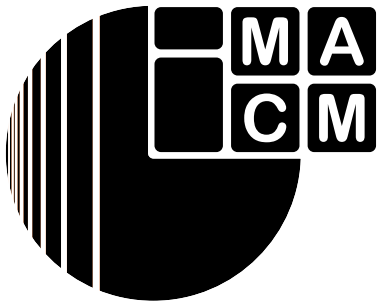

Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 13/13

Ioannis Famelis, Georgios Galanis, Matthias Ehrhardt
and Dimitrios Triantafyllou

Classical and Quasi-Newton methods for a Meteorological Parameters Prediction Boundary Value Problem

July 2013

http://www.math.uni-wuppertal.de

Classical and Quasi-Newton methods for a Meteorological Parameters Prediction Boundary Value Problem

I. Th. Famelis[a], G. Galanis[b], M. Ehrhardt[c], D. Triantafyllou[d,*]

[a] *Department of Mathematics, TEI of Athens, Greece*
[b] *Section of Mathematics, Hellenic Naval Academy, Greece*
[c] *Fachbereich C - Mathematik und Naturwissenschaften Angewandte Mathematik und Numerische Analysis Bergische Universitat Wuppertal, Germany*
[d] *Section of Mathematics, Hellenic Naval Academy, Greece*

Abstract

We study the numerical solution of a system of a quadratic form second order Boundary Value Problem which arises in the numerical prediction of meteorological parameters. In the present work, we use finite differences and focus on the numerical solution of the resulting nonlinear system. More precisely, we apply classical Newton's and Quasi-Newton methods paying attention to the special sparse form of the Jacobian matrix and modify appropriately the LU factorization in order to reduce significantly the required floating point operations. Furthermore, we implement and study in depth the behavior of all the proposed procedures in respect of their accuracy, stability and complexity using data from South East Mediterranean sea. All the methods are tested with a variety of initial values and their performance is presented and discussed leading to interesting results on the sensitivity of the selected starting point.

Keywords: Newton method, Quasi Newton methods, Information Geometry, Boundary Value Problem, Finite differences, Numerical solution of nonlinear systems, sparse matrix

1. The Physical Problem and Information Geometry

In the recent years, the need of accurate local predictions of environmental parameters has increased significantly as a result of the large number of related social and commercial activities e.g. climate change, renewable energy production, transportation, marine pollution, ship safety. This has led to the activation

*Corresponding author
Email addresses: ifamelis@teiath.gr (I. Th. Famelis), galanis@mg.uoa.gr (G. Galanis), ehrhardt@math.uni-wuppertal.de (M. Ehrhardt), dtriant@math.uoa.gr (D. Triantafyllou)

of numerous research and operational centers that develop high quality scientific tools able to provide reliable environmental predictions.

In the study of such problems one may use in site or remote sensing (e.g. satellite) observations or numerical prediction models that solve the basic equations governing the atmospheric and wave evolution based on a numerical analysis approach. Numerical weather and wave forecasting models have been proved successful for the simulation of the general environmental conditions on global or intermediate scale. However, when focusing on local conditions usually systematic errors appear [1, 2, 3, 4, 5]. Such problems are multi-parametric and several different issues are involved such as the strong dependence on the initial and lateral conditions, the inability to capture sub-scale phenomena, the parametrization of certain atmospheric or wave procedures.

To deal with these difficulties one may increase the model resolution, but it is still an open question if this leads to a considerable improvement of the forecast quality. Even if this is true, it also results to an considerable increase of the computational costs.

Alternatively, we can improve the initial conditions based on assimilation systems. Corrected analysis fields are provided to the numerical models based on available observations from meteorological stations, satellites or other sources. Post process algorithms are used for the improvement of the final results based on statistical models (MOS methods, Neural networks, Kalman filters). The latter implements techniques from the non-Euclidean Geometry in Statistics, targeting to the optimization of the solution of nonlinear problems. One of the key issues is the appropriate estimation of the "distance" between two distributions or data sets. The classical treatment of such problems is usually based on regression techniques - least squares methods. However, such an approach carries the assumption that the data processed belongs to an Euclidian - finite dimensional space.

For example, in the case of Kalman filters, the evolution in time of an unknown process x_t is described by the system equation:

$$x_t = F_t \cdot x_{t-1} + w_t.$$

Here a known process y_t is coupled to x_t by the observation equation

$$y_t = H_t \cdot x_t + v_t.$$

The best unbiased linear estimate of the unknown process at time t is a linear combination of the known process and the previous time step values

$$x_t = L_t \cdot x_{t-1} + K_t y_t$$

The filter is based on the minimization of the covariance matrix $E(x_t x_t^\top)$ of x_t.

However, the following question need to be addressed: *is the distance/cost-function correctly estimated by means of classical Euclidean Geometry tools?*

Recently, optimization techniques, in the framework of a relatively new branch of mathematics, the Information Geometry (IG) [6, 7, 8], are employed.

Methods and techniques of non-Euclidean geometry to stochastic processes provide the tools to define a notion of distance between two probability distributions or two data sets. This affects crucially the cost function estimation. IG shows that the use of Eucledian/flat geometry techniques is false in general, and provides a theoretical recipe to avoid such simplifications. Families of probability distributions are recognized as manifolds on which geometrical entities such as Riemannian metrics, distances, curvature and affine connections can be naturally introduced.

More precisely, a n-dimensional statistical manifold is a family of probability distributions

$$S = \{p = p(x;\xi) \mid \xi = [\xi_1, \xi_2, \ldots, \xi_n] \in \Xi\},$$

where each element may be parameterized using n real valued variables in an open subset Ξ of $\mathbb{R}^n$ while the mapping $\xi \to p_\xi$ is injective and smooth. The geometrical framework in a statistical manifold is characterized by the information matrix which at a point ξ is a $n \times n$ matrix $G(\xi) = [g_{ij}(\xi)]$, with elements

$$g_{ij}(\xi) = \mathbb{E}_{x|\xi}\left[\partial_i \ell(x;\xi)\, \partial_j \ell(x;\xi)\right] = \int \partial_i \ell(x;\xi)\, \partial_j \ell(x;\xi)\, p(x;\xi)\, dx,$$

where $\ell(x;\xi) = log\left[p(x;\xi)\right]$ and

$$\mathbb{E}_{x|\xi}[f] = \int f(x) p(x;\xi) dx$$

denotes the expectation with respect to the distribution p.

The matrix $G(\xi)$, called the Fisher information matrix , (see [6, 7, 8]) is symmetric and positive semi-definite. If $G(\xi)$ is positive definite, then a Riemannian metric can be defined on the statistical manifold corresponding to the inner product induced on the natural basis of the coordinate system $[\xi_i]$:

$$g_{ij} = \langle \partial_i \mid \partial_j \rangle$$

This Riemannian metric is called the "*Fisher metric*" or the "*information metric*". The corresponding geometric properties are characterized by the Christoffel symbols $\left(\Gamma^i_{jk}\right)$ of the Levi-Civita connection with respect to the Fisher metric, which are defined solving :

$$\sum_{i=1}^{2} g_{hi}\Gamma^i_{jk} = \Gamma_{jk,h}, \quad (h = 1, 2),$$

where

$$\Gamma_{jk,h}(\xi) = E_\xi\left[\left(\partial_j\partial_k\ell_\xi + \frac{1}{2}\partial_j\ell_\xi\partial_k\ell_\xi\right)(\partial_h\ell_\xi)\right], \quad j,\ k\ = 1, 2, \ldots, n.$$

The minimum distance between two elements f_1 and f_2 of a statistical manifold S is defined by the corresponding *geodesic* ω which is the minimum length curve that connects them. Such a curve

$$\omega = (\omega_i) \quad : \quad \mathbb{R} \to S \tag{1}$$

satisfies the following system of 2^{nd} order differential equations:

$$\omega_i^{''}(t) + \sum_{j,k=1}^{n} \Gamma_{jk}^{i}(t)\,\omega_j^{'}(t)\,\omega_k^{'}(t) = 0, \quad i = 1,\ 2, \ldots,\ n. \tag{2}$$

under the conditions $\omega(0) = f_1,\ \ \omega(1) = f_2$.

In particular, it has been proved (see *e.g.* [9, 10]) that the two parameter Weibull distributions is a good choice for fitting wind and wave data. These distributions form a 2-dimensional statistical manifold with $\xi = [\alpha, \beta],\ \Xi = \{[\alpha, \beta]\ \alpha \text{ and } \beta > 0\}$ (where α is the shape and β the scale parameter) and

$$p(x) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} \mathrm{e}^{-\left(\frac{x}{\beta}\right)^{\alpha}}, \quad \alpha,\ \beta > 0. \tag{3}$$

The Fisher information matrix in this case becomes

$$G(\alpha, \beta) = \begin{bmatrix} \alpha^2\beta^2 & \beta(1-\gamma) \\ \beta(1-\gamma) & \frac{6(\gamma-1)^2+\pi^2}{6\alpha^2} \end{bmatrix},$$

where γ is the Euler gamma.

Let ξ_0=[α_0,β_0] and ξ_1=[α_1,β_1] be two members of the Weibull statistical manifold. Substituting the values of the Christoffel Γ_{jk}^{i} the previous system becomes:

$$\begin{aligned} \omega_1^{''}(t) + \frac{6\left(\gamma\alpha_0 - \alpha_0 - \frac{\pi^2}{6}\right)}{\pi^2\beta_0}\left(\omega_1^{'}(t)\right)^2 + \frac{12\left(\gamma^2 - 2\gamma + \frac{\pi^2}{6} + 1\right)}{\pi^2\alpha_0}\omega_1^{'}(t)\,\omega_2^{'}(t) \quad - \\ \frac{6(1-\gamma)\,\beta_0\left(\gamma^2 - 2\gamma + \frac{\pi^2}{6} + 1\right)}{\pi^2 a^3}\left(\omega_2^{'}(t)\right)^2 \quad = \quad 0 \\ \omega_2^{''}(t) - \frac{\alpha_0^3}{\pi^2\beta_0^2}\left(\omega_1^{'}(t)\right)^2 + \frac{12\alpha_0(1-\gamma)}{\pi^2\beta_0}\omega_1^{'}(t)\,\omega_2^{'}(t) \quad - \\ \frac{6\left(\gamma^2 - 2\gamma + \frac{\pi^2}{6} + 1\right)}{\pi^2\alpha_0}\left(\omega_2^{'}(t)\right)^2 \quad = \quad 0 \end{aligned}$$

with $\omega(0) = \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix},\ \ \omega(1) = \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix}$ where $\omega(t) = \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \end{bmatrix}$.

Our intention is to study the numerical solution of the above problem for a choice of its parameters obtained from environmental data sets within the South East Mediterranean territory.

It is worth noticing that IG techniques have been, directly or not, tested on different applications. Iguzquiza and Olmo [11] utilized some of these ideas for geostatistical simulations for restricted samples. On the other hand, Cai et al. [12] applied information theoretic analysis on self-clustering of amino acids along protein chains. Finally, Resconi [13] based a risk analysis study on non-Euclidean geometric tools. However, up to the authors "knowledge", applications of this framework on meteorology/oceanography is still very rare.

2. The Numerical Solution of the BVP

In this section, we study the numerical solution of the BVP problem

$$\begin{aligned} \omega_1^{''} + a_{11}(\omega_1^{'})^2 + a_{12}\omega_1^{'}\omega_2^{'} + a_{22}(\omega_2^{'})^2 &= 0 \\ \omega_2^{''} + b_{11}(\omega_1^{'})^2 + b_{12}\omega_1^{'}\omega_2^{'} + b_{22}(\omega_2^{'})^2 &= 0 \end{aligned} \tag{4}$$

supplied with boundary conditions

$$\omega_1(0) = \omega_1^0, \quad \omega_2(0) = \omega_2^0, \quad \omega_1(1) = \omega_1^{N+1}, \quad \omega_2(1) = \omega_2^{N+1}.$$

This second order BVP can be written as

$$\widetilde{\omega}^{''} = F(\widetilde{\omega}, \widetilde{\omega}^{'}),$$

where

$$\widetilde{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$$

is defined on the interval $[0, 1]$.

2.1. Finite Differences

We divide $[0, 1]$ into $N + 1$ equal subintervals whose endpoints are at $t_i = 0+ih$, for $i = 0, 1, \ldots, N, N+1$. Assuming that the exact solution has a bounded fourth derivative we discretize and replace $\omega_1^{''}(t_i)$, $\omega_2^{''}(t_i)$, $\omega_1^{'}(t_i)$, $\omega_2^{'}(t_i)$ using the following standard finite differences (FD) :

$$\begin{aligned} \omega_1^{''}(t_i) &= \frac{\omega_1(t_{i+1}) - 2\omega_1(t_i) + \omega_1(t_{i-1})}{h^2} - \frac{h^2}{12}\omega_2^{(4)}(\xi_i) \\ \omega_2^{''}(t_i) &= \frac{\omega_2(t_{i+1}) - 2\omega_2(t_i) + \omega_2(t_{i-1})}{h^2} - \frac{h^2}{12}\omega_2^{(4)}(\xi_i) \\ \omega_1^{'}(t_i) &= \frac{\omega_1(t_{i+1}) - \omega_1(t_{i-1})}{2h} - \frac{h^2}{6}\omega_1^{(3)}(\eta_i) \\ \omega_2^{'}(t_i) &= \frac{\omega_2(t_{i+1}) - \omega_2(t_{i-1})}{2h} - \frac{h^2}{6}\omega_2^{(3)}(\eta_i) \end{aligned}$$

for some ξ_i, η_i in the interval (t_{i-1}, t_{i+1}).

The numerical difference method results from substituting the above FD to the differential equation, neglecting the error terms and employ the boundary conditions:

$$\omega_1(0) = \omega_1^0, \quad \omega_2(0) = \omega_2^0, \quad \omega_1(1) = \omega_1^{N+1}, \quad \omega_2(1) = \omega_2^{N+1},$$

with $\omega_1^i \approx \omega_1(t_i),\ \omega_2^i \approx \omega_2(t_i)$ for $i = 1, \ldots, N$.

The result is a nonlinear system of $2N$ equations with $2N$ unknowns of the form

$$\widehat{F}(\widehat{\omega}) = \mathbf{0}$$

where $\mathbf{0} = [0, \ldots, 0]^\top$ and $\widehat{\omega} = \left[\omega_1^1, \ldots, \omega_1^N, \omega_2^1, \ldots, \omega_2^N\right]^\top$.

This system can be solved using the Newton's iteration

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - J(\widehat{\omega}^{(k-1)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)}), k = 1, 2, \ldots. \tag{5}$$

where $\widehat{\omega}^{(0)}$ is given.

To apply this method we usually transform our problem

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - J^{-1}(\widehat{\omega}^{(k-1)})\widehat{F}(\widehat{\omega}^{(k-1)}) \Leftrightarrow$$
$$\widehat{\omega}^{(k-1)} - \widehat{\omega}^{(k)} = J^{-1}(\widehat{\omega}^{(k-1)})\widehat{F}(\widehat{\omega}^{(k-1)}) \Leftrightarrow$$
$$J(\widehat{\omega}^{(k-1)})(\widehat{\omega}^{(k-1)} - \widehat{\omega}^{(k)}) = \widehat{F}(\widehat{\omega}^{(k-1)}) \Leftrightarrow$$
$$J(\widehat{\omega}^{(k-1)})X = \widehat{F}(\widehat{\omega}^{(k-1)})$$

where

$$X = (\widehat{\omega}^{(k-1)} - \widehat{\omega}^{(k)}).$$

In each step it is more efficient to solve a linear system of the form:

$$J(\widehat{\omega}^{(k-1)}) \cdot X = \widehat{F}(\widehat{\omega}^{(k-1)}).$$

We apply the LU decomposition method to solve the system in each step as follows:

▷ We factorise

$$J(\widehat{\omega}^{(k-1)}) = L \cdot U$$

and solve the system

$$L \cdot U \cdot X = \widehat{F}(\widehat{\omega}^{(k-1)}).$$

▷ We set

$$Y = U \cdot X,$$

solve the lower triangular system

$$L \cdot Y = \widehat{F}(\widehat{\omega}^{(k-1)}),$$

and then the upper triangular system

$$U \cdot X = Y.$$

▷ Finally at each step we update the solution

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - X.$$

2.2. The proposal of the LU modification

In order to take advantage of the special sparse form of the Jacobian matrix J, we interchange its rows as follows.

$$J^* = \begin{pmatrix}
-8 & 4-a_1 & 0 & 0 & \dots & 0 & 0 & 0 & -b_1 & 0 & 0 & \dots & 0 & 0 \\
4+d_2 & -8 & 4-a_2 & 0 & \dots & 0 & 0 & b_2 & 0 & -b_2 & 0 & \dots & 0 & 0 \\
c_2 & 0 & -c_2 & 0 & \dots & 0 & 0 & 4+d_2 & -8 & 4-d_2 & 0 & \dots & 0 & 0 \\
0 & 4+a_3 & -8 & 4-a_3 & \dots & 0 & 0 & 0 & b_3 & 0 & -b_3 & \dots & 0 & 0 \\
0 & -c_1 & 0 & 0 & \dots & 0 & 0 & -8 & 4-d_1 & 0 & 0 & \dots & 0 & 0 \\
0 & c_3 & 0 & -c_3 & \dots & 0 & 0 & 0 & 4+d_3 & -8 & 4-d_3 & \dots & 0 & 0 \\
0 & 0 & 4+a_4 & -8 & \dots & 0 & 0 & 0 & 0 & b_4 & 0 & \dots & 0 & 0 \\
0 & 0 & c_4 & 0 & \dots & 0 & 0 & 0 & 0 & 4+d_4 & -8 & \dots & 0 & 0 \\
0 & 0 & 0 & 4+a_5 & \dots & 0 & 0 & 0 & 0 & 0 & b_5 & \dots & 0 & 0 \\
0 & 0 & 0 & c_5 & \dots & 0 & 0 & 0 & 0 & 0 & 4+d_5 & \dots & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & \dots & . & . \\
0 & 0 & 0 & 0 & \dots & 4+a_n & -8 & 0 & 0 & 0 & \dots & 0 & b_N & 0 \\
0 & 0 & 0 & 0 & \dots & c_n & 0 & 0 & 0 & 0 & \dots & 0 & 4+d_N & -8
\end{pmatrix}$$

Then we can apply a Modified LU factorization reducing significantly the computational cost.

The Modified LU factorization.

1st Step: Make zero only two elements under the main diagonal.
Update only 5 elements in rows 2 and 3 (the 2nd, 3rd, $(N+1)$-th, $(N+2)$-th, $(N+3)$-th).
Right here we have a significant reduction of floating point operations as the classical LU updates the entries of an $(2N-1)\times(2N-1)$ *submatrix.*

2nd Step: Make zero only 4 elements under the main diagonal.
Update only 6 elements in rows 3 to 6 (the 3rd,4*th*, $(N+1)$-th, $\dots$, $(N+4)$-th).

3rd Step: Make zero only 5 elements under the main diagonal.
4rd-8th row: Update only 6 elements in rows 4 to 8 (the 5th,6th, $(N+1)$-th, $\dots$, $(N+6)$-th).

Next Steps Continue similarly until the upper triangularization of J^*.

In every step the number of elements which must be zeroed is increased by 1 until the $N-2$-th step and the number of elements in every row which must be updated is increased by 1 until the (N-4)-th step. Then these numbers are decreased by 1 in every step. The required floating point operations for triangularizing the $2N\times 2N$ Jacobian matrix through modified LU factorization are $O\left(\frac{2N^3}{3}\right)$ whereas the classical one requires $O\left(\frac{8N^3}{3}\right)$ flops. So, the classical Newton's method (5) demands $O(k_0\cdot\frac{8N^3}{3})$ flops when using the classical LU factorization and $O(k_0\cdot\frac{2N^3}{3})$ flops when applying the modified LU to the modified Jacobian matrix, for k_0 iterations.

matrix dim.	classical LU	Modified LU	% of gain
200	0.0176	0.0096	45.6654
500	0.3188	0.0753	76.3882
1000	3.8151	0.7283	80.9093
2000	35.2260	8.7771	75.0834

Table 1: LU vs modified LU factorization

The reduction is achieved in the first half of the factorization, when we update specific entries and not whole submatrices. The second half of the procedure requires the same flops with classical LU. Eventually, the modified LU is 4 times cheaper than the classical one, and the execution time is expected to be even more faster due to the reduction of number of comparisons due to the zero entries of the matrix J^*.

In the next section, we compare the two LU approaches for matrices which have the form of the Jacobian J and random elements. We average the computational time needed for sets 50 matrices. The results are presented in Table 1.

3. Quasi Newton methods

A variety of modifications of the classical Newton method, substituting the Jacobian matrix with other quantities, called Quasi Newton methods can be found in literature [14, 15, 16, 17, 18, 19, 20]. Brezinski [14, 15] classified and proposed theoretically new Quasi-Newton methods. We implemented numerically four of them. Since the Jacobian matrix J^* of our system is of a special form, we adapt these methods to J^* in order to reduce the required floating point operations.

The general scheme of a Quasi Newton method reads

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)}), \quad k = 0, 1, \ldots. \tag{6}$$

where $\Lambda_k \in \mathbb{R}^{2N \times 2N}$.

Brezinski [14] studied the cases that (Λ_k) is the identity matrix multiplied by a scalar, (Λ_k) is a diagonal matrix and (Λ_k) is a full matrix.

3.1. Scalar matrix case (SMC)

In this case $\Lambda_k = \lambda_k \cdot I$, where $\lambda_k = -\frac{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), \widehat{F}(\widehat{\omega}^{(k)}))}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}))}$ [14] and I the identity matrix.

Algorithm Scalar Matrix Case (SMC)

$fd := J(\widehat{\omega}^{(0)}) \cdot \widehat{F}(\widehat{\omega}^{(0)})$
$\lambda_0 = -\frac{(fd, \widehat{F}(\widehat{\omega}^{(0)}))}{(fd, fd)}$
$\Lambda_0 = \lambda_0 \cdot I$

$\widehat{\omega}^{(1)} = \widehat{\omega}^{(0)} + \Lambda_0 \cdot \widehat{F}(\widehat{\omega}^{(0)})^t$
$k = 1$
while $||\widehat{\omega}^{(k)} - \widehat{\omega}^{(k-1)}|| > TOL$ **and** $k < nmax$
$k = k + 1$
$fd := J(\widehat{\omega}^{(k-1)}) \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$
$\lambda_{k-1} = -\frac{(fd, \widehat{F}(\widehat{\omega}^{(k-1)}))}{(fd, fd)}$
$\Lambda_{k-1} = \lambda_{k-1} \cdot I$
$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} + \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})^\top$
end while

In the previous and the following algorithms n_{max} is the maximum number of iterations and TOL is a tolerance parameter.

We consider the required time for the computation of one addition and one multiplication as one flop [21]. In case that the computation includes a single addition or only a single multiplication only a $\frac{1}{2}$ flop will be added to the complexity.

Our problem demands $O(k_1 \cdot 18N)$ flops for k_1 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration. In general, SMC requires $O(k_1 \cdot (4N^2))$ flops for solving an $2N \times 2N$ system of nonlinear equations. The reduction in complexity is due to the special structure of the matrix J^*.

3.2. Diagonal Matrix Case 1 (DMC1)

In this case Λ_k is diagonal [14]. The initial scheme can be written in the form $\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \widetilde{F}(\widehat{\omega}^{(k-1)}) \cdot \widetilde{\Lambda}_{k-1}$, where

$$\widetilde{F}(\widehat{\omega}^{(k)}) = diag(\widetilde{F}_1(\widehat{\omega}^{(k)}), \widetilde{F}_2(\widehat{\omega}^{(k)}), \ldots, \widetilde{F}_{2N}(\widehat{\omega}^{(k)})),$$

$\widetilde{\Lambda}_k = (\lambda_k^1, \lambda_k^2, \ldots, \lambda_k^{2N})^\top$ and $diag(x)$ is a diagonal matrix with diagonal entries the elements of vector x. Thus, this formula can be considered as Newton method with a diagonal preconditioner (see [14] for more details).

Algorithm Diagonal Matrix Case(DMC1)

$\lambda_1 = -diag(\frac{1}{\widetilde{F}_1(\widehat{\omega}^{(0)})}, \frac{1}{\widetilde{F}_2(\widehat{\omega}^{(0)})}, \ldots, \frac{1}{\widetilde{F}_{2N}(\widehat{\omega}^{(0)})}) \cdot J(\widehat{\omega}^{(0)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(0)})$
$\Lambda_1 = diag(\lambda_1)$
$\widehat{\omega}^{(1)} = \widehat{\omega}^{(0)} + \Lambda_0 \cdot (\widehat{F}(\widehat{\omega}^{(0)}))^\top$
while $||\widehat{\omega}^{(k+1)} - \widehat{\omega}^{(k)}|| > TOL$ **and** $k < nmax$
$k = k + 1$
$\lambda_{k-1} = -diag(\frac{1}{\widetilde{F}_1(\widehat{\omega}^{(k-1)})}, \frac{1}{\widetilde{F}_2(\widehat{\omega}^{(k-1)})}, \ldots, \frac{1}{\widetilde{F}_{2N}(\widehat{\omega}^{(k-1)})}) \cdot J(\widehat{\omega}^{(k-1)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$
$\Lambda_{k-1} = diag(\lambda_{k-1})$
$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} + \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})^\top$
end while

We use the modified LU factorization in order to compute $J(\widehat{\omega}^{(k-1)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$ reducing significantly the required flops. The required flops are $O(k_2 \cdot \frac{2N^3}{3})$ flops for k_2 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration. In general, the complexity for solving an $2N \times 2N$ system is of order $O(k_2 \cdot (\frac{8N^3}{3}))$.

3.3. Diagonal Matrix Case 2 (DMC2)

This is a modification of the previous case,

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \widetilde{F}(\widehat{\omega}^{(k-1)}) \cdot \widetilde{\Lambda}_{k-1},$$

where $\widetilde{\Lambda}_k$ is computed using forward differences. Thus, $\widetilde{\Lambda}_k = \Delta\widetilde{F}(\widehat{\omega}^{(k-1)})^{-1} \cdot \Delta\widehat{\omega}^{(k-1)}$.

The DMC2 algorithm requires significant less flops than DMC1. More precisely, DMC2 demands $O(k_2 \cdot 8N)$ flops for k_2 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration.

3.4. Full matrix case (FMC)

Now, the matrix Λ_k is a full matrix with $\Lambda_k = -\frac{\widehat{F}(\widehat{\omega}^{(k)} \cdot (\widehat{F}(\widehat{\omega}^{(k)})^\top \cdot (J(\widehat{\omega}^{(k)})^\top}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)})}$. It follows the FMC algorithm.

Algorithm Full Matrix Case(FMC)

$fd := J(\widehat{\omega}^{(0)}) \cdot \widehat{F}(\widehat{\omega}^{(0)})$
$\Lambda_0 = \frac{-\widehat{F}(\widehat{\omega}^{(0)}) \cdot fd^\top}{(fd, fd)}$
$\widehat{\omega}^{(1)} = \widehat{\omega}^{(0)} + \Lambda_0 \cdot (\widehat{F}(\widehat{\omega}^{(0)}))^\top$
while $||\widehat{\omega}^{(k+1)} - \widehat{\omega}^{(k)}|| > TOL$ **and** $k < nmax$
    $k = k + 1$
    $fd := J(\widehat{\omega}^{(k-1)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$
    $\Lambda_{k-1} = \frac{-\widehat{F}(\widehat{\omega}^{(k-1)}) \cdot fd^\top}{(fd, fd)}$
    $\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} + \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})^\top$
end while

The FMC algorithm demands $O(k_3 \cdot 9N^2)$ flops at every iteration for k_3 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration. In general, FMC requires $O(k_3 \cdot (13N^2))$ flops for solving an $2N \times 2N$ system of nonlinear equations.

4. Numerical Tests and Observations

For our numerical tests we choose data from the area of Levantive in the eastern Mediterranean Sea. For every month of a year we have modeled wind speed and wave height data following two approaches. The former includes in

	model data no current		model data with current		satellite data	
Weibull	shape α_0	scale β_0	shape α_0	scale β_0	shape α_1	scale β_1
Jan	1.600	1.010	1.726	1.095	2.523	1.441
Feb	1.500	1.400	1.571	1.464	2.450	1.762
Mar	1.462	1.132	1.578	1.225	2.560	1.509
Apr	1.564	0.695	1.719	0.754	2.140	1.012
May	1.533	0.608	1.608	0.661	1.576	0.780
Jun	2.333	0.633	2.542	0.680	3.759	0.759
Jul	2.557	0.837	2.688	0.876	3.515	0.960
Aug	3.099	0.716	3.341	0.759	4.938	0.889
Sep	2.418	0.754	2.580	0.800	3.491	0.968
Oct	1.629	0.551	1.850	0.609	2.204	0.665
Nov	1.446	0.892	1.499	0.919	1.911	1.224
Dec	1.435	1.216	1.512	1.283	2.208	1.442

Table 2: Levantine problems parameters

the simulation the first includes in the simulation the impact of see currents, while the second does not. Second source of data is the available corresponding sattelite data. The data are fitted by a 2-parameter Weibull distribution and their Weibull parameters are given in Table 2. When we consider the minimum length curve which connects the each modeled and its corresponding satellite data we conclude in 24 BVP problems. For instance for month January we have two problems, one for the curve connecting the modeled data with the presence of current and the satellite data, which we call "problem Jan with current", and one for the modeled data without current and the satellite data, which we call "problem Jan with no current".

To get reference solutions we use NDSolve of Mathematica to solve the 24 test problems. Mathematica uses the shooting method and one can set appropriate accuracy options (Working Precision, Accuracy Goal, Accuracy Goal) to get an considerably accurate solution which is in a "continuous" interpolating form. The computed solution can be substituted in the test differential equations for an abscissae on $[0, 1]$ of a desired width (*e.g.* 10^{-5}) and record the maximum residual error, the maximum defect as it is usually called. The size of the defect gives a different measure of the suitability of the approximate solution; it is the amount by which the computed solution fails to satisfy the system of differential equations. It has been suggested that monitoring the defect may be appropriate in situations where difficulties arise in estimating the global error [22].

Such solutions can be used as highly accurate reference solutions for the comparison to the other numerical methods which attain a significantly lower precision. In Figure 1, the reference solution of the problem Jun with current is presented and in Figure 2, the defect for the reference solution of the same problem which attains maximum value 1.11×10^{-15}. Moreover in Figure 3, the reference solution of the problem Aug with current is shown and in Figure 4, the defect for the reference solution of the same problem which attains a maximum value 6.25×10^{-13}. For this problem (along with a few other) the numerical integrator suggests elements of stiffness in the solution.

For our numerical tests we calculate and program the analytical form of F and Jacobian J. We choose $N = 100$ and get a system of 200 equations. For the 24 problems we produce a reference solution and for an initial guess we

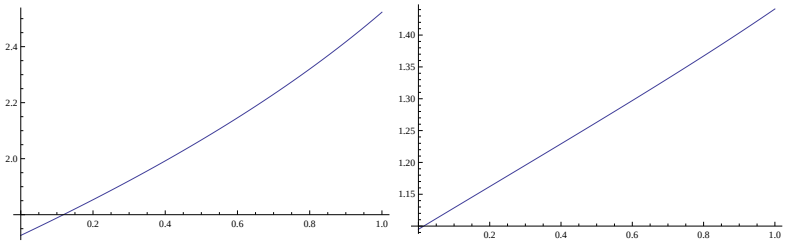

Figure 1: The reference solution $\widehat{\omega}_1$ and $\widehat{\omega}_2$ for the problem Jun with current.

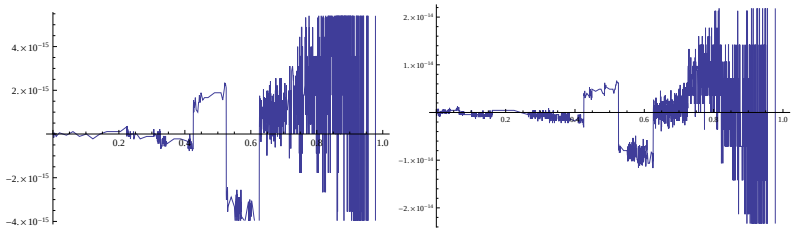

Figure 2: The defect for the reference solution $\widehat{\omega}_1$ and $\widehat{\omega}_2$ for the problem Jun with current.

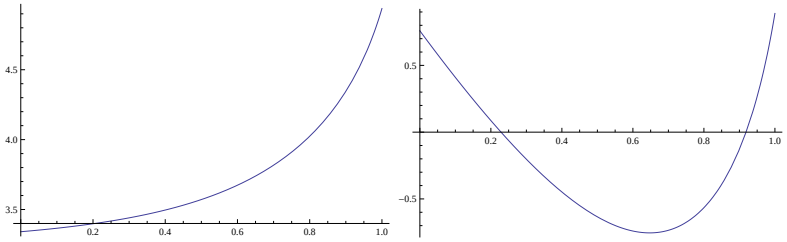

Figure 3: The reference solution $\widehat{\omega}_1$ and $\widehat{\omega}_2$ for the problem Aug with current.

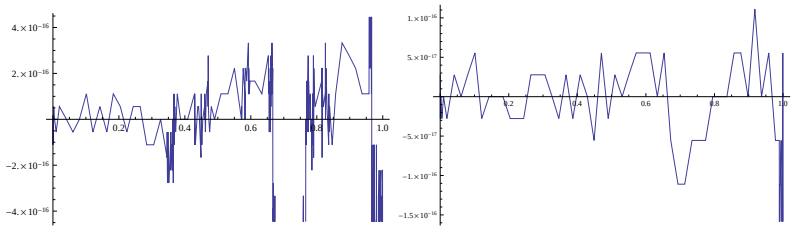

Figure 4: The defect for the reference solution $\widehat{\omega}_1$ and $\widehat{\omega}_2$ for the problem Aug with current.

use a perturbation of the initial data with random numbers. Then, we solve numerically the 24 test problems using the proposed methods for tolerances $10^{-3}, 10^{-4}, \ldots, 10^{-13}$ to compare efficiency and computational costs. We use two error measures at the grid points. The first one is $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ the maximum absolute value that the numerical solution fails to satisfy the nonlinear problem resulting from the finite difference method. The second one is the $\|\widehat{\omega}_{so} - \widehat{\omega}_{ref}\|_\infty$ maximum absolute value of the difference of the numerical solution and the reference solution. We investigate the sensitivity in the choice of the initial guess for tolerances $10^{-3}, 10^{-4}, \ldots, 10^{-13}$ with respect to its distance from the reference solution, in order to evaluate the range of convergence for each method.

We first solve the 24 problems using Newton's Method with both the classical implementation of the LU factorization and the proposed modification and average the various measures. In Table 3, it is shown that both Newton's method with classical LU and Newton's method with modified LU have the same iterations and similar error measures at the grid points for all 24 problems. In some problems the iteration diverges for both approaches (e.g. problem Aug with Current). The superiority of the proposed modified LU algorithm is obvious in Table 4, where we compare the average time to solve each of the 24 problem for 50 different choices of initial conditions using Newton's method with classical LU and Newton's method with modified LU.

	Average							
	no of iter.		time in secs		$\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$		$\|\widehat{\omega}_{sol} - \widehat{\omega}_{ref}\|_\infty$	
TOL	clas.	mod.	clas.	mod.	clas.	mod.	clas.	mod.
10^{-8}	8.33	8.04	0.1529	0.099	0.355e-10	0.176e-10	0.605e-5	0.605e-5
10^{-10}	8.7	8.67	0.1731	0.1196	0.080e-12	0.173e-12	0.123e-4	0.123e-4

Table 3: Comparing the Classical LU and the proposed modified LU factorization.

	Average time in secs		
TOL	clas.	mod.	% of gain
10^{-8}	0.199	0.135	32.70
10^{-9}	0.209	0.143	31.53
10^{-10}	0.217	0.147	32.05
10^{-11}	0.224	0.151	32.57
10^{-12}	0.362	0.169	40.44
10^{-13}	0.952	0.088	56.41

Table 4: Time comparisons for the Classical LU and the proposed modified LU.

In the following some remarks for the comparison of mod-NR, SMC, DMC1, DMC2, FMC are discussed. We solve for tolerances $10^{-3}, \ldots, 10^{-12}$ the 24 problems for a common initial condition each time and compare the average values of the results. It seems that DMC1 fails as the preconditioning matrix is singular. DMC2 succeeds only for $TOL = 10^{-3}, 10^{-4}$. For smaller tolerances it fails as in $\widetilde{\Lambda}_k = \Delta\widetilde{F}(\widehat{\omega}^{(k-1)})^{-1} \cdot \Delta\widehat{\omega}^{(k-1)}$ the denominator becomes less than than unit round of (machine accuracy ϵ). SMC and FMC algorithms work for $TOL = 10^{-3}, \ldots, 10^{-8}$. For smaller tolerances, even if the methods do not seem

to diverge, the iteration stops as the denominator of

$$\frac{\#}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)})} \tag{7}$$

becomes less than machine accuracy ϵ.

In Tables 5 and 6, the modified Newton, SMC and FMC methods are compared for various tolerances in respect of the number of iterations and and the required running time. No mater the theoretical complexity cost, the time needed for the solution using Newton's method is considerably smaller. SMC takes the longer time over the three methods. In Tables 7 and 8 the average recorded error measures for the modified Newton, SMC and FMC methods are presented. Newtons method attains a better convergence (with respect to the reference solution) and has a smaller defect for the nonlinear system compared to its competitors.

	average number of iterations		
TOL	Newton's method	SMC	FMC
10^{-3}	7.4	485	95
10^{-4}	7.6	3460	835
10^{-5}	8.4	6853	2959
10^{-6}	8.6	10057	7363
10^{-7}	9.2	13429	10596
10^{-8}	9.6	16296	13632

Table 5: Mean number of iterations for problems with convergence.

	average time in secs		
TOL	Newton's method	SMC	FMC
10^{-3}	0.098	0.166	0.443
10^{-4}	0.098	4.062	1.070
10^{-5}	0.112	14.684	8.063
10^{-6}	0.117	35.450	24.140
10^{-7}	0.126	56.277	44.580
10^{-8}	0.123	85.410	71.087

Table 6: Mean time in secs for problems which converge.

	Average $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$		
TOL	Newton's method	SMC	FMC
10^{-3}	1.55e-6	1.83e-3	9.92e-3
10^{-4}	6.77e-8	1.45e-4	1.09e-3
10^{-5}	7.29e-9	1.53e-5	1.13e-4
10^{-6}	1.93e-9	1.48e-6	1.11e-5
10^{-7}	8.24e-11	1.62e-7	1.19e-6
10^{-8}	2.82e-11	1.71e-8	1.20e-7

Table 7: Mean $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ for problems which converge.

An interesting remark is that Newton's method with either classical and modified LU factorization seem to be very sensitive in the choice of initial guess and so they have shorter interval of convergence compare to both SMC, FMC. This can be tested by solving the problems taking initial guesses further away

from the reference solution (see Tables 9, 10, 11). It must be noted that SMC and FMC do not diverge for $TOL = 10^{-9}$ and smaller (see Tables , 12, 13). The algorithms stop as the denominator of (7) becomes less than machine accuracy ϵ. Even though Newton's method is more accurate, SMC and FMC can be used as starting procedures. For problems such as Aug with current, where Newtons method diverges, we can use either SMC or FMC to get an initial point and then continue with Newton's iteration (see Table 14).

5. Conclusions

In this paper we study the numerical solution of a special boundary value problem, arising in meteorological parameters prediction. The construction and study of the cost functions that estimate the discrepancies between modeled and observed data is based on techniques developed within the framework of Information Geometry. In this way, we adopt an approach that avoids simplifications connected with the a priori acceptance of Euclidean distances for the data under study as in the least square based methods. More precisely, Weibull probability distribution functions are fitted to the data under study and treated as elements of statistical manifolds on which Riemannian metrics are defined and the distances between the different sets of data are measured by means of the corresponding geodesics (i.e. minimum length curves).

These curves are obtained as numerical solutions of second order boundary value problems using finite differences. For the numerical solution of the resulting system of nonlinear equations, we apply Newton and Quasi Newton methods in which we take into account the special form of the Jacobian matrix and modify appropriately the LU algorithm in order to reduce the computational complexity.

More precisely, the Newton method using the modified LU factorization demands the same iterations and has an absolute error of the same order with the classical one but it requires significantly less floating point operations. The error measures for Newton's method (classical or modified) are considerable smaller and the method converges for smaller tolerances compared to the SMC and FMC.

The DMC methods are not proposed for our problem, since the preconditioning matrix is almost singular.

The Newton's method using either classical or modified LU factorization is more sensitive in the selection of the initial point compared to SMC and FMC since, for some problems, it demands the initial values to be closer to the final solution than the other two methods.

The use of SMC or FMC as an initial procedure improves the behavior of Newtons method in the cases that diverges. These methods can be used as starting procedures that compute initial points for the Newton's iteration within the interval of convergence, while the modified LU factorization reduces significantly the required floating point operations resulting to an effective algorithm for solving efficiently the system of non linear equations.

	Average $\|\widehat{\omega}_{sol} - \widehat{\omega}_{ref}\|_\infty$		
TOL	Newton's method	SMC	FMC
10^{-3}	2.87e-6	2.53e-1	4.17e-1
10^{-4}	1.59e-5	3.77e-2	2.05e-1
10^{-5}	3.49e-6	3.96e-3	2.96e-2
10^{-6}	1.56e-5	3.94e-4	2.88e-3
10^{-7}	2.53e-5	6.61e-5	3.29e-4
10^{-8}	3.57e-6	7.14e-6	3.45e-5

Table 8: Mean $\|\widehat{\omega}_{sol} - \widehat{\omega}_{ref}\|_\infty$ for problems which converge.

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	24	24	21	3
10^{-4}	24	24	22	1
10^{-5}	24	24	23	2
10^{-6}	24	24	21	2
10^{-7}	24	24	23	2
10^{-8}	24	24	21	1
10^{-9}	24	24	21	1
10^{-10}	24	24	19	2
10^{-11}	24	24	23	3
10^{-12}	24	24	23	3
10^{-13}	24	22	19	3

Table 9: Newton's method sensitivity in initial condition choice. Number of convergent solution of problems (out of 24).

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	24	24	24	24
10^{-4}	24	24	24	24
10^{-5}	24	24	24	24
10^{-6}	24	24	24	24
10^{-7}	24	24	24	24
10^{-8}	24	24	24	24
10^{-9}	12	13	13	14

Table 10: SMC sensitivity in initial condition choice. Number of convergent solution of problems (out of 24).

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	24	24	24	24
10^{-4}	24	24	24	24
10^{-5}	24	24	24	24
10^{-6}	24	24	24	24
10^{-7}	24	24	24	24
10^{-8}	24	24	24	24
10^{-8}	15	14	10	14

Table 11: FMC sensitivity in initial condition choice. Number of convergent solution of problems (out of 24).

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	1.78e-3	1.61e-3	1.32e-3	9.40e-3
10^{-4}	1.00e-4	1.00e-4	1.08e-4	1.11e-4
10^{-5}	1.10e-5	1.10e-5	1.10e-5	1.11e-5
10^{-6}	1.20e-6	1.19e-6	1.22e-6	1.21e-6
10^{-7}	1.27e-7	1.07e-7	1.27e-6	1.26e-7
10^{-8}	1.40e-8	1.40e-8	1.41e-8	1.34e-8
10^{-9}	1.32e-8	1.20e-8	1.30e-8	1.25e-8

Table 12: SMC average $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ for all 24 problems.

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	7.14e-3	7.80e-3	8.16e-3	8.27e-3
10^{-4}	8.46e-4	8.31e-4	8.46e-4	8.78e-4
10^{-5}	8.99e-5	9.49e-5	9.62e-5	9.51e-5
10^{-6}	1.02e-5	1.03e-5	1.04e-5	1.04e-5
10^{-7}	1.08e-6	1.07e-6	1.09e-6	1.08e-6
10^{-8}	1.09e-7	1.11e-7	1.56e-7	1.15e-7
10^{-9}	1.30e-8	1.34e-8	1.42e-8	1.31e-8

Table 13: FMC average $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ for all 24 problems.

$TOL = 10^{-11}$			
	no of iter.	$\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$	$\|\widehat{\omega}_{sol} - \widehat{\omega}_{ref}\|_\infty$
Newton's methodR	182	4.38e+177	9.03e+88
SMC	9219	1.75e-6	7.51e-4
Newton's methodR	7	4.78e-14	3.74e-4
FMC	5602	1.39e-5	3.40e-3
Newton's methodR	7	5.59e-14	3.74e-4

Table 14: SMC and FMC as starting procedures.

Acknowledgements The authors would like to thank the "The Atmospheric Modeling and Weather Forecasting Group" of the Department of Physics, University of Athens, Greece for providing the meteorological data.

This work has been co-funded by the European Union and Greek national resources under the framework of the "Archimedes III: Funding of Research Groups in TEI of Athens" project of the "Education and Lifelong Learning" Operational Programme.

References

[1] G. Emmanouil, G. Galanis, G. Kallos, Statistical methods for the prediction of night time cooling and minimum temperature, Meteorol. Appl. 13 (2006) 169–178.

[2] G. Emmanouil, G. Galanis, G. Kallos, L. Breivik, H. Heilberg, M. Reistad, Assimilation of radar altimeter data in numerical wave models: An impact study in two different wave climate regions, Ann. Geoph. 25 (2007) 581–595.

[3] G. Galanis, M. Anadranistakis, A one dimensional kalman filter for the correction of near surface temperature forecasts, Meteor. Appl. 9 (2002) 437–441.

[4] G. Galanis, G. Emmanouil, P. Chu, G. Kallos, A new methodology for the extension of the impact in sea wave assimilation systems, Ocean Dynamics 59 (3) (2009) 523 – 535.

[5] G. Galanis, P. Louka, P. Katsafados, I. Pytharoulis, G. Kallos, Applications of kalman filters based on non-linear functions to numerical weather predictions, Ann. Geophys. 24 (2006) 2451 – 2460.

[6] S. I. Amari, H. Nagaoka, Methods of Information Geometry, American Mathematical Society, Oxford University Press, Oxford, 2000.

[7] S. Amari, Differential-geometrical methods in statistics, Lecture notes in statistics 28, Springer-Verlag, Berlin, 1985.

[8] K. Arwini, C. Dodson, Information Geometry: Near Randomness and Near Independence, Lecture Notes in Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, 2008.

[9] G. Galanis, P. C. Chu, G. Kallos, Y.-H. Kuo, C. Dodson, Wave height characteristics in the north atlantic ocean: a new approach based on statistical and geometrical techniques, Stoch Environ Res Risk Assess 26 (2012) 83 – 103.

[10] G. Galanis, D. Hayes, G. Zodiatis, P. C. Chu, Y.-H. Kuo, G. Kallos, Wave height characteristics in the mediterranean sea by means of numerical modeling, satellite data, statistical and geometrical techniques, Marine Geophysical Research 33 (2012) 1 – 15.

[11] E. Iguzquiza, M. Chica-Olmo, Geostatistical simulation when the number of experimental data is small: an alternative paradigm, Stoch. Env. Res. Risk Assess. 22 (2008) 325 – 337.

[12] Y. Cai, C. Dodson, A. Doig, O. Wolkenhauer, Information-theoretic analysis of protein sequences shows that amino acids self-cluster, J. Theor. Biology 218 (4) (2002) 409 – 418.

[13] G. Resconi, Geometry of risk analysis (morphogenetic system), Stoch. Env. Res. Risk Assess. 23 (4) (2009) 425 – 432.

[14] C. Brezinski, Projection Methods for Systems of Equations, North-Holland, Holland, 1997.

[15] C. Brezinski, A classification of quasi-newton methods, Numer, Algor. 33 (1997) 123 – 135.

[16] J. E. Dennis, Jr., J. J. More, A characterization of superlinear convergence and its application to quasi-newton methods, Math. Comp. 28 (126) (1974) 549–560.

[17] D. Zhu, Nonmonotone backtracking inexact quasi-newton algorithms for solving smooth nonlinear equations, Appl. Math. Comput. 161 (3) (2005) 875 – 895.

[18] S. Fangyu, L. Xiangfang, On an accelerating quasi-newton circular iteration, Appl. Math. Comput. 106 (1) (1999) 17 – 29.

[19] W. Cruz, M. Raydan, Nonmonotone spectral methods for large-scale nonlinear systems, Optimization Methods and Software 18 (5) (2003) 583 – 599.

[20] W. Zengxin, L. Guoyi, Q. Liqun, New quasi-newton methods for unconstrained optimization problems, Appl. Math. Comput. 175 (2) (2006) 1156 – 1188.

[21] B. N. Datta, Numerical Linear Algebra and Applications, 2nd Edition, SIAM, USA, 2010.

[22] U. Ascher, R. M. M. Mattheij, R. D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Prentice Hall, Englewood Cliffs, New Jersey, 1988.