

LAMINATED WAVE TURBULENCE: GENERIC ALGORITHMS I

ELENA KARTASHOVA*

*RISC, J.Kepler University, Altenbergerstr.69
Linz, A-4040 Austria
lena@risc.uni-linz.ac.at*

ALEXEY KARTASHOV

*AK-Soft, Pillweinstr.41 Linz, A-4020 Austria
alekxkart1@gmx.at*

Received Day Month Year

Revised Day Month Year

*Research Institute for Symbolic Computations, Johannes Kepler University, Altenbergerstr.69, Linz, A-4040 Austria

The model of laminated wave turbulence presented recently unites both types of turbulent wave systems - statistical wave turbulence (introduced by Kolmogorov and brought to the present form by numerous works of Zakharov and his scientific school since nineteen sixties) and discrete wave turbulence (developed in the works of Kartashova in nineteen nineties). The main new feature described by this model is the following: discrete effects do appear not only in the long-wave part of the spectral domain (corresponding to small wave numbers) but all through the spectra thus putting forth a novel problem - construction of fast algorithms for computations in integers of order 10^{12} and more. In this paper we present a generic algorithm for polynomial dispersion functions and illustrate it by application to gravitational water waves and oceanic planetary waves.

PACS: 47.27.E-, 67.40.Vs, 67.57.Fg

Key Words: Laminated wave turbulence, discrete wave systems, computations in integers, transcendental algebraic equations, complexity of algorithm

1. INTRODUCTION

Statistical theory of wave turbulence begins with the pioneering paper ¹ of Kolmogorov presenting the energy spectrum of turbulence as a function of vortex size and thus founding the field of mathematical analysis of turbulence. Kolmogorov regarded some inertial range of wave numbers between viscosity and dissipation, $k_0 < k < k_1$ for wave numbers k where $k = |\vec{k}|$, and suggested that in this range turbulence is locally homogeneous and isotropic which, together with dimensional analysis, allowed Kolmogorov to deduce that energy distribution is proportional to $k^{-5/3}$.

Kolmogorov's ideas were further applied by Zakharov for construction of wave kinetic equations ² which are approximately equivalent to the initial nonlinear PDEs:

$$\dot{A}_1 = \int |V_{(123)}|^2 \delta(\omega_1 - \omega_2 - \omega_3) \delta(\vec{k}_1 - \vec{k}_2 - \vec{k}_3) (A_2 A_3 - A_1 A_2 - A_1 A_3) d\vec{k}_2 d\vec{k}_3$$

for 3-wave interactions, and similar equations for i -wave interactions where δ is the Dirac delta-function and $V_{(12..i)}$ is the vortex coefficient in the standard representation of nonlinearity in the initial PDE:

$$\Sigma_i \frac{V_{(12..i)} \delta(\vec{k}_1 + \vec{k}_2 + \dots + \vec{k}_i)}{\delta(\omega_1 + \omega_2 + \dots + \omega_i)} A_1 A_2 \dots A_i. \quad (1)$$

A wave here has a standard form

$$A \exp i[\vec{k}\vec{x} - \omega t] \quad \text{or} \quad A \sin(\vec{k}\vec{x} - \omega t)$$

with amplitude A , wave vector \vec{k} and dispersion function $\omega_i = \omega(\vec{k}_i)$. In the **linear problem setting**, it is supposed that amplitude $A = A(\vec{k})$ does not depend on time t and dispersion function can be found as a solution of linear PDE, for instance

$$\psi_t + \psi_x + \psi_{xxx} = 0 \quad \Rightarrow \quad \omega(k) = k - k^3.$$

In the **weakly nonlinear problem setting**, also called wave turbulence theory, the main idea is to take into account only resonant interactions of waves described by resonance conditions

$$\begin{cases} \omega(\vec{k}_1) \pm \omega(\vec{k}_2) \pm \dots \pm \omega(\vec{k}_{n+1}) = 0, \\ \vec{k}_1 \pm \vec{k}_2 \pm \dots \pm \vec{k}_{n+1} = 0. \end{cases} \quad (2)$$

Now waves are (weakly) nonlinear, their amplitudes are slowly changing functions $A(T)$ of time $T = t/\epsilon$ where ϵ is a small parameter of nonlinearity, for instance, steepness of a wave. In this way, dispersion function keeps all the properties of the linear part of the initial nonlinear PDE, and resonance conditions describe the waves, or wave vectors, giving the greatest contribution into the nonlinear part of the initial PDE. That is the reason why Eqs.(2) are the main subject to study in the wave turbulence theory.

Statistical wave turbulence theory deals with **real solutions** of Sys.(2) and one of its most important discoveries in the statistical wave turbulence theory are stationary exact solutions of the kinetic equations first found in ³. These solutions have the form $k^{-\alpha}$ with $\alpha > 0$ and are now called Zakharov-Kolmogorov (ZK) energy spectra.

Discrete wave turbulence have been studied in the papers of Kartashova ⁴ where properties of integer solutions of Sys.(2) have been studied. It was proven in particular that (1) the spectral space of the discrete wave system is decomposed into the small disjoint groups of waves showing periodic energy fluctuations (depicted with yellow squares); (2) the most part of the waves do not take part in nonlinear interactions (depicted with blue diamonds). The model of laminated turbulence ⁵ includes both - statistical and discrete layers of turbulence, which co-exist simultaneously.

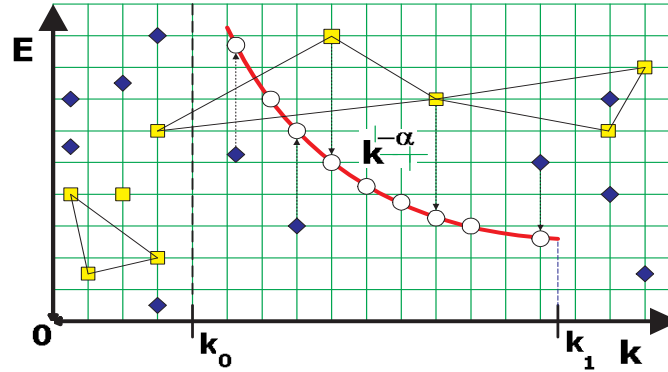


Fig. 1. **Laminated Wave Turbulence Theory, arbitrary $|\vec{k}|$** : Discrete and statistical layers of turbulence co-exist in many wave systems. ZK-energy spectrum contains "holes" in the nodes of the integer lattice which are depicted by empty circles.

The most important result of the theory of laminated wave turbulence is following: discrete effects do appear not only in the long-wave part of the spectral space (corresponding to small wave numbers) as it was supposed before but all along the wave spectra. Importance of the discrete layer of laminated turbulence is emphasized also by the fact that there exist many wave systems described **only** by discrete waves approach (for instance, most wave systems with periodic or zero boundary conditions).

From the computational point of view this theory gives rise to a completely novel problem: construction of fast algorithms for computations of integer solutions of Sys.(2) in integers of order 10^{12} and more. For instance, for 4-wave interactions

of 2-dimensional gravitational water waves this system has the form

$$\sqrt{k_1} + \sqrt{k_2} = \sqrt{k_3} + \sqrt{k_4}, \quad \vec{k}_1 + \vec{k}_2 = \vec{k}_3 + \vec{k}_4,$$

where $\vec{k}_i = (m_i, n_i)$, $\forall i = 1, 2, 3, 4$ and $k_i = |\vec{k}_i| = \sqrt{m_i^2 + n_i^2}$. It means that in a finite but big enough domain of wave numbers, say $|m|, |n| \leq D \sim 1000$, direct approach leads to necessity to perform extensive (computational complexity D^8) computations with integers of the order of 10^{12} . These computations in a substantially smaller domain $|m|, |n| \leq 128$ took 3 days ⁶ with Pentium-4. A sketch of the first fast algorithm for this problem is given in ⁷.

Probably the most often encountered context in which a physicist uses big natural numbers is generation of random numbers. However, finding numerical properties of big integer numbers is not as simple as random number generation. E.g. generation of a random number of order 10^{19} is much faster than to establish that a number of order 10^8 can be decomposed into the sum of two integer squares. Computational problems in integers present some specific challenges. First of all, the solution must be *precise* and not approximate as with "reals" (i.e. floating-point numbers). Consider a circle of some astronomic radius, say $R = 10^{100}$. Its area can be computed in microseconds with any reasonable precision. However, calculating the precise number of integer points within that same circle by computer means is an unrealistic task for modern means - "full search" for multivariate problems in integers consumes exponentially more time with each variable and size of the domain to be explored. It is not a problem of a "good approximation" - solution in integers either exists or not. And there is no general theory for finding them. Even equations of very simple form like $x^3 + y^3 = z^3$ can have no solutions at all. This last equation is the simplest case of the Last Fermat Theorem and the fact that it has no integer solutions was used in ⁸ to show that there are no three-wave resonant interactions among the capillary waves in the rectangular domain.

Last but not least, laminated turbulence problems often deal with *irrational* equations in integers. Transforming them into "normal" (Diophantine) equations in integers leads to huge powers (far beyond the reach of either ordinary personal or supercomputers) and is not always possible in principle.

Sometimes these difficulties are combined. E.g. four-wave interactions of gravitational water waves have 8 variables and full search in the domain $D \sim 1000$ would imply some 10^{24} tries. What is still worse is, that the equation includes radicals and straightforward transformation to a purely integer form would lead to operations with huge integer numbers - for the said domain, of the order of 10^{120} . All these reasons make the need for effective algorithms unavoidable.

In this paper we present a generic algorithm for computing discrete layers of wave turbulent systems with dispersion function being a function of the modulus of the

6 *E.Kartashova, A.Kartashov*

wave vector \vec{k} , $\omega = \omega(k)$. The main idea underlying our algorithm is the partition of the spectral space into disjoint classes of vectors which allows us to look for the solutions of Sys.(2) in each class separately. In Sec.2 we describe this construction in detail and with numerous examples because its brief description given in ⁸ is often misunderstood by other researchers. In Sec. 3 the generic algorithm is presented with gravitational water waves taken as our main example while in Sec. 4 modification of this algorithm is given for the oceanic planetary waves. Results of the computations and brief discussion are given at the end.

2. DEFINITION of CLASSES

For a given $c \in \mathbb{N}, c \neq 0, 1, -1$ consider the set of algebraic numbers $R_c = \pm k^{1/c}, k \in \mathbb{N}$. Any such number k_c has a unique representation

$$k_c = \gamma q^{1/c}, \gamma \in \mathbb{Z}$$

where q is a product

$$q = p_1^{e_1} p_2^{e_2} \dots p_n^{e_n},$$

while p_1, \dots, p_n are all different primes and the powers $e_1, \dots, e_n \in \mathbb{N}$ are all smaller than c .

Definition. The set of numbers from R_c having the same q is called q -class Cl_q (also called "class q "). The number q is called class index. For a number $k_{(c)} = \gamma q^{1/c}$, γ is called the weight of $k_{(c)}$.

Obviously, for any two numbers k_1, k_2 belonging to the same q -class, all their linear combinations with integer coefficients belong to the same class q . For instance, let $c = 2, q = 2, k_1 = \sqrt{8}$ and $k_2 = \sqrt{18}$, then $k_1 + k_2 = 2\sqrt{2} + 3\sqrt{2} = 5\sqrt{2}$, i.e. $k_1, k_2 \in Cl_2 \Rightarrow k_1 + k_2 \in Cl_2$. On the contrary, it is not difficult to prove that for any n numbers k_1, k_2, \dots, k_n belonging to pairwise different q -classes, the equation

$$k_1 \pm k_2 \dots \pm k_n = 0$$

has no nontrivial solutions. The general idea of the proof is very simple indeed: a linear combination of two different irrational numbers $\sqrt{q_1}, \sqrt{q_2}$ can not satisfy any equation with rational coefficients. For example, equation $a\sqrt{3} + b\sqrt{5} = 0$ has no solutions for arbitrary rational a and b .

These nice properties of the classes allow us to substitute an irrational equation by the system of linear algebraic equations, e.g. the equation

$$a_1\sqrt{8} + a_2\sqrt{12} + a_3\sqrt{18} + a_4\sqrt{24} + a_5\sqrt{48} = 0 \quad (3)$$

is equivalent to

$$2a_1\sqrt{2} + 2a_2\sqrt{3} + 3a_3\sqrt{2} + 2a_4\sqrt{6} + 4a_5\sqrt{3} = 0 \quad (4)$$

and in its turn is equivalent to the system

$$\begin{cases} 2a_1 + 3a_3 = 0 \\ 2a_2 + 4a_5 = 0 \\ a_4 = 0 \end{cases} \quad (5)$$

which is in every respect much simpler than the original equation.

The computational aspect of this transformation is an especially important illustration to the main idea of the algorithm presented in this paper. Suppose we are to find all **exact** solutions of (3) in some finite domain $1 \leq a_i \leq D$. Then the straightforward iteration algorithm needs $O(D^4)$ floating-point operations, $a_i = 1..D, i = 1, 2, 3, 4$ (even ignoring difficulties with floating-point arithmetic precision for large D). On the other hand, solutions of Sys.(5) can, evidently, be found in $O(D)$ operations with integer numbers.

3. EXAMPLE ONE: GRAVITATIONAL WATER WAVES

To show the power of the approach outlined above in practice, we proceed as follows. First we give a detailed description of the algorithm which is used to find all physically relevant four-wave interactions of the gravitational water waves. We also estimate computational complexity and memory requirements for its implementation and present results of our computer simulations. In the next section we discuss reusability of this algorithm and transform it to solve a similar problem for three-wave interactions of oceanic planetary waves. Further we briefly discuss applicability of our algorithm to other wave-type interactions.

3.1. Problem Setting

The main object of our studies are four-tuples of 2-dimensional gravitational water waves. In this case it is well-known (see, e.g. ¹⁴) that dispersion function has the form $\omega = \sqrt{k}$ where $k = |\vec{k}| = \sqrt{m^2 + n^2}$ is norm of a wave vector \vec{k} . Eqs.(2) take the following form

$$\begin{cases} \sqrt{k_1} + \sqrt{k_2} = \sqrt{k_3} + \sqrt{k_4} \\ \vec{k}_1 + \vec{k}_2 = \vec{k}_3 + \vec{k}_4 \end{cases} \quad (6)$$

where $\vec{k}_i = (m_i, n_i)$, $m_i, n_i \in \mathbb{Z} \forall i = 1, 2, 3, 4$, and $k_i = |\vec{k}_i| = \sqrt{m_i^2 + n_i^2}$. We call four wave vectors $\vec{k}_1, \vec{k}_2, \vec{k}_3, \vec{k}_4$ a *resonantly interacting four-tuple* if the resonant conditions (6) are fulfilled.

Sys. (6) is written in vector form and is equivalent to the following system of

8 *E.Kartashova, A.Kartashov*

scalar equations:

$$\begin{cases} (m_1^2 + n_1^2)^{1/4} + (m_2^2 + n_2^2)^{1/4} = (m_3^2 + n_3^2)^{1/4} + (m_4^2 + n_4^2)^{1/4} \\ m_1 + m_2 = m_3 + m_4 \\ n_1 + n_2 = n_3 + n_4 \end{cases} \quad (7)$$

Sometimes (especially in numerical examples) it is convenient to represent the solution four-tuple as

$$(m_{1L}, n_{1L})(m_{2L}, n_{2L}) = (m_{1R}, n_{1R})(m_{2R}, n_{2R}) \quad (8)$$

We are going to find all resonantly interacting four-tuples with coordinates m_i, n_i such that $-D \leq m_i, n_i \leq D$, $i = 1, 2, 3, 4$ for some $D \in \mathbb{N}$. The set of numbers $d \in [-D, D]$ is further called the main domain or simply domain.

3.2. Computational Preliminaries

3.2.1. Strategy Choice

Numerically solving irrational equations in whole numbers is always an intricate business. Basically, two approaches are widely used.

The first approach is to get rid of irrationalities (for equations in radicals typically taking the expression to a higher power, re-grouping members etc.). For an equation like $a\sqrt{x} = b\sqrt{y}$ this approach is reasonable: we simply raise both sides to power 2 and solve the equation $a^2x = b^2y$. (Some attention should be paid to the signs of a, b afterwards.) However, for Sys.(7), containing four fourth-degree roots, this approach is out of question.

The second approach is, to solve the equations using floating-point arithmetic, obtain (unavoidably) approximate solutions and develop some (domain dependent) lower estimate for the deviation, which would enable us to sort out exact solutions with deviation due only to the floating-point. As an example, consider the equation $\sqrt{x} = y$ in the domain $0 \leq x, y \leq D$. If x is *not* a square then $|\sqrt{x} - [\sqrt{x}]| \geq 1/2\sqrt{(1/D)} - 1/8D$, so each solution with smaller deviation is a perfect square. In other words, very small deviations are guaranteed to be an artefact of floating point arithmetic.

This approach is more reasonable, though for Sys.(7) the corresponding estimate would probably be not so easy to obtain. However, it has one crucial drawback, namely, its high computational complexity. Indeed, Sys.(7) consists of 3 equations in 8 variables and exhaustive search takes at least $O(D^5)$ operations, and many time consuming operations (like taking fractional powers) at that.

Our primary goal is to find all solutions in the presently physically relevant domain $D \sim 10^3$ with a possibility of extension to larger domains. The algorithm

should be *generic*, i.e. applicable to a wide class of wave types by simple transformations. Studying resonant interactions of other physically important waves we may have to deal with even more variables, e.g. for inner waves in laminated fluid $\vec{k} = (m, n, l)$ and for four-wave resonant interactions the brute force algorithm described above has computational complexity $O(D^8)$.

Clearly we need a crucially new algorithm to cope with the situation; and here classes come to our aid.

3.2.2. Application of Classes

Construction of classes, applied to the first equation of Sys.(7) readily yields the following result. For the equation

$$\sqrt[4]{t_1} + \sqrt[4]{t_2} = \sqrt[4]{t_3} + \sqrt[4]{t_4} \quad (9)$$

with $t_i \in \mathbb{N}, t_i > 0$, two situations are possible:

Case 1: all the numbers t_i , $i = 1, 2, 3, 4$ belong to the same class Cl_q .

In this case Eq.(9) can be rewritten as

$$\gamma_1 \sqrt[4]{q} + \gamma_2 \sqrt[4]{q} = \gamma_3 \sqrt[4]{q} + \gamma_4 \sqrt[4]{q} \quad (10)$$

with $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{N}$ and q an R_4 class index (i.e. a natural number not divisible by a fourth degree of any prime).

Case 2: all the numbers t_i , $i = 1, 2, 3, 4$ belong to two different classes Cl_{q_1}, Cl_{q_2} .

In this case Eq.(9) can be rewritten as

$$\gamma_1 \sqrt[4]{q_1} + \gamma_2 \sqrt[4]{q_2} = \gamma_3 \sqrt[4]{q_1} + \gamma_4 \sqrt[4]{q_2} \quad (11)$$

with $\gamma_1, \gamma_2 \in \mathbb{N}$ and q_1, q_2 being R_4 class indexes.

The fact that only these two cases are possible can be proven as a strict mathematical statement. Physical interpretation of the classes is very transparent: Case 1 describes interactions of the waves with (possibly) different lengths while Case 2 describes the interactions among the waves whose lengths are pairwise equal and only phases are different.

In this paper we concentrate on Case 1 being most interesting physically. For this case we can do computations class-by-class, i.e. for every relevant q we take all solutions of $\gamma_1 + \gamma_2 = \gamma_3 + \gamma_4$ such that $\gamma_i^4 q$ can be represented as a sum of squares $\gamma_i^4 q = m_i^2 + n_i^2$, $|m_i|, |n_i| \leq D$ and for every decomposition into such sum of squares we check the linear condition (7.2).

3.3. Algorithm Description

At the beginning we have to compute a very important domain-dependent parameter we need for the computations.

Notice that in the main domain $-D \leq m, n \leq D$ every number under the radical $t_i \leq 2D^2$ i.e. $\gamma_i^4 q \leq 2D^2$. For a given q , $\gamma^{max}(q) \leq (2D^2/q)^{1/4}$.

Definition. A number $\gamma^{max}(q)$ is called *class multiplicity* and denoted $\mathfrak{M}(q)$.

For the main domain $D = 10^3$, class multiplicities are reasonably small numbers, $\gamma^{max}(1) = 37$ being the largest. Class multiplicities for the majority of classes (starting with $q = 125002$) are equal to 1 - this fact will be later used to achieve a major shortcut in computation time.

3.3.1. Step 1. Calculating Relevant Class Indexes

Class indexes of the module R_c as defined above are numbers not divisible by any prime in c -th degree, in our case $c = 4$ not divisible by 4-th power of any prime. We can further restrict relevant class indexes as follows.

First, in (9) every number under the radical $t_i = \gamma_i^4 q$ must have a representation as a sum of two squares of integer numbers, $t_i = m_i^2 + n_i^2$. According to the well-known Euler's theorem an integer can be represented as a sum of two squares if and only if its prime factorization contains every prime factor $p \equiv 4u + 3$ in an even degree. As γ_i^4 evidently contains every prime factor in an even degree, this condition must also hold for q . This can be formulated as follows: if q is divisible by a prime $p \equiv 4u + 3$, it should be divisible by its square and should *not* be divisible by its cube.

The implementation of this step is accomplished with a sieve-type procedure. Create an array $Ar_q = [1, \dots, 2D^2]$ of binary numbers, setting the all the elements of the array to 1. Make the first pass: for all primes p in the region $2 \leq p \leq \sqrt[4]{2D^2}$ set to 0 the elements of the array $p^4, 2p^4, \dots, \kappa p^4$ where $\kappa = \lfloor 2D^2/p^4 \rfloor$. In the second pass, for all primes $p_{4u+3} \equiv 3 \pmod{4}$, $p \leq 2D^2$ and integer factors $a = 1 \dots a_{max}$ such that $ap \leq 2D^2$, do the following. If $a \not\equiv 0 \pmod{p}$ then set the ap -th element of the array Ar_q to 0. If $a \equiv 0 \pmod{p}$, then if $a \equiv 0 \pmod{p^2}$ then also set the ap -th element of the array Ar_q to 0.

Notice that in the second pass the first check should only be done for primes $p \leq \sqrt{2D^2}$ and the second one - for $p \leq \sqrt[3]{2D^2}$.

We create an array W_q of "work indexes". In the third pass, we fill it with indexes of the array Ar_q for which the elements' values have not been set to 0 in the first two passes. We also create an array of class multiplicities $\mathfrak{M}(q)$ and fill it with

corresponding class multiplicities (see previous subsection). Notice that all numbers q found above do have a representation as a sum of two squares; however, some do not have representation with $|m| \leq D$ and $|n| \leq D$. We do not look for them now: they will be discarded automatically at further steps.

The computational complexity of this step can be estimated in the following way. The number of primes $\leq x$ is, asymptotically, $\pi(x) = x/\log(x)$, so their density around x is $1/\log(x)$. The first pass takes $\lfloor 2D^2/p^4 \rfloor$ operations for each prime $2 \leq p \leq \sqrt[4]{2D^2}$ so the overall number of operations can be estimated as

$$\int_2^{\sqrt[4]{2D^2}} \frac{2D^2/x^4}{\log(x)} dx = O(D^2)$$

As for the second pass, primes $p \equiv 4u + 3 < 2D^2$ constitute about a half of all primes and are evenly distributed among them. Sieving out by a prime p requires $O(2D^2/p) + O(2D^2/p^2) + O(2D^2/p^3) = O(2D^2/p)$ operations which again boils down to overall $O(D^2)$ steps.

Evidently, the third pass requires the same $O(D^2)$ operations and the overall computational complexity of this step is $O(D^2)$. Notice that it is not so easy to give a good estimate for the number of class indices $\pi_{cl}(D)$. Of course $O(D^2/\ln(D)) \leq \pi_{cl}(D) \leq O(D^2)$ holds, and most probably $\pi_{cl}(D) = O(D^2/\ln(D))$. (This is presently under study.) Whenever we need this number for estimating computational complexity of the algorithm, we presume $\pi_{cl}(D) = O(D^2)$ to be on the safe side of things. In our main computation domain $D = 10^3$ the number of class indices $\pi_{cl}(10^3) = 384145$.

3.3.2. Step 2. Finding Decompositions into Sum of Two Squares

In 1908, G. Cornacchia¹¹ proposed an algorithm for solving the diophantine equation $x^2 + dy^2 = 4p$ with p prime, $p = 4u + 1$. This has been recently generalized to solving $x^2 + dy^2 = m$, m not necessarily prime¹⁰. To find all decompositions of a number $\gamma^4 q$ into two squares we can use a simplified variant of this setting $d = 1$. A very efficient implementation of this algorithm can be obtained thanks to the following result¹⁰:

Let $t^2 \equiv -1 \pmod{m}$, $0 < t < (m/2)$. Set $r_0 = m$ and $r_1 = t$ and construct the finite sequence $\{r_i\}$, $r_i = q_i r_{i+1} + r_i + 2$, $q_i = \lfloor r_i/r_{i+1} \rfloor$, for $0 \leq i \leq n-1$, where $r_0 > r_1 > \dots > r_n = 1 > r_{n+1} = 0$. If $r_{k-1}^2 > m > r_k^2$ then $m = r_k^2 + r_{k+1}^2$.

Now it is evident that for each $0 < t < (m/2)$ such that $t^2 \equiv -1 \pmod{m}$ we obtain one decomposition of m into two squares and the algorithm gives all decompositions with $x > y$. For our use, we also take symmetrical decompositions $x < y$ and also $x = y$ if $m = 2x^2$. The computational complexity T of the algorithm is,

12 *E.Kartashova, A.Kartashov*

basically, the complexity of finding all square roots of -1 modulo m and is logarithmic in m , i.e. $T = O(\log(m))$.

Let Dec_q be the maximal number of decompositions of $\gamma^4 q$, $\gamma = 1 \dots \mathfrak{M}(q)$ into sum of two squares. We create a three-dimensional array $Ar_D[G_q = 1 \dots \mathfrak{M}(q), D_q = 1 \dots Dec_q, 2]$ and for each G_q store the list of decompositions $(m_{G_q, D_q}, n_{G_q, D_q})$. We also create a one-dimensional auxiliary array Ar_{DtoW} storing the number of two-square decompositions for each weight. The number of decompositions of an integer into sum of two squares can be estimated as $O(\log(m))$ using the classical theorem:

Euler Theorem. Let m be a positive integer, and let

$$m = 2^r p_1^{s_1} \dots p_k^{s_k} q_1^{t_1} \dots q_l^{t_l}$$

be its factorization into prime numbers, where $p_i \equiv 1 \pmod{4}$ and $q_i \equiv 3 \pmod{4}$. Then the number of essentially different decompositions of m into sum of 2 squares is equal to the integral part of $\delta/2$ where

$$\delta = \left(\prod_{j=1}^k (s_j + 1) \right) \left(\prod \frac{(-1)^{t_j} + 1}{2} \right).$$

Now we see that filling the array Ar_D can be accomplished in

$$T = O(\log(q) + \log(2^4 q) + \dots + \log(\mathfrak{M}(q)^4 q)) \quad (12)$$

steps.

Using presentation (¹², Eq.(4.4.8.1))

$$\sum_{k=1}^n \ln(ak + b) = n \ln a + \ln \Gamma(b/a + n + 1) - \ln \Gamma(b/a + 1)$$

and the well-known

$$n! \sim \sqrt{\pi n} \left(\frac{n}{e} \right)^n$$

we obtain $T = O(\mathfrak{M}(q)(\log q + \log(\mathfrak{M}(q))) = O(\mathfrak{M}(q) \log q)$. This is much less than $O(\mathfrak{M}(q)^3)$ (see the next subsection) and contribution of this step into the overall computational complexity of the algorithm is negligible.

3.3.3. Step 3. Solving the Sum-of-Weights Equation

Consider now the equation for the weights

$$\gamma_1 + \gamma_2 = \gamma_3 + \gamma_4 \quad (13)$$

with $1 \leq \gamma_i \leq \mathfrak{M}(q)$ (see 10). For convenience we change our notation to γ_{1L}, γ_{2L} (left) and γ_{1R}, γ_{2R} (right) and introduce weight sum $S_\gamma = \gamma_{1*} + \gamma_{2*}$. Without loss of generality we can suppose

$$\gamma_{1L} \leq \gamma_{1R} \leq \gamma_{2R} \leq \gamma_{2L}. \quad (14)$$

Notice that we may not assume strict inequalities because even for $\gamma_i = \gamma_j$ there may exist two distinct vectors $(m_i, n_i), (m_j, n_j)$ with $m_i^2 + n_i^2 = m_j^2 + n_j^2 = \gamma^4 q$ either due to the possibility of representing $\gamma^4 q$ as sum of two squares in multiple ways or even for a single two-square representation - to the possibility of taking different sign combinations $(\pm|m|, \pm|n|)$ left and right.

Now we may encounter the following four situations:

$$(1) \quad \gamma_{1L} < \gamma_{1R} < \gamma_{2R} < \gamma_{2L}$$

The general, physically most interesting case. Every solution yields four waves with pairwise distinct modes.

$$(2) \quad \gamma_{1L} = \gamma_{1R} < \gamma_{2R} = \gamma_{2L}$$

$$(3) \quad \gamma_{1L} < \gamma_{1R} = \gamma_{2R} < \gamma_{2L}$$

$$(4) \quad \gamma_{1L} = \gamma_{1R} = \gamma_{2R} = \gamma_{2L}$$

The "most degenerate" case.

The search is organized as follows. Each admissible sum of weights S_γ ; $2 \leq S_\gamma \leq \mathfrak{M}(q)$ is partitioned into sum of two numbers $S_\gamma = \gamma_{1L} + \gamma_{2L}$, $1 \leq \gamma_{1L} \leq \gamma_{2L} \leq \mathfrak{M}(q)$. Then the same number is partitioned into sum of γ_{1R}, γ_{2R} , $\gamma_{1L} \leq \gamma_{1R} \leq \gamma_{2R} \leq \gamma_{2L}$. Evidently, if $S_\gamma \leq \mathfrak{M}(q) + 1$ then the minimal γ_{1L} is 1, otherwise it is $S_\gamma - \mathfrak{M}(q)$ (to provide $\gamma_{1R} \leq \mathfrak{M}(q)$). The maximal γ_{1L} is always $\lfloor S_\gamma/2 \rfloor$ and similarly $\gamma_{1R} \leq \lfloor S_\gamma/2 \rfloor$.

The computational complexity of this step can be estimated as $T = O(\mathfrak{M}(q)^3)$ due to $O(\mathfrak{M}(q))$ possibilities for each of three values $S_\gamma, \gamma_{1L}, \gamma_{1R}$. As $\mathfrak{M}(q) = \lfloor (2D^2/q)^{1/4} \rfloor$, $T = O((2D^2/q)^{3/4})$.

This step contains an evident redundancy. Indeed, the equation 13 need not be solved independently for each class. Instead, its solutions for all S_γ could be computed in advance and stored in a look-up table. However, this involves significant computational overhead (e.g. the lookup procedure includes computing the minimal $\gamma_{1L} = \max(1, S_\gamma - \mathfrak{M}(q))$, which **must** be done for each class) wiping out the gains of this approach, at least for our basic domain $D = 10^3$. Nevertheless, this approach should be kept in view if need for computations in much larger domains, say $D = 10^6$, arises. On the other hand, the general case is not really so general - most classes have small multiplicities and then degenerate cases prevail. The overall

distribution is given below:

<i>Case</i>	1	2	3	4
<i>Classes</i>	24368	57666	13987	63778

3.3.4. Step 4. Discarding "Lean" Classes

In the main domain $D \leq 10^3$ we encounter 384145 classes. This sounds like a lot - however, most of these can be processed without computations or with very simple computations. Notice the simple fact that if a class has multiplicity 1, Sys.(7) takes the form

$$\begin{cases} q = m_{1L}^2 + n_{1L}^2 = m_{2L}^2 + n_{2L}^2 = m_{1R}^2 + n_{1R}^2 = m_{2R}^2 + n_{2R}^2 \\ m_{1L} + m_{2L} = m_{1R} + m_{2R} \\ n_{1L} + n_{2L} = n_{1R} + n_{2R} \end{cases} \quad (15)$$

and for any nontrivial solution the four vectors (m_i, n_i) should be pairwise distinct. In terms of the weight equation of the previous section it means that solutions, if any, have to belong to the fourth ("most degenerate") case. It is evident that no solution of Sys.(15) with pairwise distinct (m_i, n_i) exist for q having few decompositions into sum of two squares: one ($q = m^2 + m^2$), two ($q = m^2 + n^2 = n^2 + m^2$) and three ($q = m^2 + n^2 = n^2 + m^2 = l^2 + l^2$). It can be shown by means of elementary algebra that this also holds for q having four decompositions. It is very probable that for classes of multiplicity 1 no nontrivial solutions exist, whatever the number of decompositions into sum of two squares. The question is presently under study. In the main domain $D = 10^3$ we encounter 357183 classes of multiplicity 1 (1-classes). This is about 93% of all classes in the domain. Among them, the number of decompositions into sum of two squares is distributed as follows:

<i>Dec(q)</i>	0	1	2	3	4	5	6	7	8
<i>Classes</i>	110562	256	138044	163	78886	3	8727	2	16595

<i>Dec(q)</i>	14	16	18	20	24	26	32	9	10	12
<i>Classes</i>	38	1015	84	1	75	1	1	31	269	2429

Table 1. Distribution of decomposition number $\mathfrak{M}(q)$ for 1-classes q in the main domain $D = 1000$

It follows that 327911 1-classes can be discarded without any computations at all and only 29272 must be checked for probable solutions.

3.3.5. Step 5. Checking Linear Conditions: Symmetries and Signs

Sum-of-weights equation solved and decompositions into sum of two squares found, we need only check the linear conditions to find all solutions. On the face of it, the

step is trivial, however some underwater obstacles have to be taken into account. Having found a four-tuple of vectors (m_i, n_i) satisfying the first equation of Sys.(7) with both coordinates *non-negative*, solutions of the system will be found taking all combinations of signs satisfying

$$\begin{cases} \pm m_1 \pm m_2 = \pm m_3 \pm m_4 \\ \pm n_1 \pm n_2 = \pm n_3 \pm n_4 \end{cases} \quad (16)$$

Even the straightforward approach does not need more than 2^8 comparisons, so this step does not consume very much computing time. However, a few points may not be overlooked in order to organize correct, exhaustive and efficient search:

- For the system to represent a four-wave interaction, all the four waves must be pairwise unequal. For the first degenerate case we must provide $m_{1L} \neq m_{1R}$ and $m_{2R} \neq m_{2L}$. For the second one - $m_{1R} \neq m_{2R}$ and for total degeneration - $m_{1L} \neq m_{2L} \neq m_{1R} \neq m_{2R}$.
- One and the same solution may not occur among the 256 sign combinations twice. First, this could happen due to some m_i or n_i being 0 (evidently, the \pm variation should not be done for any 0 coordinate). Next, sign variation could lead to a transposition of wave vectors. For example, for $q = 1$ and $\gamma_1 + \gamma_2 = 10$ we obtain solutions

$$\begin{cases} (0, -9)(0, 49) \Rightarrow (15, 20)(-15, 20) \\ \text{and} \\ (0, -9)(0, 49) \Rightarrow (-15, 20)(15, 20) \end{cases}$$

which really represent one and the same four-tuple.

- The set of solutions possesses some evident symmetries: if

$$(m_{1L}, n_{1L})(m_{2L}, m_{2L}) \Rightarrow (m_{1R}, n_{1R})(m_{2R}, m_{2R}) \quad (17)$$

then, of course,

$$(-m_{1L}, n_{1L})(-m_{2L}, n_{2L}) \Rightarrow (-m_{1R}, n_{1R})(-m_{2R}, n_{2R}) \quad (18)$$

$$(m_{1L}, -n_{1L})(m_{2L}, -n_{2L}) \Rightarrow (m_{1R}, -n_{1R})(m_{2R}, -n_{2R}) \quad (19)$$

$$(-m_{1L}, -n_{1L})(-m_{2L}, -n_{2L}) \Rightarrow (-m_{1R}, -n_{1R})(-m_{2R}, -n_{2R}) \quad (20)$$

Taking into account these points, an effective search is constructed easily.

4. EXAMPLE TWO: OCEANIC PLANETARY WAVES

In this Section we demonstrate the flexibility of our algorithm. Namely, we solve essentially the same problem - finding all integer solutions in a finite domain $-D \leq m, n \leq D$ for three-wave interactions and another wave type - oceanic planetary

waves. These are oceanic waves existing due to the Earth rotation and described by barotropic vorticity equation

$$\frac{\partial \Delta \psi}{\partial t} + \beta \frac{\partial \psi}{\partial x} + J(\psi, \Delta \psi) = 0 \quad (21)$$

with boundary conditions $\psi = 0$ for $x = [0, 1]$ and $y = [0, 1]$ (β is a constant, so-called Rossby number.) In this case dispersion relation has the following form¹³

$$\omega = \frac{2\beta}{\sqrt{m^2 + n^2}}$$

and resonance conditions (2) can be re-written as

$$\begin{cases} \frac{1}{\sqrt{m_1^2 + n_1^2}} + \frac{1}{\sqrt{m_2^2 + n_2^2}} = \frac{1}{\sqrt{m_3^2 + n_3^2}} \\ m_1 + m_2 = m_3 \end{cases} \quad (22)$$

where $|m_i|, |n_i| \leq D$.

4.1. Steps that Stay

- Step 1 - sieving out possible class bases - undergoes minimal changes. Now $c = -2$ and each q should be a square-free number and not divisible by any prime $p = 4u + 3$. Evidently, for this wave type the set of class indices is a subset of class indices of the previous section.
- Step 2 - decomposition into two squares - can be preserved one-to-one. Indeed, there are sophisticated algorithms for representing square-free numbers as sums of two squares that are slightly more efficient than in the general case (one used in the previous section) but this step is not the bottleneck of the algorithm.

4.2. Steps to be Modified

- Step 3 - the weight equation is in this case

$$\frac{1}{\gamma_1} + \frac{1}{\gamma_2} = \frac{1}{\gamma_3} \quad (23)$$

or

$$\gamma_3 = \frac{\gamma_1 \gamma_2}{\gamma_1 + \gamma_2} \quad (24)$$

which has relatively few solutions in integers. Indeed, even for class 1 with multiplicity 1414 we obtain only 3945 solutions.

Remark. For this example it makes sense to generate and store the set of triads $(\gamma_1, \gamma_2, \gamma_3)$ which constitute integer solutions of the Eq.(23) for $1 \leq \gamma_i \leq \mathfrak{M}(1)$ and for each class q just take its subset $1 \leq \gamma_i \leq \mathfrak{M}(q)$.

- Step 4 - discarding "lean" classes - becomes trivial: no class with multiplicity 1 yields an integer solution of 23. We need only consider 63828 classes ($q_{63828} = 499993$, $q_{63829} = 500009$) from 243143 in the main domain.

- Step 5 - checking linear conditions - is also much easier than in the previous example, i.e. one equation with three variables instead of two equations with four variables each.

5. DISCUSSION

Our algorithm has been implemented in VBA programming language; computation time (without disk output of solutions found) on a low-end PC (800 MHz Pentium III, 512 MB RAM) is about 4.5 minutes for Example 1 and 1.5 minutes for Example 2. Some overall numerical data for both examples is given in the Tables and Figures below:

<i>Domain</i>	≤ 200	≤ 400	≤ 600	≤ 800	≤ 1000
<i>Solutions</i>	263648	800435	932475	1127375	1389657

Table 2. **Gravitational water waves:** Distribution of the number of solutions depending on the chosen main domain D .

It is interesting that though the overall number of solutions grows sublinearly as we extend the domain, the number of asymmetrical solutions ($\gamma_1 \neq \gamma_2 \neq \gamma_3 \neq \gamma_4$), physically most important ones, grows faster than linearly:

<i>Domain</i>	≤ 200	≤ 400	≤ 600	≤ 800	≤ 1000
<i>Solutions</i>	96	344	744	1328	2088

Table 3. **Gravitational water waves:** Distribution of the number of the asymmetrical solutions depending on the chosen main domain D .

Notice that considerable part of them (185 of the overall 2088) lie outside of the $D = 950$ area, e.g.:

$$\begin{aligned}
 &(-150, -25)(990, 945) \Rightarrow (294, 49)(546, 871) \\
 &\text{where } q = 37, \quad \gamma_1 = 5, \quad \gamma_2 = 15, \quad \gamma_3 = 7, \quad \gamma_4 = 13, \\
 &\quad (128, 256)(990, 180) \Rightarrow (400, 200)(718, 236) \\
 &\text{where } q = 20, \quad \gamma_1 = 8, \quad \gamma_2 = 15, \quad \gamma_3 = 10, \quad \gamma_4 = 13, \\
 &\quad (-80, -76)(980, 931) \Rightarrow (180, 171)(720, 684) \\
 &\text{where } q = 761, \quad \gamma_1 = 2, \quad \gamma_2 = 7, \quad \gamma_3 = 3, \quad \gamma_4 = 6
 \end{aligned}$$

etc. As a whole, asymmetrical solutions are distributed not uniformly along the wave spectrum but are rather grouped around some specific wave numbers. For instance, the first group of asymmetrical solutions (containing 8 solutions) appears in the domain $D = 50$, with solution

$$(-4, -4)(49, 49) \Rightarrow (9, 9)(36, 36),$$

and others, while in the domains $D = 60, 70, 80, 90$ there are no new asymmetrical solutions. The next new group (16 solutions) appears in the domain $D = 100$, and so on. From the physical point of view, asymmetrical solutions are the most interesting ones because they generate new wave lengths and, therefore, distribute energy through the scales. As it was pointed out quite recently ¹⁴, asymmetrical solutions play an extremely important role in wave turbulence. Indeed, no profound understanding of turbulence can be achieved without studying their properties which is in our agenda.

Numerical data for the case of planetary waves are given in the Table 4 below:

<i>Domain</i>	≤ 200	≤ 400	≤ 600	≤ 800	≤ 1000
<i>Solutions</i>	1099	3137	5664	8565	11795

Table 4. **Oceanic planetary waves:** Distribution of the number of solutions depending on the chosen main domain D .

This data is presented graphically in figures below. Number of asymmetric solutions for Example 1 (gravitational water waves) and total solutions for Example 2 (oceanic planetary waves) show smooth power growth and probably *are* asymptotically power functions of the domain size D . On the contrary, the total solution number for Example 1 has an unexpected twist about $D = 350$ shown in Fig.2, this phenomenon is presently under study.

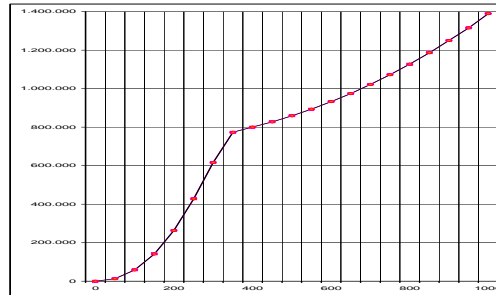


Fig. 2. Gravitational water waves: Number of all solutions in partial domains

Notice that the algorithm presented here allows to find all solutions for wave vectors belonging to the same class. For three-wave interactions of arbitrary wave types this is always the case. For n -wave interactions with $n > 3$, however, interacting waves may belong to $\lfloor \frac{n}{2} \rfloor$ different classes ¹⁵. Consider for example the four-wave

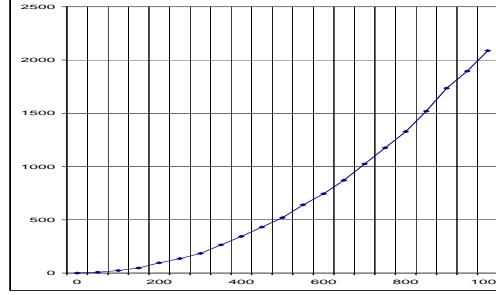


Fig. 3. Gravitational water waves: Number of asymmetric solutions in partial domains

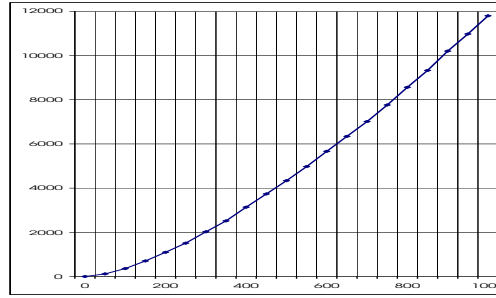


Fig. 4. Oceanic planetary waves: Number of all solutions in partial domains

system

$$\begin{cases} \sqrt{k_1} + \sqrt{k_2} = \sqrt{k_3} + \sqrt{k_4} \\ \vec{k}_1 + \vec{k}_2 = \vec{k}_3 + \vec{k}_4 \end{cases}$$

where k_1 and k_3 belong to one class and k_2 and k_4 - to another one, i.e. the first equation breaks up into two independent equations

$$\sqrt{k_1} = \sqrt{k_3} \quad \text{and} \quad \sqrt{k_2} = \sqrt{k_4}. \quad (25)$$

It is important to realize that construction of classes is not just a mathematical trick allowing to reduce drastically the computational time but has profound physical background. One has to remember that wave length λ is inversely proportional to the length of the wave vector, $\lambda = 2\pi/k$, then it is obvious that asymmetrical solutions, all belonging to the same class and all having different k_i , describe the waves which transport the energy over the scales of the wave field. On the other hand, solutions of Eqs.(25) are all symmetrical, they do not generate new wave lengths and transport the energy not over the scales but over the phases presenting

circle structures in the spectral space. For computing these symmetric solutions, a modified form of our generic algorithm can be applied. This will be dealt with in our next paper.

Acknowledgement. E.K. acknowledges the support of the Austrian Science Foundation (FWF) under projects SFB F013/F1304.

References

1. A.N. Kolmogorov. *Dokl. Akad. Nauk SSSR* (30), 301-305 (1941). Reprinted: *Proc. R. Soc. Lond. A* (434), 9-13 (1991)
2. V.E. Zakharov, V.S. L'vov, G. Falkovich. *Kolmogorov Spectra of Turbulence* (Series in Nonlinear Dynamics, Springer, 1992)
3. V.E. Zakharov, N.N. Filonenko. *J. Appl. Mech. Tech. Phys.* (4), 500-515 (1967)
4. E.A. Kartashova. *Physica D* (46), 43 (1990); E.A. Kartashova. *Physica D* (54), 125 (1991); E.A. Kartashova. *Phys. Rev. Letters* (72), 2013 (1994); E.A. Kartashova. *TMPH* (99), 675 (1994); E.A. Kartashova. *AMS Transl.* (182), 2, 95 (1998) and others
5. E.A. Kartashova. *JETP Letters* (83) 7, 341 (2006)
6. S. Nazarenko. *Private communication* (12.2005)
7. E.A. Kartashova. *JLTP* (to appear). E-print arXiv.org:math-ph/0605067 (2006)
8. E.A. Kartashova. *AMS Transl.* (182), 2, 95 (1998)
9. I. Besikovitch. *J. London Math. Society* **15**, 3 (1940)
10. J.M. Basilla. *Proc. Japan. Acad.* **80**, Ser. A, 40 (2004)
11. H. Cohen, *A Course in Computation Number Theory* (Grad. Texts in Math. **138**. Springer-Verlag, New York, 1993) p.34
12. N.P. Prudnikov et.al. *Integrals and Rows. Vol. I* (Nauka, Moscow, 1981, in Russian)
13. E.A. Kartashova, G.M. Reznik. *J. Oceanology* **31**, 385 (1992)
14. Y.V. Lvov, S. Nazarenko, B. Pokorni. *Physica D* (218) 24-35 (2006)
15. E.A. Kartashova. *Resonant interactions of the water waves with discrete spectra - Proc. of Int. Nonlinear Water Waves Workshop (NWWW)*, ed. D.H.Peregrine (University of Bristol, Bristol, UK, 1992), p.43