NRL/MR/7320--16-9705

Recent Updates to SWANFAR®, a 5DVAR Data Assimilation System for SWAN

Mark Orzech
Jayaram Veeramony
Hans Ngodock

Ocean Dynamics and Prediction Branch
Oceanography Division

Stylianos Flampouris

IM Systems Group
College Park, Maryland

Innocent Souopgui

University of Southern Mississippi
Stennis Space Center, MS

November 10, 2016

REPORT DOCUMENTATION PAGE

1. REPORT DATE (DD-MM-YYYY) 10-11-2016	2. REPORT TYPE Memorandum Report	3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE

Recent Updates to SWANFAR®, a 5DVAR Data Assimilation System for SWAN

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
06011435N

6. AUTHOR(S)

Mark Orzech, Jayaram Veeramony, Hans Ngodock, Stylianos Flampouris,[1] and Innocent Souopgui[2]

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
73-4950-06-5

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Oceanography Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/7320--16-9705

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
One Liberty Center
875 North Randolph Street, Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

[1]IM Systems Group, College Park, MD
[2]University of Southern Mississippi, Stennis Space Center, MS

14. ABSTRACT

SWANFAR®, a five-dimensional variational (5DVar) data assimilation system, was built from the architecture of the spectral wave model SWAN. Initially designed to solve only the stationary, homogeneous wave action equation, the system has been updated to include adjoint and tangent-linear subroutines for nonlinear source/sink components of SWAN version 40.81, to perform nonstationary assimilations, and to apply covariance multipliers. The initial development of the stationary system, excluding source and sink terms, is described in Orzech et al. (2013). In the present report, we first discuss the building and testing of the nonstationary system with source and sink terms, detailing the status of all related subroutines. We then briefly summarize the implementation of radial, spectral, and temporal covariance expressions, which are more fully detailed in Veeramony et al. (2016). Results from twin experiments and field tests of the system at different stages are presented, followed by discussion of the advantages and limitations of spectral data assimilation built upon the SWAN architecture. Guidelines for performing an assimilation with SWANFAR®, along with samples of relevant scripts and files, are provided in several appendices.

15. SUBJECT TERMS

Data assimilation	Wave spectra
5DVar	Covariance
SWAN	SWANFAR®

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Mark Orzech
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	40	19b. TELEPHONE NUMBER (include area code) (228) 688-5974

Contents

List of Figures

1. INTRODUCTION

This report describes recent modifications and upgrades to SWANFAR®, a data assimilation system based on the SWAN spectral wave model (Booij et al., 1999).[1] The fundamental wave action equation solved by SWAN is generally espressed as

$$\frac{\partial N}{\partial t} + \vec{\nabla}((\vec{c}_g + \vec{U})N) = \sum_i \frac{S_i}{\sigma} \tag{1}$$

where N is wave action (i.e., wave energy normalized by standard deviation σ), t is time, $\vec{c}_g$ is wave group velocity, and $\vec{U}$ is mean ocean current velocity. Each of the source and sink terms S_i is also normalized by σ to make units consistent. For the assimilation system described herein, the source terms on the right side of (1) are expanded as

$$\sum_i \frac{S_i}{\sigma} = \frac{1}{\sigma}(S_{br} + S_{tr} + S_{wi} + S_{qu} + S_{wc} + S_{bo}) \tag{2}$$

The expanded source/sink terms in (2) represent depth-limited wave breaking, triad interactions, wind forcing, quadruplet interactions, whitecapping, and bottom stress, respectively. Only source and sink terms listed in (2) are included in this assimilation system. Additional terms incorporated into more recent versions of SWAN, such as ice and mud dissipation, have not yet been incorporated into SWANFAR®.

Historically, the assimilation of surface wave data was used to improve estimates of large-scale ocean models such as WAM (WAMDI, 1998; Komen et al., 1994). Early assimilation methods primarily made use of simple wave statistics like significant wave height and mean period, extracted from selected portions of the observed frequency-directional spectra. The statistics were used in weighted optimal interpolation schemes that nudged the model results closer to each measured data value (e.g., Hasselmann et al., 1997; Voorips et al., 1997; Aouf et al., 2006). These simpler schemes did not account for the stochastic nature of the wave environment, in which wave energy spectra are five-dimensional functions of space (i.e., x, y), time, frequency, and direction.

In a more sophisticated approach, Walker (2006) developed a variational data assimilation system for two-dimensional spectra of surface waves, creating a limited adjoint to the SWAN model. The system implements a quasi-strong-constraint approach, in which the adjoint to a stationary homogeneous form of (1) (i.e., with the first term and right hand side equal to zero) is used to project spectral differences between model and data outward to domain boundaries. An average of the projected spectra along the domain's offshore boundary is used to correct the boundary condition input to the (fully nonlinear) forward SWAN model, and an iterative conjugate gradient process is used to minimize the cost function. The partially linearized system was tested with datasets from Duck, NC, by Walker (2006) and later by Veeramony et al. (2010). Both studies demonstrated that, for interior regions away from lateral boundaries, the assimilation produced significant improvement in SWAN estimates of observed wave statistics and good agreement between measured and post-assimilation wave spectra.

The SWANFAR® spectral wave data assimilation system described here is based on the variational approach of Walker (2006) and Veeramony et al. (2010) and built from SWAN version 40.81. It is still a quasi-strong constraint system, controlled by the wave action values at the boundary and interior. Unlike the earlier efforts, however, the linearized adjoint is paired with a linear form of the forward model in order to achieve consistency between forward and adjoint components. The upgraded system's capabilities are now also more extensive than those of the Walker system, including adjoints to most of SWAN's nonlinear source and sink terms as well as nonstationary assimilation. To improve efficiency and accuracy, SWANFAR® has been configured to work with **perturbations** of wave action and related quantities (i.e., model-data error values only) rather than with "full-size" tangent-linear model estimates of actual measured wave quantities. In the linearized forward model, these perturbation arrays and the subroutines that

---

[1] The name "SWANFAR" and associated symbol were awarded a registered trademark on 15 September 2015.

operate on them have been relabeled as "representer-perturbation" (RP). In the linear adjoint, they retain the label ADJ. Post-assimilation spectral estimates are obtained by adding the RP perturbation spectra to the background spectra (normally pre-calculated by nonlinear SWAN). The SWANFAR® assimilation system operates in the two spatial, two spectral, and one time dimensions listed above and is thus described as 5D variational, or 5DVAR.

The first stage of development of this system, including construction and testing of adjoint and tangent linear routines for the stationary homogeneous wave action equation (i.e., with no sources, sinks, or time dependence, similar to Walker, 2006) is described in Orzech et al. (2013). After summarizing this earlier work, the present report details the implementation and testing of components for nonlinear source and sink terms and nonstationary assimilation in SWANFAR®. It also briefly discusses the development and initial basic tests of covariance multipliers for the system (these are more fully detailed in Veeramony et al., 2016). Additional tests for nonstationary scenarios, time covariance, and forecasting will be performed in FY17 as part of a Verification and Validation (V&V) study funded by the Naval Oceanographic Office; results of that study will be provided in a separate report in late 2017.

The following section describes the creation and validation of system components, including linearizations and approximations used in creating individual adjoint and RP subroutines, validations based on the representer method, and implementation of covariances. It concludes with a brief description of input and output files associated with the new system. Section 3 first presents comparative results from twin-experiment tests of the SWANFAR® system as source/sink terms are added. The system is then applied to stationary and nonstationary scenarios at two field sites, and finally a semi-idealized nonstationary case is provided to demonstrate the effects of covariance multipliers. Discussion and conclusions are offered in Section 4, including some comments on the limitations of data assimilation based on the SWAN wave model. Appendices A - F offer guidance for those wishing to conduct an assimilation with SWANFAR®, accompanied by samples of relevant scripts and files.

2. SYSTEM DEVELOPMENT

As mentioned above, the SWANFAR® system was initially developed in a linear, stationary format, with adjoint and RP components created only for the parts of the code associated with stationary wave advection and dispersion (Orzech et al., 2013). A strong constraint approach was used, under which only the boundary conditions were controlled. Orzech et al. (2013) conducted twin experiment validations of the linear, homogeneous SWANFAR® system using data from sites along the U.S. Gulf Coast and near Duck, NC. While results compared well to those from the original system developed by Walker (2006), the absence of nonlinear source and sink terms in the adjoint still contributed to model-data errors in system estimates of significant wave height and directional spread at Duck, particularly for cases with significant wave height (H_s) greater than 1 m. The system performed worse when assimilating spectra from several different instrument types (e.g., AWACs, Datawell Buoys, and the 8-m pressure array at Duck, NC), but it did significantly better when all assimilated spectra were all from the same instrument type or were artificially generated by a modified SWAN run.

2.1. Overall Development Status of SWANFAR®

RP and adjoint subroutines have now been developed and validated for nearly all of the components in SWAN v.40.81 that play a role in solving (1) and (2), along with numerous routines performing other functions.[2] As described in Orzech et al. (2013), these routines were primarily developed using the parametric FORTRAN compiler (PFC) utility (Erwig et al., 2007), results from which were modified by the

---

[2]A large number of routines did not require adjoints because they were not associated with solving the wave action equation. A list of SWAN routines that remain unchanged in SWANFAR is provided in Appendix F.

authors to improve efficiency and remove errors. Subroutines for which the adjoint and RP were not fully completed and/or tested include:

(*i*) Several source/sink routines with more complicated numerics (e.g., SWFLXD);

(*ii*) Two matrix solvers (SOLMT1, SWSIP) for conditions with currents, for which a different adjointing technique was used that has not yet been fully validated;

(*iii*) Several routines that cannot be adjointed due to arbitrary value changes they impose upon wave action or other parameters (e.g., PHILIM, HJLIM);

(*iv*) All routines relating to unstructured grids.

The status of all SWAN subroutines that have been modified for SWANFAR® is summarized in Table 1. For a number of "administrative" subroutines (e.g., SWREAD, SWOUTP, BCFILE, SWOEXD), special "ADJ" and/or "RP" versions were created to handle adjoint and/or RP arrays; they are not themselves adjoint/RP versions of the original subroutines. As part of a separate project to couple SWANFAR® with 4DVAR NCOM, additional adjoint and RP subroutines were created for parts of SWAN that work with current velocities (e.g., SPROXY, SPROSD, DSPHER), in order to allow the system to exchange adjoint and RP velocity arrays with the coupled model. These will be described in a future journal article and will not be addressed further in this report. Tests presented in Section 3 will demonstrate that the newly incorporated assimilation features significantly improve system performance in comparison to the linear homogeneous SWANFAR® described above.

2.2. Validations of Individual and Grouped Subroutines

Validation of the consistency of each pair of adjoint-RP subroutines relies on basic tensor identities to construct a partial representer matrix (Bennett, 2002). The action of each linearized RP subroutine upon the active variables is expressed as $Au = v$, in which the vector u of active variables is acted upon by the subroutine tensor A to generate vector output v. The adjoint subroutine is then written as $A^T v = u$, in which A^T is the transpose of tensor A. The inner product identity $< Au, v > \equiv < u, A^T v >$ is evaluated by using multiple initializing values of u to generate the representer matrix. This matrix is symmetric for properly constructed RP and adjoint subroutines. For further discussion, see Orzech et al. (2013).

Following their development, each pair of adjoint/RP subroutines was validated using this symmetry test. In each validation, the adjoint subroutine was initialized with a unit impulse (i.e., one spectral bin at one location and time was set to 1.0 while all other bins in the domain were set to zero). The RP subroutine was initialized with adjoint output at all domain locations, and RP output was then stored. The test was repeated three times using different bins, locations, and times. A 4×4 representer matrix was constructed by recording the output of the RP at each of the four test locations/times. When the matrix was found to be symmetric within machine accuracy ($O(10^{-12})$), the adjoint/RP pair was determined to be consistent. Similar validations were completed for selected groups of subroutines (e.g., all subroutines called by SOURCE) in both stationary and nonstationary formats. Representer matrix symmetry tended to be of lower order ($O(10^{-6})$) for nonstationary tests.

2.3. Covariance Implementation

A separate and extensive analysis has been conducted to develop estimates for the ten-dimensional covariance for the SWANFAR® system (Veeramony et al., 2016). In the analysis, it is assumed that covariance is independent and separable into each of the five dimensions. Additionally, analysis results indicated that covariance in space is actually a function of three dimensions: x, y, and, for shallower coastal regions, the (known) water depth $d(x, y)$. The resulting expression for overall covariance is given by Veeramony et al. (2016) as

$$C_{NN'}(x, y, \theta, f, t, x', y', \theta', f', t') = C_d[C_{xx'} + C_{yy'}]C_{\theta\theta'}C_{ff'}C_{tt'} \qquad (3)$$

Table 1: Development/Validation Status of SWAN Routines

Subroutine(s)	Adj/RP Complete?	Validation Done?	Notes
SWCOMP, SWOMPU, SOURCE, ACTION, SWREAD, SWOUTP	Y[a]	Y	Higher-level control routines
STRSXY, SORDUP, SANDL, STRSSI, STRSSB, STRSD	Y	Y	Advection/Diffusion
SWFLXD	Y	N	Flux-limiting advec/diffus.
SOLMAT	Y	Y	Primary solver (implicit, no currents)
SOLMT1, SWSIP	Y	N	Additional solvers (currents)
INIT, CGINIT, SNEXTI, SOLPRE	Y	Y	Setup/Intermediate
RESCALE, PHILIM, HJLIM	N	N	NOT USED (not adjointable)
SWLTA, SSURF, SWCAP	Y	Y	Sources/sinks (triads, breaking)
WNDPAR, WINDP3, SWIND3, SWIND4, SWIND5	Y	Y	Sources/sinks (wind)
SWSNL1, SWSNL2, SWSNL3, SWSNL4, SWSNL8, FILNL3	Y	Y	Sources/sinks (quadruplets)
SBOT	N	N	Bottom stress (adjointing not required)
SPROXY, SPROSD	Y	Y	Propagation vels in X, Y, f, θ[b]
DSPHER	Y	Y	Propagation vels in θ, spherical[b]
SWSENDAC, SWRECVAC	Y	Y	ADJ only; send/receive adjoint wave action in parallel.[c]
BCFILE	Y	N	ADJ only; innovation input
SWSPEC, WRSPEC, SWORDC, SWREOQ, SWREPS, SPROUT, FOR	Y	N	Various i/o[c]
SWOEXD	Y	N	Calc quantities for output[d]
SWCLME, MSGERR	Y	N	Admin routines; ADJ/RP versions created for more convenient records/troubleshooting.

[a]Not actual adjoints. These primarily have order of calls reversed and some adjoint/RP-specific modifications.
[b]Adjoint current velocities; needed for coupling with NCOM.
[c]Not actual adjoints. ADJ/RP routines modified to work with ADJ/RP quantities, respectively.
[d]Not actual adjoint. ADJ_SWOEXD just computes adjoints to current velocities for adjoint SWAN.

where $C_{NN'}$ is the covariance of the wave action density $N \equiv N(x, y, \theta, f, t)$, C_d is covariance with depth, and each term $C_{ii'}$ on the right-hand side of (3) represents covariance in dimension i. For each of the five dimensions, extensive datasets of buoy-measured and SWAN-generated wave spectra were processed to determine their covariance matrices and identify dependencies. An empirical function was developed from each covariance matrix to represent error covariance in each corresponding dimension. The resultant covariance functions for d, x and y, θ, f, and t are

$$C_d = 10^{\left[-1.54exp\left(-\frac{d+1.25}{5.33}\right)^2\right]} \tag{4}$$

$$C_{xx'} + C_{yy'} = exp\left(-\frac{|x-x'|}{L_x}\right)^{\beta_x} + exp\left(-\frac{|y-y'|}{L_y}\right)^{\beta_y} \tag{5}$$

$$C_{\theta\theta'} = e^{-\left[\left(\frac{\theta-\theta_p}{2\sigma_\theta}\right)^2+\left(\frac{\theta'-\theta'_p}{2\sigma_{\theta'}}\right)^2\right]} \tag{6}$$

$$C_{ff'} = a_f e^{-\left[b_f(f-f_p)^2+2c_f(f-f_p)(f'-f'_p)+d_f(f'-f'_p)^2\right]} \tag{7}$$

$$C_{tt'} = exp\left(-\left[\frac{t-t'}{\tau}\right]^2\right) \tag{8}$$

where $[L_x, L_y]$ are spatial correlation lengths, $[\beta_x, \beta_y]$ are empirical parameters for spatial covariance, σ is standard deviation, subscript p indicates the spectral peak value, τ is a temporal correlation length ($\approx 30 - 96$ hr, depending on conditions), and a_f, b_f, c_f, and d_f are fitted empirical coefficients for frequency covariance.

The above covariance functions were implemented within a module that runs concurrently with but separately from the SWANFAR® code.[3] A configuration file accompanying the module allows the user to turn "on" or "off" the application of covariance in each dimension and modify their correlation lengths (see Section 5.2). As noted by Veeramony et al. (2016), assumptions inherent in the development of this covariance function and the supporting analysis have resulted in an imperfect yet ultimately beneficial tool that demonstrably improves the performance of the SWANFAR® data assimilation system (see Section 3.3). However, extensive additional analysis is required to fully validate and fine tune these functions, involving a systematic and consistent examination of much larger global wave datasets and many more model simulations at global, regional, and local scales. Covariance multiplication in the time dimension is not presently functional in SWANFAR®, but this is being addressed in the NAVO V&V project mentioned earlier and should be resolved by late FY17.

3. TESTING AND APPLICATION

A range of tests were conducted to evaluate the performance of SWANFAR® under stationary and nonstationary conditions. For each test, the following skill score was computed to evaluate the accuracy of pre- and post-assimilation spectra:

---

[3]In the implementation, the C_d, $C_{xx'}$, and $C_{yy'}$ multipliers are combined into a single radial covariance multiplier, $C_{rr'}$, for which users may specify a radial correlation length.
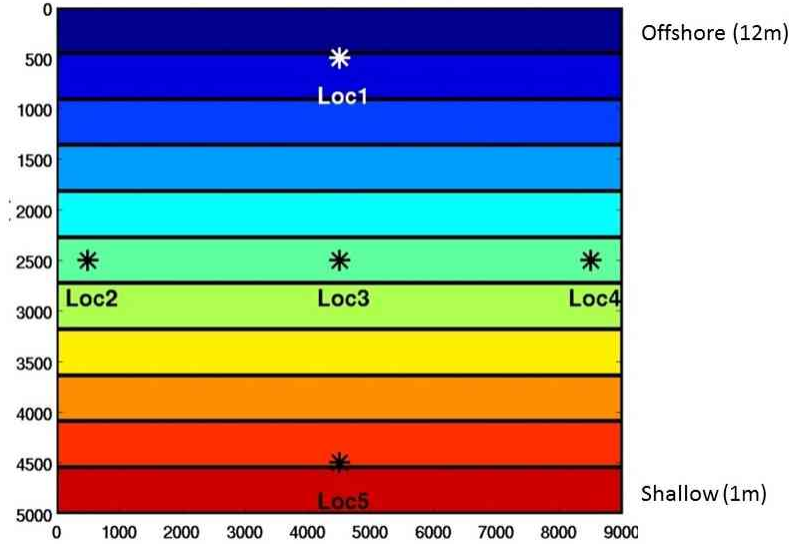
Figure 1: Idealized planar beach with five assimilation locations

$$skill = 1 - \frac{\sqrt{\sum_{i,j}(E_{mod}(f_i, \theta_j) - E_{obs}(f_i, \theta_j))^2}}{\sqrt{\sum_{i,j}(E_{obs}(f_i, \theta_j))^2}} \tag{9}$$

In (9), E_{mod} and E_{obs} are modeled and observed spectral energy densities respectively for discrete frequencies f_i and directions θ_j. A perfect match between modeled and observed spectra produces a skill value of 1, while poor results have skill values less than or equal to zero.

3.1. Stationary Tests

3.1.1. Idealized Twin Experiment

A validation of the complete SWANFAR® system in stationary mode (excluding covariances) was conducted by running a twin experiment for five locations on an idealized planar beach (Figure 1). Both background and pseudo-observations were generated by fully nonlinear SWAN (using slightly different boundary conditions). Only selected sources or sinks were activated in SWANFAR® in order to investigate the relative importance of different source and sink terms to the assimilation results.

In general, SWANFAR®'s assimilation performance for the planar beach was significantly improved by inclusion of nonlinear source/sink terms, when compared to its previous linear homogeneous version. Figure 2 displays model skill scores and scatter plots of basic statistics at each of the five assimilation locations for the homogeneous system (without accounting for sources and sinks). Figure 3 shows the same results when all source and sink terms from (2) are included in the SWANFAR® assimilation. Spectra were assimilated from all five locations. Mean skill score for the assimilation improved from -0.98 with the homogeneous system to +0.80 with sources and sinks included. The majority of this improvement was determined to come from inclusion of depth-limited wave breaking (S_{br} in (2)); with only this sink term included, mean skill at the five locations was +0.70.

Model skill at the shallow water surfzone location (Loc5 in Figure 1) was consistently lower than at other locations. Additionally, when spectra were only assimilated from this location, overall model skill dropped significantly (to a mean of 0.05), despite running the system with all source and sink terms activated. In

contrast, when spectra were only assimilated from mid-domain (Loc3), mean skill remained high at 0.79. This result stems from an arbitrary treatment of wave breaking in SWAN that cannot be adjointed; it is discussed further in Section 4.

3.1.2. Trident Warrior Tethered Buoys

For the 2013 Trident Warrior field experiment near Norfolk, VA, five tethered mini-buoys were deployed by a team from Scripps Institution of Oceanography (Terrill, 2013). The buoys used GPS to record their positional time series at roughly fixed locations offshore of Norfolk, VA, from July 8 - 19, 2013. To generate background spectra for the assimilation, the nonlinear SWAN model was run in stationary mode with offshore and lateral boundary spectra generated by WAVEWATCH III® (Tolman, 2014) for a 100×80 km^2 domain (Figure 4).

The SWANFAR® system assimilated data simultaneously from all five minibuoy locations. All nonlinear source and sink terms from (2) were activated, but spatial and spectral covariance multipliers were not applied. Post-assimilation spectra had consistently more accurate significant wave heights, lower RMS error, and higher skill at all assimilated locations than was obtained for the background estimates from nonlinear SWAN. Sample results are presented in Figure 5. Post-assimilation skill scores obtained by SWANFAR® were generally positive and consistently better than those obtained with SWAN background spectra at all fixed buoy locations during this experiment. However, skill values with the measured data were usually 0.5 or less, and performance never approached the more accurate results obtained under the idealized planar beach scenario described in Section 3.1.1 above.

3.2. Nonstationary Tests

3.2.1. Trident Warrior Free-Floating Buoy

In Trident Warrior '13, an additional free-floating mini-buoy was deployed on July 19 and allowed to drift with the current for roughly one day. Spectra returned by this buoy were provided to the SWANFAR® system at multiple times and locations. Although the system is generally designed for nonstationary assimilations from *fixed* locations, in this case SWANFAR® was temporarily reconfigured to assimilate data from the *moving* buoy for 15 hours on July 19. The buoy motion was primarily limited to the northwest corner of the model domain (Figure 6).

Model performance under this more challenging setup was relatively poor, but the SWANFAR® system nevertheless managed to correctly shift estimated spectra toward the observations from the moving buoy. While post-assimilation spectral shapes were little changed from those of the forward estimate, the corrected wave heights were consistently shifted closer to the wave heights observed at the buoy (e.g., Figure 7). Post-assimilation skill scores consistently improved, but they generally still remained negative. Results suggest that the SWANFAR® system is better limited to nonstationary assimilations from *fixed* locations.

3.2.2. Duck, NC

In a more typical nonstationary test, wave spectra were next assimilated at three-hour intervals from three fixed nearshore instruments at Duck, NC, over a one-week period from August 20-27, 2011. To provide background estimates, SWAN was initialized at the offshore boundary of a 6 km × 2.5 km domain using spectra from the Field Research Facility's 26 m Datawell Waverider buoy. Post-assimilation corrected results were compared with spectra from the 17 m Waverider buoy, the 11 m Acoustic Wave and Current (AWAC) sensor, and the FRF's 8 m array (Figure 8). The cost function was minimized in SWANFAR® for the entire one-week period.

Post-assimilation wave statistics (H_s, T_m, and D_m) were closer to observed values for nearly all times during the one-week period at all three instrument locations (Figures 9 - 11). Mean skill scores, which ranged
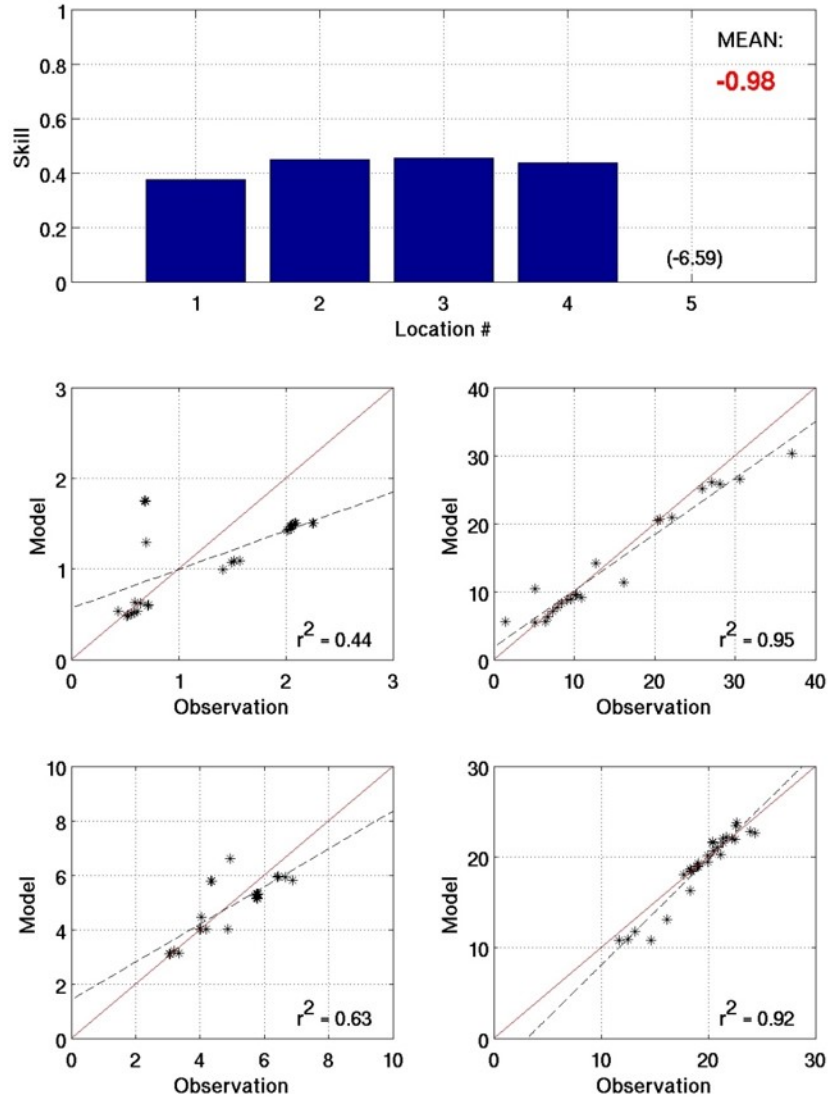
Figure 2: Skill scores and statistics for planar beach with NO source/sink terms activated in SWANFAR®
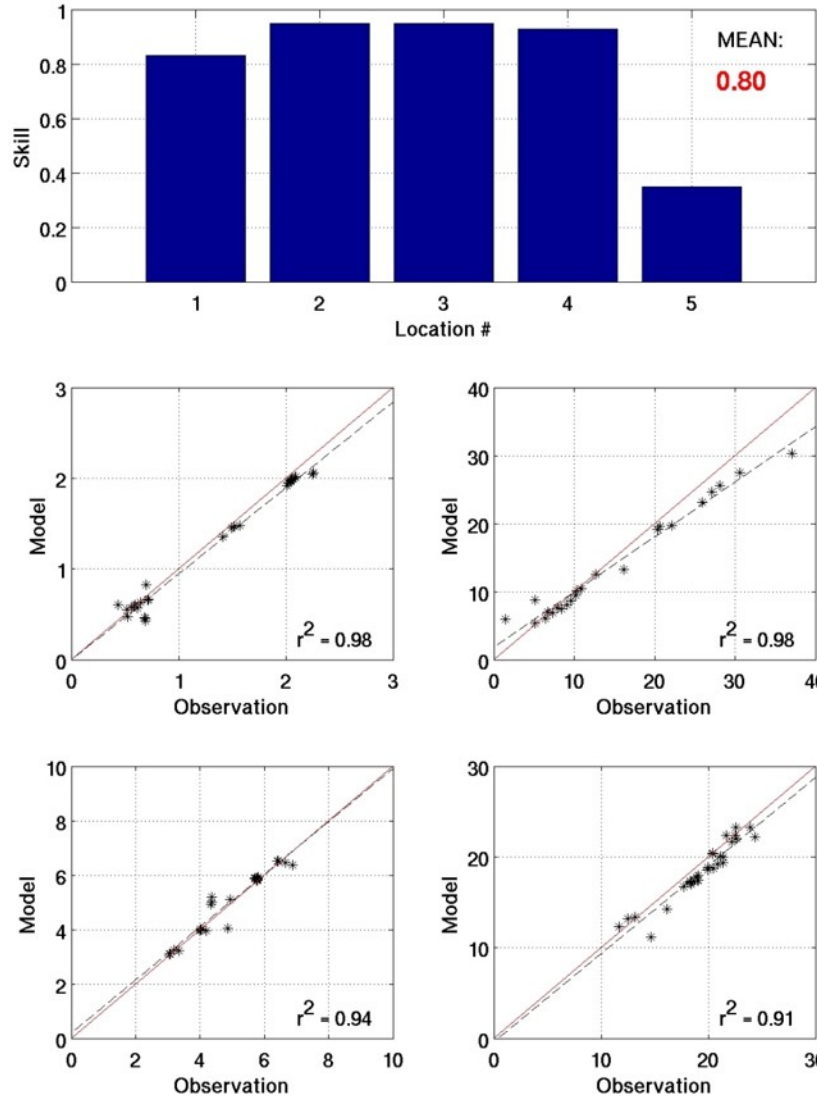
Figure 3: Skill scores and statistics for planar beach with ALL source/sink terms activated in SWANFAR®

Figure 4: Domain boundaries for SWANFAR® stationary assimilation. Approximate locations of Scripps mini-buoys are marked as yellow circles.
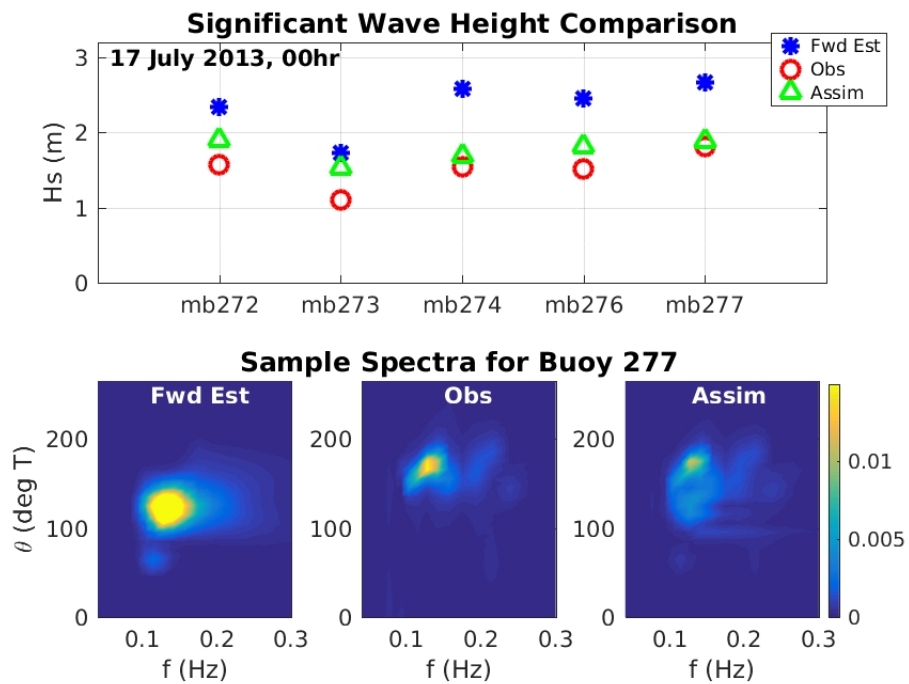
Figure 5: Sample results for stationary assimilation of spectra from five Scripps mini-buoys at Trident Warrior '13. Top panel: significant wave heights at each buoy location at midnight on 17 July. Bottom panels: estimated, observed, and post-assimilation spectra for buoy 277 at that time. For the spectra, skill of forward estimate is -1.4, while post-assimilation skill is +0.53.

Figure 6: Assimilation locations of Scripps mini-buoy #275 (yellow circles) in model domain, 17 July 2013. Buoy began at 0200 hr at initial (easternmost) position, then moved west and north to final (westernmost) position at 1700 hr.
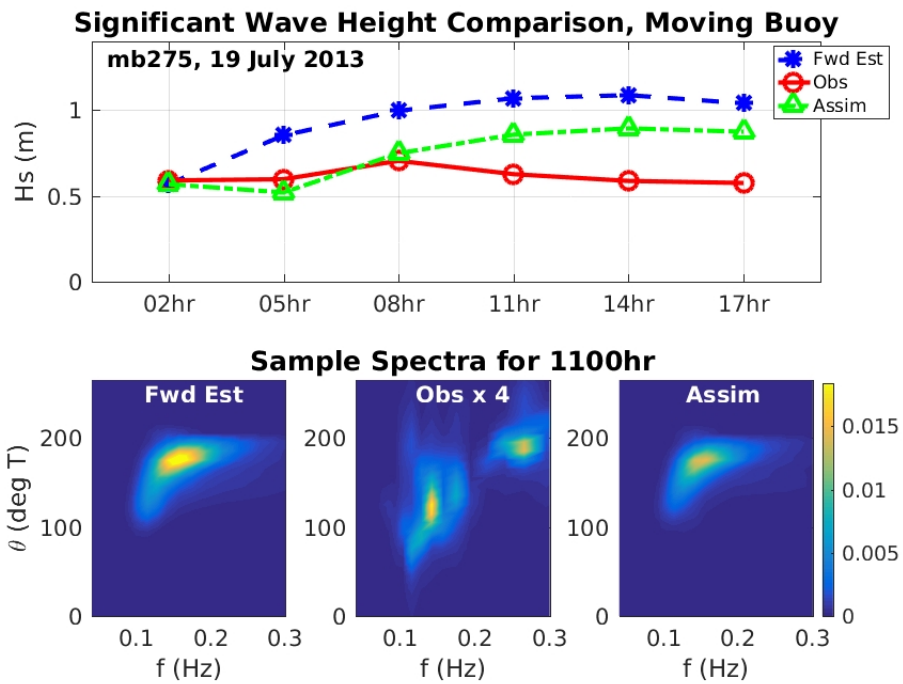
Figure 7: Sample results for nonstationary assimilation of free-floating mini-buoy at multiple locations and times during Trident Warrior '13. Top panel: significant wave heights between 0200 - 1700 hr on 17 July. Times correspond to locations shown as yellow circles in Figure 6. Bottom panels: estimated, observed (rescaled), and post-assimilation spectra for buoy 275 at 1100 hr. For the spectra shown, skill of forward estimate is -3.1, while post-assimilation skill is -1.0.
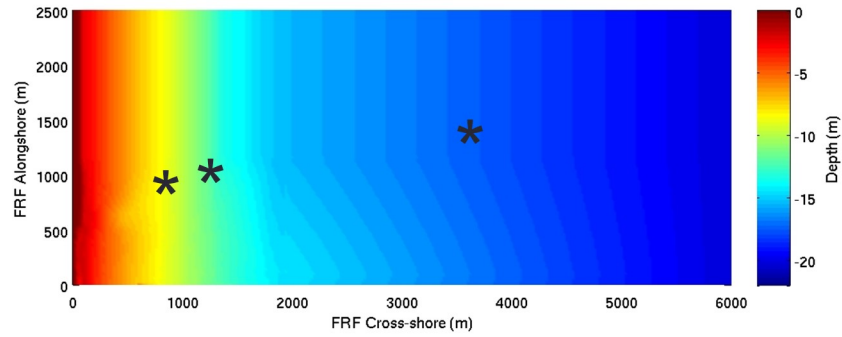
Figure 8: Domain for one-week non-stationary assimilation at Duck, NC. Color contours indicate bathymetry. From right to left, assimilation instrument locations marked by asterisks include Waverider buoy (17 m depth), AWAC (11 m), and FRF pressure array (8 m).
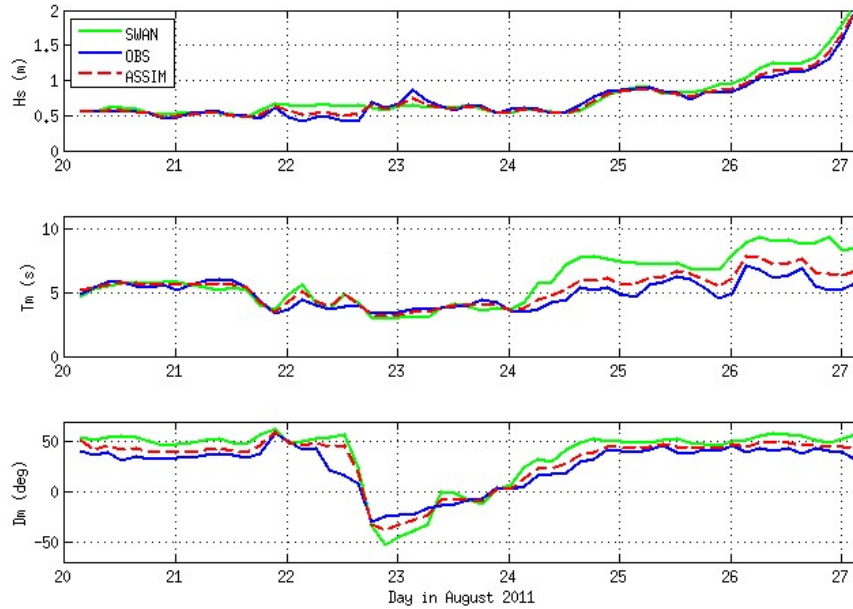
Figure 9: Wave statistics (H_s, T_m, and D_m) for one-week nonstationary assimilation at Duck, NC, Waverider buoy location (17 m). Results plotted for original forward SWAN estimates (solid green), instrument observations (solid blue), and SWANFAR® assimilation results (red dashed).

12
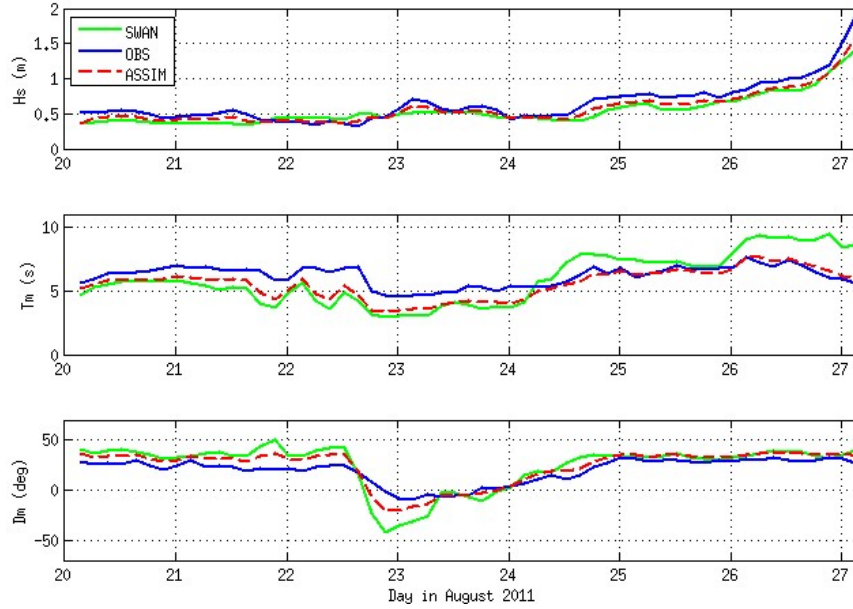
Figure 10: Wave statistics (H_s, T_m, and D_m) for one-week nonstationary assimilation at Duck, NC, AWAC location (11 m). Same format as Figure 9.
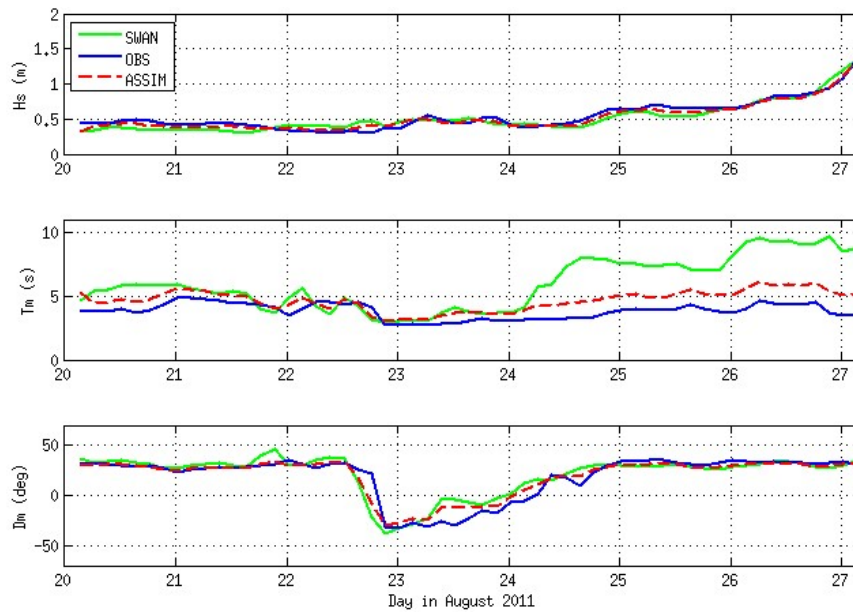
Figure 11: Wave statistics (H_s, T_m, and D_m) for one-week nonstationary assimilation at Duck, NC, FRF pressure array location (8 m). Same format as Figure 9.
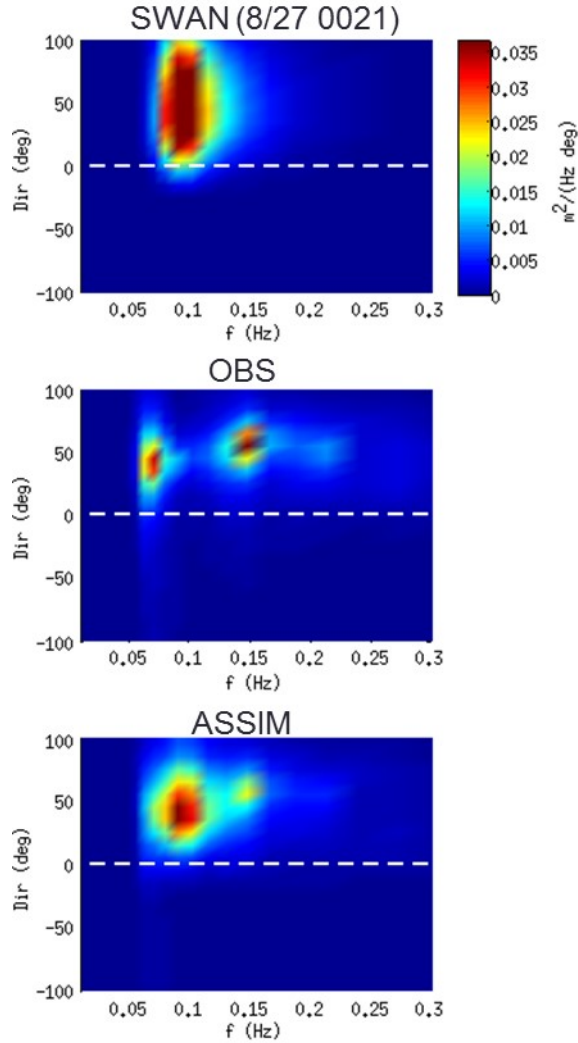
Figure 12: Sample of estimated, observed, and post-assimilation spectra from one-week nonstationary assimilation at Duck, NC. These results are from 0021 hr on 27 August 2011, at the FRF 8m array location. Post-assimilation spectrum displays some of the bimodality of the observed spectrum, which is not seen in the original estimated spectrum.

from -0.23 to -0.14 for original estimated spectra, were in the range of +0.33 to +0.38 for assimilated spectra. As suggested by the skill score results, the frequency-directional distributions of assimilated spectra also shifted to more closely resemble observed spectra. An example of this shift is presented in Figure 12, in which the post-assimilation estimated spectrum at the FRF 8m array (bottom panel) has incorporated some of the bimodality of the observed spectrum (middle panel), in contrast to the unimodal original estimated spectrum (top panel).

3.3. Nonstationary Assimilation with Covariances

As a preliminary evaluation of the effectiveness of spatial and spectral covariances in SWANFAR®, an idealized test case was set up using the bathymetry and boundary conditions from a domain slightly to the north of the Trident Warrior '13 domain described in Sections 3.1.2 and 3.2.1. Original boundary spectra generated for 11 - 13 July 2013 by WAVEWATCH III® were used to initialize nonlinear forward SWAN for a three-day nonstationary simulation. Resulting output spectra from the forward model were saved at

14
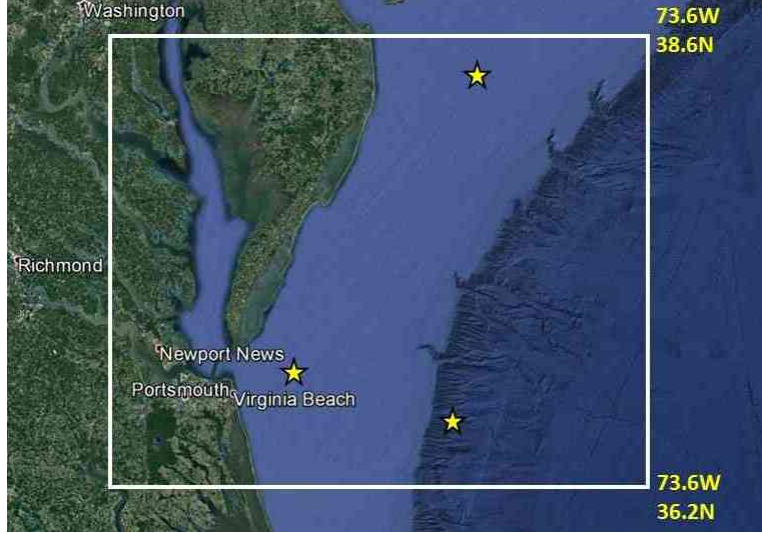
Figure 13: Map of domain used for idealized test case with covariance multipliers applied in SWANFAR®. Domain boundaries are indicated by white square. Observation locations are marked with yellow stars.

three "observation" locations (marked as stars in Figure 13). Spectra were also saved at sixteen locations surrounding each observation at distances ranging from $0.01° - 0.25°$, to enable evaluation of the effectiveness of covariance multipliers in SWANFAR®.

To perform the evaluation, the SWANFAR® system was initialized with new boundary spectra, created by randomly shifting the original WAVEWATCH boundary spectra by roughly ± 0.1 Hz on the frequency axis and $\pm 10°$ on the directional axis. Spectral magnitude was randomly modified by up to $\pm 20\%$. Two nonstationary assimilations were conducted for the three-day period using the "observed" spectra from the original SWAN simulation as input. Output background spectra generated by nonlinear SWAN and post-assimilation corrected spectra from SWANFAR® were saved at each observation location and its sixteen surrounding locations. In the first set of tests, covariances in space (i.e., radial $C_{rr'}$), frequency, and direction were all activated in SWANFAR®. Using default values recommended by Veeramony et al. (2016), spatial correlation length was set to 5×10^4 m, frequency correlation length was set to 0.02 Hz, and directional correlation length was set to $5.0°$. In a second set of tests, the same boundary conditions were used but all covariances were deactivated.

Test results are evaluated by comparing the skill scores achieved by forward SWAN (Sk_f), SWANFAR®-without-covariances (Sk_{woc}), and SWANFAR®-with-covariances (Sk_c) at output locations surrounding the original observed locations. These scores, summarized in Figure 14, demonstrate the effectiveness of SWANFAR® at improving model accuracy, particularly when covariances are applied. There is a gradual decrease in skill with increasing distance from observation locations, but Sk_c is consistently higher than Sk_{woc} and Sk_f. For neighboring locations roughly $0.01°$ from assimilated locations, the mean skill values for $Sk_f/Sk_{woc}/Sk_c$ were 0.72/0.74/0.79 over the three-day period. For neighboring locations roughly $0.10°$ away, respective mean skill values were 0.72/0.73/0.78. When distance was increased to $0.25°$ away from observation locations, respective skill values dropped slightly more to 0.71/0.72/0.76. Combining all 66 surrounding non-observation locations tested, the overall mean skill value for Sk_f remained relatively constant at 0.72. Mean skill for Sk_{woc} was 0.73, and mean skill for Sk_c was 0.78.

At observation locations, assimilation with covariances also achieves higher skill scores at observation locations than assimilation without covariances (Figure 15). For this more limited case, the mean skill for Sk_f, Sk_{woc}, and Sk_c were 0.72, 0.74, and 0.79, respectively.

Depending on how they are configured, the application of covariances can sometimes reduce overall model

15
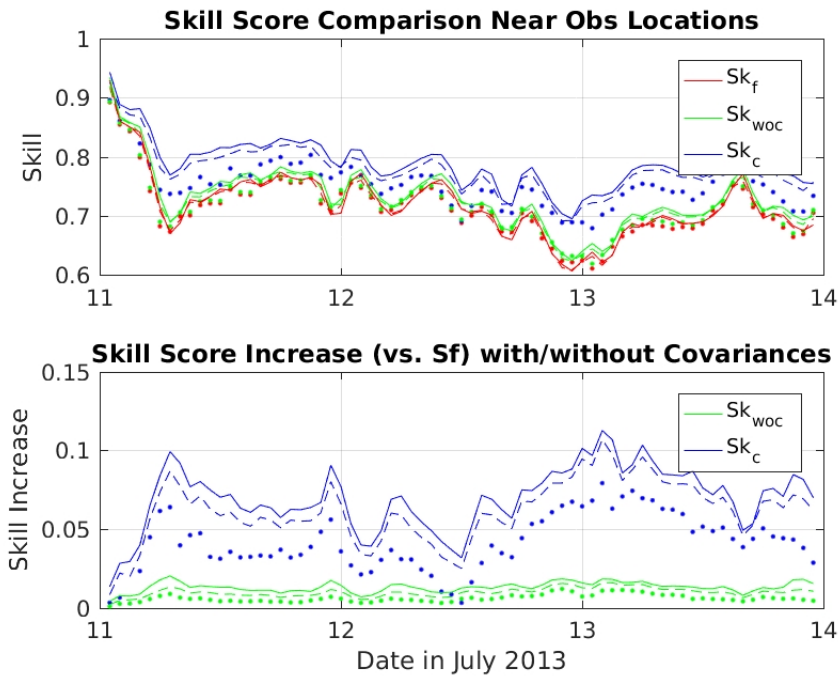
Figure 14: Skill score comparison for locations surrounding the three observation locations where spectra were assimilated. Colors show results for Sk_f (red), Sk_{woc} (green), and Sk_c (blue). Solid, dashed, and dotted lines show results for locations $0.01°$, $0.10°$, and $0.25°$ away from observation location, respectively.
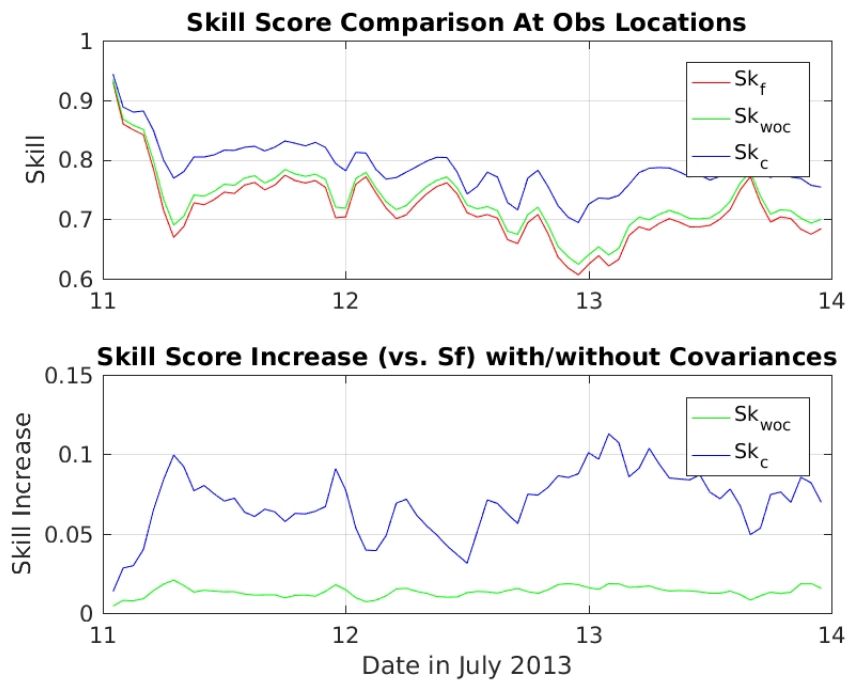
Figure 15: Skill score comparison for the three observation locations where spectra were assimilated. Colors show results for Sk_f (red), Sk_{woc} (green), and Sk_c (blue).

16

Skill Change (vs Sk$_f$) with Different Spectral Corr Lengths
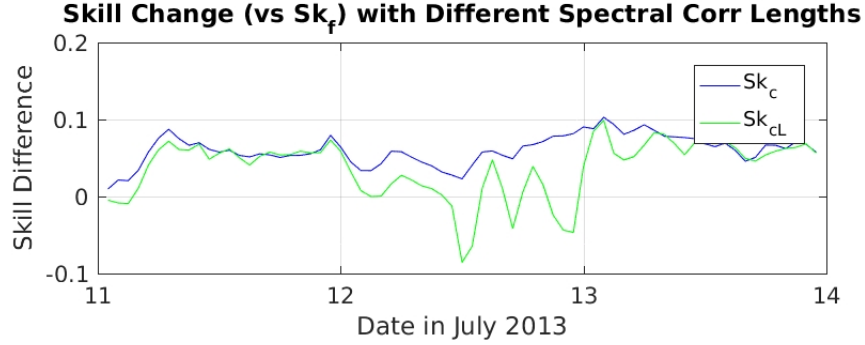
Figure 16: Skill score change (vs Sk_f) for different spectral correlation lengths, averaged over all locations. *Blue line: Sk_c with ORIGINAL settings [$\sigma_\theta = 5°$, $\sigma_f = 0.02$ Hz]. Green line: Sk_{cL} with LARGE settings [$\sigma_\theta = 20°$, $\sigma_f = 0.08$ Hz].*

accuracy, both at assimilated observation points and at nearby locations. In particular, when spectral covariance length values are larger than appropriate, the correction spectra can be excessively smoothed along the frequency and directional axes. This effectively reduces the resolution of the corrected spectrum, by spreading the effects of individual spectral bin corrections onto neighboring bins. To illustrate this, the above assimilation is rerun with significantly larger frequency and directional correlation lengths. For the revised test case, directional correlation length σ_θ is increased from its original (default) value of $5°$ to a significantly broader $20°$, and frequency correlation length σ_f is increased from 0.02 to 0.08 Hz. With the increased values, the SWANFAR® assimilation scores consistently lower than its original output and performs worse than forward SWAN for much of 12 July (Figure 16).

An examination of observed and post-assimilation spectra for this date indicates that, with larger correlation length values, SWANFAR® overcorrected spectral estimates for spectral bins in the adjoint surrounding the peaks. Figure 17 illustrates the additional spreading applied to adjoint boundary spectra with the use of larger correlation lengths. The broader correction spectra are passed from the adjoint to the RP model and ultimately produce broader corrections at observation points and other interior domain locations. By forcing the system to assume larger than appropriate uncertainties, the corrected spectrum in this case was reshaped to have an excessively broad peak and a poorer overall match to the observed spectrum. This result highlights the importance of the (somewhat subjective) assignment of correlation lengths to covariance multipliers in each dimension. Whenever possible, assignment of correlation lengths for a specific domain should be based on statistics (such as spatial and spectral variances) obtained from extensive analysis of wave data from that domain. As this is not always feasible, it is recommended that users generally stick with the default correlation length values provided in Appendix E (based on the analysis of Veeramony et al., 2016).

In addition to covariances, a general variance-based multiplier is available to adjust the magnitude of the adjoint correction spectra. This value is assigned by the user along with the correlation lengths in the **input.cov** file. Also somewhat subjectively determined, based loosely on the estimated variance of the spectral energy densities, this multiplier can help to reduce a general bias in the corrected spectra. It cannot fully correct for excessive frequency and directional spreading. For example, in the case illustrated by Figure 17, the broader adjoint spectrum in the lower panel also has increased in overall magnitude (especially the positive corrections). Although it would not have fully resolved the spectral mismatch, use of a somewhat smaller variance multiplier could have reduced overall error magnitude and may have been appropriate for this case. For further discussion, see Veeramony et al. (2016).

In the spatial dimensions, the covariance multiplier acts to spread the innovation energy in the adjoint output radially from every location. In the RP model, these modified innovations might be expected to produce radial spreading of the total energy correction in the vicinity of observation locations and, to an extent, they do (see Figure 18). However, because it is the adjoint results that are smoothed and then
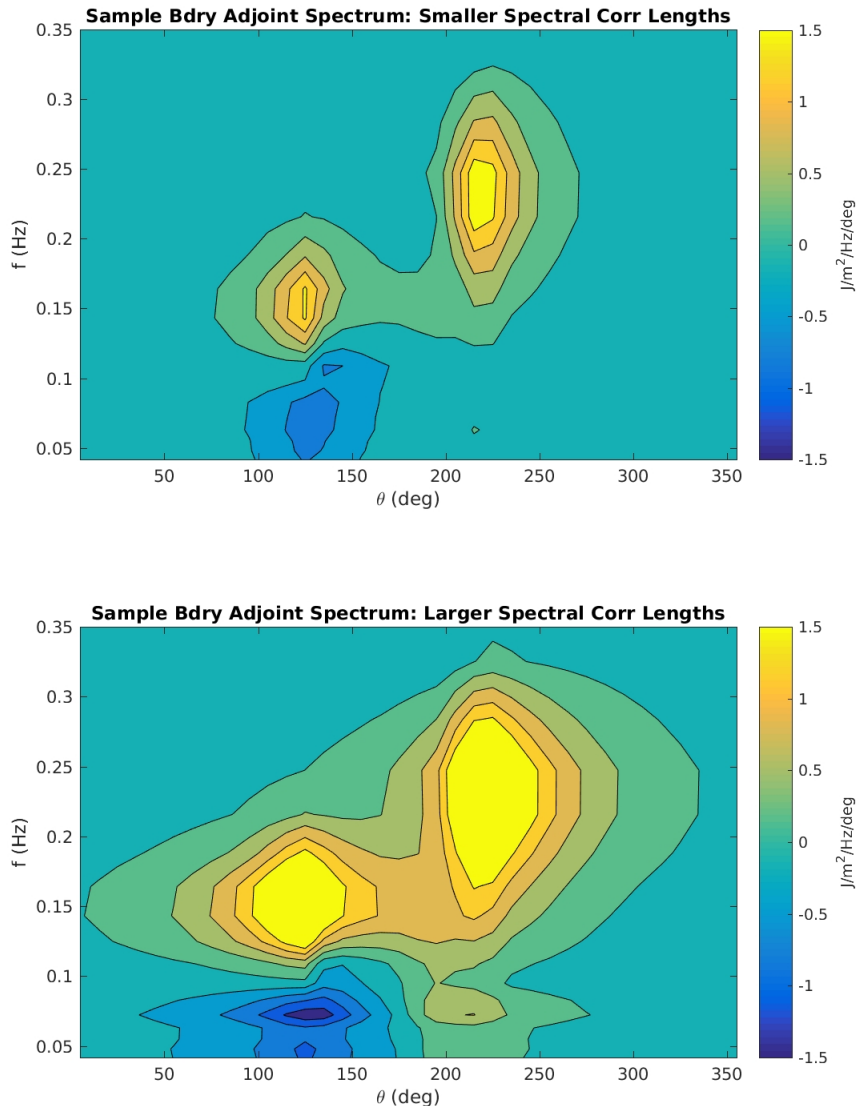
Figure 17: Sample adjoint spectra from a central location along offshore boundary (Lon: , Lat:) for 1200hr on 12 July 2013. *Top panel:* Spectrum from simulation with $(\sigma_\theta, \sigma_f) = (5°, 0.02Hz)$. *Bottom panel:* Spectrum from simulation with $(\sigma_\theta, \sigma_f) = (20°, 0.08Hz)$
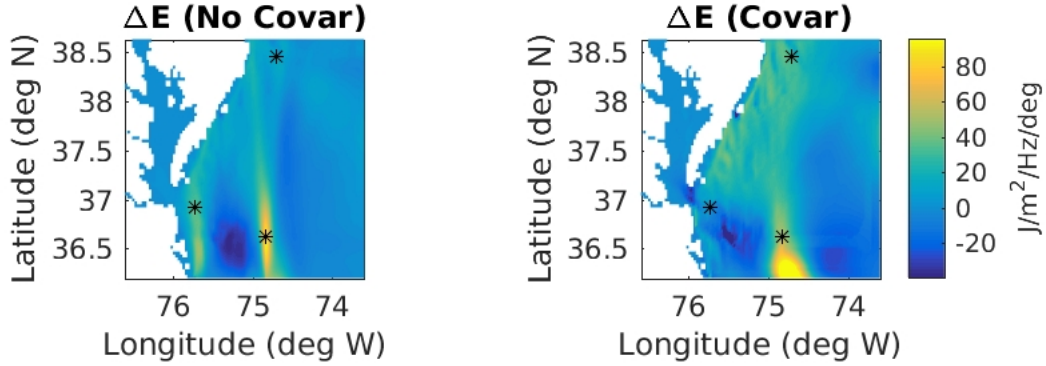
Figure 18: RP-generated total energy correction at all domain locations for domain shown in Figure 13. Left panel shows correction for 11 July 2013, 1200hr, without covariances applied. Right panel shows correction after spatial and spectral covariances were applied. Asterisks are observation locations.
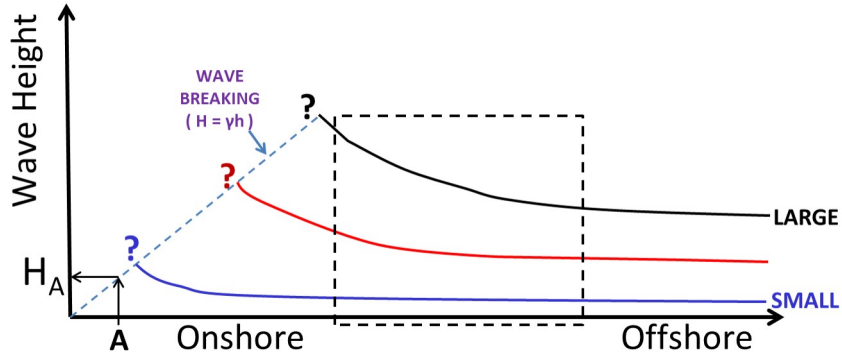
Figure 19: Limitations of wave assimilation inside surf zone. Blue, red, and black lines illustrate heights of small, medium, and large waves as they approach shore (from right to left). If wave heights H_A or spectra are assimilated at location A only, it is not possible for the assimilation system to determine whether they resulted from small, medium, or large offshore waves. Only assimilation from outside the surf zone (such as within the dashed box region) will accurately reflect offshore wave heights.

propagated through the RP model, the overall effect in the RP is more complex. The resulting spatial distribution of energy in the RP domain also depends on the magnitude of energy in specific directional bins of the smoothed adjoint spectra. As illustrated in the figure, energy in the RP model may be spread from an observation location toward a specific boundary (e.g., near 74.8W along S boundary), or the spatial spreading of negative corrections from one area (e.g., near 75.3W, 36.5N) may wash out a positive correction at an observation point (e.g., 75.8W, 37N).

4. DISCUSSION AND CONCLUSIONS

Efforts to assimilate wave spectra are gradually progressing, but numerous challenges remain. Some challenges relate to the design of SWAN and other spectral wave models and the manner in which they are configured to solve the general wave action equation (1). The efficiency of the SWANFAR® system is limited by the parallelized architecture of the SWAN model. SWAN's parallelization divides the physical model domain among available processors. This division is restricted to one dimension (i.e., only the longest of the two horizontal dimensions is partitioned). thus, for example, if a 10 km × 5 km domain is

utilized with 100 processing nodes, it will be split into 100 subdomains that are each 100 m × 5 km. It cannot be divided into 100 subdomains of size 1000 m × 500 m. Because SWAN's (and SWANFAR®'s) computations sweep through the domain from each of the four corners, many processors must sit idle until the sweep computation reaches their domain. In the adjoint, this delay is more significant, as a greater number of computations must be completed for each subdomain. While these structural limitations do not necessarily affect assimilation accuracy and quality, they can implicitly limit the applicability of SWANFAR® to shorter time periods in smaller domains.

In contrast, WAVEWATCH III® divides the model domain first spatially and then spectrally. It uses a multi-step process to first compute spatial propagation at each frequency and directional bin, then uses the results to compute intra-spectral propagation at each spatial location. The computation of source/sink terms is addressed in an additional fractional step. These extra domain subdivisions will likely improve the efficiency of a WAVEWATCH III® adjoint, when one is created.

Other challenges for SWAN-based assimilation stem from the linearization of often complicated nonlinear expressions for source/sink terms and variables that are used in matrix solver routines. The development of tangent-linear and RP forms of many nonlinear subroutines requires a degree of approximation. Some complicated nonlinear functions or expressions (e.g., hyperbolic tangent) that could not be adjoined were either replaced in SWANFAR® by a representative linear expression (e.g., $tanh(x) \approx x$) or constrained to operate only on background variables. This reduces the accuracy and effectiveness of the spectral corrections provided by the system, but it is essential in order to produce a fully consistent RP/adjoint pair. For this reason, the SWANFAR® system should be expected to assimilate spectral data most effectively from milder (i.e., non-stormy) wave environments outside of the surf zone, where most spectral wave components are quasi-linear. When possible, we recommend that the implicit solver in directional space (SOL-MAT) be used (keep default settings or specify "**NUMER ... SIGIM ...**" in **INPUT** file), together with first order propagation (STRSXY; specify "**PROP BSBT**" in **INPUT** file).

There are also specific instances in SWAN where subroutines arbitrarily adjust modeled wave energy levels to conform to various rules. One example of this occurs for SWAN subroutines related to wave breaking in the surf zone. As mentioned in Section 3.1.1, the SWANFAR® system performs relatively poorly when data are only assimilated from within the surf zone. This is a consequence of the wave breaking process, which constrains all breaking wave heights to follow the rule $H_{br} = \gamma h$ (Figure 19). In SWAN, this limiter is enforced by reducing the overall energy of any spectrum for which the computed energy is greater than the allowed E_{max}. This reduction is arbitrarily imposed on all spectra, and it is not possible to create a consistent adjoint to it. For this reason, one cannot use adjoint computations to determine the specific offshore waves that shoaled to produce a given nearshore breaking wave. To overcome this limitation, one or more spectra must be assimilated from completely outside the surf zone. These offshore assimilation locations must be chosen carefully, as the width of the surf zone depends on wave energy levels.

A final challenge comes not from the structure of SWAN, but from the availability of observed spectra. At present, it is unusual to find a moderate-sized coastal ocean domain (e.g., 10^4 sq. km) that includes more than $5 - 10$ spectral observation locations. This scarcity of data harshly limits the extent to which spectral assimilation can be used to improve forward model estimates in such domains. While the application of covariance multipliers (with its inherent assumptions) can effectively spread SWANFAR® corrections some distance from observation locations, large areas will see little or no improvement from such sparse assimilations. While a solution to this problem has not yet arrived, an answer of sorts may eventually come from ongoing efforts to assimilate SAR-based wave spectra (e.g., Ren et al., 2016). If satellite or aerial SAR instruments can be configured to regularly target a domain of interest for high-resolution (e.g., X-band) swath mapping, it may become possible to extract accurate wave spectra for hundreds of locations rather than just a few. Only with this advance will it be possible for SWANFAR® to fully correct wave spectra effectively for all locations in moderate-sized domains. Without it, corrections will be limited to smaller areas surrounding observation points.

In conclusion, we have completed a series of important upgrades to SWANFAR®, a 5DVAR spectral wave assimilation system based on the model SWAN. The new system now includes adjoint and tangent linear

subroutines for all major advection, diffusion, source/sink, and matrix solver routines in version 40.81 of SWAN. The basic implicit matrix solver (SOLMAT) is fully functional, but the other two solvers (SOLMT1 and SWSIP) require additional validation. The system operates in both stationary and nonstationary modes, although the time covariance in the nonstationary assimilation is not yet fully functional. SWANFAR® has also incorporated a coupled package of routines that apply configurable covariance multipliers to adjoint output for each of the model's five dimensions (see Veeramony et al., 2016). Test results presented here demonstrate the significant improvement in model performance due to these modifications. For the present, it is recommended that SWANFAR® assimilations be limited to 2-3 day scenarios with relatively mild wave conditions, using default settings for covariance, in domains including a relatively high density of observation locations.

5. APPENDIX A - Performing an assimilation with SWANFAR®

5.1. General Operation

At present, the most updated version of SWANFAR® is integrated into a larger COAMPS-based "Coupled DA" system together with NCOM, which operates on UNIX/LINUX-based architecture. The coupled code consists of a number of binary executables and scripts which are called in a specific order. The SWANFAR® executables assimilate and process the spectral wave data, while the Coupled DA system performs the conjugate gradient computations to minimize the cost function. Once properly configured, three linked shell scripts (**run_cycle.sh**, **run_fcst.sh**, and **run_assim.sh**, modified from versions by M. Carrier, NRL 7321) allow users to start uninterrupted multi-day assimilations with a single command-line entry. An example of each of these script files is provided in Appendix B.

In the **run_cycle.sh** script, users may configure working directory locations[4], number of processors, conjugate gradient iterations, and date-time range of the assimilation. This script controls the full assimilation, first calling **run_fcst.sh** to run the forward SWAN model, and then calling **run_assim.sh** to run the SWANFAR® assimilation system and covariance multiplier. The **run_fcst.sh** script runs the forward model for the cycle time period specified in **run_cycle.sh**, saving background spectral estimates for the full domain and at observation locations as specified in the data assimilation input parameter file (see Section 5.2 below). The **run_assim.sh** script then utilizes the saved background files together with observed wave spectra to initialize the conjugate gradient minimization for the specified cycle. The minimization is limited to the number of iterations specified by the user in **run_cycle.sh**. When the minimization is completed, SWANFAR® saves optimal spectral corrections for the full domain and for individual observation locations (see Section 5.2). Control is then returned to **run_cycle.sh**, which steps forward to the next assimilation time period and calls **run_fcst.sh**.

The Coupled DA code executables, accompanying shell scripts, and additional guidance are available from M. Orzech upon request.

5.2. Input and Output

As currently configured, the SWANFAR® system requires three separate ASCII text input parameter files. The **coamps.rc** file is a namelist file containing basic settings used by the Coupled DA system (e.g., set time range for assimilation, turn on/off NCOM, etc.). The **input.cov** parameter file provides information for the covariance multiplication. Adjusting the settings in this file allows the user to turn on/off covariance multipliers in each of the five dimensions and specify related values such as correlation lengths, variance multiplier, etc. Both of these files are created at runtime by the **create_nl.sh** script, which is called by **run_cycle.sh**. An example **create_nl.sh** script with comments describing each section is provided in Appendix E.

The third input parameter file is the data assimilation parameter file (a text file often named simply **INPUT**). In the current configuration, the forward SWAN model uses its own separate SWAN-format input file, while the adjoint and RP components of SWANFAR® share a second version. The adapted latter input file includes many original keywords from SWAN (e.g., *SET, READGRID, COMPUTE*, etc.), as well as additional keywords specifically for the adjoint or RP SWAN modules (e.g., *INNOV, ASPECOUT, RPOINTS*, etc.). At runtime, each component ignores keywords that do not apply to its specific tasks.

---

[4]The primary working directory, EXP_DIR in **run_cycle.sh**, must contain subdirectories **run**, **bin**, **obs**, **parms**, and **output**. All executables will be located in **bin**. Spectral observation input files should be in **obs**. Input files for SWANFAR® are stored in **parms**. The **output** directory is used for storage of temporary assimilation files. Computations are performed in the **run** directory, which should contain the three run scripts, the **create_nl.sh** script, hotstart and boundary spectral files (as needed) for forward SWAN, bathymetry files, and a text file specifying observation point locations (often called "obspts.txt").

A full list of new, SWANFAR®-specific keywords and syntax is provided in Appendix C, and a sample **INPUT** file for the adjoint and RP modules is provided in Appendix D. As indicated by comments in the sample file, some settings are presently hard-wired into the assimilation system, and it is not recommended that they be modified by the user. In the present configuration, the **run_fcst.sh**/**run_assim.sh** script copies a generic form of the appropriate input file from the **parms** directory to the working (**run**) directory at the beginning of each forward/assimilation cycle, then modifies it as necessary to apply to the date range of that cycle (see Appendix B).

Similar to SWAN, most SWANFAR® assimilations also require several additional input data files, including observed/assimilated and boundary spectra (preferably as SWAN-format ASCII text files), bathymetry grid and depths, and a text file listing the coordinates of all assimilation locations in two columns (generally the same format and values as the coordinates provided in the observed spectra file).

Output of the system is configured by the parameters specified in the three input files. Like SWAN, the SWANFAR® system can save spectra from its forward, adjoint, and RP components at individual or multiple locations and times, in either ASCII text files or binary flatfiles. Several output "work" files are produced as part of the data assimilation (spec_dif.frw, spec_dif.in, swan_preproc_done.txt, CG_wav_vct.dat) and may be deleted once results have been obtained.

Post-assimilation output is generally saved in smaller text files for individual assimilation locations and in larger binary flatfiles for the entire domain (depending on user specifications in **INPUT** file). As noted at the beginning of this section, SWANFAR® is presently hard-wired to work with specific filenames for certain input and output. RP-generated spectral corrections are generally stored in the SWAN-format ASCII file "**spec_rpobs.out**" (at assimilation locations) and binary flatfiles "**spcr2d_[A]_[B]_specfld**" (in which [A] represents the starting 10-digit year/month/date/hour – YYYYMMDDHH – of the dataset and [B] represents the additional 8-digit day/hour/minute/second – ddhhmmss – time increment needed to reach this specific output time). Background spectra generated by forward SWAN are stored in the SWAN-format ASCII file "**spec_frw.out**" (at assimilation locations) and binary flatfiles "**spec2d_[A]_[B]_specfld**" for the full domain (same naming syntax as "spcr2d" filenames). Each set of flatfiles for a given YYYYM-MDDHH is accompanied by a corresponding header file, "**sp*2d_[A]_[B0]_spechdr**" (in which [B0] is the initial time of the cycle, most often simply "00000000") providing basic dimensional information about the data. To obtain corrected model spectra at any given location and time, the background spectrum (from forward SWAN) must be added to the corresponding correction spectrum (from RP SWAN). For further information on binary flatfiles, see documentation for the latest NRL version of the SWAN model code. Matlab scripts for reading the binary flatfiles are available from M. Orzech or T. Campbell (NRL 7322).

6. APPENDIX B - Examples of assimilation shell scripts

run_cycle.sh:

```ksh
#!/bin/ksh
fatal() {
   print "====================================================="
   print "Script stopped on error condition at " `date +%T`
   print "====================================================="
   exit 1
}
#
################################################################################
#                          BEGIN USER MODIFICATIONS
#
################################################################################
#
# tiling (NB: SWANFAR will only use ipr*jpr for 1D tiling):
export ipr=4
export jpr=2
#
# cycle parameters
export ITERMAX=5             ## maximum number of CG iterations
export CRDATE=2013071100   ## start date
export CRDATE0=$CRDATE         ## preserve orig start date
export ENDDTG=2013071200   ## end date
export cyclen=24                ## cycle length in hours
export cpl_freq=1           ## frequency of coupling
#
# directories
export EXP_DIR=/u/COAMPS/coastalDA/TW_3km2
#
################################################################################
#                          END USER MODIFICATIONS
#
################################################################################
#
# Global DIRs
export RUNDIR=$EXP_DIR/run
export BINDIR=$EXP_DIR/bin
#
# WAV/OCN DIRs
export INCOM_OBSDIR=$EXP_DIR/obs
export INCOM_PARDIR=$EXP_DIR/parms
export INCOM_OUTDIR=$EXP_DIR/output
export INCOM_IO_PREC=r4
#
# WAV DIRs
export SWANFAR_OBSDIR=$EXP_DIR/obs
#
# General DIRs
export ODIR=$RUNDIR/out
#
# set executable commands
alias DTG=$BINDIR/dtg
run_assim=$RUNDIR/run_assim.sh
run_fcst=$RUNDIR/run_fcst.sh
#
# set tiling file (spmd.D) for multiprocessors
export iprsum=$ipr
export jprsum=$jpr
mkdir -p $ODIR
jqr=`expr ${ipr} \* ${jpr}`
cat << eof > $ODIR/spmd.D
 ipr  jpr  jqr  iprsum  jprsum
$ipr $jpr $jqr $iprsum $jprsum
eof
```

run_cycle.sh (cont):

```
cat << eof > $ODIR/spmd_001.D
 ipr  jpr  jqr  iprsum  jprsum
  1    1    1    1       1
eof
#
# create assimilation log directory
mkdir -p $RUNDIR/LOGS
##
#######################################
## CYCLE THROUGH ASSIMILATION WINDOWS #
#######################################
#
# source the site.env file
source $RUNDIR/site.env
##
## while loop on assimilation cycle
##
while [[ $CRDATE -lt $ENDDTG ]]; do
      CURDTG=$CRDATE
      ANADTG=`DTG -h $cyclen 2>/dev/null`
      export ANADTG=$ANADTG
       ## run forecast to create background
      TDATE=`DTG -h $cyclen +%Y%m%d 2>/dev/null`
      TTIME=`DTG -h $cyclen +%H%M%S 2>/dev/null`00
      export TDATE=$TDATE
      export TTIME=$TTIME
##
        ${run_fcst}
        ## SWAN - Process and update first guess
        Afile=$RUNDIR/backup_${TDATE}_000000
        ## kill if run_fcst did not produce SWAN forecasts
        if [ ! -e ${Afile}_specfld ]; then
         print " ${Afile}_specfld does not exist"
         print "error: SWAN did not create first-guess restart file!"
         fatal
        fi
        ## SWAN - copy restart (first guess) files to spectral output dir
        print " *** NOT copying ${Afile} to ffout dir ***"
##@@        /bin/cp ${Afile}_* $RUNDIR/ffout/
##@@      fi
      export CRDATE=$CURDTG
       ## run 4dvar
       ${run_assim}
        Afile=$RUNDIR/backup_${TDATE}_000000
        Bfile=$RUNDIR/forcst_${TDATE}_000000
        ## SWAN - kill if previous run_assim did not produce SWAN forecasts
        if [ ! -e ${Bfile}_specfld ]; then
         print " ${Bfile}_specfld does not exist"
         print "error: TLM SWAN did not create forecast restart file!"
         fatal
        fi
        ## SWAN - overwrite orig first-guess file with forecast file
        print " *** NOT overwriting orig ${Afile}_specfld ***"
##@@        /bin/mv ${Bfile}_specfld ${Afile}_specfld
##        fi
        export CRDATE=$CURDTG
       ## update CRDATE by cyclen
       CRDATE=`DTG -h $cyclen 2>/dev/null`
       export CRDATE=$CRDATE
       ## end loop ##
done
date
##
```

run_fcst.sh:

```ksh
#!/bin/ksh
fatal() {
    print "===================================================="
    print "Script stopped on error condition at " `date +%T`
    print "===================================================="
    exit 1
}
################################################################################
## THIS SCRIPT WILL RUN A SINGLE 24-hour FORECAST (using the RESTART files)
################################################################################
##
## set binary paths and executable commands
jqr=`expr ${ipr} \* ${jpr}`
fwd_exe=$BINDIR/coamps_nlm.exe
run_namelists=$RUNDIR/create_nl.sh
run_fwd="mpirun -np $jqr $fwd_exe"
alias DTG=$BINDIR/dtg
## begin forecast loop
export CRDATE=$CRDATE
ERDATE=`DTG -h $cyclen 2>/dev/null`
tau=24
while [[ $CRDATE -lt $ERDATE ]]; do
      ## set namelists coamps.rc and oparm_1.D
      ${run_namelists}
      ## copy time-tagged SWAN input file to generic
      Afile=${INCOM_PARDIR}/swan.inp.fwd.orig
      if [ ! -e $Afile ]; then
         print " $Afile does not exist"
         print "error: 4DVAR copy fwd SWAN input has failed!"
         fatal
      fi
      TDATE0=`DTG -h 0 +%Y%m%d 2>/dev/null`
      TDATE=`DTG -h $cyclen +%Y%m%d 2>/dev/null`
      sed "s/20130002/${TDATE}/g" $Afile > tmp.txt
      sed "s/20130001/${TDATE0}/g" tmp.txt > $RUNDIR/INPUT
      rm tmp.txt
      ## run fwd model for 24 hours
      export NCOM_SPMD_0D=$ODIR/spmd.D
      ${run_fwd} 2>&1 |tee ${ODIR}/fcst.log ## runs fwd model
      TDATE=`DTG -h $tau +%Y%m%d 2>/dev/null`
      TTIME=`DTG -h $tau +%H%M%S 2>/dev/null`00
      ## copy SWAN output to time-tagged file; DON'T reorient directions
      cp spec_frw.out ${ODIR}/spec_frw.out.${CRDATE}
      cp spec_frw.all ${ODIR}/spec_frw.all.${CRDATE}
      ## save forecast log file to LOGS
      cp ${ODIR}/fcst.log ${RUNDIR}/LOGS/fcst.${CRDATE}.log
      ## update time counter
      CRDATE=`DTG -h $tau 2>/dev/null`
      export CRDATE=$CRDATE
   ## continue loop until all ## forecast days are complete
done
##
##
```

run_assim.sh:

```ksh
#!/bin/ksh
fatal() {
    print "=================================================="
    print "Script stopped on error condition at " `date +%T`
    print "=================================================="
    exit 1
}
################################################################################
## THIS SCRIPT WILL RUN A SINGLE ASSIMILATION OUTER LOOP
################################################################################
##
alias DTG=$BINDIR/dtg
##
## print environmental variables
print " Environment Info: "
which mpirun
which pgf90
##
## set binary paths and executable commands
##
obss_exe=$BINDIR/swan_preprocess.exe
var_exe=$BINDIR/coamps_vda.exe
run_namelists=$RUNDIR/create_nl.sh
jqr=`expr ${ipr} \* ${jpr}`
arg=""
run_obss="$obss_exe"
run_var="mpirun -np $jqr $var_exe"
##
## set namelists
${run_namelists}
## make working directory for covariances
mkdir -p $RUNDIR/runtime
##
## process individual SWAN observation files into two files for the 4dvar
Afile=$INCOM_OBSDIR/spec_obs.in.${CRDATE}
Bfile=$ODIR/spec_frw.out.${CRDATE}
if [ ! -e $Afile ]; then
    print " $Afile does not exist"
    print "error: 4DVAR process SWAN obs has failed!"
    fatal
fi
if [ ! -e $Bfile ]; then
    print " $Bfile does not exist"
    print "error: 4DVAR process SWAN obs has failed!"
    fatal
fi
## copy time-tagged SWAN obs and est files to generic, then process obs/est
cp $Afile $RUNDIR/spec_obs.in
cp $Bfile $RUNDIR/spec_frw.out
## remove any previous copy of observation file from earlier run
rm -f $RUNDIR/swanfar_obs_4DV.bin
## remove flag file indicating that swan_preprocess has completed
rm -f $RUNDIR/swan_preproc_done.txt
${run_obss} 2>&1 |tee ${ODIR}/proc_swan_obs.log ## processes observations into two
files
Afile=$RUNDIR/swan_preproc_done.txt
if [ ! -e $Afile ]; then
    print " $Afile does not exist"
    print "error: swan_preprocess.exe has failed!"
    fatal
fi

##
```

run_assim.sh (cont):

```
## copy SWANFAR input file to generic INPUT
Afile=${INCOM_PARDIR}/swan.inp.swanfar.orig
if [ ! -e $Afile ]; then
   print " $Afile does not exist"
   print "error: 4DVAR copy SWANFAR input has failed!"
   fatal
fi
TDATE0=`DTG -h 0 +%Y%m%d 2>/dev/null`
TDATE=`DTG -h $cyclen +%Y%m%d 2>/dev/null`
sed "s/20130002/${TDATE}/g" $Afile > tmp.txt
sed "s/20130001/${TDATE0}/g" tmp.txt > $RUNDIR/INPUT
rm tmp.txt

##
## process global out3d files into tiled files for 4dvar based on spmd.D
TDATE=`DTG +%Y%m%d 2>/dev/null`
TTIME=`DTG +%H%M%S 2>/dev/null`00
##
## run the 4dvar
${run_var} 2>&1 |tee ${ODIR}/coamps.log ## runs the 4dvar
cp ${ODIR}/coamps.log $RUNDIR/LOGS/coamps.${ANADTG}.log
Afile=$INCOM_OUTDIR/analinc_${TDATE}_${TTIME}.A
##
## Post-process the 4dvar
Afile=$RUNDIR/spec_rpobs.out
Bfile=$RUNDIR/spec_rpobs.all
if [ ! -e $Afile ]; then
   print " $Afile does not exist"
   print "error: 4DVAR copy SWANFAR rp output has failed!"
   fatal
fi
cp $Afile ${ODIR}/spec_rpobs.out.${CRDATE}
cp $Bfile ${ODIR}/spec_rpobs.all.${CRDATE}
##
## remove latest adjoint flatfile output to save space
## NO. Need all spca2d files for covariances!     WAS:
print "***** NOT Removing all files spca2d_${CRDATE} "
## /bin/rm $RUNDIR/ffout/spca2d_${CRDATE}*
```

7. APPENDIX C

Keywords specific to SWANFAR® (for use in **INPUT** file):

Adjoint:

APOINTS = Specify coordinates of desired adjoint output points. Same syntax as POINTS keyword used by SWAN. Paired with ASPECout.

ASPECout = Specify details of output spectral files written by adjoint only. Same syntax as SPECout keyword used by SWAN. Paired with APOINTS.

INNOV = (REQUIRED) Provide filename of SWAN-format ASCII text file containing innovations (i.e., model-data difference spectra) at all observation locations and times. SWANFAR® is presently configured to accept only 'spec_dif.in' for this filename (see Appendix D for syntax).

RP Model:

RPOINTS = Specify coordinates of desired RP output points. Same syntax as POINTS keyword used by SWAN. Paired with RSPECout.

RSPECout = Specify details of output spectral files written by RP model only. Same syntax as SPECout keyword used by SWAN. Paired with RPOINTS.

Timing-Related Specifications:

TBKG = Specify times at which to load background spectra (generated by preceding run of forward SWAN) into adjoint and RP SWAN. For use with nonstationary assimilations involving multiple time steps. Syntax is similar to that of the COMPUTE command for nonstationary scenarios. The keyword TBKG is followed by a start time, a timestep, and an end time. See Appendix D for an example of this format in an INPUT file. Generally, the time values specified here should be the same as the times specified as output times for the forward SWAN computation corresponding to this assimilation.

TRPAC = Specify times at which RP model will load adjoint spectra (generated by preceding iteration of adjoint SWAN) for initialization of the next iteration of RP SWAN. For use with nonstationary assimilations involving multiple time steps. Syntax is same as that of TBKG described above. See Appendix D for an example of this format in an INPUT file. Generally, the time values specified here should be the same as the times specified as output times for the adjoint SWAN computation (using ASPECout) for to this assimilation.

8. APPENDIX D

Sample **INPUT** file for SWANFAR®:

```
*************************************************************
$——————————————————————————————————$
$ Start-up
$——————————————————————————————————$
SET 0 90 0.1 200 2 9.81 1025.0 99999 1 0.10 CART
COORDINATES SPHERICAL CCM
OUTPUT OPTIONS FLATFILE COAMPS 'ffout'
$——————————————————————————————————$
$ Computational grid and initial conditions
$——————————————————————————————————$
CGRID CURV 100 81 EXCEPTION 999 999 CIRCLE 36 0.0418 0.9708114547513104
READGRID COORD 1.0 'grid.dat' 4 UNFORMATTED
INPGRID BOTTOM CURV EXCEPTION -999
READINP BOTTOM 1.0 'bottom.dat' 4 UNFORMATTED
$ **Will be read by Forward SWAN only**
INITIAL HOTSTART FLATFILE DATETIME '20130714_000000'
$——————————————————————————————————$
$ Inputs **DO NOT MODIFY**
$——————————————————————————————————$
$ SWANFAR innovation input file (Will be read by ADJOINT only)
INNOV 'spec_dif.in'
$——————————————————————————————————$
$ Physics
$——————————————————————————————————$
SSWELL ARDHUIN 1.2
FRIC JON 0.019
OFF QUAD
BREAK CON 1.00 0.80
TRIAD
$————————————————————————————————
$ Numerics **DO NOT MODIFY BSBT or ALPHA=0.00 BELOW **
$————————————————————————————————
$ Propagation scheme
PROP BSBT
$ Numerics
NUM ACCUR 0.02 0.02 0.02 98.0 STAT 1 0.00 0.1
$————————————————————————————————
$ Forward SWAN Outputs **DO NOT MODIFY except time/date**
$————————————————————————————————
$ Flatfile spectra output settings
SPECOUT 'FLATFILE' SPEC2D RELATIVE OUTPUT 20130714 3600 SEC
$ Forward estimates at observation points
POINTS 'LOC1' FILE 'obspts.txt'
SPEC 'LOC1' SPEC2D REL 'spec_frw.out' OUTPUT 20130714.000000 1.0 HR
$————————————————————————————————
$ Adjoint/RP Outputs **DO NOT MODIFY except time/date **
$————————————————————————————————
$ Flatfile ADJOINT spectra output settings
ASPECOUT 'FLATFILE' SPEC2D RELATIVE OUTPUT 20130714.000000 1.0 HR
$ flatfile RP spectra output settings
RSPECOUT 'FLATFILE' SPEC2D RELATIVE OUTPUT 20130714 12.0 HR
```

$ ASCII text RP spectra output settings
RPOINTS 'LOCR' FILE 'obspts.txt'
RSPEC 'LOCR' SPEC2D ABS 'spec_rpobs.out' OUTPUT 20130714.000000 1.0 HR
$——
$ Timing-related Specifications
$——
$ Specify background file times to load in adjoint/rp modules
$ **REQUIRED.** Will be read by ADJOINT/RP only**
TBKG 20130714.000000 1.0 HR 20130715.000000
$
$ Times to tell RP model when to load nonstationary outputs from ADJOINT
$ **REQUIRED.** Will be read by RP only**
TRPAC 20130714.000000 1.0 HR 20130715.000000
$
$——
$ Computation
$——
$
TEST ITEST= 1 ITRACE= 1
COMPUTE NONSTAT 20130714.000000 600 SEC 20130715.000000
STOP
****

9. APPENDIX E

Example **create_nl.sh** script, which creates **coamps.rc** file with parameters for coupled DA and **input.cov** file with parameters for covariance multiplication in SWANFAR®:[5]

————————————————————————

```
#!/bin/ksh
##
## Create coamps.rc namelist
cat << eof > coamps.rc
verbose: t
time_step: 1 0 0
start_time: 2013 7 14 00 00 00
run_duration: 24 0 0
ATM_active: f
OCN_active: f
WAV_active: t
OCN_type: ncom
WAV_type: swan
pet_layout_option: sequential
pet_count: 8
OCN-TO-WAV_type: bilinr
WAV-TO-OCN_type: bilinr
eof
##
## Create input.cov namelist
cat << eof > input.cov
&NAM_wave
    ! Paths to working directories; data file prefixes and suffixes[6]
    path_work = '/u/COAMPS/coastalDA/TW_3km2/run',
    path_data = '/u/COAMPS/coastalDA/TW_3km2/run/ffout',
    path_runtime = '/u/COAMPS/coastalDA/TW_3km2/run/runtime',
    path_parameter = '/u/COAMPS/coastalDA/TW_3km2/run/runtime',
    path_test = '/u/COAMPS/coastalDA/TW_3km2/run/runtime',
    prefix_adj = 'spca2d',
    prefix_bkg = 'spec2d',
    suffix_header = 'spechdr',
    suffix_field = 'specfld',

    ! Timing and bathymetry grid information
    starting_datetime = '2013071400',
    timestep = 1,
    n_time_step = 24,
    bathymetry_FName = 'bottom.dat',
    bathymetry_fmt = .false.,
    grid_in_degrees = .true.,

    ! **Temporary files for spectral data**
    d_corr_lngth_filename = 'dspr.bin',
    s_corr_lngth_filename = 'frqwdth.txt',

    ! Correlation length values [space(m), dir(deg),freq(Hz),time(hr)]
```

————————————————————————

[5]When **create_nl.sh** is called by **run_cycle.sh**, some of the values shown may be replaced by runtime variables; e.g., $YR, $MO, $DY.

[6]Comment lines (with !) may need to be removed before this file will be accepted for an assimilation. Consult an expert before changing parameter values that follow comment lines with asterisks (**).

r_corr_lngth_value = 50000.,
d_coor_lngth_value = 20.0,
s_corr_lngth_value = 0.02 0.04,
t_corr_lngth_value = 12.,

! **Resolution (grid cells) of r_corr computation (interpolate others)**
r_normalization_step = 10,

! Normalization variance
variance = 0.07

! Switches to turn on/off covariance in each dimension
apply_R_cov = .true.,
apply_D_cov = .true.,
apply_S_cov = .true.,
apply_T_cov = .false.,

! **Tolerance and max iterations for diffusion/normalization**
rbcg_tol = 1.0d-16,
rbcg_NiterMax = 20,
NRMLZ_nIterMax = 100
/
eof

————————————————————————————

10. APPENDIX F

SWAN 40.81 subroutines that were **not** modified in SWANFAR®:

swancom1.ftn: SWPRSET, SACCUR, INSAC, SWSOR, SWMTLB, SWSTPC, SETUPP, SETUP2D, SINTGRL

swancom2.ftn: SBOT[7], SVEG[8], FRABRE, BRKPAR, PLTSRC

swancom4.ftn: SWINTFXNL, FAC4WW, RANGE4

swancom5.ftn: SPREDT

swanout1.ftn: SWODDC, SWOEXC, SWIPOL, SWOEXA, SWOINA, SWOEXF

swanout2.ftn: SWBLOK, SBLKPT, SWBLKP, SWTABP, SWCMSP

swanparll.ftn: SWINITMPI, SWEXITMPI, SWSYNC, SWSENDNB, SWRECVNB, SWBROADC, SWGATHER, SWREDUCE, SWREDUCI, SWREDUCR, SWSTRIP, SWPARTIT, SWBLADM, SWDECOMP, SWEXCHG, SWCOLLECT, SWCOLOUT, SWCOLTAB, WREXCV, SWCOLSPC, SWCOLBLK

swanpre1.ftn: SINPGR, SREDEP, SSFILL, SWDIM, CGBOUN, SEPARAREA, INITVA, BACKUP

swanpre2.ftn: SWNMPS, SVARTP, SWBOUN, BCWAMN, BCWW3N, SWBCPT, RETSTP

swanser.ftn: READXY, REFIXY, DISTR, KSCIP1, AC2TST, CVCHEK, CVMESH, NEWTON, NEWT1D, EVALF, SWOBST, SWOBSTO, OBSTMOVE, SWTRCF, REFLECT, SSHAPE, SIN-TRP, HSOBND, CHGBAS, SWACC

w2a.ftn: OUTBETA, AIRSEA, STRESS

ocpcre.ftn, ocpids.ftn, and ocpmix.ftn: RDINIT, NWLINE, INKEYW, INREAL, INDBLE, ININTG, INCSTR, INCTIM, ININTV, LEESEL, GETKAR, PUTKAR, UPCASE, WRNKEY, IG-NORE, NXTREC, OCPINI, OCDTIM, DTSTTI, DTTIST, DTINTI, DTRETI, REPARM, INAR2D, STRACE, TABHED, LSPLIT, BUGFIX

swmod1.ftn & swmod2.ftn: Various arrays added to existing modules (no ADJ/RP versions)

*.ftn90 files: None of these files were changed for SWANFAR®.

---

[7]While the wave action, AC2, is used in SBOT to determine which bottom stress will be applied, there are no dependent variables which are computed or changed based on this active variable. An adjoint to SBOT would not affect AD_AC2.

[8]The active variable AC2 is not present in subroutine SVEG, so its adjoint would not have an effect on AD_AC2.

11. REFERENCES

Aouf, L., Lefevre, J.-M., Hauser, D., 2006. Assimilation of directional wave spectra in the wave model WAM: an impact study from synthetic observations in preparation for the SWIMSAT satellite mission. J. Atm. Oceanic Tech. 23(3), 448-463.

Bennett, A.F., 2002. Inverse Modeling of the Ocean and Atmosphere. Cambridge U. Press, New York, 234 pp..

Booij, N., Ris, R.C., Holthuijsen, L.H., 1999. A third-generation wave model for coastal regions 1. Model description and validation. J. Geophys. Res. 104(C4), 7649-7666.

Erwig, M., Fu, Z., Pflaum, B., 2007. J. Software Maint. Evol. Res. Pract. 19, 155-182.

Hasselmann, S., Lionello, P., Hasselmann, K., 1997. J. Geophys. Res. 102(C7), 15823-15836.

Komen, G., Cavaleri, L., Donelan, M., Hasselmann, K., Hasselmann, S., Janssen, P.A.E.M., 1994. Dynamics and Modelling of Ocean Waves. Cambridge U. Press, New York.

Orzech, M.D., Veeramony, J., Ngodock, H., 2013. A variational assimilation system for nearshore wave modeling. J. Atm. & Oceanic Tech. 30, 953-970.

Ren, L., Yang, J., Zheng, G., Wang, J., 2016. A joint method to retrieve directional ocean wave spectra from SAR and wave spectrometer data. Chin. J. Oceanology & Limnology 34(4), 847-858.

Terrill, E.J., 2013. Trident Warrior Buoy Testing. ONR FY13 Research Report, Award N00014-13-1-0576.

Tolman, H.L., 2014. User manual and system documentation of WAVEWATCH III® version 4.18. NOAA/NWS/NCEP/MMAB Tech. Note 316, 282 pp. + appendices.

Veeramony, J., Walker, D., Hsu, L., 2010. A variational data assimilation system for nearshore applications of SWAN. Oc. Mod. 35, 206-214.

Veeramony, J., Flampouris, S., Orzech, M., Ngodock, H., 2016. Development of covariance multipliers for data assimilation with the SWANFAR® system. NRL Research Report NRL/MR/7320-16-XXXX,1-40.

Voorips, A.C., Makin, V.K., Hasselmann, S., 1997. Assimilation of wave spectra from pitch-and-roll buoys in a north sea wave model. J. Geophys. Res., 102(C3), 5829-5849.

Walker, D.T., 2006. Assimilation of SAR imagery in a nearshore spectral wave model. Tech. Rep. 2002 36, GDAIS.

Wave Model Development and Implementation (WAMDI) Group, 1998. The WAM model - a third generation ocean wave prediction model. J. Phys. Oceanog. 15, 566-592.