SOFT

Satellite-Based Ocean Forecasting
EVK3-CT-2000-00028
Coordinator: Dr. Joaquín Tintoré

Wave Forecasting Report – v. 1.0

WP 3 – Task 3.2
Hybrid SOFT tracking method
September 2003

Paolo Cipollini, Peter Challenor
Southampton Oceanography Centre, U.K.
cipo@soc.soton.ac.uk

Introduction

This report (`SOFT_WP32_rep1.pdf`) describes the first version of the wave forecasting code developed within Work Package 3, task 3.2 (implementation of a hybrid SOFT tracking system) of the SOFT Project. The forecasting of westward-propagating signals (planetary waves or westward-travelling eddies), using the fields of tracked wave from Work Package 3, task 3.1, is one of the two components of the hybrid system which is the overall deliverable of task 3.2. The results presented here are provisional and are likely to be replaced as research proceeds.

Related to this report are two other documents:

- the progress report on task 3.1 released in January 2003 (`SOFT_WP31_report.pdf`), which presents the rationale to the study and gives ample details on the scheme adopted for the fitting of elementary waves (according to a Gaussian wave shape model) to altimeter data (see also the paper by Cipollini, 2003);

- the handbook `SOFT_WP31_handbook.pdf` describing the suite of MATLAB programs developed within Work Package 3, task 3.1 of the SOFT Project, for the **tracking of large-scale, westward propagating features** (planetary waves or westward-travelling eddies) in altimeter data and the removal of the identified features from the datasets. The suite has been applied to TOPEX/POSEIDON data over the Azores region (one of the SOFT study regions) and the output results have been used for the forecast'

All documents and data relevant to Work Package 3 of the SOFT Project, including all the code described in this report, can be downloaded via anonymous FTP from ftp.soc.soton.ac.uk under directory `/pub/soft/distribution2`. That directory is further subdivided into:

- `…/data` : fitting results from task 3.1 plus some ancillary data

- `…/fitting` : fitting routines (task 3.1)

- `…/forecast` : forecast code

- `…/utils` : various utilities for data management, screening and plotting

- `…/docs` : documentation (incl. this report) and SOFT scientific papers

Data used

The forecast described in this report is based on the tables of planetary waves obtained with the software developed in task 3.1, and applied to the merged CLS MSLA product from CLS Space Oceanography Division in Toulouse. This is a merged TOPEX/Poseidon + ERS altimetric product, consisting of optimally-interpolated SSH anomalies on a 0.25° x 0.25° x 10 day grid, and produced in the framework of the Environment and Climate EU AGORA (ENV4-CT9560113) and DUACS (ENV44-T96-0357) Projects. The details of the optimal interpolation are covered in Le Traon and Dibarboure (1999), Ducet et al. (2000) and Le Traon et al. (2001). The current distribution (distribution 2.0) of the SOFT WP 3.1 data is based on the above dataset and covers the SOFT study area 20°N to 40°N and 75°W to 8°W, using 232 10-day cycles of CLS MSLA from early April 1995 to early August 2001.

In the final round-up of Work Package 3, task 3.2 we are going to use the newly available 1/3° x 7 day data from CLS, obtained from optimal interpolation and merging of all current altimeter missions (including Geosat Follow-On and ENVISAT). This should maximise the accuracy of the forecasted fields. Morevover, CLS are providing these data with just a delay of just a few days from acquisition, so they can be used for real operational forecasting with the SOFT techniques. Again, the codes described in this handbook lend themselves to be used on this novel dataset virtually without any modifications.

Description of the forecasting software

The forecasting is made with **soft_waveforecast** which is available under `/pub/soft/distribution2/forecast`. The code first finds which waves are active at the starting time t_0 of the forecast; only these waves will be propagated (forecast). Then the program builds a matrix where every column contains the parameters of an elementary wave from the set of all waves at that latitude. For any wave to be propagated (which we will call the *actual* wave), the code computes a distance of that wave at instant t_0 from every other elementary wave in parameters' space, that is in the space defined by amplitude, slope, width, longitude, using the matrix created above. The distance computation involves a set of weights, one for each one of the parameters. The wave closest to (that is with the minor distance to) the one to be propagated is then assumed as the *model* wave for the forecast of the actual wave at the next longitude value. (Note that, because the wave tracking has been implemented on discrete longitude values as waves propagate, we must now forecast the parameters of the wave when it gets to the next longitude value, rather than when it gets to the next time value, or in other words we must forecast by stepping in space rather than in time). Many different strategies are then possible for the forecast; we have tested two of them:

1) Proportional change: the parameters of the actual wave are changed by looking at the relative change of the parameters of the model wave (possibly introducing a damping exponent τ). So, if A_n is a parameter of the actual wave at longitude step n, and M_n is the corresponding parameter for the model wave, the parameter for the actual wave at the next longitude step (longitude indices increase westward here) is $A_{n+1} = A_n (S_{n+1}/S_n)^{\tau}$

2) Gradual attraction: the actual wave parameter at step $n+1$ is a combination of the one at step n on plus the model wave parameter at $n+1$, that is $A_{n+1}=(1-\tau)A_n+\tau M_{n+1}$ with $\tau \in [0,1]$

It must be noted that once the actual wave has been propagated by one step, a new set of distances from that wave to all the other waves is computed, and a new model wave (the closest) is selected to continue the forecast.

We have used as testbed for the development of the forecast code **soft_waveforecast** the CLS MSLA dataset, with the waves found by means of the wave fitting software. Cycle 196 has been taken as the initial time of the forecast (which is therefore more appropriate to call 'hindcast') and the previsions have been made over cycles 197-214 (a total of 18 cycles, corresponding to 6 months). Figure 3 shows the result of the hindcast with the 'gradual attraction' strategy and τ=0.6. Figure 3 a) displays the original longitude/time plot (after westward-filtering), figure 3 b) the plot of the fitted waves reconstructed with **soft_reconstruct** up to cycle 196, and the hindcast field after then. Figure 3 c) shows the difference of b) and a), having a standard deviation (RMS error of the hindcast) of 6.37 cm over the final six months.
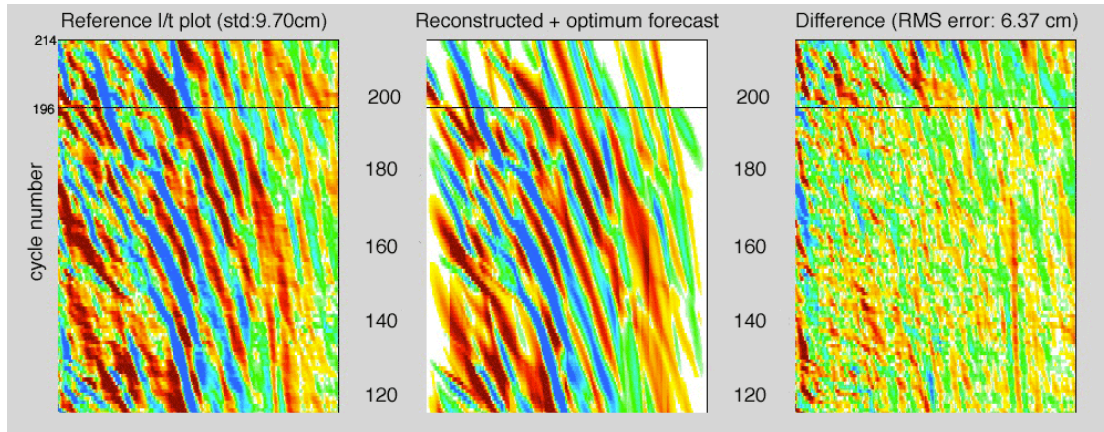
Figure 3: example of hindcast using **soft_waveforecast**. a) the original longitude/time plot (after westward-filtering) of CLS MSLA data; b) the plot of the fitted waves reconstructed with **soft_reconstruct** up to cycle 196, and the hindcast field obtained with **soft_waveforecast** after then; c) shows the difference b) minus a).

A comparison of the errors in the hindcast with the standard deviation of the original (westward-filtered) datasets is in figure 4, which also reports the error from simple persistence hindcast of the waves, i.e. propagating those waves existing at t_0 without changing their amplitude, slope and width (this can be also seen as particular case of the proportional change strategy, that is the case $\tau=0$). It is clear that within 2 and 4 months into the hindcast the 'gradual attraction' approach used above performs better than the simple 'persistence of wave properties' approach.
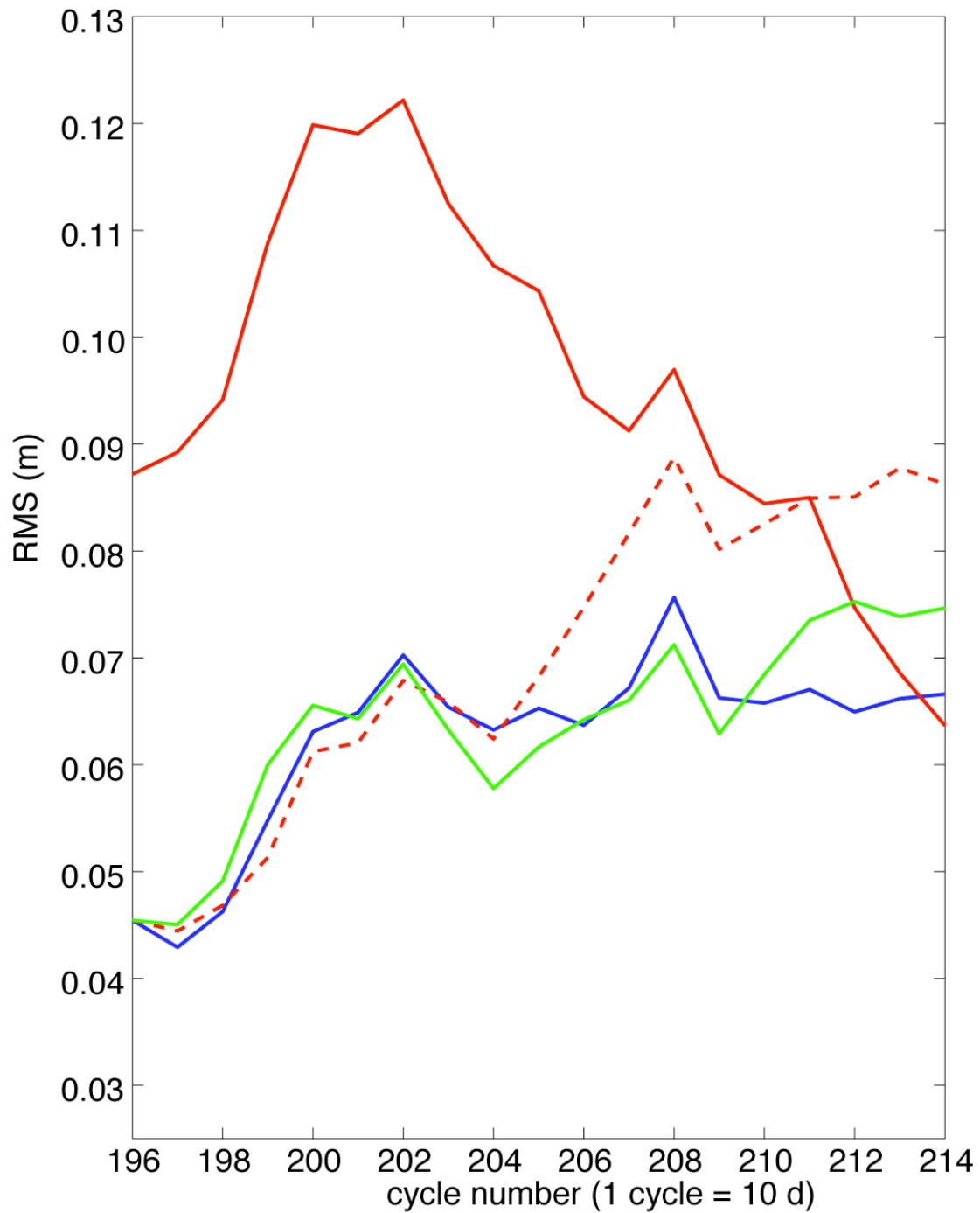
Figure 4. Comparison of RMS errors in different hindcasts. The blue and green curves represent the RMS error in two hindcasts based on the 'gradual attraction' strategy with τ=0.6 and two different sets of weight for the distance in parameters' space. The dashed red line is the RMS error from a hindcast based on simple persistence of wave parameters. The solid red line is the RMS of the original (westward-filtered) dataset, plotted as reference.

References

Cipollini, P., 2003: Multiple satellite observations of oceanic planetary waves: techniques and findings. *Proceedings of 2003 Tyrrhenian International Workshop on Remote Sensing*, ed. by E. Dalle Mese, pp 134-143, Edizioni Plus, Uni. of Pisa, Italy. ISBN 88-8492-291-7

Ducet, N., P.-Y. Le Traon, and G. Reverdin, 2000: Global high resolution mapping of ocean circulation from TOPEX/Poseidon and ERS-1 and -2. *J. Geophys. Res*., **105,** 19477-19498.

Le Traon, P.-Y. and G. Dibarboure, 1999: Mesoscale mapping capabilities of multi-satellite altimeter missions. *J. Atmos. Oceanic Technol*., **16,** 1208-1223.

Le Traon, P.-Y., G. Dibarboure, and N. Ducet, 2001: Use of a High-Resolution Model to Analyze the Mapping Capabilities of Multiple-Altimeter Missions. *J. Atmos. Oceanic Technol*., **18,** 1277-1288.

APPENDIX – Program listing

gapfill_gaussian_2d.m

```
function rwtfore=soft_waveforecast(rwtable,lon,ts,fwdts,dwamp,dwslo,dwwdt,dwlon)
% SOFT_WAVEFORECAST Forecast westward-propagating waves
%
%    RWTFORE = SOFT_WAVEFORECAST(RWT,LON,TS,FWDTS,DWAMP,DWSLO,DWWDT,DWLON)
%    forecasts the waves contained in structure RWT (created by one of the
%    SOFT wave-fitting programs) and outputs them in structure RWTFORE. LON (in
%    degrees) and TS (in cycles) are vectors specifying the definition domain of
%    the waves. The forecast is done over FWDTS cycles starting from T0=end(TS).
%    Two forecast strategies are possible (see in the code), both based on identifying
%    the 'most similar' wave to the one being forecasted using a distance in
%    parameters' space (parameters are amplitude, slope, width, longitude)
%    DWAMP,DWSLO,DWWDT,DWLON are the distance weights, taken here as multipliers of
the
%    squares under the square root in the Euclidean distance formula.
%
% This code is part of a suite of programs developed at the
% Southampton Oceanography Centre, U.K., within the framework
% of the Energy, Environment and Sustainable Development EU
% Project SOFT (Satellite-based Ocean ForecasTing) - contract
% number EVK3-CT-2000-00028
%
% Codes and documentation are available via anonymous FTP from
% ftp.soc.soton.ac.uk under /pub/soft
%
% v. 1.0 Paolo Cipollini 16/07/2003 (rev. 09/03 with comments)


rwtfore=rwtable;
nc=length(lon);
t0=ts(end);
tfore=(t0+1):(t0+fwdts);
%ltfore=zeros(fwdts,length(lon));

% 1 - find which waves to propagate

rwtoprop=rwtable;
for irw=length(rwtable):-1:1
    if (rwtoprop(irw).wave.intercept(end)-rwtoprop(irw).wave.slope(end))<t0,
        rwtoprop(irw)=[];
    end
end

% 2 - build buffermatrix (used to compute distances and locate minimum
% distance wave

bufmat=[];

for irw=1:length(rwtable)
    bufamp=rwtable(irw).wave.amplitude;
    bufslo=rwtable(irw).wave.slope;
    bufitc=rwtable(irw).wave.intercept;
    bufwdt=rwtable(irw).wave.width;
    buflon=rwtable(irw).wave.longitude;
    lospan=length(bufamp);
    % bufmat: every column is an elementary wave
    bufmat=[bufmat,...
            [bufamp; bufslo; bufitc; bufwdt; buflon; irw*ones(1,lospan); 1:lospan]];
    % lblvec: used below to join forecast with appropriate wave
    lblvec(irw)=str2num(rwtable(irw).wave.label(6:9));
end

% 3 - wave propagation
for irw=1:length(rwtoprop),
    Aamp=rwtoprop(irw).wave.amplitude(end);
    Aslo=rwtoprop(irw).wave.slope(end);
    Aitc=rwtoprop(irw).wave.intercept(end);
    Awdt=rwtoprop(irw).wave.width(end);
    Alon=rwtoprop(irw).wave.longitude(end);
    Alabel=rwtoprop(irw).wave.label;
    ip=find(lblvec==str2num(Alabel(6:9)));

    while (Alon>lon(1) & (Aitc-Aslo)<tfore(end))
```

```matlab
        % compute distance

        D = sqrt(dwamp*(bufmat(1,:)-Aamp).^2 + ...
                dwslo*(bufmat(2,:)-Aslo).^2 + ...
                dwwdt*(bufmat(4,:)-Awdt).^2 + ...
                dwlon*(bufmat(5,:)-Alon).^2);
           % penalty if signs are different
           % D=D+1e8*(sign(bufmat(1,:))~=sign(Aamp));
        % find and extract 'most similar' wave
        D(find(D==0))=1e10;  % necessary to exclude itself!
        isim=find(D==min(D));
        irw0=bufmat(6,isim); irw1=bufmat(7,isim);

        if irw1==length(rwtable(irw0).wave.longitude),break,end % exit if 'end of
wave'

        % STRATEGY 1 - PROPORTIONAL CHANGE AS MOST SIMILAR, WITH EXPONENT
        % changes are made by looking at relative change of most similar wave
        % but with a damping exponent 'de'
        de=0.5;
        newamp=Aamp * (rwtable(irw0).wave.amplitude(irw1+1) / ...
            rwtable(irw0).wave.amplitude(irw1)).^de;
        newslo=Aslo * (rwtable(irw0).wave.slope(irw1+1) / ...
            rwtable(irw0).wave.slope(irw1)).^de;
        %newitc=Aitc-rwtable(irw0).wave.slope(irw1);
        newitc=Aitc-newslo;
        newwdt=Awdt * (rwtable(irw0).wave.width(irw1+1) / ...
            rwtable(irw0).wave.width(irw1)).^de;
        newlon=Alon-1;

%           % STRATEGY 2 - GRADUAL ATTRACTION
%           % new strategy: gradual attraction by most similar wave
%           de=0.5;
%           newamp=(1-de)*Aamp + de*sign(Aamp)*abs(rwtable(irw0).wave.amplitude(irw1+1))
;
%           newslo=(1-de)*Aslo + de*rwtable(irw0).wave.slope(irw1+1);
%           %newitc=Aitc-rwtable(irw0).wave.slope(irw1);
%           newitc=Aitc-newslo;
%           newwdt=(1-de)*Awdt + de*rwtable(irw0).wave.width(irw1+1);
%           newlon=Alon-1;
%
        % propagate wave
        rwtfore(ip).wave.amplitude(end+1)=newamp;
        rwtfore(ip).wave.slope(end+1)=newslo;
        rwtfore(ip).wave.intercept(end+1)=newitc;
        rwtfore(ip).wave.width(end+1)=newwdt;
        rwtfore(ip).wave.longitude(end+1)=newlon;

        % update and iterate
        Aamp=newamp;
        Aslo=newslo;
        Aitc=newitc;
        Awdt=newwdt;
        Alon=newlon;

    end

end

return
```