MODELING ROGUE WAVES IN FULLY DIRECTIONAL SEA STATES: HYPERFAST SIMULATIONS OF THE BOUSSINESQ EQUATIONS IN THE COASTAL ZONE

A. R. Osborne and M. Onorato

Dipartimento di Fisica Generale dell'Università
Via Pietro Giuria 1, Torino 10125, Italy

Donald T. Resio and Charles E. Long

U. S. Army Engineer Research and Development Center
Coastal and Hydraulics Laboratory, 3909 Halls Ferry Road
Vicksburg, MS 39180-6199, U.S.A.

10th International Workshop on Wave Hindcasting and Forecasting
&
Coastal Hazard Assessment

Oahu, Hawaii
11-16 November 2007

MODELING ROGUE WAVES IN FULLY DIRECTIONAL SEA STATES: HYPERFAST SIMULATIONS OF THE BOUSSINESQ EQUATIONS IN THE COASTAL ZONE

A. R. Osborne and M. Onorato

Dipartimento di Fisica Generale dell'Università
Via Pietro Giuria 1, Torino 10125, Italy

Donald T. Resio and Charles E. Long

U. S. Army Engineer Research and Development Center
Coastal and Hydraulics Laboratory, 3909 Halls Ferry Road
Vicksburg, MS 39180-6199, U.S.A.

1. INTRODUCTION

We study a new type of oceanic rogue wave which arises in the theory and modeling of directional sea states in shallow water coastal zones. In contrast to deep-water rogue waves, which are caused by the Benjamin-Feir instability, the shallow water rogue waves discussed herein do not arise from any instability: Instead they occur because of nonlinear interactions among cnoidal (Stokes) waves in the nonlinear directional spectrum. The basic nonlinearity is that of a *Mach stem*, a phenomenon identified by John Miles [Miles, 1977]. We show that the appearance of the Mach stem (and its much less nonlinear counter part) is a natural consequence of the spectral nature of directionally spread waves in shallow water. We give several examples of shallow water rogue waves.

The formulation begins with that of the Kadomtsev-Petviashvili (KP) equation [Novikov, et al 1980; Ablowitz and Segur, 1981] and its nonlinear spectral representation in terms of multi-dimensional Fourier series [Belokolos, et al, 1994]. We then show that the numerical spectral method for KP can be rendered "almost linear" by this spectral approach. This perspective provides a numerical model for KP which is about three orders of magnitude faster than the traditional leap-frog FFT spectral model. We refer to the algorithm as "hyperfast numerical integration" of the KP equation. We give an overview of this new approach and discuss how to extend it to the order of the Boussinesq equations. A number of directionally spread numerical simulations of the method are discussed. The appearance of the new type of rogue wave is invariably found in these simulations and we discuss their prediction using the new algorithms given herein.

We first discuss (Section 2) the numerical integration of *linear* partial differential wave equations using the fast Fourier transform. We show a novel way to program such a model which is then quite useful in the conception of a new class of nonlinear models (Section 3). Section 4 discusses the KP equation and its numerical model. A simple extension to a leading order Boussinesq equation is also discussed. Implementation of the new nonlinear model is discussed in Section 5 and some numerical results are given in Section 6. Finally in Section 7 we give some physical perspective about the formation of shallow water rogue waves. Section 8 is a summary and discussion of the results.

2. LINEAR NUMERICAL MODELING

In this Section we discuss the modeling of *linear partial differential equations*. The procedure is based upon the fast Fourier transform (FFT) algorithm. One way to program this approach is shown in the flow chart of Fig. 1. Since we are dealing with a linear wave equation its solution is a *linear superposition of directionally spread sine waves*. Each wave has direction described by a wave number pair $(k_x, k_y) = (k_m, l_n)$ where the indices vary in the usual way $m, n = -N/2 ... -2, -1, 0, 1, 2 ... N/2$. Here N is the number of points on each side of the (x, y) domain, for example in typical numerical simulations one might take $N = 128$ or 512 points. From Fig. 1 we see that the inputs to the algorithm are the total time of the simulation, T, the wave number pair, $(k_x, k_y) = (k_m, l_n)$, the linear Fourier amplitudes, A_{mn}, the Fourier phases, ϕ_{mn} (often chosen to be random), and the frequencies, $\omega_{mn} = \omega_{mn}(k_m, l_n)$ (this is the linear dispersion relation). As seen from the flow chart the Fourier transform for each value of time, t, is then computed. A film of 500 or 1000 frames or time values is a typical output. In the last box of Fig. 1 we take the FFT of the 1000 Fourier transforms to get the wave surface elevation at the 1000 time points; then the film is made from these 1000 surfaces.

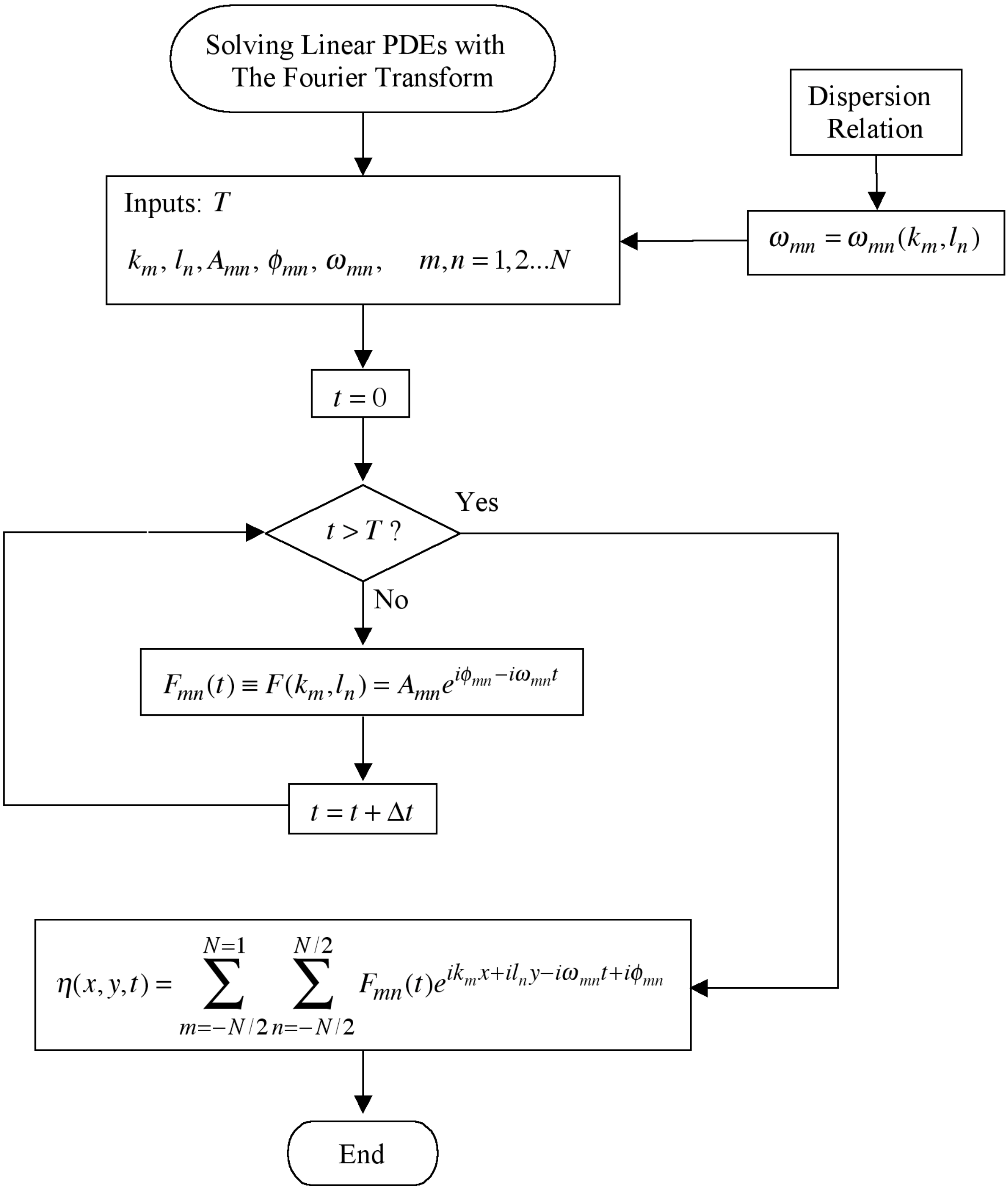

Figure. 1. Simple flow chart of a numerical model for a ***linear partial differential equation*** with a well-defined dispersion relation.

There are a number of important points to learn from Fig. 1. There are two steps in the algorithm. The first step, call it a *preprocessor* (red box of Fig. 1), computes the Fourier transform of the wave field for all values of time. The second step, call it the *time evolution of the wave field*, computes the wave field for the values of time using the inverse FFT. So, as we have seen above the preprocessor step is trivial and therefore very fast. The time evolution step is also very fast because one is computing, say, 1000 FFTs and on a modern desktop computer this step takes only a few seconds at most. Note that the second step could just as well be placed inside the preprocessor do-loop. This is not the case for the nonlinear problem as discussed below. Please note that the algorithm in the flow chart of Fig. 1 has been designed to be compatible with the nonlinear problem discussed in the next section.

We see that a fast preprocessor step and ~1000 calls of the FFT are required for the linear problem. Can we ever hope to solve nonlinear wave equations at the speed of the linear problem? Probably not, but the similarities between the two problems can be made evident as discussed in the next section and the speed savings can be quite dramatic with respect to conventional FFT simulations.

3. NONLINEAR MODELING

The nonlinear modeling project that we are discussing herein is quite new. A first attempt at programming the algorithm was made early this year in February and March. However, the algorithm discussed herein has been completely redesigned and reprogrammed in the last couple of months! And just explaining what is being done is still quite a challenge. The algorithm, remarkably, has an overall structure very similar to the linear approach discussed above (see Fig. 2). Basically this means that the *first step* consists of a preprocessor part which determines the *nonlinear time evolution* of the linear Fourier spectrum, i.e. it computes the linear Fourier spectrum with nonlinear interactions among the components at some large number (say $N \sim 1000$) of desired time steps (once again to make a movie for studying the nonlinear wave behavior). The *second step* is to take the inverse FFT of the time varying Fourier transforms. Both steps parallel the linear problem as in Fig. 1. However, for the nonlinear problem the preprocessor is much more complex and time consuming. Additionally the mathematics (soliton theory, Riemann theta functions and algebraic geometry) is not that ordinarily studied by physical oceanographers and this may be an impediment that could test the patience of those who might want to apply the method. As we shall see however, the speed gains are quite dramatic, i.e. two or three orders of magnitude faster than the direct numerical integration of nonlinear wave equations by the FFT.

As with the linear problem the nonlinear method proposed here for *integrable nonlinear partial differential equations* is also exact, i.e. there is no degradation of the numerical accuracy with increasing time. This is because time is a parameter in the theory and one is simply evaluating the solution at each value of time. As mentioned above the linear algorithm could be easily modified to place the inverse FFTs inside the preprocessor itself. On the contrary the nonlinear algorithm cannot be so easily modified. It is for this reason that we give Figs. 1 and 2 in the form shown, i.e. with the pre- and post-processing maintained separately.

Let us now briefly return to Fig. 2. The preprocessor is unique for each ***integrable*** *wave equation* as discussed in detail in a later section. Furthermore, for each ***nonintegrable*** *equation* a sub-processor must be added as shown in the blue box of Fig. 2. This detail is also discussed below.

Finally it is worthwhile noting that the numerical integration of nonlinear wave equations depends primarily upon the preprocessor and its design and implementation. The development of the preprocessor is therefore the most active part of present research connected with the method proposed herein.

4. NONLINEAR INTEGRABLE MODEL WAVE EQUATIONS

There exist large classes of nonlinear partial differential wave equations that are completely solvable by the inverse scattering transform (Ablowitz and Segur [1981]; Whitham [1974]). One of the most important of these equations is the nonlinear Kortweg-deVries (KdV) equation in one space (x) and one time (t) (1+1) dimensions:

$$\eta_t + c_o\eta_x + \alpha\eta\eta_x + \beta\eta_{xxx} = 0 \tag{1}$$

Here $\eta(x,t)$ is the wave amplitude as a function of space and time. The constants are given by $c_o = \sqrt{gh}$, $\alpha = 3c_o / 2h$ and $\beta = c_o h^2 / 6$; h is the water depth and g is the acceleration of gravity. The solution to the KdV equation is given by the following expression:

$$\eta(x,t) = \frac{2}{\lambda}\frac{\partial^2}{\partial x^2}\ln\theta(x,t\,|\,\mathbf{B},\mathbf{k},\boldsymbol{\omega},\boldsymbol{\phi}) \tag{2}$$

where $\lambda = \alpha / 6\beta$. The generalized Fourier series, $\theta(x,t\,|\,\mathbf{B},\mathbf{k},\boldsymbol{\omega},\boldsymbol{\phi})$, has the form:

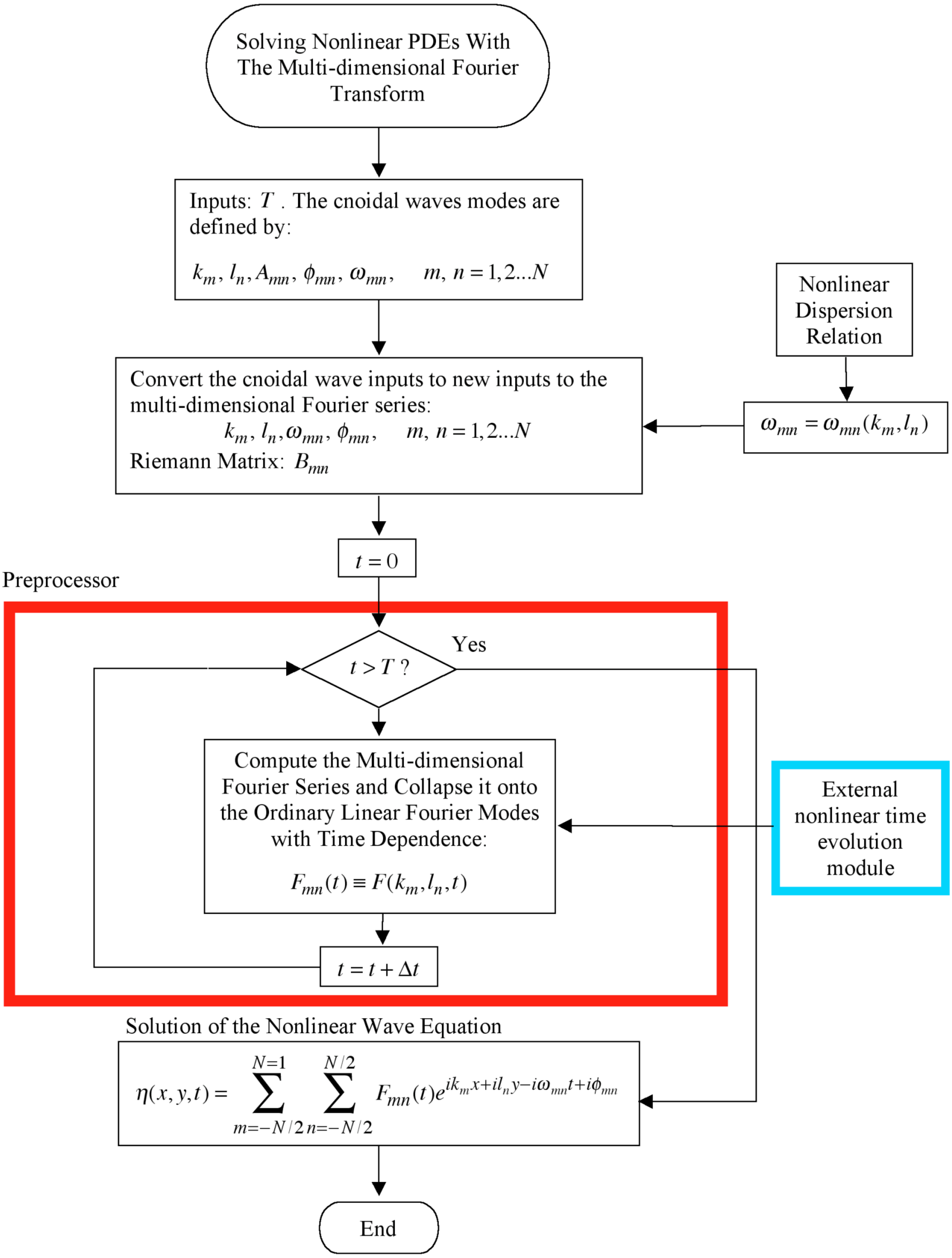

Figure. 2. Simple flow chart of the numerical model for *integrable and nonintegrable nonlinear partial differential equations* with a well-defined dispersion relation.

$$\theta\left(X_1, X_2, \ldots X_N\right) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \cdots \sum_{m_N=-\infty}^{\infty} e^{i\sum_{n=1}^{N} m_n X_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} m_m m_n B_{mn}} \tag{3}$$

where $X_n = k_n x - \omega_n t + \phi_n$. The *inverse problem* associated with (2) (not discussed in detail herein) allows one to determine the period or Riemann matrix, B_{mn}, wave numbers, k_n, frequencies, ω_n, and phases, ϕ_n, appropriate to solving the *Cauchy problem* for KdV, i.e. given the spatial variation of the solutions at $t = 0$, namely $\eta(x,0)$, compute for all time the solution, $\eta(x,t)$.

It is important to recognize that (2), for small amplitude waves, gives the usual ordinary Fourier transform as a solution to the *linearized* KdV equation: $\eta_t + c_o\eta_x + \beta\eta_{xxx} = 0$. Thus the generalized Fourier expression (2), (3), in the small amplitude limit, is just the ordinary Fourier approach commonly used for everyday data analysis problems. However, the main advantage of (2), (3) is that when the waves are *not* small in amplitude one is able to fully generalize to a nonlinear set of basis functions (cnoidal waves, see for example [Weigel, 1964], [Mei, 1983]) and to include nonlinear interactions among these nonlinear modes.

The Kadomtsev and Petviashvili (KP) equation, KdV generalized to the case of directional spreading, is given by

$$\eta_t + c_o\eta_x + \alpha\eta\eta_x + \beta\eta_{xxx} + \gamma\ \partial_x^{-1}\eta_{yy} = 0 \tag{4}$$

Where $\eta(x,y,t)$ is the wave amplitude as a function of the two spatial variables, x, y and time, *t*. The constants c_o, α, β are given after (1) and $\gamma = c_o / 2$. The KP equation (4) is a natural two-space-dimension extension of the KdV equation (1). The periodic KP solutions include *directional spreading* in the wave field:

$$\eta(x,y,t) = \frac{2}{\lambda}\frac{\partial^2}{\partial x^2}\ln\theta(x,y,t\,|\,\mathbf{B},\mathbf{K},\mathbf{\Omega},\mathbf{\Phi}) \tag{5}$$

Here the generalized Fourier series has the same form as above, but the phase has the *two dimensional* expression:

$$\mathbf{X}(x,y,t) = \mathbf{K}x + \mathbf{L}y - \mathbf{\Omega}t - \mathbf{\Phi} \tag{6}$$

The spatial term $\mathbf{K}x$ has been joined by the lateral spatial term $\mathbf{L}y$, which allows wave spreading to be taken into account. The KP equation is the first nonlinear step toward a directional sea state; KP is however limited to small directional spreading. Improving the directional spreading characteristics of the KP equation requires the addition of physically important corrections to the equation as discussed below.

5. IMPLEMENTATION OF THE NEW NONLINEAR MODEL

We now discuss certain aspects of the nonlinear preprocessor in the new approach for obtaining numerical solutions to nonlinear wave equations. See Osborne [1995, 2000, 2002] for useful references. *First* note that one begins with the Riemann theta functions and makes a transformation. The simplest type of transformation is given by equation (5). In the numerical results discussed below we implement (5). *Second* we will use the following result, i.e. that the theta function (3) can be reduced to an ordinary linear Fourier series with time varying coefficients:

$$\theta(x,y,t) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\theta_{mn}(t)e^{ik_m x + il_n y + i\phi_{mn}} \tag{7}$$

Here the time varying coefficients $\theta_{mn}(t)$ can be written analytically in terms of the Riemann matrix, phases and frequencies. Note that the above expression is rather simple, i.e. it is just an ordinary linear Fourier series, which in numerical applications has N^2 terms. The theta function itself is quite different in behavior for it has an exponential number of terms $\sim 10^N$, which can be a huge number. Consider for example a case where $N = 30$, i.e. 30 cnoidal waves in the spectrum (see below for an example of a cnoidal wave). Then the number of terms in the theta function is far greater then the number of grains of sand on the earth, the number of galaxies in the universe, greater than Avogardro's number and greater then the number of seconds since the big bang! Clearly the preprocessor has got to be pretty efficient to reduce 10^N ($\sim 10^{30}$) to N^2 (900)! We call the mathematical process to reduce the theta function onto the ordinary linear Fourier modes a "collapse", i.e. the 10^N modes of the theta function are collapsed onto the N^2 modes of the linear Fourier transform (for each value of time *t*). Indeed we generally say that *we collapse the theta function onto the linear Fourier modes*. This process requires some knowledge of the algebraic geometry of theta functions [Belokolos, et al, 1994] that we do not go into here.

Another way to describe what we do in the preprocessor is to note that the theta function, which effectively has $\sim 10^N$ phase space dimensions, for all but a set of zero measure has only a *polynomial number of active phase space dimensions*, i.e. $N^2 + aN^3 + bN^4 + cN^5 + ...$ The proof of this statement is lengthy and will be documented in the future. We are therefore able to reduce the theta function from a calculation which is exponential in the number of modes (10^N) to one which is polynomial in the number of modes ($N^2 + aN^3 + bN^4 + cN^5 + ...$) for computations of physical interest. The preprocessor algorithm must therefore be able to find this polynomial number of phase space dimensions or must have *a priori* knowledge of what they are in order to collapse the theta function onto the linear Fourier modes. We are investigating a number of approaches at the present time to do this and now have four that are useful for the collapsing process. Only one of these methods (clearly the best up to now) is *not* exponential. Fig. 2 shows a flow chart of some of the significant steps in the integration of nonlinear wave equations. In order to illustrate the polynomial behavior for the number of theta function modes ($N + aN^2 + bN^3 + cN^4 + ...$) we show in Fig. 3 a typical example which works quite well for a fifth order polynomial. Thus roughly 30,000 active theta function degrees of freedom are required to do the simulations herein where the Riemann matrix was taken to be 25x25. For the particular limits taken in the theta function we had a total of 5 trillion degrees of freedom! The reduction from 5 trillion to 30 thousand active modes provides the basis for the fast nature of the present algorithm. The assessment of sparseness and the determination of the actual active degrees of freedom in the Riemann theta function is one of the most fundamental of the preprocessor tasks.

6. NUMERICAL RESULTS

We now consider some of the numerical results. To be concrete we have chosen the KP equation, i.e. directionally spread waves in shallow water that are distributed over relatively small angles. An example of a directionally spread wave train is given in Fig. 4. The significant wave height is 1.5 m and the water depth is 8 m. There are 24 cnoidal waves in the spectrum with a maximum modulus of 0.84 (these are strongly nonlinear waves, but they are still less nonlinear than solitons which have a modulus of 1). This latter cnoidal wave of modulus 0.84 is shown in Fig. 5 and its corresponding profile is shown in Fig. 6. In the latter figures we see that the wave is really Stokes-like in its shape, a result of the nonlinearities in the KP equation.

We now turn to the modified or ***extended KP equation***, which we refer to as xKP. This equation is found from the KP equation by extending it an additional two orders of approximation beyond the KP equation and for many ***purposes xKP behaves essentially like the Boussinesq equation***. To add this extended feature to the model in Fig. 2 we must of course add a ***time varying Riemann matrix and phases***. To this end the model of Fig. 2 must be modified by adding the dynamics of the Riemann matrix and phases to the blue box in the figure. This step is quite mathematical and will therefore be documented in detail at a later time.

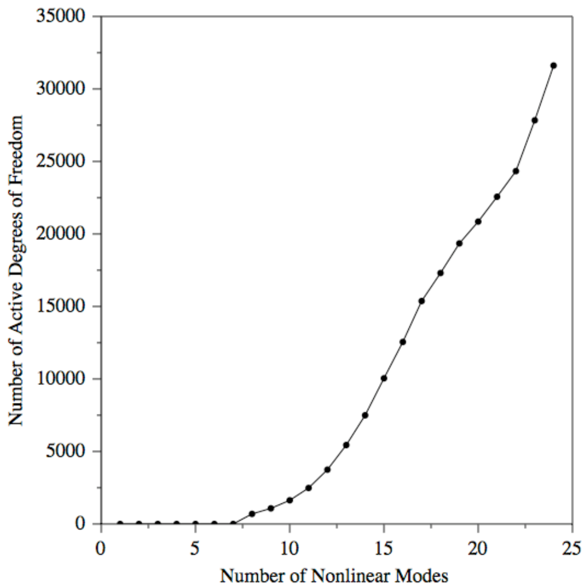

Figure. 3. Number of active degrees of freedom as a function of the number of nonlinear modes (dimension of Riemann theta function), for a typical simulation with a 256x256 spatial grid for 500 time points.

The numerical results for the Boussinesq model are shown in Figs. 7 through 11. One of the major new results which have come from these simulations is the appearance of a *new kind of rogue wave* which occurs only in shallow water wave dynamics and is not related to the Benjamin-Feir instability. Shallow water rogue waves of the type observed in the simulations occur due to strong nonlinear interactions between two or more cnoidal waves and in their simplest form are a mildly nonlinear precursor to the Mach stem.

The other major result is the small amount of computer time required for these simulations. The waves are significantly nonlinear with the ratio of significant wave height to water depth given by $H_s / h = 1.5m / 8m = 0.1875$ and the maximum wave height to depth is $H_{\max} / h = 3.4m / 8m = 0.425$. The integration domain is 500 m by 500 m and has 128 x 128 spatial bins. A total of 500 time values were computed to make a film. The total cpu time was 19 sec on a Macintosh G5 running at 2.5 GHz (this result is considerably faster than previous versions of the code). This compares to a split-step FFT code which took 11 hours to compute the same problem on the same computer.

Thus the preprocessor multi-dimensional Fourier model which we have developed is over 2000 times faster than the split-step FFT code. For a number of reasons which we will not mention here, we anticipate additional improvement in the preprocessor

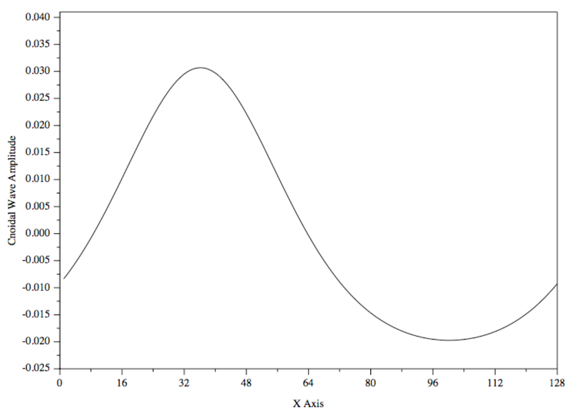

Figure. 6. A single cnoidal wave basis function (see Fig. 5) showing the narrow peak and broad trough. The wave modulus (a number between 0 and 1 indicating nonlinearity) is $m = 0.84$.

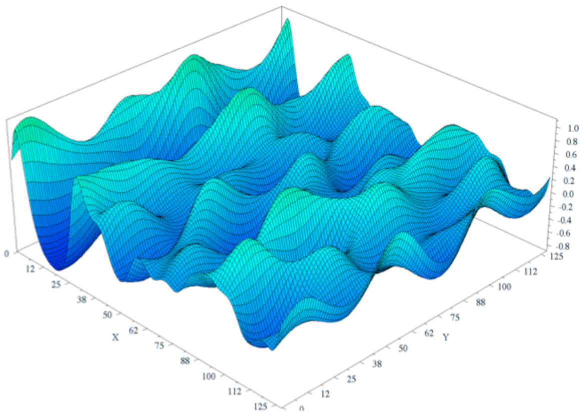

Figure. 4. Typical wave field in simulation of KP equation. The significant wave height is 1.5 m and the water depth is 8 m.

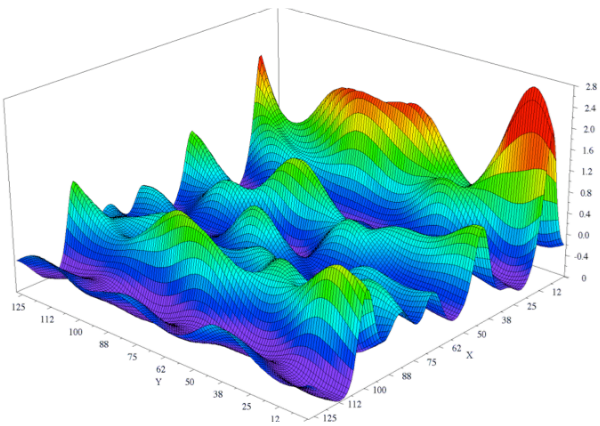

Figure. 7. A large rogue wave of height 3.4 m in the Boussinesq (xKP) simulation. The significant wave height is 1.5 m. Note the presence of the rogue waves whose maxima are shown in red.

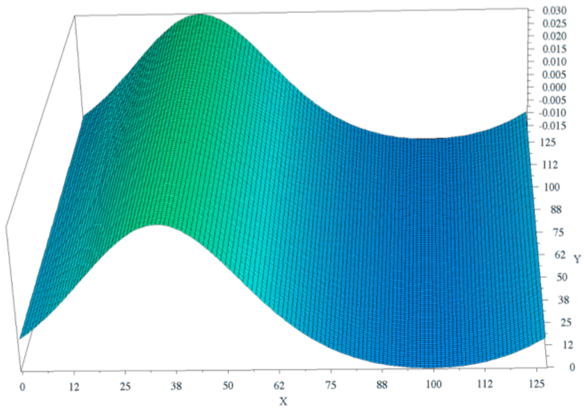

Figure. 5. A single cnoidal wave basis function is effectively a Stokes wave which is up-down asymmetric, having a narrow peak and a broad trough.

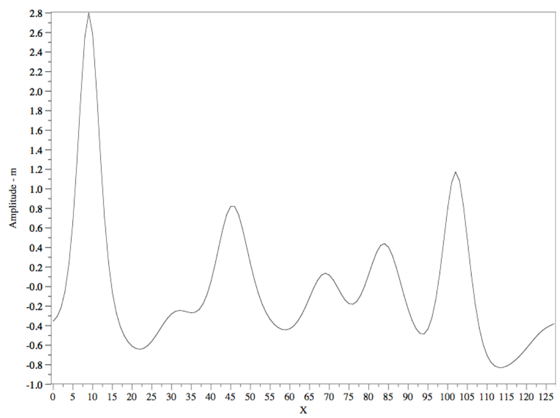

Figure. 8. Space series of the shallow water rogue wave of Fig. 7. The height is 3.4 m and the wave length is 145 m.

time, which for the present run was 4 sec. The remaining 15 sec was the iteration of the computation of the 500 points in time for FFTs of 128 x 128 points (this last step is equivalent to the linear model).

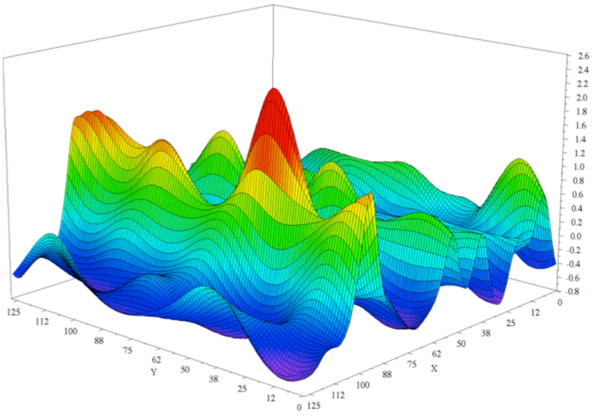

Figure. 9. Another large rogue wave of height 3.3 m in the Boussinesq (xKP) simulation. The significant wave height is 1.5 m.

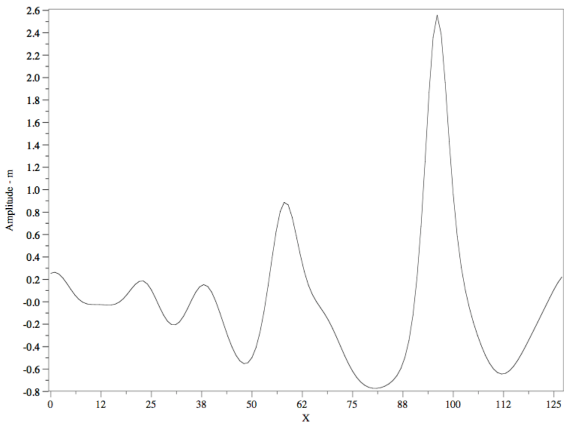

Figure. 10. Space series of shallow water rogue wave of Fig. 7. The height is 3.3 m and the wave length is 120 m.

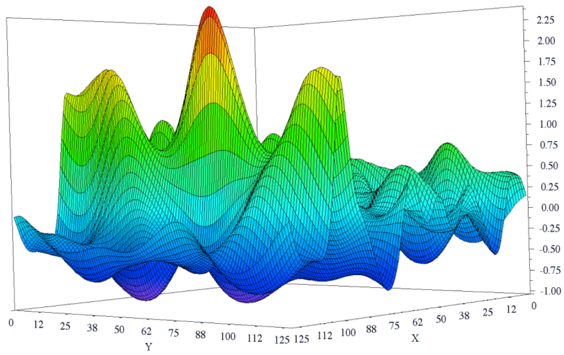

Figure. 11. Another view of a shallow water rogue wave as seen from the mean water level.

Figs. 7 and 8 show a single frame from the above xKP run. A rogue wave of height 3.4 m is seen on the right hand side of the graph. The height 3.4 m divided by the significant wave height of 1.5 m gives a rogue wave height of 2.27 H_s. Figs. 9 and 10 show another rogue wave from another frame in the film. Finally in Fig. 11 we see another rogue wave, this time from the perspective of an observation at the mean water level.

7. THE PHYSICS OF SHALLOW WATER ROGUE WAVES

The physics of shallow water rogue waves is most well-known in terms of soliton interactions and the creation of the Mach stem. This is illustrated in Figs. 12 and 13. Fig. 12 is a case with moderate nonlinearity while Fig. 13 has the strongly nonlinear case for two interacting soliton trains.

We now turn to other cases which are sufficiently complex (the number of nonlinear modes have been reduced to 15, i.e. the Riemann matrix is 15 x 15) and which have less directional spreading than the examples given above. Figs. 14 and 15 show the sea surface and the contours for such a case. The larges wave is easily seen to be at the junction of a Mach stem.

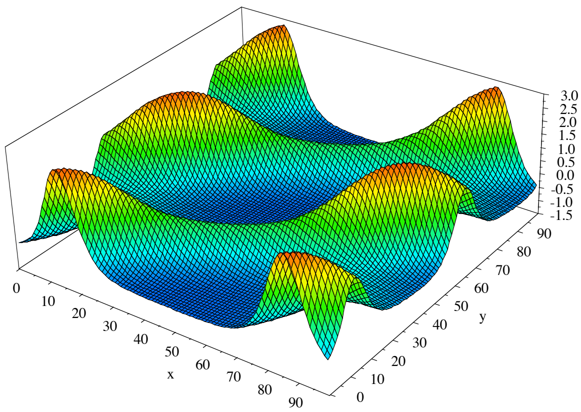

Figure. 12. Surface elevation due to the interaction of two cnoidal waves which results in a large wave amplitude caused by the Mach stem effect.

Another similar case is shown in Figs. 16 and 17. Note the crossed lines, indicating the presence of a Mach stem, in the contour graph Fig. 17. However in this case another large wave, at the bottom of Fig. 17, can also be categorized as a rogue wave, but no Mach stem is observed! Clearly the characterization of shallow water rogue waves is not yet well understood.

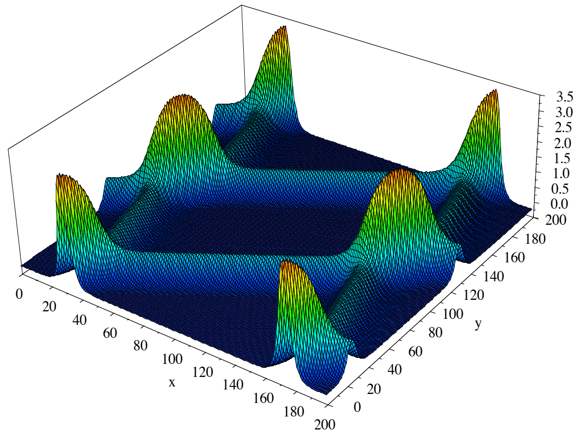

Figure. 13. Surface elevation due to the interaction of two cnoidal waves (so large in amplitude that they are effectively two soliton trains) resulting in a large wave at the intersections of the soliton trains which is caused by the Mach stem effect.

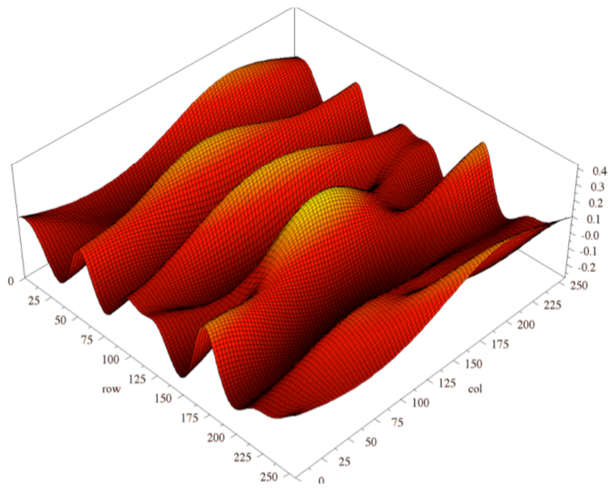

Figure. 14. Surface elevation of a large wave caused by the Mach stem effect.

To shed some light on this last observation we note the single rogue wave, clearly in the absence of any kind of Mach stem effect, shown in Figs. 18-20. This simulation was conducted with a Riemann matrix of 15 x 15 with a spatial grid of 256 x 256. The initial Riemann phases were set to zero and the wave train was advanced in time to capture the occurrence of the rogue wave. The procedure is similar to that used in wave tanks to construct a large amplitude wave train by preselecting the phases, not as random numbers, but with specific values in order to elicit the occurrence of a large amplitude wave somewhere in the center of the tank. This approach is seen to work equally well for the nonlinear problem. However, in our case one must preselect the Riemann phases to have appropriate values.

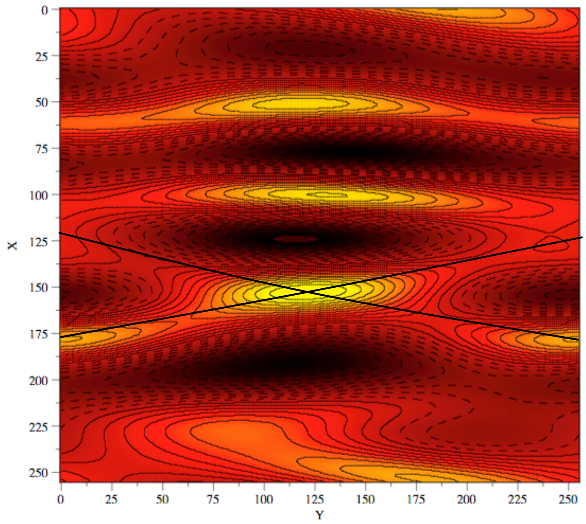

Figure. 15. Contours of Fig. 14. One can actually see the Mach stem surrounding the largest wave as shown by the crossed lines.

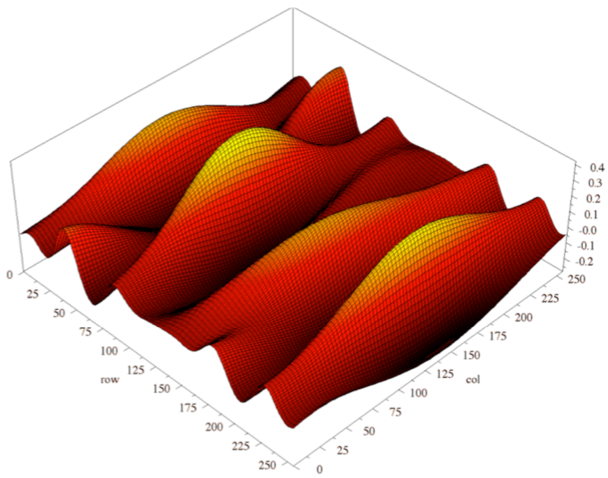

Figure. 16. Surface elevation of a large wave caused by the Mach stem effect.

8. SUMMARY AND DISCUSSION

We have developed a new approach for the numerical modeling of nonlinear partial differential wave equations. The method is based upon the inverse scattering transform for periodic/quasi-periodic boundary conditions. In this formulation one uses the Riemann theta function (a kind of multi-dimensional Fourier series) to construct the nonlinear dynamics of the waves. We have found an approach which allows us to treat the numerics in a way which is quite similar to the linear problem, but with the addition of a preprocessor step which is based upon a fast numerical computation of the theta functions and on their reduction to ordinary time-dependent Fourier series. The results of our numerical algorithm are about three orders of magnitude faster than the split-step FFT algorithm.

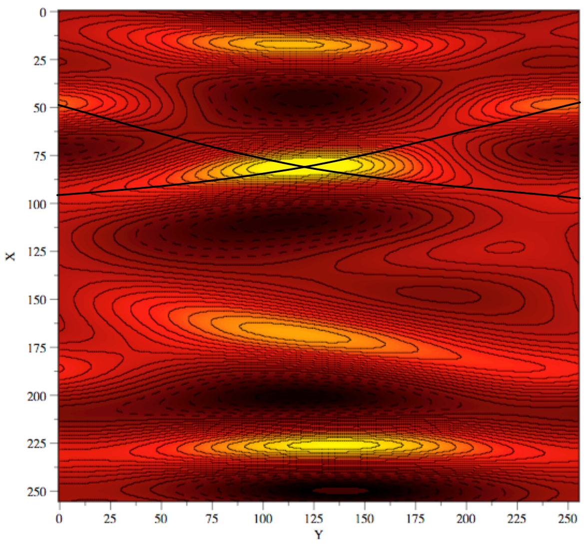

Figure. 17. Contours of Fig. 16. One can actually see the Mach stem surrounding the largest wave as shown by the crossed lines.

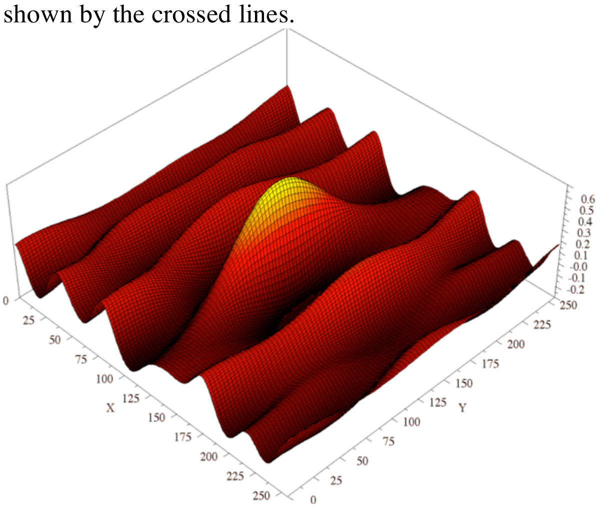

Figure. 18. Another type of rogue wave identified in the simulations. No Mach stem is evident.

How accurate is our algorithm? In principle it is exact. This is because it is simply an algorithm which evaluates the solution at particular values of the time t. We do not have to numerically integrate the motion in any way. Since t is a parameter one can rapidly and exactly compute the solution at any desired time. We have not evaluated the round off errors in our approach, but they should be roughly the same as with the standard Fourier transform evaluation.

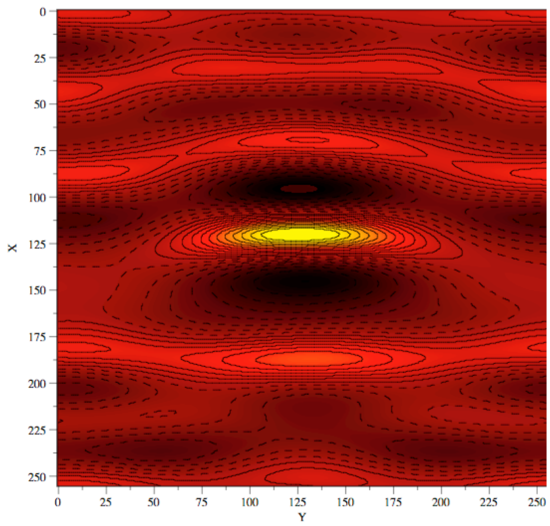

Figure. 19. Contours of rogue wave of Fig. 18. No Mach stem is evident.

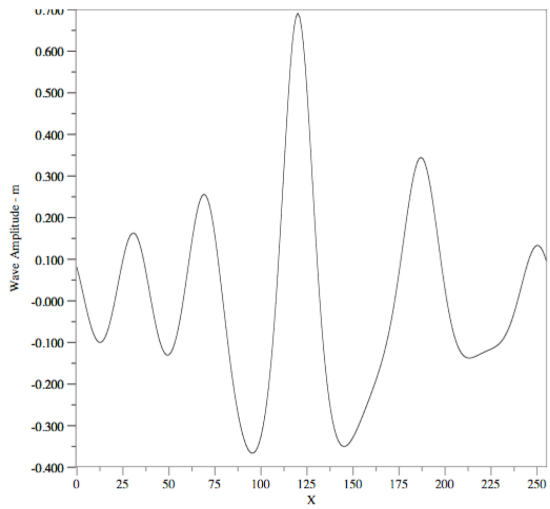

Figure. 20. Profile of rogue wave of Figs. 18, 19.

The model will require considerable work in the next year or two to make it "user friendly." At present one must specify the Riemann matrix and phases, whereas we are all more familiar with and prefer to use the standard Pierson-Moskowitz and JONSWAP power spectra. In addition to an interface providing spectra of this type as input (including the computation of the Riemann matrix from the standard input oceanographic spectra) one must also offer the possibility of integrating the motion out to any order and in particular to provide deep water capability. Additionally we plan to add arbitrary bathymetry and coast lines, and wind forcing and dissipation to the model.

This work was supported in part by the Army Corp of Engineers, The Office of Naval Research, the European Community, and a Cofin Grant and two MURST (Ministero Università e Ricerca Scientifica Tecnologica) grants from Italy.

REFERENCES

Ablowitz, M. J. and H. Segur, 1981: *Solitons and the Inverse Scattering Transform* (SIAM, Philadelphia).

Belokolos, E. D., Bobenko, A. I., Enol'skii, V. Z., Its, A. R. and Matveev, V. B., *Algebro-Geometric Approach to Nonliner Integrable Equations* (Springer-Verlag, Berlin, 1994).

Mei, C. C., 1983: *The Applied Dynamics of Ocean Surface Waves* (John Wiley, New York).

Miles, J. W., 1977: J. Fluid Mech., Vol. 79, 157.

Novikov, S. P., S. V. Manakov, and M. P. Pitaevskii, V. E. Zakharov, 1980: *Theory of solitons. The Inverse Scattering Method*, (Consultants Bureau, New York).

Osborne, A. R., 1995: Solitons in the periodic Korteweg-deVries equation, the θ-function representation, and the analysis of nonlinear, stochastic wave trains, Phys. Rev. E, Vol. 52(1), 1105-1122.

Osborne, A. R., Onorato, M., Serio, M., 2000: The nonlinear dynamics of rogue waves and holes in deep-water wave trains, Phys. Lett. A 275, 386.

Osborne, A. R., 2002: Nonlinear Ocean Waves and the Inverse Scattering Transform, *Scattering*, ed. by Roy Pike and P. Sabatier, Academic Press.

Weigel, R., 1964: *Oceanographical Engineering*. (Prentice-Hall, Englewood Cliffs, N.J.).

Whitham, G. B, 1974: *Linear and Nonlinear Waves* (John Wiley, New York).