Using differential properties of the green function in seakeeping computational codes

A.H. Clement

USING DIFFERENTIAL PROPERTIES OF THE GREEN FUNCTION IN SEAKEEPING COMPUTATIONAL CODES

A.H Clément
[1]Laboratoire de Mécanique des Fluides, Ecole Centrale de Nantes
1 Rue de la Noë, B.P. 92101, 44321 Nantes Cedex 3, France
Fax: 332 40 37 25 23; E-Mail: Alain.Clement@ec-nantes.fr

ABSTRACT

The Green function of linear time-domain hydrodynamics has been recently proven to satisf y a simple fourth order ordinary differential equation (ODE) [5]. As a fist application of this remarkable property, the present paper shows how it can be used to evaluate the kernel of the convolution integrals in the BEM solution of seakeeping problems, in the linear time-domain approach. The Green function is obtained by integrating these ODE from one time-step to another instead of being computed by the usual methods. Several Runge-Kutta algorithms were tested and compared; it is shown how an optimal time step can be derived from a stability analysis. The *cpu*-time savings brought by this new method are then evaluated. Numerical results are given for three test cases: the heaving hemisphere, a standard series-60 ship hull, and a TLP offshore platform. Impulse response functions are calculated and Fourier transformed to recover the classical frequency domain hydrodynamic coefficients as a check for validity and accuracy.

INTRODUCTION

The solution of time-domain seakeeping problems in the framework of linear potential flow theory generally requires to resolve a boundary integral problem involving convolution integrals. These integrals may be regarded as the memory of the free-surface fluid; their kernel features the time-domain Green-function of linear free-surface hydrodynamics, and its spatial gradient. The numerical evaluation of this function, analytically defined as an integral over an unbounded domain, is quite time consuming by itself. Furthermore, since the convolution integrals extend from the initial state of rest up to the current time t, the mass storage and *cpu* time required for their computation grow roughly quadratically with time. Since 1995 [3], we are developing a "system" approach to the problem, which is aimed at speeding up these calculations and at reducing the storage demand. It is based on a major result stating that the time-domain Green-function of linearized free-surface hydrodynamics satisfies an exact fourth order ordinary differential equation [4, 5]. If the coefficients of this ODE were constant with respect to the time variable, the convolution integrals could be completely suppressed and replaced by a simple filtering numerical process. Unfortunately, the ODE coefficients being polynomial with respect to the time variable, such a simplification does not occur. Nevertheless, this differential property provides us with an alternative method for the on-line computation of the Green function during the calculation of the convolution integrals. In this paper, we present an application of this approach to the computation of the free-surface flow resulting from an impulsive motion of a floating body around its equilibrium position; the practical output of such computations being the matrix of impulse response functions (IRFs) of the body. The aim of this very first application of our ODEs was primarily to establish the feasibility of the method, and to estimate the savings it could bring in terms of computation time. The problem is posed in the linear potential theory, and solved by a direct, zeroth order, BEM method. The Green functions in the kernel of the convolution integrals are computed by either the classical series expansion, a tabulation procedure, or integrating the new ordinary differential equation. The computational aspects of these three methods are presented and compared.

MATHEMATICAL FORMULATION

Let us make the usual assumptions of linearized potential flow theory. We shall denote by $\Phi(x,y,z;t)$ the velocity potential which depends on the nondimensional space variables (x,y,z) and the time variable t. The fluid velocity at a field point $M(x,y,z)\,;z \leq 0$ is then given by $\mathbf{V} = \nabla\Phi$. All space variables will be reduced by a typical length h, and the time variables by $\sqrt{h/g}$, g being the gravity acceleration.

We shall focus here on the impulsive wave radiation problem which may be formulated as : given a floating body at rest in its equilibrium position $\mathcal{C}$, it is impulsively set into motion, at $t = 0$, in one of its six degrees of freedom. The resulting velocity potential associated with the waves radiated from the body will be the unknowns of the problem. Let us denotes by $\mathbf{V}_i$ the velocity vector for a motion on the ith mode.

The velocity potential is sought as the solution, in the fluid domain $\mathcal{D}$, of the following initial boundary value problem (IBVP).

Laplace's equation

$$\Delta\Phi(x,y,z;t) = 0 \quad ; M \in \mathcal{D},\, t \geq 0 \qquad (1)$$

Free-Surface condition:

$$\frac{\partial^2\Phi(x,y,0;t)}{\partial t^2} + \frac{\partial\Phi(x,y,0;t)}{\partial z} = 0 \quad ; t \geq 0 \qquad (2)$$

No-Flux body boundary condition:

$$\frac{\partial\Phi(x,y,z;t)}{\partial n} = \mathbf{V}_i.\mathbf{n} \quad ; M \in \mathcal{C} \qquad (3)$$

Initial conditions:

$$\Phi(x,y,z;t) = \frac{\partial\Phi(x,y,z;t)}{\partial t} = 0 \quad ; M \in \mathcal{D},\, t \leq 0 \qquad (4)$$

Let us now introduce the Green function of the impulsive source problem which satisfies intrinsically Laplace's equation (1), the linearized free surface condition (2) and the initial conditions (4). It may be written as the sum of an impulsive part and a memory part, like :

$$G(M,t,M',t') = -\frac{1}{4\pi}\left[\delta(t-t')G_0(M,M')\right]$$

$$-\frac{1}{4\pi}\left[H(t-t')\mathcal{F}(M,t,M',t')\right] \qquad (5)$$

where δ and H refer respectively to the Dirac and to the Heaviside distributions. Full expressions of this function, together with some interesting differential properties, will be given in the next section. Applying the Green's formula to the unknown potential Φ and to this function yields the following Fredholm-Volterra integral equation :

$$\frac{\Phi(M,t)}{2} - \iint_{\mathcal{C}} \Phi(M',t)\frac{\partial}{\partial n'}G_0(M,M')d\mathcal{C} =$$

$$-\iint_{\mathcal{C}} G_0(M,M')\mathbf{V}_i.\mathbf{n}(M',t)d\mathcal{C}$$

$$+\int_0^t dt' \iint_{\mathcal{C}} \left[\Phi(M',t')\frac{\partial}{\partial n'}\mathcal{F}(M,t,M',t')\right.$$

$$\left. -\,\mathcal{F}(M,t,M',t')\mathbf{V}_i.\mathbf{n}(M',t')\right] d\mathcal{C} \qquad (6)$$

A direct Boundary Element Method may then be derived from this integral equation by discretizing the body surface into plane panels, and representing the unknown functions over each panel by a suitable functional approximation. The details of the numerical solution of (6) will be presented in a later section.

The j^{th} component of the transient hydrodynamic forces consecutive to the i^{th} impulsive motion can be computed afterwards by integrating the potential over the body surface, namely :

$$M_{ij}(t) = \iint_{\mathcal{C}} \Phi_i(M',t)n_j(M')d\mathcal{C} \qquad (7)$$

and differentiating with respect to the time variable to return to the pressure on the body surface :

$$L_{ij}(t) = \iint_{\mathcal{C}} \frac{\partial\Phi_i(M',t)}{\partial t}n_j(M')d\mathcal{C} = \frac{\partial M_{ij}(t)}{\partial t} \qquad (8)$$

This *impulse response function* (IRF) is homogeneous to a force and will be given, in the following sections, as a coefficient defined by : $CL_{ij} = L_{ij}/\rho g h^3$.

Before examining the numerical method for the solution of the boundary integral equation (6), let us first recall some important results about the impulsive time-domain Green function and its differential properties.

THE TIME-DOMAIN GREEN FUNCTION

The function (5) which satisfies (1), (2) and (4) is the basic element of the so-called Kelvin BEM methods. Its impulsive part G_0 is nothing but the

free space Green-function associated with its anti-mirror image about the linearized free-surface, namely :

$$G_0(M,M') = \frac{1}{R} - \frac{1}{R_1} \qquad (9)$$

where : $r = \sqrt{(x-x')^2+(y-y')^2}$, $R = \sqrt{r^2+(z-z')^2}$, $R_1 = \sqrt{r^2+(z+z')^2}$
while the memory part is given by :

$$\mathcal{F}(M,t,M',t') = F(r,\zeta,t) = R_1^{-\frac{3}{2}}\widetilde{F}(\mu,\tau) \qquad (10)$$

with $\widetilde{F}(\mu,\tau) =:$

$$2\int_0^\infty J_0\left(\lambda\sqrt{1-\mu^2}\right)\mathrm{e}^{-\lambda\mu}\sqrt{\lambda}\sin\left(\sqrt{\lambda}\tau\right)d\lambda \qquad (11)$$

The set of variable $(r,\zeta = z+z',t)$ will be referred to as the *initial variables*, while (μ,τ), defined as :

$$\mu = -\frac{z+z'}{R_1} \quad and \quad \tau = t/\sqrt{R_1} \qquad (12)$$

will be named the *natural variables* of the function. One may notice that the geometric parameter μ lies in the bounded range $[0,1]$. Approaching this limit, the function is more and more oscillating but remains convergent at infinity, as illustrated by fig.1. In the limit $\mu = 0$, where both source and field points belong to the free-surface $z = 0$, and only in this case, it is divergent, linearly in t.

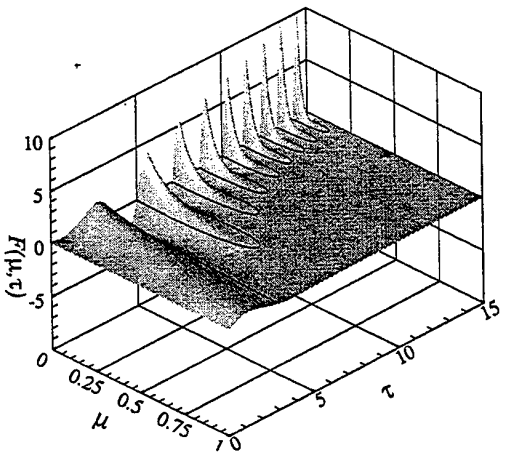


Figure 1: The Green function (eq.11) in natural variables (μ,τ)

In a recent paper [5], we derived a general lemma stating that the family of functions $A_{\nu,l}(\mu,\tau)$ of the form:

$$A_{\nu,l}(\mu,\tau) = \int_0^\infty \lambda^l e^{-\lambda\mu}J_\nu(\lambda\sqrt{1-\mu^2}\sin(\sqrt{\lambda}\tau)d\lambda$$

satisfy a fourth order differential equation with polynomial coefficients of second degree with respect to the time variable. The time-domain Green function and its space derivatives belonging to the family, they are shown to satisfy the following ODEs:

$$(r^2+\zeta^2)\frac{\partial^4 F}{\partial t^4} - \zeta t\frac{\partial^3 F}{\partial t^3} + \left(\frac{t^2}{4} - 4\zeta\right)\frac{\partial^2 F}{\partial t^2} + \frac{7t}{4}\frac{\partial F}{\partial t} + \frac{9}{4}F = 0 \qquad (13)$$

$$(r^2+\zeta^2)\frac{\partial^4 F_r}{\partial t^4} - \zeta t\frac{\partial^3 F_r}{\partial t^3} + \left(\frac{t^2}{4} - 6\zeta\right)\frac{\partial^2 F_r}{\partial t^2} + \frac{11t}{4}\frac{\partial F_r}{\partial t} + \frac{21}{4}F_r = 0 \qquad (14)$$

$$(r^2+\zeta^2)\frac{\partial^4 F_\zeta}{\partial t^4} - \zeta t\frac{\partial^3 F_\zeta}{\partial t^3} + \left(\frac{t^2}{4} - 6\zeta\right)\frac{\partial^2 F_\zeta}{\partial t^2} + \frac{11t}{4}\frac{\partial F_\zeta}{\partial t} + \frac{25}{4}F_\zeta = 0 \qquad (15)$$

where F_r (resp. F_ζ) denotes the horizontal (resp. vertical) gradient of $F(r,\zeta,t)$. The initial conditions, also derived in their general form in [5], become in this case:

$$\begin{cases} F(r,\zeta,0) = 0 \\ \frac{\partial F}{\partial t}(r,\zeta,0) = -2\frac{\zeta}{(r^2+\zeta^2)^{\frac{5}{2}}} \\ \frac{\partial^2 F}{\partial t^2}(r,\zeta,0) = 0 \\ \frac{\partial^3 F}{\partial t^3}(r,\zeta,0) = 2\frac{r^2-2\zeta^2}{(r^2+\zeta^2)^{\frac{5}{2}}} \end{cases}$$

$$\begin{cases} F_r(r,\zeta,0) = 0 \\ \frac{\partial F_r}{\partial t}(r,\zeta,0) = 6\frac{r\zeta}{(r^2+\zeta^2)^{\frac{5}{2}}} \\ \frac{\partial^2 F_r}{\partial t^2}(r,\zeta,0) = 0 \\ \frac{\partial^3 F_r}{\partial t^3}(r,\zeta,0) = 6\frac{r(4\zeta^2-r^2)}{(r^2+\zeta^2)^{\frac{7}{2}}} \end{cases} \qquad (16)$$

$$\begin{cases} F_\zeta(r,\zeta,0) = 0 \\ \frac{\partial F_\zeta}{\partial t}(r,\zeta,0) = \frac{4\zeta^2-2r^2}{(r^2+\zeta^2)^{\frac{5}{2}}} \\ \frac{\partial^2 F_\zeta}{\partial t^2}(r,\zeta,0) = 0 \\ \frac{\partial^3 F_\zeta}{\partial t^3}(r,\zeta,0) = -6\frac{\zeta(3r^2-2\zeta^2)}{(r^2+\zeta^2)^{\frac{7}{2}}} \end{cases}$$

We shall see in a later section that the major part of the computation time required to solve the

integro-differential integral equation (6) is spent in evaluating this function (and its gradient) a huge number of time ($\tilde{}O(10^8)$). Then, the choice of the method for the calculation of F, F_τ and F_ζ requires a particular attention if one wants to keep the global *cpu*-time within reasonable limits. Two families of methods were available up to now. The new method proposed here constitutes a third alternative .

•***series expansions method***. Obviously, the original expression (11) is not well suited for direct numerical evaluation; then, from the early eighties, several authors [12, 15, 13, 20, 19] have developed numerical procedures to compute the Green function in its natural variables. The best choice among all the available methods is local in both space and time; it depends on the values of μ and τ. Asymptotic expansions are used in the large time range, series expansions for moderate μ and τ, Filon quadrature for moderate τ and larger μ, and finally recursive Bessel relations in the vicinity of $\mu = 1$ (see [13] for details). Following B.W. King (1987), a set of subroutines devoted to this task has been developed in our laboratory, and will be used in the sequel as the first alternative for the numerical solution of (6).

•***tabulation method***. A second numerical method for the evaluation of (11) is based on a bi-linear interpolation in a pre-computed table [9, 16]. This is made possible by the fact, first pointed out by Jami (1981), that the Green function is a function of only two natural variables (μ, τ), the first one varying in a bounded domain. Because the maximum of the τ variable reached during a given computation is not known a priori, the table must be quite large in that direction, and the preceeding (series expansion) method must be also available in case of the function should be evaluated beyond the table boundary τ_{max}. Nevertheless, we shall see later that this method yet allows considerable saving in *cpu*-time compared to the first one.

•***ODE-integration method***. Because eq. (6) is to be solved sequentially in a time marching procedure from given initial conditions, the above set of ODEs may be used to update the right hand side from a time step to the next one rather than evaluating the Green function by one of the two above mentionned methods as usual. Runge-Kutta ODE integration algorithms are used for this purpose in the present study. Details about the stability of this new method will be given in a forthcoming section.

SOLUTIONS TO THE BOUNDARY INTEGRAL PROBLEM

Since the early eighties, several authors have proposed numerical algorithms to solve the above boundary integral equation, or some variants[15, 13, 14, 9, 17, 2, 1]. From (6), one may consider the potential in the fluid domain as generated by a distribution of both sources and doublets with density distributions : $\sigma = \frac{\partial \Phi}{\partial n}$, resp $\nu = -\Phi$, on the body surface $\mathcal{C}$. Discretizing this surface into simple sub-elements and approximating the unknown functions on each one by some low-order functional, the continuous integral equation (6) is transformed into a discrete set of linear algebraic equations which can be solved numerically afterwards. Various order of approximations for both the type of geometric sub-element and the functional representation of the solutions may be found in the related literature.

The present implementation of the method follows a previous work done by Ferrant (1988b), who used a zeroth order direct BEM method with constant singularity distributions (σ, ν) over quadrilateral and/or triangular plane panels. Because his code has served as a starting basis in the present study, this constant distribution option was kept . Furthermore, a step velocity excitation of the body hull, $\mathbf{V_i} = \mathbf{n_i} \mathrm{H}(t)$, is considered here for numerical convenience; the hydrodynamic responses for an actually impulsive velocity input then follow straithforwardly by simple differentiation. Due to the constant density distribution over the panels, the discretized integral equation (6) is written at the centroid of each panel, leading to the discrete set of linear algebraic equations :

$$[D_{ij}]\,[\nu_j(t)] = -\mathrm{H}(t)\,[S_{ij}]\,[\sigma_j(0)] -$$

$$- \int_0^t [s_{ij}(t')]\,[\sigma_j(0)]\,dt' + \int_0^t [d_{ij}(t-t')]\,[\nu_j(t')]\,dt' \tag{17}$$

where :

$$D_{ij} = \iint_{\mathcal{C}_j} \frac{\partial}{\partial n_j} G_0(M_i, M_j)\, d\mathcal{C}_j$$

$$S_{ij} = \iint_{\mathcal{C}_j} G_0(M_i, M_j)\, d\mathcal{C}_j \tag{18}$$

$$d_{ij}(t) = \iint_{\mathcal{C}_j} \frac{\partial}{\partial n_j} \mathcal{F}(M_i, t, M_j, 0)\, d\mathcal{C}_j$$

$$s_{ij}(t) = \iint_{C_j} \mathcal{F}(M_i, t, M_j, 0) dC_j \qquad (19)$$

The constant coefficient matrices $[D_{ij}]$ and $[S_{ij}]$ are computed by the classical Hess & Smith formula. Being time independent, they can be evaluated once for all at the onset. Furthermore, $[D_{ij}]$ is inverted once for all by a Gaussian procedure, and stored.

The time variable is also discretized into equal time-steps. The solution of (17) at each time step is then obtained by a simple matrix product after updating the RHS. This later task requires the computation of convolution products of the past solution on the body with the Green function and its gradient. The convolution integrals are computed by a simple trapezoidal rule, and, in this very first application of the method, the surface integrals in (19) are computed by means of single point Gauss quadrature. Despite of these low order algorithms, these computations remain the most time consuming part of the overall numerical process due to the complexity of the Green function [14, 17]. All the memory coefficients (19) from the first to the current time step are theoretically needed to perform the time integration in (17). The best choice is therefore to keep all these coefficient matrices in RAM memory during the execution, as long as the computer memory size permits doing so; when it does not, which essentially occur when running the *tabulation* method on a small computer, we are constrained to store them on disk, resulting in slowing down the program by multiplicating disk input/output (I/O) (see fig.9left).

In the present study, the three above mentioned computational methods were implemented. In the *series expansion* approach, several methods are used to compute the Green function in its natural variables, depending on the values of μ and τ. These methods do not share the same performances in terms of *cpu*-time, and during the same time step, the choice of an algorithm may be different from one couple of points to an other due to the dependency of the time variable $\tau = t/\sqrt{R_1}$ upon the relative position of field point and source point. The local *cpu* time thus depends on time and meshing, making the global *cpu* time quite unpredictible.

In the *tabulation* method, the Green function $\widetilde{F}(\mu, \tau)$ is precomputed on a grid of the (μ, τ) plane and stored in a permanent file. In the present study, a $(1000, 24000)$ grid was used with $\delta\mu = 0.001$ and $\delta\tau = 0.005$, resulting in a 96Mbyte file in single precision, in order to cover the whole time-range in the simulations. This long time range tabulation avoids the algorithm to switch to asymptotic expansion method in long time simulations, and permits a better comparison with the ODE-*integration* method afterwards. The table has to be implemented in RAM memory during execution to avoid to much disk access ; then, depending upon the total memory available to him, the user may be obliged to store the memory coefficient matrices (19) on disk files as mentioned above. We happened to be in that case in the computations reported herein, as we shall see later.

STABILITY AND CONVERGENCE

ODE integration scheme

The new method proposed above for the online evaluation of the Green function is based on the numerical integration of ordinary differential equations featuring time-varying polynomial coefficients. In the wide choice of numerical schemes available to perform this task, we payed a special attention to the class of Runge-Kutta algorithms which has the advantages of being sufficiently robust and which does not require the knowledge of previous values to advance in time as it is the case with predictor-corrector method. Furthermore, the evaluation stage of the ODEs right hand side, in the present case, is so unexpensive (in term of cpu time indeed) that economizing the number of evaluations is not a key point as it may be in other circumstances. (see matrix eq.20).

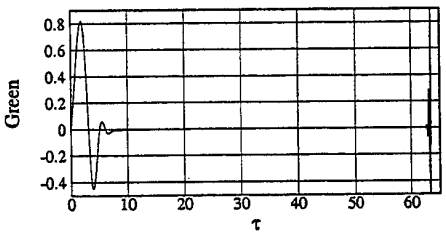


Figure 2: Typical instabilities in the determination of the Green function by ODE integration ($\mu = 0.5$, $d\tau = 0.1$), method=RK4

The most important point to be investigated carefully here is the stability of the integration scheme which must be ensured whatever the kind of body being tested and the level of mesh re-

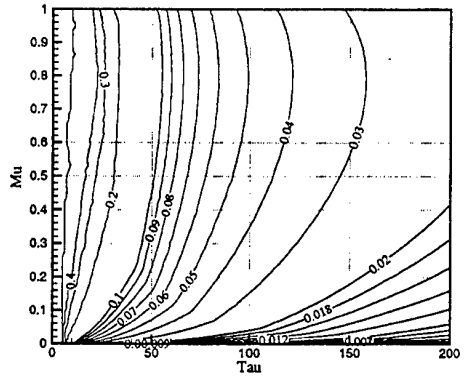


Figure 3: stability limits RK2 method

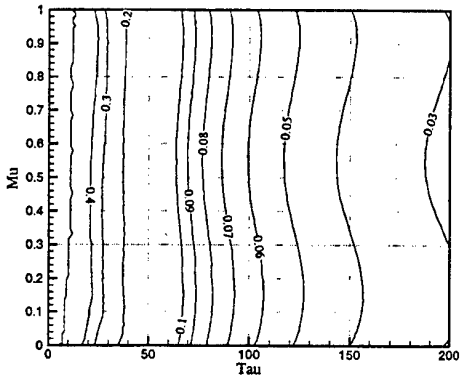


Figure 5: stability limits RK4 method

finement.The figure (2) shows a typical instability occuring at $\tau \approx 62$ when computing $\widetilde{F}(0.5, \tau)$ by a standard fourth order Runge-Kutta method. Such an instability will occur systematically at some (large) value of the time variable τ for a given value of the geometrical parameter μ. Then, before using this method to compute the whole set of s_{ij} and d_{ij} influence coefficients, one must address the problem of finding the optimum time step to avoid such a phenomenon in a given computation range $0 \leq t \leq t_{max}$, for a given meshing of the floating body.

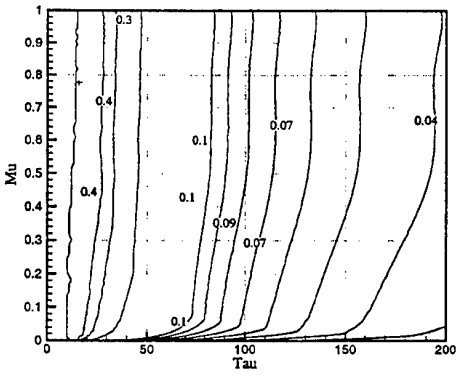


Figure 4: stability limits RK5 method

Fisrt of all, one must remember that the Green function is a function of two variables in the system of natural variables (μ, τ) defined by (12), and of three variables (r, ζ, t) when we considered the function in its initial set of variables. The computation of the convolution integral in (17) must be performed in the later one while the stability can be investigated in the former one. The natural variable τ depends on the time variable t through the relation $\tau = t\left(r^2 + \zeta^2\right)^{-\frac{1}{4}}$; therefore, it is scaled by the space variables depending on the couple of source and field points being considered. It means that the maximum (natural) time step $\Delta\tau_{max}$ determined by the stability analysis has to be converted into a maximum (system) time step Δt_{max} by considering the minimum value of the distance R_1 among all the couples of points defined by the hull panelization.

Usually, the stability analysis of differential equations starts from examining some basic features of the matrix [A] defined by expressing the high order differential equation as a system of first order ODE, namely:

$$\dot{\mathbf{F}} = [\mathbf{A}]\,\mathbf{F}$$

For the Green function ODE (13) under consideration, the so-called companion matrix [A] would read:

$$[\mathbf{A}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{9}{4} & \frac{-7\tau}{4} & \left(4\mu - \frac{\tau^2}{4}\right) & -\mu \end{bmatrix} \qquad (20)$$

The most common stability criterion requires all the eigenvalues of [A] to have a negative real part. But this theorem holds only for constant coefficients matrix systems, and not for varying matrices as in the present case (see [18] p113). This prevent us to proceed to the standard analysis straigthforwardly.

Nevertheless, this point is not crucial because the Green function, which is the solution we are

concerned with, is known to be stable everywhere except at $\mu = 0$. Furthermore, this parameter μ belongs to the bounded range $[0, 1]$.

The stability of the various Runge-Kutta integration algorithms was therefore investigated from this argument, by numerical experiments in the (μ, τ) plane. Three schemes were tested: a second order (RK2) a fourth order (RK4) and a fifth order (RK5). The first and the second one are the classical algorithms which can be found in the literature. According to Press et al. [21] we chose the set of Cash-Karp parameters for the fifth order scheme. This permits a precise time step optimization by using an embedded fourth order formula and the Fehlberg method. The coefficients for this fifth order scheme are given in Appendix. With each method, the parameter range $0 < \mu < 1$ was thoroughly swept, and the time step leading to the stability limit was determined numerically for each couple (μ, τ_{max}) by detecting instabilities like in fig:2. The corresponding results are plotted in figures 3, 5 and 4 for RK2, RK4 and RK5 methods respectively. In these plots, the labels on the curves indicates the time step. As it can be observed, we recover the expected behaviour : the smaller the time step, the later the divergence for a given RK scheme, and also : the higher the scheme order, the later the divergence for a given time step. Considering now the behaviour in the vicinity of the axis $\mu = 0$, the fourth order scheme appears to be far more robust than the two others. This point is important and it must be kept in mind when one designs the time-step maximization algorithm, as we did in the following manner. As a starting point, the mesh and the maximum simulation time are given by the user. From the former, one may derive the minimum value of R_1, and then, from both of them, a map of all the points in the

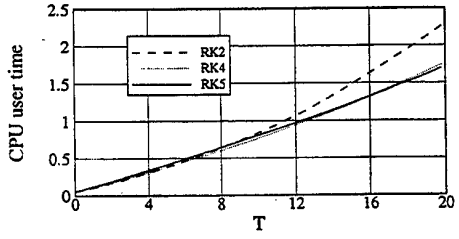


Figure 6: Heaving hemisphere. comparison of overall cpu time using various RK schemes to integrate the Green function ODEs.

scheme	maximum time-step	local error
RK2	.083	$O(7.10^{-3})$
RK4	.11	$O(1.5 \cdot 10^{-4})$
RK5	.15	$O(7.6 \cdot 10^{-5})$

Table 1: stability limit time-steps and error magnitude for the three considered Runge-Kutta methods. 4*25 panels, floating hemisphere.

(τ_{max}, μ) plane may be established. Now, for a given RK scheme, a $(d\tau, \mu)$ map may be drawn with the help of the corresponding figure 3, 5 or 4 depending on the algorithm being considered; returning, at last, to the (dt, μ) map, the smaller value of dt ensuring the stability over the whole mesh can be determined easily.

A preprocessor of our time-domain code **ACHIL3D** was built on this model. When applied to the heaving hemisphere test problem discretized into 4*25 panels, it gave the stability limit time-steps shown in table 1. Nevertheless, the final choice of a method cannot be based on these items only. The natural advantage of RK5 should be moderated by the fact that this scheme needs six evaluations of the RHS of the ODE while it is four with RK4 and only two with RK2. Therefore, the comparisons must be done on the overall computation time to obtain a more objective ranking. Figure 6 finally shows that RK4 and RK5 are practically equivalent in terms of *cpu* time when each method is run at its stability limit time-step (minus epsilon of course!). The choice left to the user is therefore based on other arguments like time-integration accuracy, which gives the leadership to RK5, or robustness of the time-step maximization which rather favours RK4 (compare fig.5 and 4). Furthermore, the accuracy argument in favour of the fifth order scheme, should be moderated by consistancy consideration, if we remember that we have chosen a zeroth order BEM with one Gauss point by panel for this first implementation of our ODE method . The two RK algorithms were finally implemented in our code in such a way that the user may easily switch from one to the other; the fourth order algorithm is the most commonly used.

Convergence with mesh refinement

Being a well documented test case, the floating hemisphere was chosen as the first model body for the convergence tests. The two geometric symmetries of this simple hull were naturally taken into

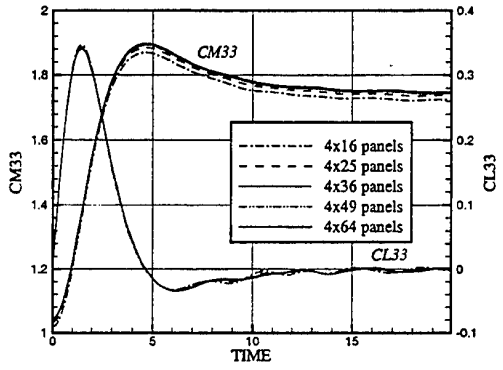


Figure 7: Floating hemisphere : impulse response function / mesh refinement

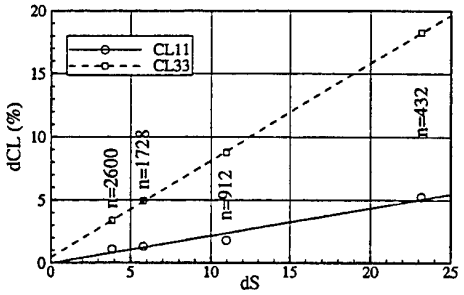


Figure 8: ISSC TLP impulse response fonction in heave and sway. Global convergence with respect to typical panel size (n the total number of panels)

account and only a quarter-body was meshed. The impulse response function for an impulsive heave motion CL_{33} was selected as the output for checking the overall accuracy. Several different meshing were used with 16, 25, 36, 49 and 64 panels on a quarter-body, and the convergence with regard to the mesh size (characterized by the average panel area) was investigated. The curves of $CL_{33}(t)$ (eq.8) and $CM_{33}(t)$ (eq.7) are plotted in figure 7. The accuracy was checked by comparing with the same result obtained with 144 panels by Korsmeyer (1988) [14]. An excellent agreement was observed. Zooming around the first peak at $CL_{33}(1.475)$ (see fig.7) revealed that a 1% local relative error was achieved with 100 panels (4x25) on the hemisphere. The oscillations observed in the tail of the response for $t \geq 6$ are the time-domain counterpart of the well known "irregular frequencies", and arise from the same origin. Since we were mainly interested here in *cpu* time statistics, we did not tried to suppress this phenomenon by the help of the usual dedicated methods. This is left as a further development of the code. A linear convergence rate was observed when plotting the error on the impulse response fonction CL as a function of the typical average area of the panels (at the peak value where a maximum discrepancy is expected). The same convergence tests were also achieved on the ISSC TLP and the results are plotted in figure 8 which illustrates this linear behaviour perfectly well.

CPU-TIME SAVING WITH ODE METHOD

The fist goal of this study was to assess the feasibility of the ODE-*integration* method. The results presented in the previous section show that it has been reached. The second one was to give a first estimate of the savings it can bring in terms of computing time. The three methods were then implemented in three parallel "brother" codes differing only by this point, and tested intensively varying the mesh, the time-step, the Runge-Kutta method order,... Impulse response functions were computed not only for the floating hemisphere, but also on the ISSC TLP platform (fig.12) and a standard Series60-06 ship hull (fig.14).

To sets of curves are plotted in Fig:9. On the left side, the total *cpu*-time including the system part is shown, whereas the user part only is plotted in the right figure. The difference between these two measures is spent mainly in I/O operations to and from the disk storage. As pointed earlier, we were obliged to write the coefficient matrices (19) on disk files when using the *tabulation* method, whereas they stayed in memory in the two other approaches. This explains the differences between the dashed lines in the left and in the right frames of figure 9. For the two other methods, the time spent in I/O is negligible and the curves are quite the same in both sides. In other words, one may assume that the right view of figure 9 can always be obtained provided one run a computer with a sufficient amount of memory. Let us thus focus on this view as a basis for comparisons.

The structure of the convolution process sug-

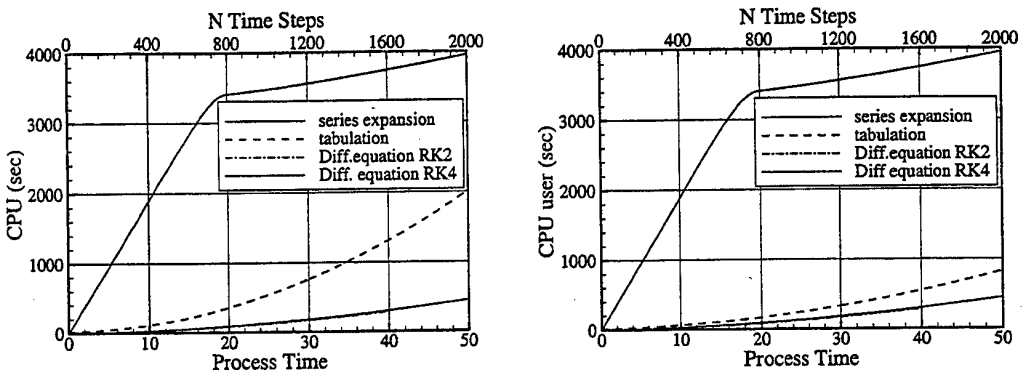


Figure 9: Floating Hemisphere (4x49 panels; δt=0.025) ; **left** : total *cpu*-time (user+I/O) ; **right** : user *cpu*-time only.

gests a quadratic growth of the computing time. Such a behaviour can be observed with both the *tabulation* and the ODE-*integration* methods, whereas the series expansion method presents two different regimes. For $t < 18$, we observe a quasi-linear growth of the *cpu*-time. In that range, the Green function is evaluated by different algorithms, according to the relative position of source and field points as explained before. These methods are far more time consuming than the asymptotic expansion which is used later, when $\tau \geq 14$. So, once all the couples of points satisfy this condition, the program speeds up and a quadratic behaviour is recovered. The benefit of using the two other methods is clearly illustrated by Fig:9. Between them, the advantage go to the ODE-*integration* method ; the *cpu*-time ratio with the *tabulation* method lies in the range [2,4], depending on wether or not you can keep all the arrays in RAM during the program execution.

In the present comparison all the methods were run using the same constant time-step $dt = 0.025$. We limit ourselves to RK2 and RK4 methods; RK5 was disregarded. The fourth-order, four steps Runge-Kutta method requires twice as many floating operations as the second order two steps method. Nevertheless, the tiny difference between the RK4 and RK2 curves in Fig:9 proves that, with the ODE-*integration* method, the time spent in Green function evaluations has became marginal compared with the time for the overall process. On the contrary, it means that, by selecting RK5 algorithm, we could increase the precision of the ODE integration at practically no extra *cpu* cost.

One should notice that these curves correspond to quite long simulations. In the present case of a heaving hemisphere, a simulation up to $t = 20$ should be sufficient from a practical point of view (see Fig:7). It would result in computing times shown in the table below, when running a DEC Alpha 500 workstation, at 330MHz.

Method	user+I/O	user (sec)
Series	3408	3405
Tabulation	356	164
ODE-RK4	88	87
ODE-RK2	82	82

Table 2. cpu-time requirements (sec), floating hemisphere (4x49 panels), t=20.

The ISSC Tension Leg Platform

The computer code was used afterwards to compute the IRFs of a more realistic body. Results for the ISSC platform in heave and surge motions were presented by Ferrant (1988)[9]. We therefore selected it as a test geometry for our code **ACHIL3D**. Several meshes were generated using the mesh generator **MACAO**. All the results reported here were obtained with a 912 panels mesh illustrated by Fig:12. The fourth-order Runge-Kutta integrator was used with a fixed time step $\delta t = 0.02$.

As a check for the computer code, our surge/surge impulse response function CL_{11} (fig.10a) was compared with the results obtained by Ferrant using a 1200 panels mesh; the agreement was found excellent.

In figures 10b and 11, the diagonal IRF for heave, roll and yaw motions are presented together with their Fourier transform which are nothing but the classical *added mass* and *damping* coefficients of the frequency domain approach to

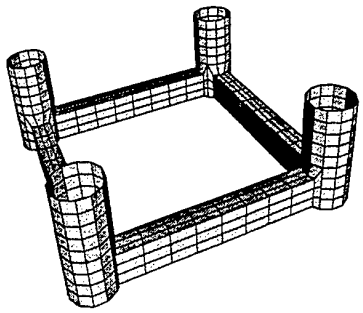


Figure 12: Meshing of the ISSC TPL. (912 panels)

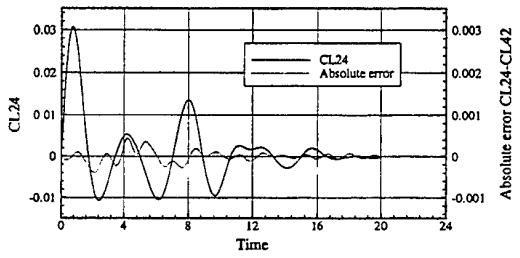


Figure 13: ISSC Tension Leg Platform - 912 panels Sway/Roll IRF CL24 ; symmetry (absolute) error

the seakeeping problem, the correspondence being given by :

$$A_{ij}(\omega) = M_{ij}(0) + \int_0^\infty L_{ij}(t)\cos(\omega t)dt$$

$$\frac{B_{ij}(\omega)}{\omega} = \int_0^\infty L_{ij}(t)\sin(\omega t)dt \qquad (21)$$

These frequency domain coefficients CA_{ij} CB_{ij}, deduced from the present time-domain approach by Fourier transformation have been compared to those obtained directly by running frequency domain panel codes like **AQUADYN** (ECN) and **WAMIT** (MIT) (data in [10]). The agreement is again excellent, within a few percent. In figure 13, the off-diagonal IRF coefficients CL_{24} and CL_{42} and the difference between them (ten times magnified) are presented. These coefficients should theoretically be equal; the gap between is therefore a good measure of the overall accuracy of the computation method. In the present case, the maximum difference is approximately one percent of the maximum of the CL_{24}/CL_{42} curve.

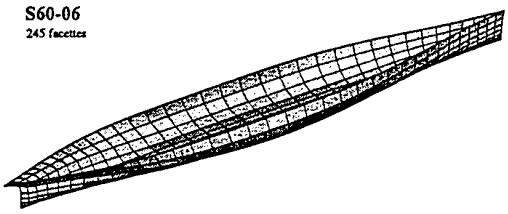


Figure 14: S60-06 meshing; 2x245 panels

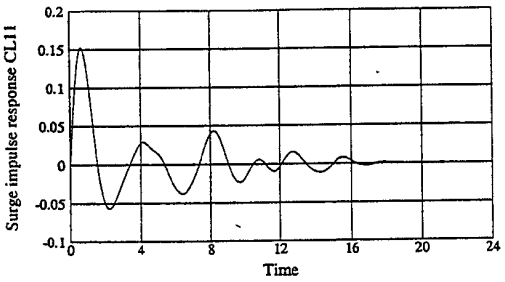
SERIES60-06 SHIP HULL

The bodies used in the previous section have two symmetry plane. This property was naturally exploited to reduce the number of unknowns, hence the system matrix size. The computer code was afterwards extended to bodies with a single symmetry plane. The standard Series 60-06 ship (fig.14) hull was then used as a test case for this further release. The hull was panelized with 2x245 plane quadrilateral panels.

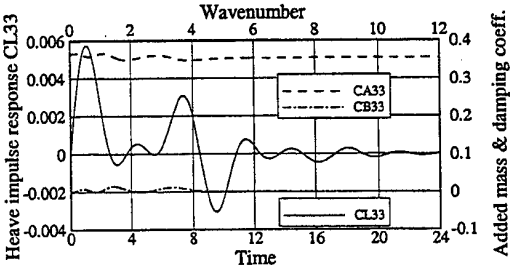
The optimal time step determined by the preprocessor as explained previously was found to be $\delta t = 0.00484$ when the ship length is taken as the reference length. A RK5 scheme was used in the reported computations. The impulse response functions were computed up to $T = 10$ which was found sufficient to reach a quasi-null response state as shown in fig.15 to 20. These time-domain results were then Fourier transformed by (21) in order to be compared to those obtained directly in the frequency domain by the diffraction-radiation code **AQUADYN** [7, 8] developped at LMF laboratory in the eigthies. In terms of order of approximation, this code is equivalent to **ACHIL3D** but in the frequency domain. It solves the linearized (to the first order) seakeeping problem by a zeroth-order BEM featuring a mixed sources-dipoles distribution of Kelvin type Green function on the discretized hull.

Time-domain and frequency domain results are plotted in figures 15 to 20 hereafter. We give only diagonal coefficients here, to save space, but the whole set of non-zero responses has, of course, been computed.

The fit between the frequency domain and the time-domain approaches is generally excellent ($O(10^{-2})$), except near the "irregular" frequencies already mentionned in this paper. This phenomenon is particularly sensitive on the frequency-domain results for heave and pitch motions, in figures 17 and 19. These results con-

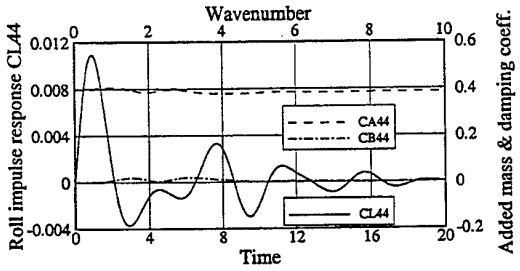


(a) CL11. Surge/Surge IRF

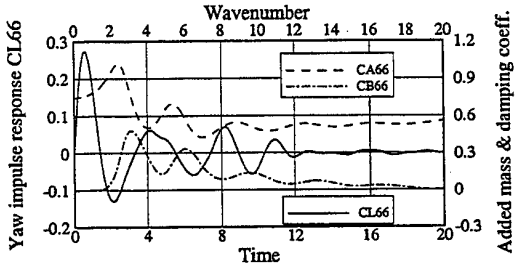


(b) CL33, CA33 and CB33 Heave/Heave coefficients

Figure 10: ISSC Tension Leg Platform - 912 panels



(a) CL44, CA44 and CB44 Roll/Roll coefficients



(b) CL66, Yaw/Yaw coefficient

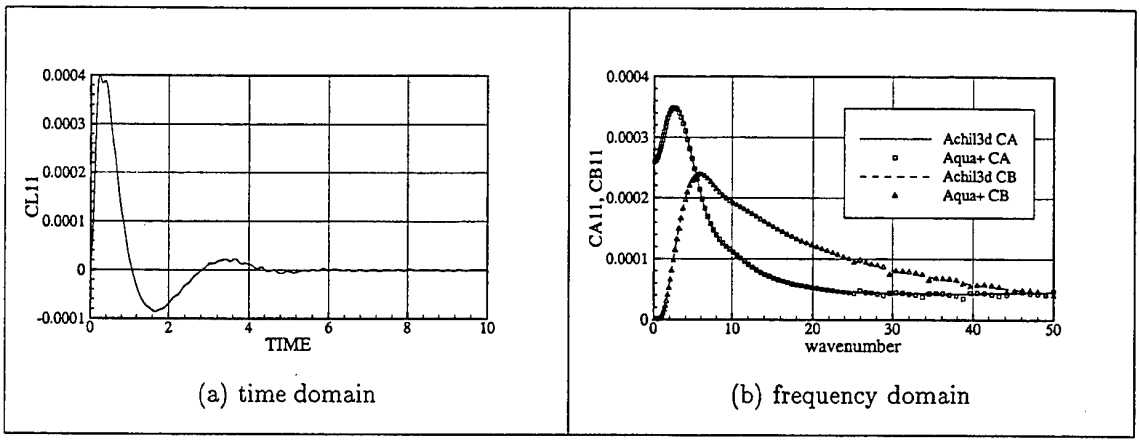
Figure 11: ISSC Tension Leg Platform - 912 panels

firm that the present time-domain approach can be used as a robust alternative method to obtain the usual frequency domain hydrodynamic coefficients. The opposite is not true due to the slow asymptotic decrease of these coefficients as the wavenumber tends to infinity (see fig.15 to 20). The inverse Fourier transform which has to be performed numerically from these data imposes a truncation at very high frequencies where the convergence of the frequency domain solution would surely be problematic.

CONCLUSION

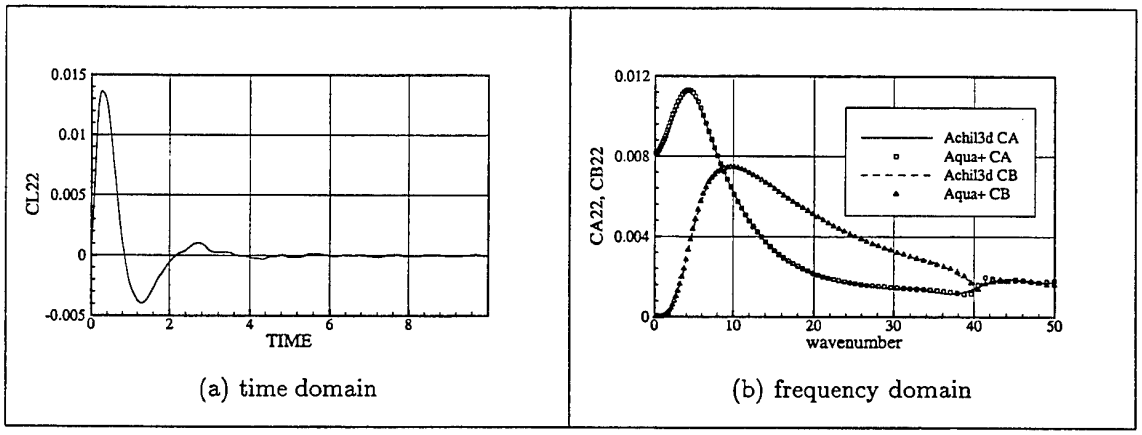
A new method for the calculation of the Green function during the computation of convolution integrals occuring in time-domain seakeeping simulations has been proposed. It is based on differential properties of the time-domain Green function which are used to speed up this numerical process. The performance, in terms of *cpu* time requirements, of the new alternative method was compared with the two usual approaches to the problem, based either on series expansions or tabulation of the Green function. The proposed ODE-*integration* method runs faster than the tabulation method, in a ratio between two and four, depending on the memory capacity of the computer. The stability analysis of the Green function ODE provides a mean to optimize the choice of the time step depending on the RK algorithm finally selected. The accuracy of the code has been checked by tests on well documented bodies as the hemisphere, the ISSC TLP platform and a serie60-06; the frequency domain coefficients of these bodies were recovered by Fourier transform with an excellent accuracy.

This work was financially supported by the *Direction des Recherches, Etudes et Techniques* of the French Ministry of Defence (DGA).



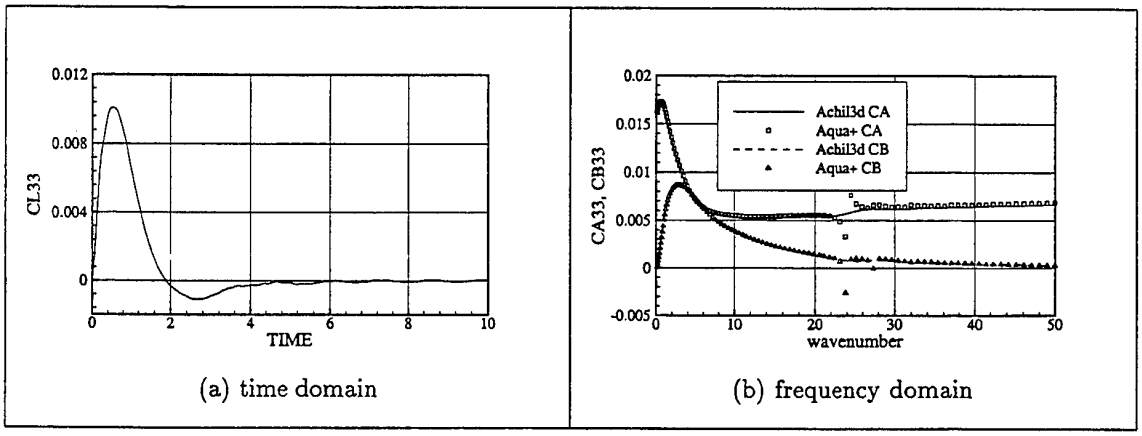
(a) time domain (b) frequency domain

Figure 15: Series 60-06 (2x245 panels). Surge/Surge hydrodynamic coefficients



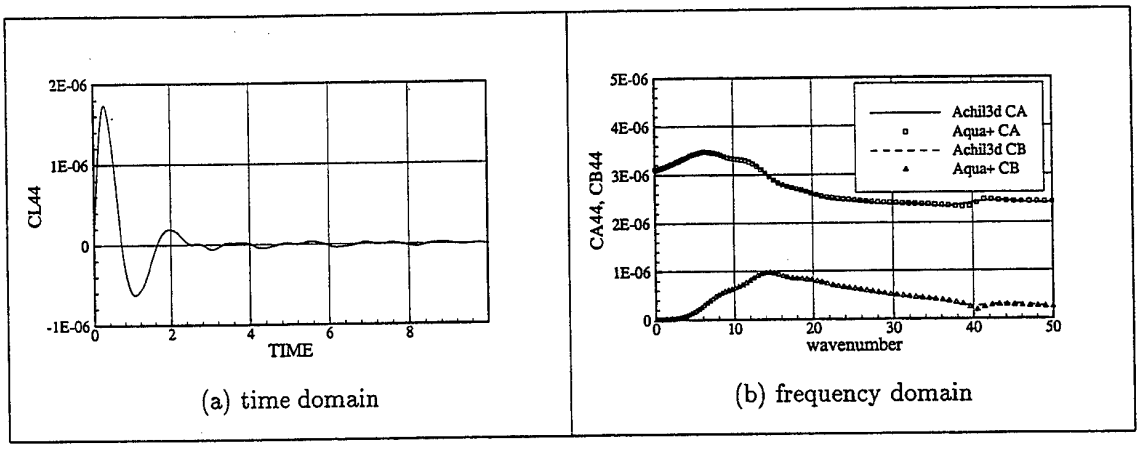
(a) time domain (b) frequency domain

Figure 16: Series 60-06 (2x245 panels). Sway/Sway hydrodynamic coefficients



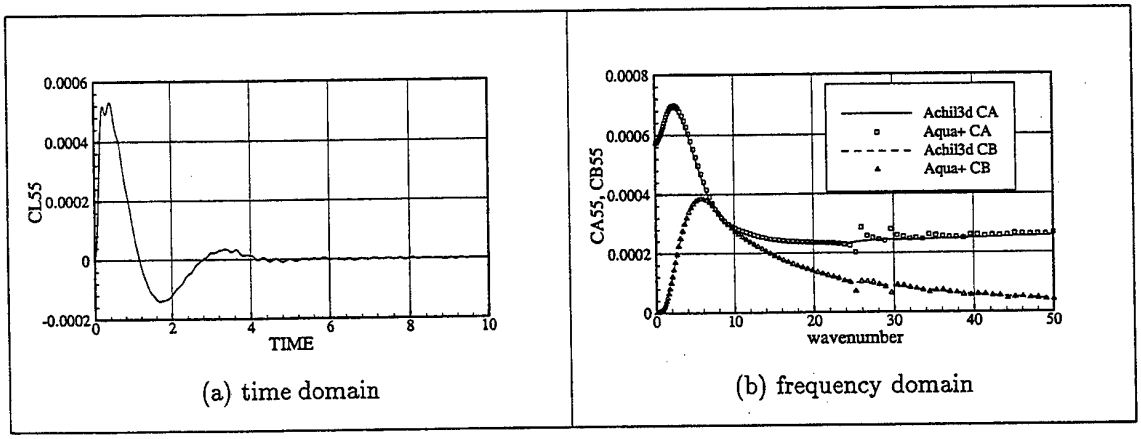
(a) time domain (b) frequency domain

Figure 17: Series 60-06 (2x245 panels). Heave/Heave hydrodynamic coefficients



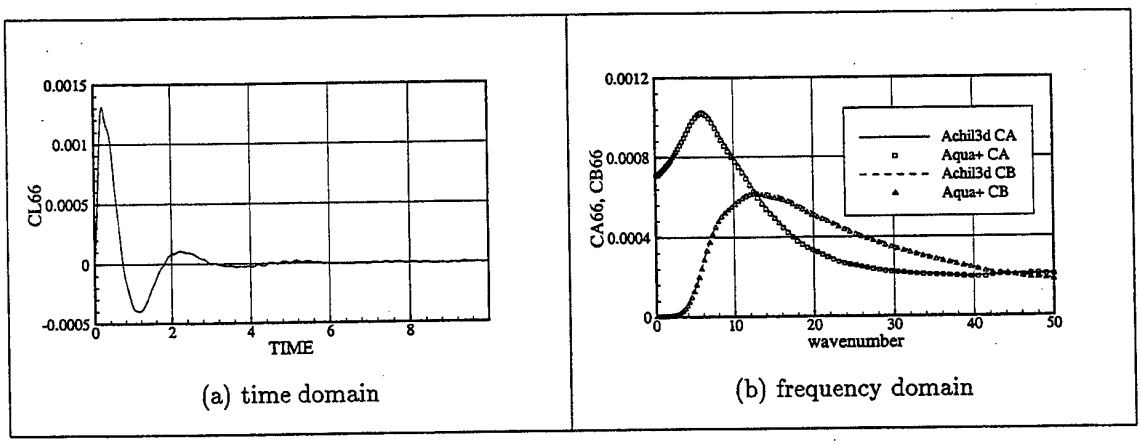
(a) time domain (b) frequency domain

Figure 18: Series 60-06 (2x245 panels). Roll/Roll hydrodynamic coefficients



(a) time domain (b) frequency domain

Figure 19: Series 60-06 (2x245 panels). Pitch/Pitch hydrodynamic coefficients



(a) time domain (b) frequency domain

Figure 20: Series 60-06 (2x245 panels). Yaw/Yaw hydrodynamic coefficients

REFERENCES

1. H.B. Bingham (1994), Simulating ship motions in the time domain. PhD . M.I.T..

2. H.B. Bingham, F.T Korsmeyer, J.N. Newman and G.E. Osborne (1993), The simulation of ship motions. In : Proc. 6th Intl Conf. Num. Ship Hydrod, Iowa City.

3. A.H. Clément (1995): Identification de la fonction de Green de l'hydrodynamique transitoire par des modèles continus . Proc. 5èmes Journées de l'Hydrodynamique, Rouen, pp.319/332.

4. A.H. Clément (1997) : A shortcut for computing time-domain free-surface potentials avoiding Green function evaluations. - Proc. 12th Int. Workshop on Water Waves and Floating Bodies;, Marseille, pp.37-43.

5. A.H. Clément (1998a) : An ordinary differential equation for the Green function of time-domain free-surface hydrodynamics. Journal of Engineering Mathematics. 33 (2), pp201-217.

6. A.H. Clément (1998b) : Computation of impulse response functions using differential properties of the time-domain Green function. - Proc. 13th Int. Workshop on Water Waves and Floating Bodies; Alphen aan den Rijn, pp.21-24.

7. G. Delhommeau (1989), Amélioration des performances des codes de calcul de diffraction radiation au premier ordre. in Proc. 2èmes Journées de l'Hydrodynamique, Nantes, pp.69-88.

8. G. Delhommeau, P. Ferrant and M. Ferrant (1992), Calculation and measurements of forces on a high speed vehicle in forced pitch and heave. Applied Ocean Research, 14-2, pp.119-126.

9. P. Ferrant (1988a), A fast computational method for transient 3D wave-body interaction. Proc. Int. Conf. Comp. Model. Ocean Engng. Venice.

10. P. Ferrant (1988b), Radiation d'ondes de gravité par les déplacements de grande amplitude d'un corps immergé: comparaison des approches fréquentielles et instationnaires. Thèse de Doctorat de l'Université de Nantes.

11. A. Hulme (1982), The wave forces acting on a floating hemisphere undergoing forced periodic oscillations. J. Fluid Mech., 121, pp. 443-463

12. A. Jami (1981), Etude théorique et numérique de phénomènes transitoires en hydrodynamique navale. Thèse de Doctorat ès Sciences. ENSTA, Paris.

13. B.W. King (1987) Time-domain analysis of wave exciting forces on ships and bodies. PhD Univ. Michigan.

14. F.T. Korsmeyer (1988), The first- and second order transient free surface wave radiation problem. PhD Diss. M.I.T.

15. S.J. Liapis (1986), Time-domain analysis of ship motions. PhD Diss. Univ. Michigan.

16. A.R. Magee and R.F. Beck (1989). Vectorized Computation of the Time-Domain Green Function. Proc. 4th Workshop Water Waves & Floating Bodies. Oystese.

17. A.R. Magee (1991). Large amplitude ship motions in the time domain. PhD Thesis. Univ. Michigan.

18. R.M.M. Mattheij and J. Molenar (1996), Ordinary Differential Equations in Theory and Practise. John Wiley & Sons Eds.

19. J.N. Newman (1992). The approximation of free-surface Green functions. in: Wave Asymptotics. P.A. Martin & G.R. Wickham edt, pp. 107-135. Cambridge University Press.

20. J.N. Newman (1985). The evaluation of free-surface Green functions. Proc. 4th Intl Conf. Numer. Ship Hydrod. Washington.

21. W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery (1992), Numerical Recipes. Cambridge University Press Eds.

APPENDIX. COEFFICIENTS OF THE 5TH ORDER RUNGE-KUTTA SCHEME

The differential equation being symbolically written as

$$\dot{y}(t) = f(t, y)$$

and the time step being denoted by h. Let us define the i^{th}stage of the method by:

$$k_i = hf\left(t_n + a_i h, y_n + \sum_{j=1}^{i-1} b_{ij} kj\right)$$

and the final estimate as

$$y_{n+1} = y_n + \sum_{i}^{I} c_i k_i$$

The Cash-Karp coefficient set of the fifth order Runge-Kutta scheme we have selected is given in

the tables below [21]. Applying the stared coefficients c_i^* provides the embedded fourth-order estimation necessary to optimize the time-step by the Fehlberg method.

i	a_i	b_{ij}				
1						
2	$\frac{1}{5}$	$\frac{1}{5}$				
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$			
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$		
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$	
6	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$
$j=$		1	2	3	4	5

i	c_i	c_i^*
1	$\frac{37}{378}$	$\frac{2825}{27648}$
2	0	0
3	$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{125}{594}$	$\frac{13525}{55296}$
5	0	$\frac{277}{14336}$
6	$\frac{512}{1771}$	$\frac{1}{4}$