Tutorial 4:
Standard Deviations for *Prior* Error Correlations

Prior Error Covariance Modeling

Recall: $\mathbf{B}_x = \mathbf{K}_b \boldsymbol{\Sigma} \mathbf{C} \boldsymbol{\Sigma}^T \mathbf{K}_b^T$

where $\boldsymbol{\Sigma}$ is a diagonal matrix of *prior* error standard deviations.

Several possible ways to estimate $\boldsymbol{\Sigma}$:

(i) Climatological variance:

Consider an ensemble of randomly chosen ocean states with covariance $\mathbf{B}_c$.

Before doing any data assimilation, the *prior* $\mathbf{x}_b$ will also be a valid member of this ensemble, so we can use $\mathbf{B}_c$ as an estimate of the 4D-Var *prior* error covariance $\mathbf{B}_x$.

Prior Error Covariance Modeling

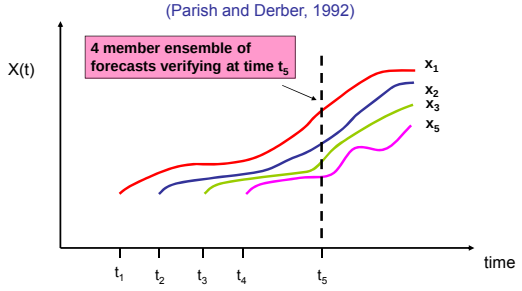
(ii) Variance from overlapping forecast ensembles (the "NMC method"):

After a few cycles of 4D-Var, the ROMS ocean state estimate will be more accurate than climatology, so $\mathbf{B}_c$ will not be an appropriate estimate for the *prior* error covariance $\mathbf{B}_x$ of $\mathbf{x}_b$.

In this case, an ensemble of overlapping forecasts that all verify at the initial time of the 4D-Var analysis can be used to estimate $\mathbf{B}_x$, if one is available.

The NMC Method

(Parish and Derber, 1992)



$$\mathbf{B}_{\mathbf{x}} \approx \frac{1}{4}\sum_{i=1}^{4}\left(\mathbf{x}_i(t_5)-\overline{\mathbf{x}}(t_5)\right)\left(\mathbf{x}_i(t_5)-\overline{\mathbf{x}}(t_5)\right)^{\mathrm{T}}$$

Used in ROMS 4D-Var by Powell et al (2008)

4D-Var Standard Deviation Files

- Four different standard deviation NetCDF files are required in ROMS 4D-Var algorithms to convert modeled, prior error correlations (C) to error covariances (B_x):
  - Model error standard deviation file, if weak constraint
  - Initial conditions standard deviation file
  - Open boundary conditions standard deviation file, if ADJUST_BOUNDARY
  - Surface forcing standard deviation file, if ADJUST_WSTRESS and/or ADJUST_STFLUX
- These standard deviation NetCDF files are specified in 4D-Var input script as:

```
STDnameM == ../Data/wc13_std_m.nc
STDnameI == ../Data/wc13_std_i.nc
STDnameB == ../Data/wc13_std_b.nc
STDnameF == ../Data/wc13_std_f.nc
```

Model Error and Initial Conditions Metadata

```
Variables:
  . . .
        double zeta(ocean_time, eta_rho, xi_rho) ;
                zeta:long_name = "free-surface standard deviation" ;
                zeta:units = "meter" ;
                zeta:time = "ocean_time" ;
                zeta:coordinates = "lon_rho lat_rho ocean_time" ;
        double ubar(ocean_time, eta_u, xi_u) ;
                ubar:long_name = "vertically integrated u-momentum component standard deviation" ;
                ubar:units = "meter second-1" ;
                ubar:time = "ocean_time" ;
                ubar:coordinates = "lon_u lat_u ocean_time" ;
        double vbar(ocean_time, eta_v, xi_v) ;
                vbar:long_name = "vertically integrated v-momentum component standard deviation" ;
                vbar:units = "meter second-1" ;
                vbar:time = "ocean_time" ;
                vbar:coordinates = "lon_v lat_v ocean_time" ;
        double u(ocean_time, s_rho, eta_u, xi_u) ;
                u:long_name = "u-momentum component standard deviation" ;
                u:units = "meter second-1" ;
                u:time = "ocean_time" ;
                u:coordinates = "lon_u lat_u s_rho ocean_time" ;
        double v(ocean_time, s_rho, eta_v, xi_v) ;
                v:long_name = "v-momentum component standard deviation" ;
                v:units = "meter second-1" ;
                v:time = "ocean_time" ;
                v:coordinates = "lon_v lat_v s_rho ocean_time" ;
        double temp(ocean_time, s_rho, eta_rho, xi_rho) ;
                temp:long_name = "potential temperature standard deviation" ;
                temp:units = "Celsius" ;
                temp:time = "ocean_time" ;
                temp:coordinates = "lon_rho lat_rho s_rho ocean_time" ;
        double salt(ocean_time, s_rho, eta_rho, xi_rho) ;
                salt:long_name = "salinity standard deviation" ;
                salt:time = "ocean_time" ;
                salt:coordinates = "lon_rho lat_rho s_rho ocean_time" ;
```

Matlab script: matlab/4dvar/c_std.m
CDL script: Data/ROMS/CDL/s4dvar_std_i.cdl

Open Boundary Conditions Metadata

```
dimensions:

        xi_rho = 56 ;
        eta_rho = 55 ;
. . .
        IorJ = 56 ;
        boundary = 4 ;

variables:
. . .
        double zeta_obc(bry_time, boundary, IorJ) ;
                zeta_obc:long_name = "free-surface, open boundaries conditions standard deviation" ;
                zeta_obc:units = "meter" ;
                zeta_obc:time = "bry_time" ;
        double ubar_obc(bry_time, boundary, IorJ) ;
                ubar_obc:long_name = "vertically integrated u-momentum component, open boundaries conditions standard deviation" ;
                ubar_obc:units = "meter second-1" ;
                ubar_obc:time = "bry_time" ;
        double vbar_obc(bry_time, boundary, IorJ) ;
                vbar_obc:long_name = "vertically integrated v-momentum component, open boundaries conditions standard deviation" ;
                vbar_obc:units = "meter second-1" ;
                vbar_obc:time = "bry_time" ;
        double u_obc(bry_time, s_rho, boundary, IorJ) ;
                u_obc:long_name = "u-momentum component, open boundaries conditions standard deviation" ;
                u_obc:units = "meter second-1" ;
                u_obc:time = "bry_time" ;
        double v_obc(bry_time, s_rho, boundary, IorJ) ;
                v_obc:long_name = "v-momentum component, open boundaries conditions standard deviation" ;
                v_obc:units = "meter second-1" ;
                v_obc:time = "bry_time" ;
        double temp_obc(bry_time, s_rho, boundary, IorJ) ;
                temp_obc:long_name = "potential temperature, open boundaries conditions standard deviation" ;
                temp_obc:units = "Celsius" ;
                temp_obc:time = "bry_time" ;
        double salt_obc(bry_time, s_rho, boundary, IorJ) ;
                salt_obc:long_name = "salinity, open boundaries conditions standard deviation" ;
                salt_obc:time = "bry_time" ;

// global attributes:
. . .
                :boundary_index = "West=1, South=2, East=3, North=4"
```

Matlab script: matlab/4dvar/c_std_bry.m
CDL script: Data/ROMS/CDF/s4dvar_std_b.cdl

Surface Forcing Metadata

```
dimensions:

        xi_rho = 56 ;
        xi_u = 55 ;
        xi_v = 56 ;
        eta_rho = 55 ;
        eta_u = 55 ;
        eta_v = 54 ;
        s_rho = 30 ;
        ocean_time = UNLIMITED ; // (1 currently)

variables:
. . .
        double sustr(ocean_time, eta_u, xi_u) ;
                sustr:long_name = "surface u-momentum stress standard deviation" ;
                sustr:units = "newton meter-2" ;
                sustr:time = "ocean_time" ;
                sustr:coordinates = "lon_u lat_u ocean_time" ;
        double svstr(ocean_time, eta_v, xi_v) ;
                svstr:long_name = "surface v-momentum stress standard deviation" ;
                svstr:units = "newton meter-2" ;
                svstr:time = "ocean_time" ;
                svstr:coordinates = "lon_v lat_v ocean_time" ;
        double shflux(ocean_time, eta_rho, xi_rho) ;
                shflux:long_name = "surface net heat flux standard deviation" ;
                shflux:units = "watt meter-2" ;
                shflux:negative = "upward flux, cooling" ;
                shflux:positive = "downward flux, heating" ;
                shflux:time = "ocean_time" ;
                shflux:coordinates = "lon_rho lat_rho ocean_time" ;
        double ssflux(ocean_time, eta_rho, xi_rho) ;
                ssflux:long_name = "surface net salt flux (E-P)*SALT standard deviation" ;
                ssflux:units = "meter second-1" ;
                ssflux:time = "ocean_time" ;
                ssflux:coordinates = "lon_rho lat_rho ocean_time" ;


// global attributes:

                :type = "ROMS/TOMS 4D-Var surface forcing error covariance standard deviation" ;
                :title = "California Current System, 1/3 degree resolution (WC13)" ;
                :Conventions = "CF-1.4" ;
                :grd_file = "test/WC13/Data/wc13_grd.nc" ;
```

Matlab script: matlab/4dvar/c_std_frc.m
CDL script: Data/ROMS/CDF/s4dvar_std_f.cdl

Standard Deviation Matlab Scripts

There are several scripts in the matlab/4dvar repository to process standard deviations (https://www.myroms.org/svn/src):

- **c_std.m** creates model error or initial conditions NetCDF file
- **c_std_bry.m** creates open boundary conditions NetCDF file
- **c_std_frc.m** creates surface forcing NetCDF file

and associated drivers templates:

- **d_std.m**
- **d_std_unbalanced.m**
- **d_std_bry.m**
- **d_std_frc.m**

d_std.m

```
%
%  D_STD:  Driver script to compute and write 4D-Var initial condition
%          or model error (weak constraint) standard deviations.
%
%  This a user modifiable script that can be used to prepare ROMS 4D-Var
%  standard deviations which are used to convert modeled error correlations
%  to error covariances.
%
%  The first step is to run the model application for a period that is
%  long enough to compute meaningful circulation statistics, like mean
%  and standard deviations for all prognostic state variables: zeta, u,
%  v, temp, and salt.
%
%  As an example, we compute the 4D-Var standard deviations for the WC13
%  application from daily history files from 1/1/2000 to 12/25/2004.
%

% svn $Id: d_std.m 489 2010-07-10 19:08:20Z arango $
%=========================================================================%
%  Copyright (c) 2002-2010 The ROMS/TOMS Group                            %
%    Licensed under a MIT/X style license                                 %
%    See License_ROMS.txt                           Hernan G. Arango      %
%=========================================================================%

%  Set standard deviation NetCDF file. The file name is edited and the
%  month will be appended as *i_jan.nc:

STDfile = 'wc13_std_i.nc';

mstr = {'jan', 'feb', 'mar', 'apr', 'may', 'jun', ...
        'jul', 'aug', 'sep', 'oct', 'nov', 'dec'};
```

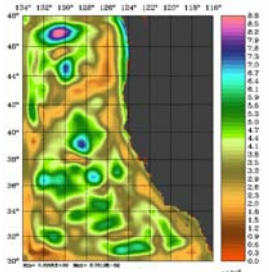
d_std_unbalanced.m

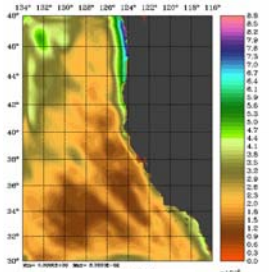
```
%
%  D_STD_UNBALANCED:  Driver script to compute and write 4D-Var unbalaced
%                     standard deviations.
%
%  This a user modifiable script that can be used to prepare ROMS 4D-Var
%  standard deviations when the balance operator is activated. They are
%  used to convert modeled error correlations to error covariances.
%
%  The first step is to run the model application for a period that is
%  long enough to compute meaningful circulation statistics, like mean
%  and standard deviations for all prognostic state variables: zeta, u,
%  v, temp, and salt.
%
%  As an example, we compute the 4D-Var standard deviations for the WC13
%  application from daily history files from 1/1/2000 to 12/25/2004.
%

% svn $Id: d_std_unbalanced.m 489 2010-07-10 19:08:20Z arango $
%=========================================================================%
%  Copyright (c) 2002-2010 The ROMS/TOMS Group                            %
%    Licensed under a MIT/X style license                                 %
%    See License_ROMS.txt                           Hernan G. Arango      %
%=========================================================================%

%  Set standard deviation NetCDF file. The file name is edited and the
%  month will be appended as *iu_jan.nc:

STDfile = 'wc13_std_iu.nc';

mstr = {'jan', 'feb', 'mar', 'apr', 'may', 'jun', ...
        'jul', 'aug', 'sep', 'oct', 'nov', 'dec'};
```

d_std_frc.m

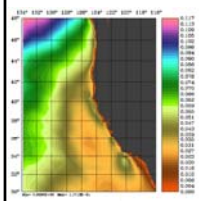
```
%
%  D_STD_frc:  Driver script to compute and write 4D-Var surface forcing
%              standard deviations.
%
%  This a user modifiable script that can be used to prepare ROMS 4D-Var
%  surface forcing standard deviations which are used to convert modeled
%  error correlations to error covariances.
%
%  The first step is to run the model application with air/sea bulk
%  fluxes (BULK_FLUXES option) for a period that is long enough to
%  compute meaningful circulation statistics, like mean and standard
%  deviations for surface forcing state variables: sustr, svstr,
%  shflux, and ssflux.
%
%  As an example, we compute the 4D-Var standard deviations for the WC13
%  application from daily history files from 1/1/2000 to 12/25/2004.
%

% svn $Id: d_std_frc.m 489 2010-07-10 19:08:20Z arango $
%=========================================================================%
%  Copyright (c) 2002-2010 The ROMS/TOMS Group                            %
%    Licensed under a MIT/X style license                                 %
%    See License_ROMS.txt                           Hernan G. Arango      %
%=========================================================================%

%  Set standard deviation NetCDF file. The file name is edited and the
%  month will be appended as *i_jan.nc:

STDfile = 'wc13_std_f.nc';

mstr = {'jan', 'feb', 'mar', 'apr', 'may', 'jun', ...
        'jul', 'aug', 'sep', 'oct', 'nov', 'dec'};
```

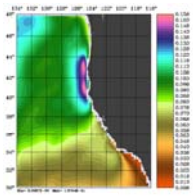
CCS Standard Deviation (January)
Initial Conditions

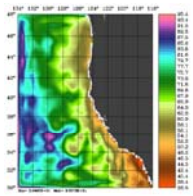

Free-surface


Free-surface (unbalanced)

CCS Standard Deviation (January)
Surface Forcing

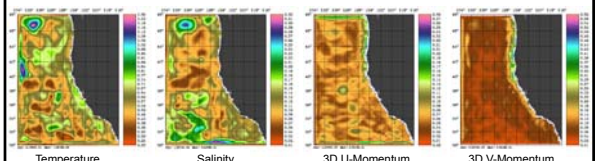

T_x


T_y

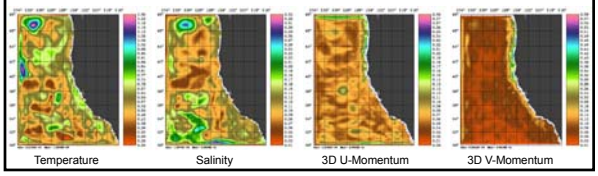

Net Heat Flux

CCS Standard Deviation (January)
Initial Conditions

Balanced

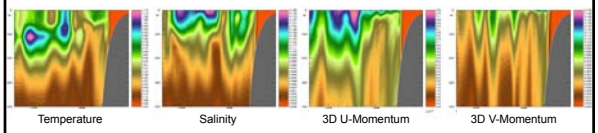

Temperature | Salinity | 3D U-Momentum | 3D V-Momentum

Unbalanced

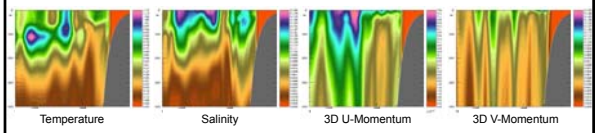

Temperature | Salinity | 3D U-Momentum | 3D V-Momentum

CCS Standard Deviations (January)
Initial Conditions, along 37ºN

Balanced



Unbalanced



References

- Parrish, D. F., Derber, J. C., 1992: The National Meteorological Center's Spectral Statistical-Interpolation Analysis System. *Mon. Wea. Rev.*, **120**, 1747-1763.
- Powell, B.S., H.G. Arango, A.M. Moore, E. DiLorenzo, R.F. Milliff and D. Foley, 2008: 4DVAR Data Assimilation in the Intra-Americas Sea with the Regional Ocean Modeling System (ROMS). *Ocean Modelling*, **23**, 130-145.