.

Detailed Description

STM Code Development Plan

- Flexible, modular design
- Generalize Eulerian transport that could be adapted to other constituents
  - New fixed frame of reference transport model
  - Clean slate, develop testable code
  - Combine resources with data assimilation project
  - Future benefit: baroclinic term in HYDRO, salinity based op rules
- Self documenting with Doxygen
- Companion code testing routine

STM=Sediment Transport Module Eulerian=fixed frame of reference

STM Code Testing

STM Code

Testing Code
Calls STM functions

Each function in STM has tests

Code and analytical tests

Test wide range of scenarios

Produce report of pass/fail result

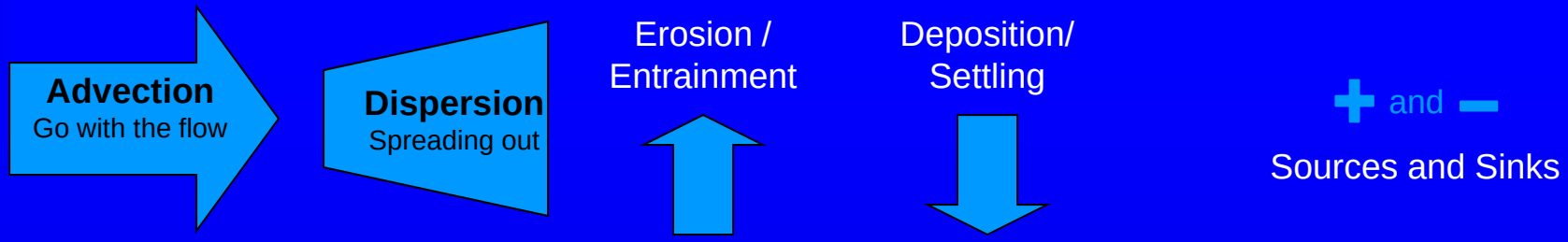
Run tests regularly

Unit testing using FRUIT
(Fortran Unit Test Framework)

Test each “unit” of code

STM=Sediment Transport Module

Sediment Transport Processes



Advection | Dispersion | Erosion | Deposition | Source/Sink | Tributaries

Progress to Date: Single Channel

Advection | Dispersion | Erosion

Next step: Complete single channel model

Advection | Dispersion | Erosion | Deposition | Source/Sink | Tributaries

Next step: Extend model to a channel network

Thank You!

Jamie Anderson
jamiea@water.ca.gov
916-654-5455

Fabian Bombardelli
fabianbombardelli2@gmail.com
530-752-0949

Photo by Ralph Finch Jan 4, 2006