Technical Report UNH-OPAL-1996-003

Hindcasting in the Gulf of Maine Wind Field: A Case Study

Hui Feng and Wendell S. Brown

Ocean Process Analysis Laboratory
Institute for the Study of Earth Ocean and Space
Department of Earth Sciences
University of New Hampshire
Durham, NH 03824

Technical Report UNH-OPAL-1996-003

Hindcasting in the Gulf of Maine

Hui Feng and Wendell S. Brown

Ocean Process Analysis Laboratory
Institute for the Study of Earth Ocean and Space
Department of Earth Sciences
University of New Hampshire
Durham, NH 03824

Summary

This report describes the optimal interpolation methodology that was used to derive nowcast and hindcast wind fields from the suite of hourly NDBC/NWS wind measurements in the Gulf of Maine. The techniques described by *Mooers et al.* [1976] were combined with computed spatial correlations for spring and summer 1994 respectively to produce weights for deriving winds at locations where there were no observation. The programs for the computations are included.

I. Database Description and Preprocessing

The time series of wind records from 1 April to 15 May 1994 (SPRING) and 18 July to 4 September 1994 (SUMMER) were selected for this case study. The observed hourly wind speed and direction series from all of the NDBC (National Data Buoy Center) buoys and C-man stations in Figure1, except Nantucket, were used here. Spurious and missing values in the original data were replaced with interpolated values. The measured wind speeds were adjusted to a common height of 10 m using an iterative scheme assuming a neutrally stable atmosphere [*Large and Pond*, 1981]. The one-and-one-half-year wind time series was divided into nine bimonthly segments because the correlation characteristics of the wind fields in the Gulf of Maine vary seasonally. Separate hindcasts were completed for each segment.

II. Wind Field Hindcasting Algorithm

The *Mooers et al.* [1976] hindcasting scheme employed here uses the wind observations at a few stations and spatial statistical properties of the observed wind field to determine winds at many other places (i.e., everywhere). The formalism is based on an:

- Input $f_i(t)$: wind component time series at an observation point ($i = 1,2 \ldots NO$, where NO is the number of observation stations)
- Output $g_j(t)$: wind component time series at a hindcast point ($j = 1,2 \ldots NH$, where HN is the number of hindcast points)

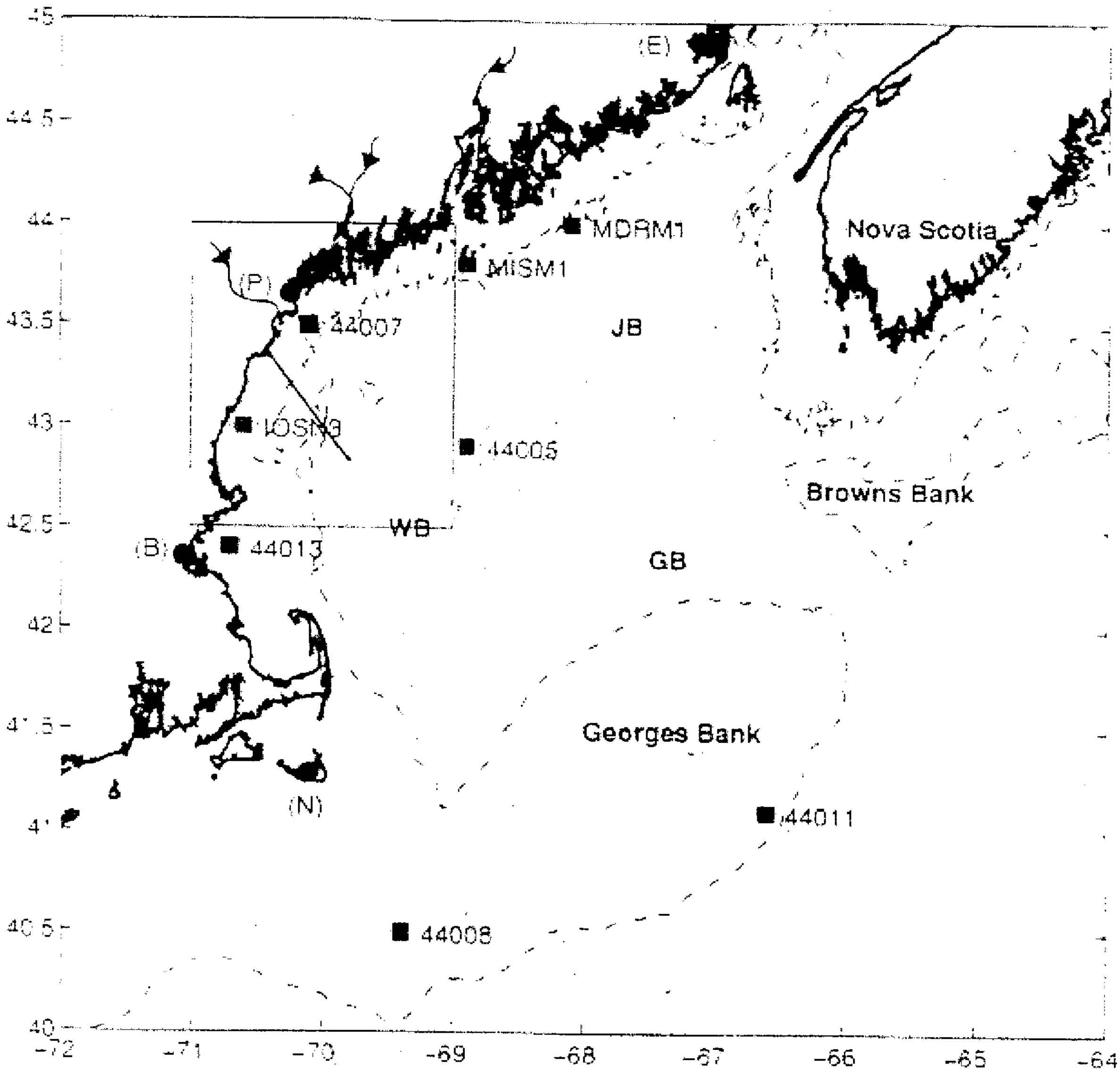

Figure 1. Map of the Gulf of Maine, showing the principal basins of the gulf and the major offshore banks. The NDBC/NWS meteorological stations (squares) and the NOS and MEDS coastal sea level stations (solid circles) are shown. (See the text for abbreviations.)

General linear estimation allows us to express the jth wind hindcast series in terms of

$$g_j(t) = \sum_{i=1}^{NO} W_{ij} f_i(t) \quad , \qquad (1)$$

where W_{ij} is the weight relating the time series at the jth hindcast point to the ith observed time series. Actually, we applied a spatial version of the *Mooers et al.* [1976] hindcast algorithm without considering time-lag information.

Equation (1) is based on the assumption that at time t the hindcast series $g_j(t)$ can be expanded in terms of a weighted sum of observed series $f_i(t)$. In general, the relative contributions of the $f_i(t)$s to $g_j(t)$ differ with respect to the characteristics of the wind field under consideration.

To determine the weight function, a least square fitting procedure is used to relate the weights to the cross-covariance (correlation) properties of the wind component fields under consideration according to

$$\hat{R}_{ji} = \sum_{k=1}^{NO} W_{kj} R_{ki} \quad , \qquad (2)$$

where R_{ki} is the cross correlation function between observations at the kth and ith locations respectively, and $\hat{R}_{ji}$ is the estimate of the cross correlation function between the jth hindcast point and the ith observation point. Equation (2) can be written in matrix form for the jth hindcast site according to

$$\hat{R}_j = [R] W_j \quad , \qquad (3)$$

where $\hat{R}_j$ and W_j are NO by 1 column vectors and $[R]$ is a NO by NO observation-only cross-correlation matrix which can be computed directly for each observation pair (i,k = 1,2 … NO) with zero time lag. Before we can determine the weights W_{ij}, the $\hat{R}_{ji}$ must be estimated. The Matlab programs for doing these calculations can be found in the section V Appendix.

For this case study, we have assumed that the wind-component fields in the Gulf of Maine are spatially isotropic. Specifically, this assumption means that the spatial correlation characteristics of wind component field (described by R_{ki}) depends only on the separation distance $S_{ki} = S_k - S_i$ between the observation stations, where S_k and S_i are the position vectors of the stations k and i.

The isotropic spatial cross-correlation function was computed for the east-west and north-south wind components for SPRING and SUMMER respectively. Exponential regressions were fit to the spatial cross-correlations of the east-west and north-south NDBC wind components for computed for SPRING, yielding $R_{ki} = \exp(-0.00178\ S_{ki})$ and $R_{ki} = \exp(-0.00235\ S_{ki})$ respectively(see Figure 2). The spatial correlation scales (SCS - defined in terms of the 95% confidence zero correlation value of 0.2) were 678km and 513km respectively.

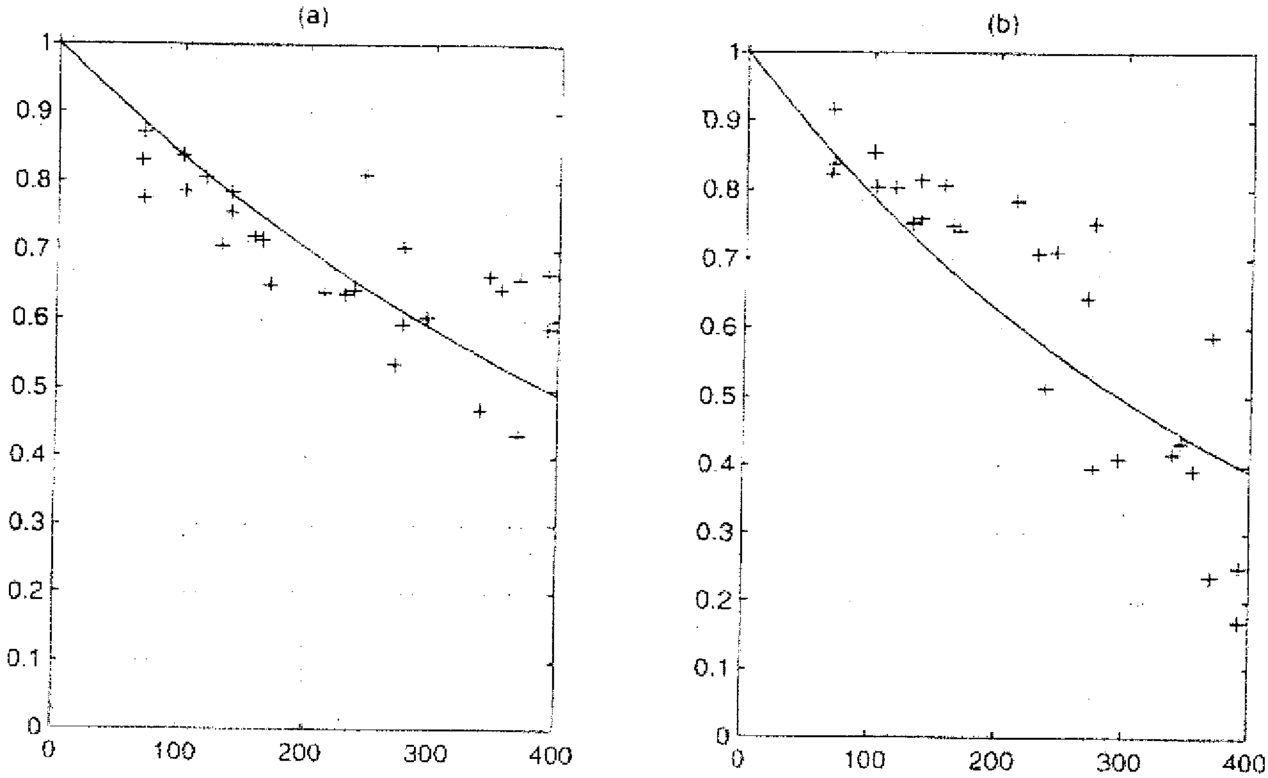

Figure 2. Spatial cross-correlations R_{ki} of the (a) east-west and (b) north-south NDBC wind components versus station separation distance S_{ki} (in km) for the Gulf of Maine for SPRING 1994.

The corresponding exponential fits to the east-west and north-south data in SUMMER were $R_{ki} = \exp(-0.00285\ S_{ki})$ and $R_{ki} = \exp(-0.00213\ S_{ki})$ respectively (see Figure 3). The spatial correlation length scale for the east-west winds in SPRING was 400 km (at R_{ki} = 0.50) versus 250 km in SUMMER.

The spatial correlation functions $\hat{R}_{ji}$ were estimated using the exponential fit with the appropriate $S_{ji} = S_j - S_i$ between any hindcast point j and each observation point i. Finally, the weights were estimated by inverting Eq.(2) for each hindcast point j, and then the hindcast series $g_j(t)$ is computed using Eq.(1). We used the data from the eight NBBC/NWS observation stations in Figure1 as inputs.

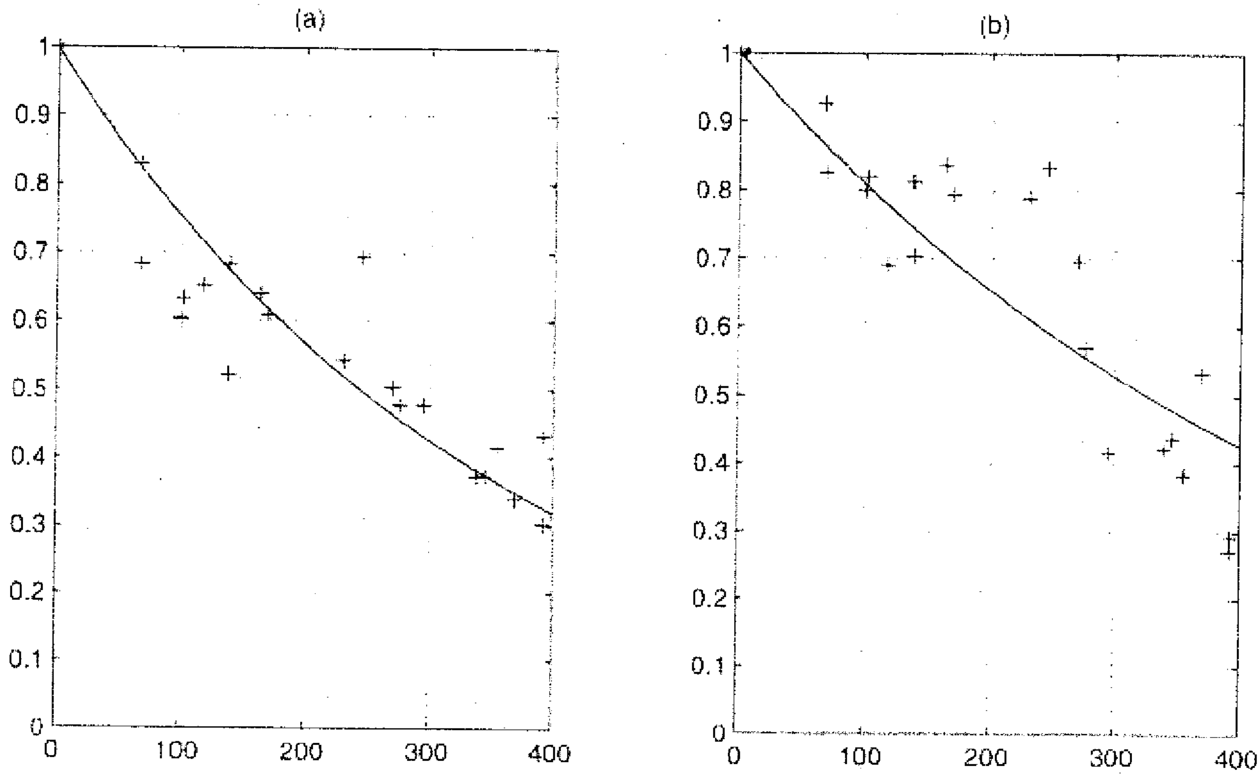

Figure 3. Spatial cross-correlations R_{ki} of the (a) east-west and (b) north-south NDBC wind components versus station separation distance S_{ki} (in km) for the Gulf of Maine for SUMMER 1994.

The hindcast wind vector field for the Gulf of Maine compares favorably with the observed winds (from which they were produced) in the representative 8 May 1994 SPRING example shown in Figure 4. Note the importance of the coastal C-Man station winds in inducing the structure in the cross-coastline winds. The same is true for the somewhat weaker winds in the corresponding SUMMER 20 July 1994 example shown in Figure5.

III. Acknowledgements

The research described here benefited from the contributions of Yalin Fan, who produced the wind correlation results for 1994. This research was supported by the Regional Marine Research Program under NOAA grant NA56RM0555, University of Maine

subcontract UM-S277; UNH/UMaine Sea Grant under NOAA grant NA56RG0159 and University of Maine subcontracts UM-S221 and UM-S304.

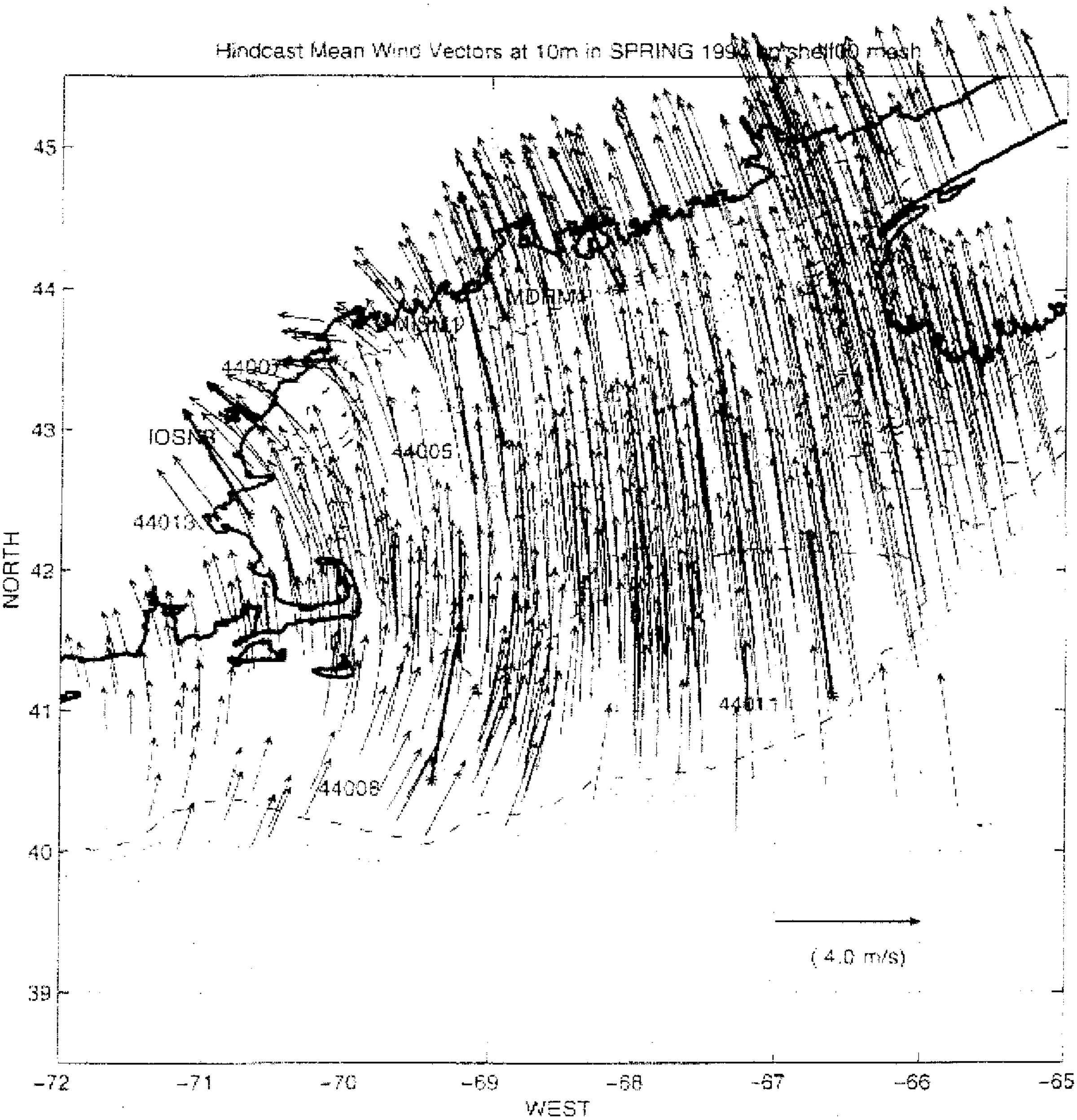

Figure 4. The hindcast wind velocity vectors are compared with the observed winds (bold) at the NDBC/NWS stations for the 8 May 1994.

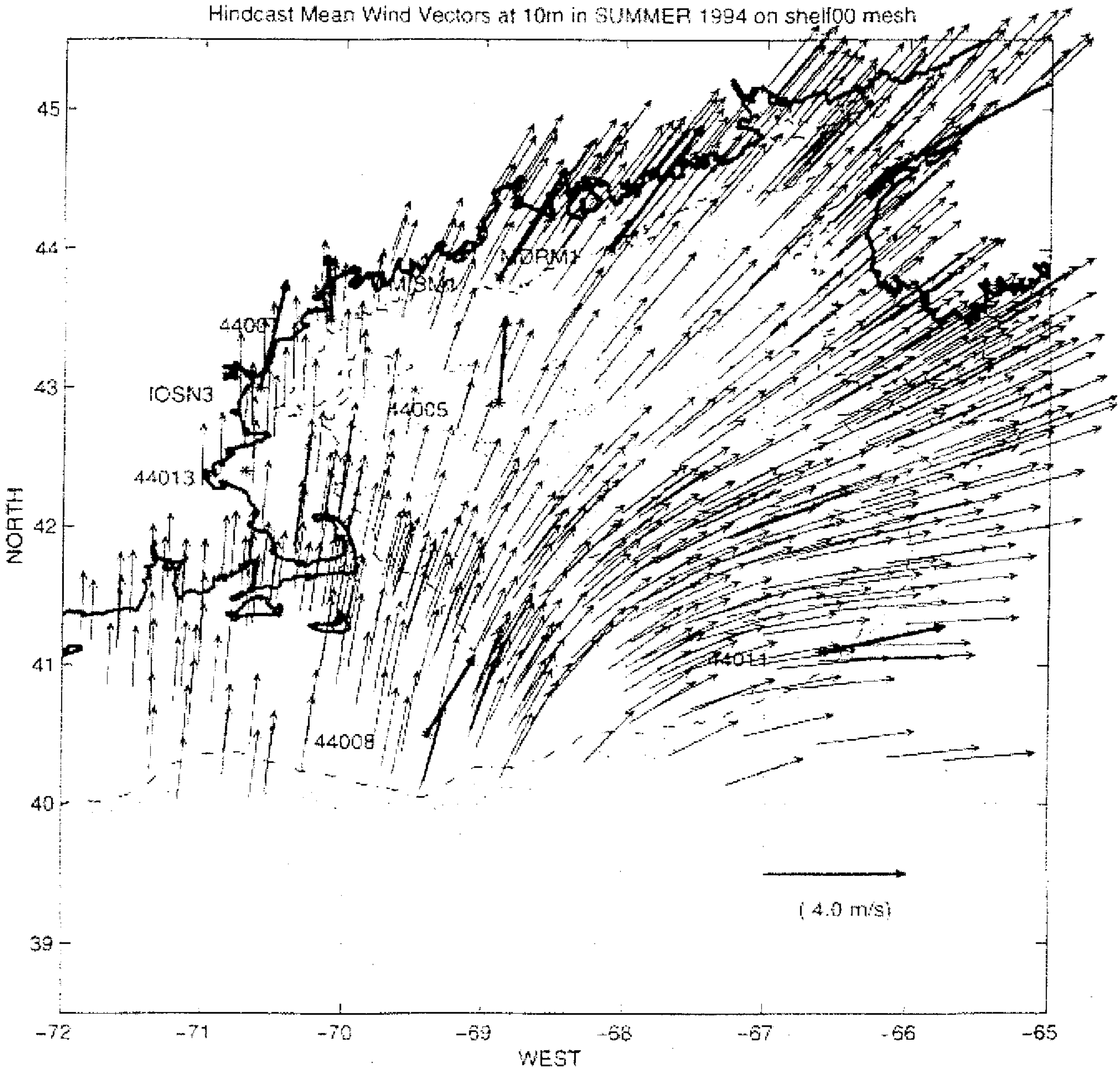

Figure 5. The hindcast wind velocity vectors are compared with the observed winds (bold) at the NDBC/NWS stations for the 20 July 1994.

IV. References

Large, W.G., and S. Pond, 1981. Open ocean momentum flux measurements in moderately strong winds, *J. Phys. Oceanogr.*, *11*, 639-657.

Mooers, C.N.K., J. Fernandez-Partages, and J.F. Price, 1976. Meteorological Forcing Fields of the New York Bight, Univ. Miami Tech. Rpt., University of Miami, Miami, Fl.

V. Appendix The Programs

A set of Matlab functions have been developed to compute the hindcast algorithm. They are:

swssub.m: which is used to calculate wind speed at 10m, and windstress for a neutral stability.

hindcast.m: which is used to calculate the hindcast winds.

hindcastw.m: which used to calculate the hindcast weights.

And can be found below.

****

swssub.m

****

```
function [tauu,tauv,u10,v10] = swssub(uo,vo,ho,delta,padv);
%     by Hui Feng   March 16,1995
%  SWSSUB: compute 1) wind speed at 10m
%                  2) the surface wind stress for a neutral stability
%  Algorithm: using a neutral steady-state drag coeff and an iterative
%             method by Large Pond (JPO,113,1981), the surface wind
%             stress is
%
%             tau=rho*cdn*(w10)^2*1e4   (dynes/cm**2)      (1)
%
%
%     rho    -- air density: 1.22e-3 gm/cm**3
%   neutral stability specific variables:
%        w10    -- wind speed at 10 meters (m/s)
%        hobs   -- observation height (m)
%        k      -- constant=0.40 Von Karman's constant
%        cdn    -- neutral steady state drag coefficient
%
%          cdn*1000=       1.2       0.0< w10 <= 11.0      (2)
%                      .49+.065*w10  11.0< w10 < 25.0
%
%          w10=wo/(1+(sqrt(cdn)/k*alog(ho/10))           (3)
%
%     (2) and (3) are used to compute cnd and w10 in an iterative
```

```
%     scheme;
%         and then cnd and w10 is used to compute the stress of (1)
%
%      input: uo,vo-- obs. u nad v wind component time series (m/s)
%             ho-- observation height (m)
%             padv-- pad value
%             delta-- iteration accuracy using diff of the wind speed
%                       as criterion
%                       (i.e. 0.001 would mean iterate to an accuracy
%                       of 1mm/s)
%
%      output: tauu and tauv --surface wind stresses in u and v dir.
%              u10 and v10 --wind comps in u and v dir (m/s)   at 10m
num=length(uo);
k=0.4; rho=1.22e-3;degr=57.295780;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% do the calculations for speed(x) and direction(y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:num
 spd=sqrt(uo(i)*uo(i)+vo(i)*vo(i));
 x(i)=spd; % which is the series of the wind speed
        if(spd==0.0)
            y(i)=0.0;  % which is the series of the wind direction
          elseif(uo(i) == 0.0)
            y(i)=0.0;
            if(vo(i)<0.0), y(i)=180.0; end
        else
            theta=atan(vo(i)/uo(i));
            y(i)=90.0-degr*theta;
            if(uo(i)<0.0),y(i)=y(i)+180.0; end
        end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  do calculation for speed at 10m and stress(x)
%     --direction will not change
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diff=10;
for i=1:num
spd=x(i);
uobs=spd;
  while 1
    if ( diff<delta), break, end
    if(spd >=0.0 & spd <11.0)

          cdn=1.2*.001;
        elseif(spd > 11.0 & spd < 25.0)
          cdn=(0.49+0.065*spd)*.001;
     end
          ucomp=uobs/(1.0+sqrt(cdn)/k*log(ho/10.0));
          diff=abs(spd-ucomp);

          spd=ucomp;
   end             % for while
  tau(i)=rho*cdn*spd*spd*10000.0;
% which is the series of the amplitude of wind stress
  w10(i)=spd; % which is the series of the wind speed at 10m
```

```
  diff=10;
end                 % end  for

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculate wind speed components u10 and v10 from speed w10 and dir y
% calculate wind stress components tauu and tauv from stress tau and
% dir y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:num
 theta= (90-y(i))/degr; % becoming
        xi=w10(i)*cos(theta);
        yi=w10(i)*sin(theta);
        u10(i)=xi;
        v10(i)=yi;
        xi=tau(i)*cos(theta);
        yi=tau(i)*sin(theta);
        tauu(i)=xi;
        tauv(i)=yi;
 end
end
```

****

hindcast.m

****

```
function [hs,W,RR,Disho]=hindcast(fs,r,ll,grid,a)
%
%  MATLAB UNIX function, hindcast.m, is used to realize an algorithm
%  of Hindcast-analysis (G. Halliwell and C.Moores, 1976), which has
%  been modified by Hui Feng ( See details in the Appendix B of Hui
%  Feng's MS. thesis, 1996 )
%
%    INPUTS ---
%        FS:  Time series matrix from obs. points in the form of
%               f11 f12 f13.....f1T
%               f21 f22 f23.....f2T
%           FS=  .   .   .       .
%                .   .   .       .
%               f71 f72 f73.....f7T
%
%             size = 7xT where T is length of time series, and
%             7 is # of obs. sites ( which could be arbitary) ;


%      R:  Obs.-only (zero-lag) cross-correlation matrix in the form of
%                 c11 c21 c31 c41 c51 c61 c71
%                 c12 c22 c32 c42 c52 c62 c72
%            R=   c13 c23 c33 c42 c53 c63 c73
%                 c14 c24 c34 c44 c54 c64 c74
%                 c15 c25 c35 c45 c55 c65 c75
%                 c16 c26 c36 c46 c56 c66 c76
%                 c17 c27 c37 c47 c57 c67 c77
%            size=7x7 (square matrix) , consistent with # of obs. sites
%
%        LL:  lat/lon matrix in 7 obs. points in the form of
%                   42.9  68.9  1
```

```
%                   44.0  68.1  2
%                   43.8  68.9  3
%                   43.5  70.1  4
%                   43.0  70.6  5
%                   42.4  70.7  6
%                   41.1  66.6  7
%
%        Grid: lat/lon matrix which has an arbitrary size  in the
%              same form as LL
%           A: a parameter associated with an
%              empirical spatial exponential function
%              obtained from the obs. points
%
%     OUTPUT----
%       HS: time series matrix from hindcast  points in the form of
%               h11 h12 h13.....h1T
%               h21 h22 h23.....h2T    size = HNxT where
%          hs=  .   .   .       .     T is length of time series
%               .   .   .       .     HN is # of obs. points,could be
%          arbitary
%
%       W:  Weight
%
%
%    REQUIRED: m-function, great_circle.m, which is called the distance
%               between two points
%     hs=hindcast(FS,R,LL,Grid,A)
%     [hs,W,RR]=hindcast(fs,r,ll,grid,a)

NH=length(grid(:,1)); % NH is # of hindcast points
NO=length(ll(:,1));   % NO is # of observed points


 for i=1:NH

% calculating the distances bet. obs and hs.points

   for j=1:NO
     [a1 b1]=great_circle(grid(i,1),-grid(i,2),ll(j,1),-ll(j,2));
     Disho(j)=a1;
    end % end j

% estimating the cross_correlation RR between a hindcast point and each
% of observational points using the estimated exponential function

  RR= exp(a.*Disho);
  W=inv(r)*RR'
  T = length(fs(1,:));
  for t=1:T;
    hs(i,t) =sum(W(:).*(fs(:,t)));
  end % t
   hhs(i,:)=mean(fs(1,:))+hs

end                %end i
```

```
*****************************************************************
hindcastw.m

*****************************************************************
function [W]=hindcastw(r,ll,grid,a) %hs=hindcast(R,LL,Grid,A)
%
%  MATLAB UNIX function, hindcastw.m, is used just to calculate
%  the weights for all input nod points
%
%    INPUT---
%
%         R:  Obs.-only cross-correlation matrix with (zero time lag)
%                 c11 c21 c31 c41 c51 c61 c71
%                 c12 c22 c32 c42 c52 c62 c72
%             R=  c13 c23 c33 c42 c53 c63 c73    size=7x7 here
%                 c14 c24 c34 c44 c54 c64 c74    which could be Nx[N];
%                 c15 c25 c35 c45 c55 c65 c75
%                 c16 c26 c36 c46 c56 c66 c76
%                 c17 c27 c37 c47 c57 c67 c77
%        LL:  lat/lon matrix in 7 obs. points in the form of
%                   42.9  68.9  1
%                   44.0  68.1  2
%                   43.8  68.9  3
%                   43.5  70.1  4
%                   43.0  70.6  5
%                   42.4  70.7  6
%                   41.1  66.6  7
%        Grid: lat/lon matrix which has an arbitrary size  in the
%              same form as LL  ( or other form )
%           A: a (or a set of ) parameter(s) corresponding to
%              empirical spatial cross correlation function
%              obtained from the obs. points
%
%    OUTPUT---- W: weights ( whose size = # of obs. points)
%
%    REQUIRED: m-function, great_circle.m, which is called the distance
%               between two points


%hs=hindcast(FS,R,LL,Grid,A)
%[hs,W,RR]=hindcast(fs,r,ll,grid,a)

NH=length(grid(:,1)); % = size of nod's #
NO=length(ll(:,1));   % = size of obs.' #
 for i=1:NH
   for j=1:NO
     [a1 b1]=great_circle(grid(i,1),-grid(i,2),ll(j,1),-ll(j,2));
     Disho(j)=a1; % distance bet. a hc point j
    end % end j

% end calculating the distances bet. obs and hs.points
% estimating the cross_correlation RR( between a hindcast point and
% each observational point
RR= exp(a.*Disho);
  W=inv(r)*RR';
end                      %end i
```


UNHMP-TR-SG-97-31