Solution of Kirchhoff's Time-Domain Integral Equation in Acoustics

Dr. George Benthien and Dr. Stephen Hobbs

March 29, 2005

E-mail: george@gbenthien.net

In this paper we will look at the time-domain equivalent of the Helmholtz integral equation in the frequency domain. This integral equation involves delayed or retarded values of the surface pressure, the time derivative of surface pressure, and the normal acceleration and is called Kirchhoff's integral equation. The numerical solution of Kirchhoff's equation reduces to a time recursion relation for the surface pressures instead of the solution of a system of linear equations at each frequency as was the case in the frequency domain. It turns out that there are stability problems associated with the solution of Kirchoff's equation when the time step is too small. A numerical technique will be presented in this paper that greatly improves the stability.

The Kirchhoff integral equation can be obtained from the Helmholtz integral equation by means of the Fourier transform. The Helmholtz integral equation in the frequency domain can be written

$$\tfrac{1}{2}P(\zeta,\omega) = \int_S P(\xi,\omega)\frac{\partial G}{\partial n_\xi}(\zeta,\xi,\omega)\,dS(\xi) + \rho\int_S G(\zeta,\xi,\omega)a(\xi,\omega)\,dS(\xi) \quad (1)$$

where

$$G = \frac{e^{-ikr}}{4\pi r} = \frac{e^{-i\omega r/c}}{4\pi r} \quad (2)$$

$$\frac{\partial G}{\partial n_\xi} = -\frac{e^{-i\omega r/c}}{4\pi r^2}(1+i\omega r/c)\frac{\partial r}{\partial n_\xi}. \quad (3)$$

Here r is the distance between the surface points ξ and η, $P(\cdot,\omega)$ is the Fourier transform of the surface pressure, and $a(\cdot,\omega)$ is the Fourier transform of the normal acceleration on the surface. Delays in the time domain correspond to complex exponential multipliers in the frequency domain under the Fourier transform operation. In particular,

$$p(t-\tau) \sim e^{-i\omega\tau}P(\omega). \quad (4)$$

Taking the inverse Fourier transform of the Helmholtz integral equation, we get the Kirchhoff integral equation

$$\tfrac{1}{2}p(\zeta,t) = -\int_S \frac{1}{4\pi r^2}\frac{\partial r}{\partial n_\xi}\Big[p(\xi,t-r/c) + \frac{r}{c}\,\dot{p}(\xi,t-r/c)\Big]\,dS(\xi) + \rho\int_S \frac{1}{4\pi r}a(\xi,t-r/c)\,dS(\xi). \quad (5)$$

Moreover, it is easily shown that

$$\frac{\partial r}{\partial n_\xi} = \frac{(\xi - \zeta) \cdot n(\xi)}{r}. \tag{6}$$

The procedure we use for solving Kirchhoff's integral equation numerically is an extension of the one used by Mitzner [Mitzner, K.M., *Numerical Solution for Transient Scattering from a Hard Surface—Retarded Potential Technique*, J. Acoust. Soc. Am., vol. 42, No. 2, pp 391–397 (1967)].

Surface Subdivision We assume that the radiating surface S is subdivided into sub areas S_k on which $p(\xi, t)$, $\dot{p}(\xi, t)$, and $a(\xi, t)$ can be taken as constant for any fixed time t. We will denote these constant values on S_k by $p_k(t)$, $\dot{p}_k(t)$, and $a_k(t)$. We do not, however, assume that the delays r/c are constant over each sub area. Evaluating the Kirchhoff integral equation at the center ζ_j of area S_j and using the relation in equation (6), we get

$$\begin{aligned}\tfrac{1}{2} p_j(t) = -\sum_{k=1}^{K} \int_{S_k} \frac{(\xi - \zeta_j) \cdot n(\xi)}{4\pi r_j^3} \Big[p_k(t - r_j/c) + \frac{r_j}{c}\, \dot{p}_k(t - r_j/c) \Big]\, dS(\xi) \\ + \rho \sum_{k=1}^{K} \int_{S_k} \frac{1}{4\pi r_j} a_k(t - r_j/c)\, dS(\xi). \end{aligned} \tag{7}$$

Approximation of Integrals We assume that each surface area S_k is defined by a one-to-one smooth mapping from a two-dimensional base region into three-dimensional space. We generally take the base region to be a rectangle. The integration over S_k involves choosing quadrature points in the base region and mapping these onto the surface. The integrals over S_k are approximated using a quadrature formula of the form

$$\int_{S_k} f(\xi)\, dS(\xi) \doteq \sum_{l=1}^{L} w_{kl} f(\xi_{kl})\, dS(\xi_{kl}). \tag{8}$$

Here ξ_{kl} is the image of the l-th quadrature point under the mapping defining the surface region S_k, $dS(\xi_{kl})$ is the surface Jacobian of this mapping evaluated at the quadrature point, and w_{kl} is the quadrature weight corresponding

to ξ_{kl}. We generally use two-dimensional Gaussian quadrature over the base rectangles.

Using these integral approximations in equation (7) gives

$$\tfrac{1}{2}p_j(t) = -\sum_{k=1}^{K}\sum_{l=1}^{L}\Omega_{jkl}\big[p_k(t-\hat{t}_{jkl}) + \hat{t}_{jkl}\,\dot{p}_k(t-\hat{t}_{jkl})\big] + \rho\sum_{k=1}^{K}\sum_{l=1}^{L}\Phi_{jkl}a_k(t-\hat{t}_{jkl}) \quad (9)$$

where r_{jkl} is the distance between ζ_j and ξ_{kl}, $\hat{t}_{jkl} = r_{jkl}/c$, and

$$\Omega_{jkl} = w_{kl}\,dS(\xi_{kl})\,\frac{(\xi_{kl}-\zeta_j)\cdot n(\xi_{kl})}{4\pi r_{jkl}^3} \qquad \Phi_{jkl} = \frac{w_{kl}\,dS(\xi_{kl})}{4\pi r_{jkl}}. \quad (10)$$

Delay Approximations The delay $\hat{t}_{jkl}$ in the above equation is not generally an integral multiple of the time step Δt. Let

$$\hat{t}_{jkl} = n_{jkl}\Delta t + \gamma_{jkl}\Delta t \qquad \text{with } 0 \le \gamma_{jkl} < 1 \quad (11)$$

where n_{jkl} is an integer. If t_0 is the initial time, then we make the following linear approximation at the m-th time step

$$\begin{aligned} p_k(t_0 + m\Delta t - \hat{t}_{jkl}) &= p_k(t_0 + m\Delta t - n_{jkl}\Delta t - \gamma_{jkl}\Delta t) && (12)\\ &\doteq (1-\gamma_{jkl})\,p_k(t_0 + m\Delta t - n_{jkl}\Delta t) + \\ &\qquad \gamma_{jkl}\,p_k(t_0 + m\Delta t - n_{jkl}\Delta t - \Delta t) && (13)\\ &\equiv (1-\gamma_{jkl})\,p_k^{(m-n_{jkl})} + \gamma_{jkl}\,p_k^{(m-n_{jkl}-1)}. && (14) \end{aligned}$$

In this equation $p_k^{(n)}$ denotes the value of p_k at the n-th time step. The quantities $\dot{p}_k$ and a_k are approximated similarly. As a result of these approximations, equation (9) becomes

$$\begin{aligned} \tfrac{1}{2}p_j^{(m)} = -\sum_{k=1}^{K}\sum_{l=1}^{L}\Omega_{jkl}\Big\{ &\big[(1-\gamma_{jkl})p_k^{(m-n_{jkl})} + \gamma_{jkl}\,p_k^{(m-n_{jkl}-1)}\big] + \\ &\hat{t}_{jkl}\big[(1-\gamma_{jkl})\dot{p}_k^{(m-n_{jkl})} + \gamma_{jkl}\,\dot{p}_k^{(m-n_{jkl}-1)}\big]\Big\} + \\ &\rho\sum_{k=1}^{K}\sum_{l=1}^{L}\Phi_{jkl}\Big[(1-\gamma_{jkl})a_k^{(m-n_{jkl})} + \gamma_{jkl}\,a_k^{(m-n_{jkl}-1)}\Big]. \end{aligned} \quad (15)$$

Approximation of Time Derivatives Special care must be taken in the approximation of the time derivatives present in equation (15) since this is where the stability problems mentioned previously originate. Mitzner approximated the time derivatives by a second order backward difference formula. Whereas it is necessary to use a backward formula to approximate derivatives at the present time, better formulas can be used to approximate derivatives at past times. The method we use incorporates smoothing to reduce the effect of errors in the pressure values. The approximation is illustrated in figure 1. We take an equal number of points on either side of the point where we want to approximate the derivative and we least-square fit a quadratic to the values at these points. The derivative of the quadratic at the center point is taken as the derivative approximation. This approximation can be expressed as a weighted sum of the function values involved.

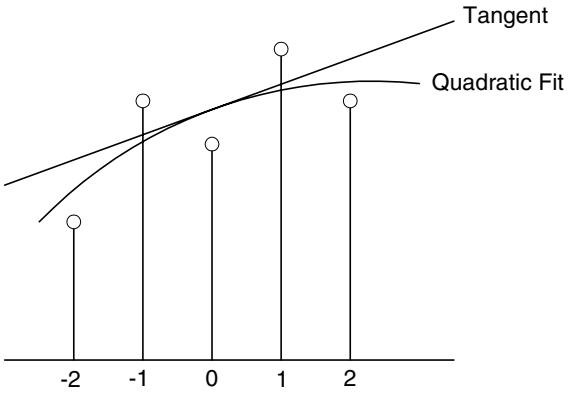

Figure 1: Derivative approximation incorporating smoothing

In equation (21) we need to approximate terms of the form $\dot{p}_k^{(m-i)}$ in terms of values $p_k^{(n)}$, $n \leq m$ for various values of the delay index i. Given a user supplied nonnegative integer value Q, we approximate $\dot{p}_k^{(m-i)}$ by an expression of the form

$$\dot{p}_k^{(m-i)} \doteq \frac{1}{\Delta t} \sum_{q=Q_0(i)}^{Q_1(i)} C_q^i \, p_k^{(m-i-q)} \tag{16}$$

where

$$Q_0(i) = -\min(i, Q) \tag{17}$$

$$Q_1(i) = \begin{cases} 2 & i = 0 \text{ or } Q = 0 \\ \min(i, Q) & \text{otherwise.} \end{cases} \tag{18}$$

The coefficients C_q^i are given by [see Appendix]

$$C_0^i = 1.5 \quad C_1^i = -2 \quad C_2^i = 0.5 \qquad \text{for } i = 0 \text{ or } Q = 0 \tag{19}$$

$$C_q^i = \frac{3q}{Q_1(i)\big(Q_1(i)+1\big)\big(2Q_1(i)+1\big)} \qquad \text{for } i > 0 \text{ and } Q > 0. \tag{20}$$

For $i = 0$ or $Q = 0$ this corresponds to a second-order backward difference approximation. For $i > 0$ and $Q > 0$ the formula was obtained by taking $\min(i, Q)$ points on either side of the time index $m - i$ where the derivative is desired and least-square fitting a quadratic to the values at these points. The derivative of this quadratic at the center point yields the above formula. Approximating the derivative by means of a least-squares fit of a quadratic to a number of points centered around the desired point is a common smoothing technique for computing derivatives from noisy empirical data. In general, this approximation yields better accuracy for small values of Q and better stability for larger values. Larger values of Q are usually needed as the time step becomes smaller in order to maintain stability.

Using the above derivative approximations in equation (15), we obtain

$$\begin{aligned} \tfrac{1}{2}p_j^{(m)} = &- \sum_{k=1}^{K}\sum_{l=1}^{L}\Omega_{jkl}\Bigg\{\Big[(1-\gamma_{jkl})p_k^{(m-n_{jkl})} + \gamma_{jkl}\,p_k^{(m-n_{jkl}-1)}\Big] + \\ &\frac{\hat{t}_{jkl}}{\Delta t}\Big[(1-\gamma_{jkl})\sum_{q=Q_0(n_{jkl})}^{Q_1(n_{jkl})} C_q^{n_{jkl}}\,p_k^{(m-n_{jkl}-q)} + \\ &\gamma_{jkl}\sum_{q=Q_0(n_{jkl}+1)}^{Q_1(n_{jkl}+1)} C_q^{n_{jkl}+1}\,p_k^{(m-n_{jkl}-1-q)}\Big]\Bigg\} + \\ &\rho\sum_{k=1}^{K}\sum_{l=1}^{L}\Phi_{jkl}\Big[(1-\gamma_{jkl})a_k^{(m-n_{jkl})} + \gamma_{jkl}\,a_k^{(m-n_{jkl}-1)}\Big]. \end{aligned} \tag{21}$$

Matrix Formulation If we group by delay in the l-sum , then equation (21) can be written in the form

$$p_j^{(m)} = \sum_{k=1}^{K} \sum_{i=0}^{I} A_{jk}^i a_k^{(m-i)} + \sum_{k=1}^{K} \sum_{i=0}^{I} B_{jk}^i p_k^{(m-i)} \tag{22}$$

where I is the maximum delay over the surface. The maximum delay is finite since all the delays depend on r_{jkl}/c which is bounded. For any pair of values j and k there are only a small number of admissible delays. Therefore, most of the coefficients A_{jk}^i and B_{jk}^i in this equation are zero. It is difficult to write general expressions for A_{jk}^i and B_{jk}^i, but it is easy to construct them in a computer program. Clearly there may be several terms in equation (21) corresponding to a given delay. The A and B matrices are first initialized to zero. As the terms in equation (21) are computed, they can be added into the correct position in the A and B matrices.

Equation (22) can be rearranged to give

$$p_j^{(m)} - \sum_{k=1}^{K} B_{jk}^0 p_k^{(m)} = \sum_{k=1}^{K} \sum_{i=0}^{I} A_{jk}^i a_k^{(m-i)} + \sum_{k=1}^{K} \sum_{i=1}^{I} B_{jk}^i p_k^{(m-i)} \tag{23}$$

where all the pressure terms at the present time $m\Delta t$ are grouped together on the left-hand side. All the values of p on the right-hand-side are at retarded times. Thus, equation (23) is a recursion relation for computing the pressure at successive time steps from computed values at previous time steps. We need to solve a system of equations at each time step for the pressures $p_1^{(m)}, \ldots, p_K^{(m)}$. However, the matrix for this system of equations only needs to be factored once since it is independent of time. In addition, the matrix is very sparse and heavily diagonally dominant. In fact this matrix is diagonal if the time step Δt is chosen so that

$$\Delta t \leq \frac{\min_{j,k,l}(r_{jkl})_{j \neq k}}{c}.$$

In order to determine the maximum delay we define

$$n_0(j,k) = \min_l(n_{jkl}) \tag{24}$$

and

$$n_1(j,k) = \max_l(n_{jkl}) \tag{25}$$

where n_{jkl} is defined in equation (11). The quantities $n_0(j,k)$ and $n_1(j,k)$ can be easily computed once the geometry is prescribed. Let j and k be fixed. For each l, the maximum delay in equation (21) is given by $n_{jkl}+1+Q_1\big(n_{jkl}+1\big)\big)$. Since $i + Q_1(i)$ is a nondecreasing function, the maximum delay over l is $n_1(j,k) + 1 + Q_1\big(n_1(j,k) + 1\big)$. For each l, the minimum delay in equation (21) is given by $n_{jkl} + Q_0(n_{jkl})$. Since $i + Q_0(i)$ is a nondecreasing function, the minimum delay over l is $n_0(j.k) + Q_0(n_0(j.k))$. The maximum delay over j and k is given by

$$I = \max_{j,k} n_1(j,k) + \begin{cases} Q & \text{if } Q > 0 \\ 2 & \text{if } Q = 0 \end{cases} \tag{26}$$

Sparse Matrix Storage Since most of the elements in the matrices A and B are zero, it is convenient to store the nonzero elements of each matrix in a vector as is often done for sparse matrices. The method we use is sometimes called the skyline method. Let Avec and Bvec denote the vectors where A and B are to be stored. We first define a single index n corresponding to the pair of indices j, k by

$$n = (j-1)K + k \qquad j,k = 1,\ldots,K. \tag{27}$$

The inverse of this correspondence is given by

$$j = \mathrm{Int}\left(\frac{n-1}{K}\right) + 1 \tag{28}$$

$$k = n - (j-1)K \tag{29}$$

where Int( ) is the operator giving the integer part of the real argument.

We now think of the A and B matrices as being two-dimensional matrices with row index i and column index n. We will only store a range of elements in each column corresponding to the admissible delays i corresponding to the pair (j,k). Let $n_B(n)$ denote the index in Bvec where the first nonzero element in the n-th column of B is stored. Similarly, let $n_A(n)$ denote the

index in Avec where the first nonzero element in the n-th column of A is stored. Define

For each j and k the delay index i in B is greater than or equal to $n_0(j,k) + Q_0\big(n_0(j,k)\big)$ and less than or equal to $n_1(j,k) + Q_1\big(n_1(j,k)+1\big) + 1$. It follows that there are $n_1(j,k) - n_0(j,k) + Q_1\big(n_1(j,k)+1\big) - Q_0\big(n_0(j,k)\big) + 2$ delay terms in B associated with each j and k. Therefore, $n_B(n)$ can be generated recursively from

$$\begin{aligned} n_B(1) &= 1 \\ n_B(n+1) &= n_B(n) + n_1(j,k) - n_0(j,k) + Q_1\big(n_1(j,k)+1\big) \\ &\quad - Q_0\big(n_0(j,k)\big) + 2. \end{aligned} \tag{30}$$

For B, the term in column n with the smallest delay $n_0(j,k) + Q_0\big(n_0(j,k)\big)$ is stored in the location $n_B(n)$ of Bvec. Thus, the term in column n corresponding to the delay index i is stored in the location $n_B(n) + i - n_0(j,k) - Q_0\big(n_0(j,k)\big)$.

For each j,k the delay index i in A is greater than or equal to $n_0(j,k)$ and less than or equal to $n_1(j,k)+1$. Therefore, there are $n_1(j,k) - n_0(j,k) + 2$ delay terms in A associated with each j,k. It follows that $n_A(n)$ can be generated recursively from

$$\begin{aligned} n_A(1) &= 1 \\ n_A(n+1) &= n_A(n) + n_1(j,k) - n_0(j,k) + 2. \end{aligned} \tag{31}$$

For A, the term in column n with the smallest delay $n_0(j,k)$ is stored in the location $n_A(n)$ of Avec. Thus, the term in column n corresponding to the delay index i is stored in the location $n_A(n) + i - n_0(j,k)$.

In computing $p^{(m)}$ using equation (23) we do not perform the multiplications corresponding to zero elements of A and B. Thus, for any pair of values j and k in equation (23), we only sum over the corresponding admissible i-values.

Storage of Surface Values Since there are only $I+1$ possible delays in equation (23), it is not necessary to keep the entire time history of the surface pressures and accelerations in memory. We keep the pressure and acceleration histories in two-dimensional matrices Pres and Acc where the

first index goes over the number of surface subdivisions and the second index runs from 0 to I. As $a_j^{(m)}$ and $p_j^{(m)}$ are computed they are stored in $\mathsf{Acc}\,(j, m')$ and $\mathsf{Pres}\,(j, m')$ where $m' = m \mod (I+1)$. The pressure and acceleration histories are retrieved using the same circular indexing.

Stability As mentioned previously, there are stability problems with the recursion relation of equation (23) when the time step is too small. It may seem odd that stability problems arise for small time steps since, in solving differential equations, the stability problems generally occur for large time steps. However, in solving differential equations the derivative approximations are not really used to calculate derivatives, but to calculate values of the underlying function. In fact, the derivative values are given by the differential equations. In the Kirchhoff integral equation we are calculating derivatives at past times in terms of previously calculated pressure values. Since there are errors in the computed pressure values that do not go to zero as the time step gets smaller (e.g., spatial discretization errors), small time steps only serve to amplify these errors. The smoothing we use in the derivative approximation reduces the effects of these errors. It should also be mentioned that it probably doesn't make sense to take arbitrarily small time steps for a given spatial breakup. The time and spatial dependencies are related. If one really needs accuracy at small time steps, the spatial subdivisions need to be chosen appropriately small. It should probably take less than ten time steps for an acoustic wave to cross a spatial subdivision. In addition, the spacing between quadrature points should not be too large relative to the time step.

Results The equations developed in the previous section have been implemented into a computer program that can handle quite general surfaces. The program takes advantage of the sparseness of the matrices involved as well as any geometric symmetries that are present. The surface geometry portion of the program was taken from the frequency-domain program CHIEF [G. Benthien, D. Barach, and S. Hobbs, *CHIEF2000 Users Manual*, Technical Document 3104, SPAWAR Systems Center, San Diego, March 2000]. In this section we will show computed results for several sample problems. In order to assess the accuracy of the method we have chosen problems where the solution can also be computed from an analytical expression. The prescribed acceleration waveform for each of these problems is a pulse consisting of one

half cycle of a sine wave of angular frequency ω.

The first problem consists of a sphere of radius r that is uniformly excited in the radial direction. The time is normalized by the time required for a sound wave to travel across the diameter of the sphere. The pulse width in this normalized time is one. Figure 1 shows the computed surface pressure at the pole of the sphere. The analytic solution is included for comparison. For this computation there were 8 subdivisions in the θ-direction and 12 subdivisions in the ϕ-direction. The time step in normalized time was 1/30 and the derivatives were approximated using $Q = 1$. The surface pressure is normalized by $\rho ca/\omega$ where a is the amplitude of the acceleration pulse. It can be seen that the agreement is excellent. Figure 2 shows the error for this same problem. It can be seen that the error at large times is not random, but has a definite pattern. Figure 3 shows the results from taking an FFT of the error. It can be seen that there are narrow peaks corresponding to the frequencies where the solution to the corresponding Helmholtz integral equation is nonunique. The computed frequencies are shown on the plot with the exact values of the interior resonances shown in parentheses. Figure 4 shows the computed pressure when the normalized time step is reduced to 1/90. Notice that the solution is unstable at large times. Figure 5 shows the results when Q is increased to 2. Notice that now the solution is stable.

The second problem is similar to the first except that the motion consists of rigid body acceleration in the direction of the polar axis. For this problem we used 24 subdivisions in the θ-direction and 12 subdivisions in the ϕ-direction. The normalized pulse width was 8/10. The pressure is normalized by ρra where r is the radius of the sphere. Figure 6 shows a comparison between the computed pressure at the pole and the analytic solution. Here again the agreement is excellent. Figure 7 shows the frequency spectrum of the error. Again there are peaks at the frequencies where the Helmholtz integral equation has a nonunique solution. In this case the interior resonances are the roots of $j_1(kr) = 0$ where j_1 is the spherical Bessel function of order one.

The third problem consists of a circular piston at the pole of a sphere. The piston's angular width is 30°. To simplify the analytical solution there is a 30° transition region where the acceleration varies linearly in $\cos\theta$ from one down to zero. Again we are using 24 subdivisions in the θ-direction and a normalized pulse width of 8/10. The normalizations are the same as for

the uniformly pulsed sphere. Figure 8 shows the pressure near the pole (the center of the piston), near the equator, and near the opposite pole. The analytic solution was computed in the frequency domain and numerically transformed to the time domain. Here again the agreement is excellent.

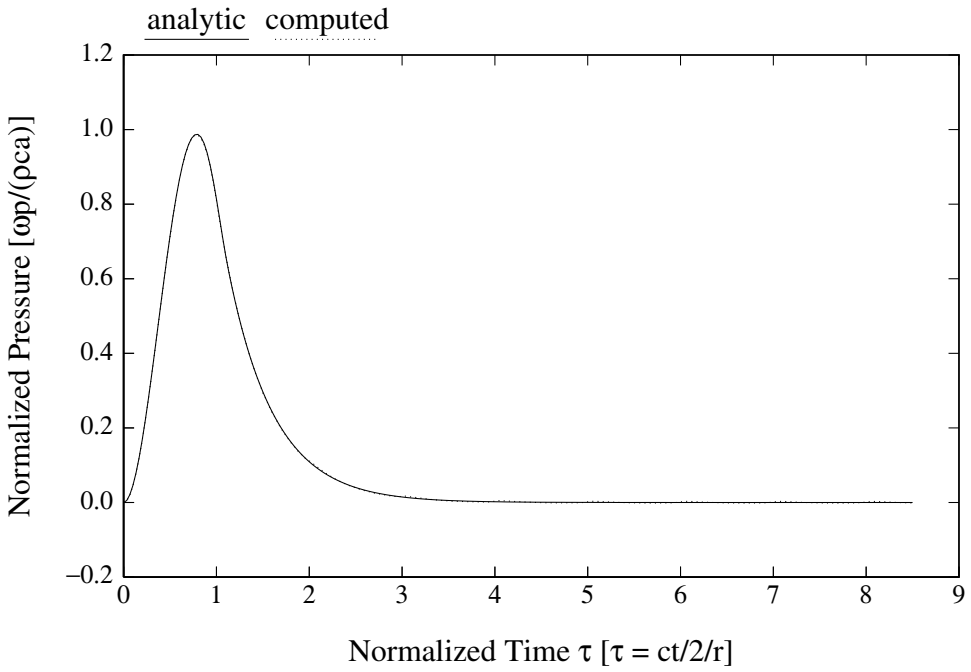

Figure 2: Surface pressure due to uniformly pulsed sphere

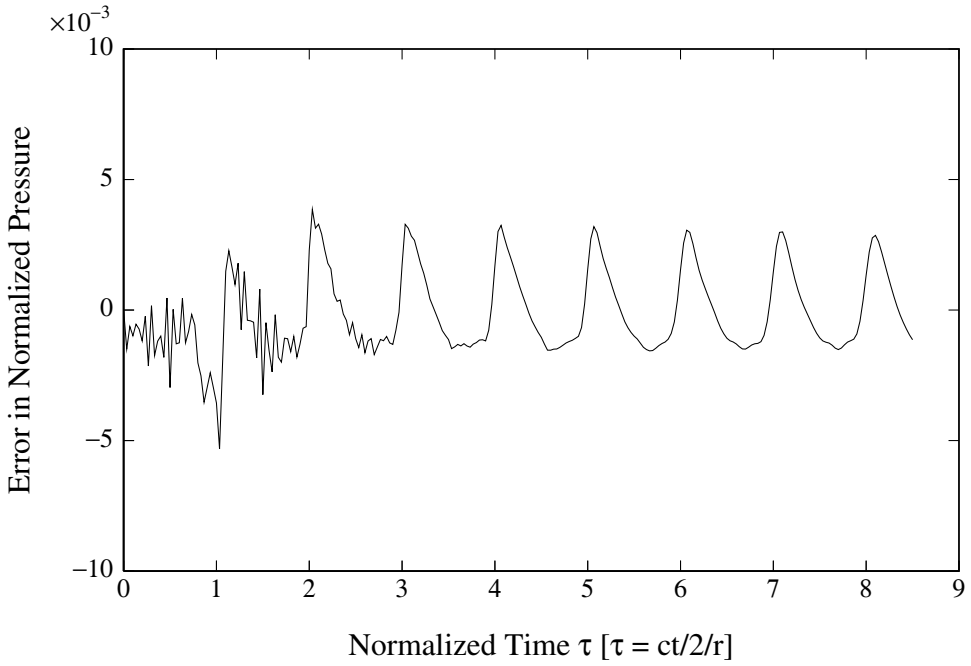

Figure 3: Error for uniformly pulsed sphere

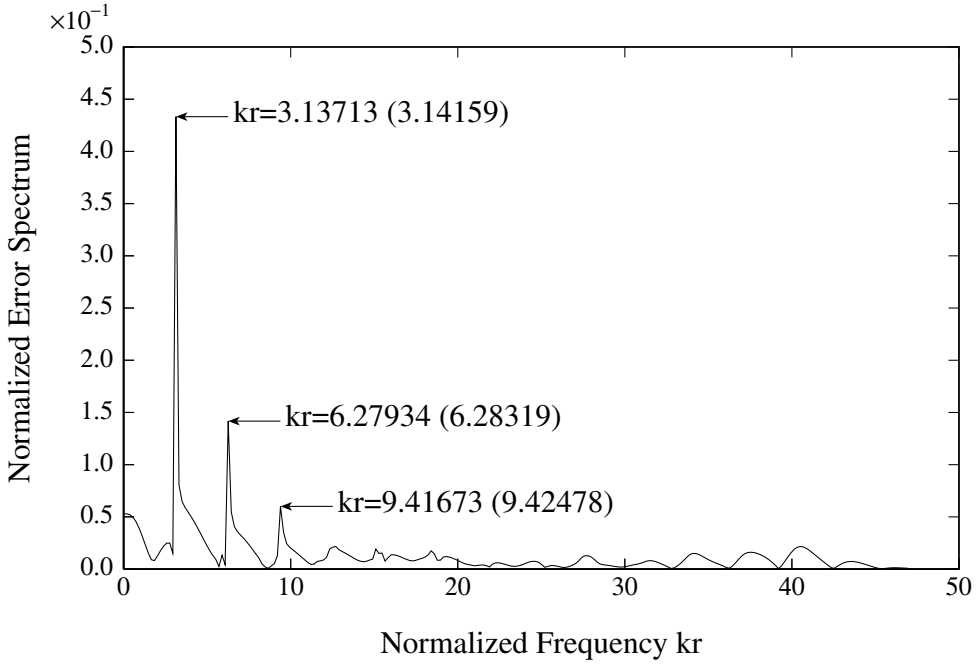

Figure 4: Error spectrum for uniformly pulsed sphere

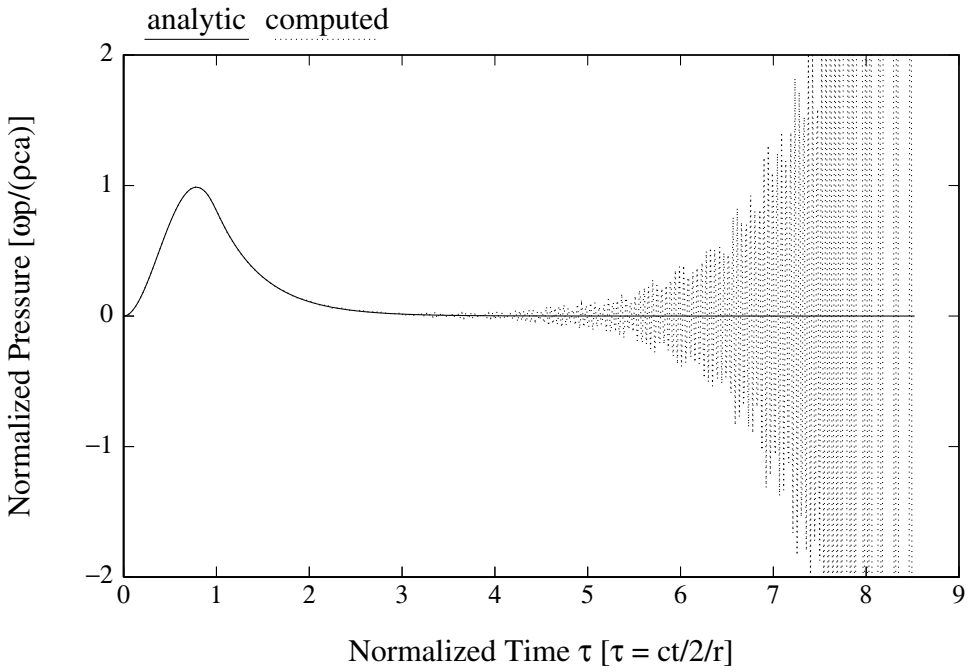

Figure 5: Unstable behavior of uniformly pulsed sphere for small time steps

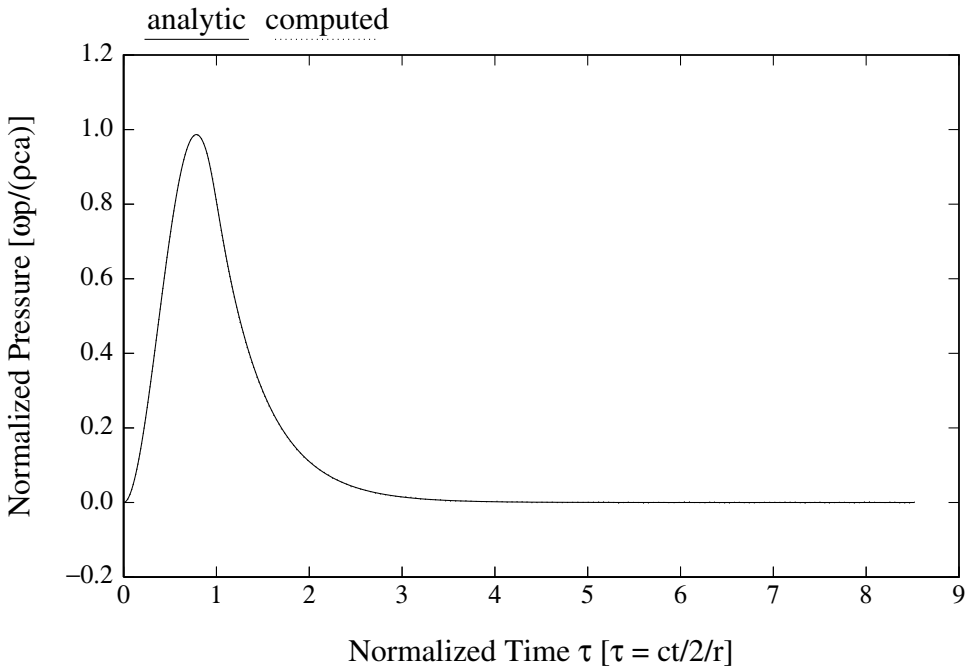

Figure 6: Stabilization of uniformly pulsed sphere for small time steps

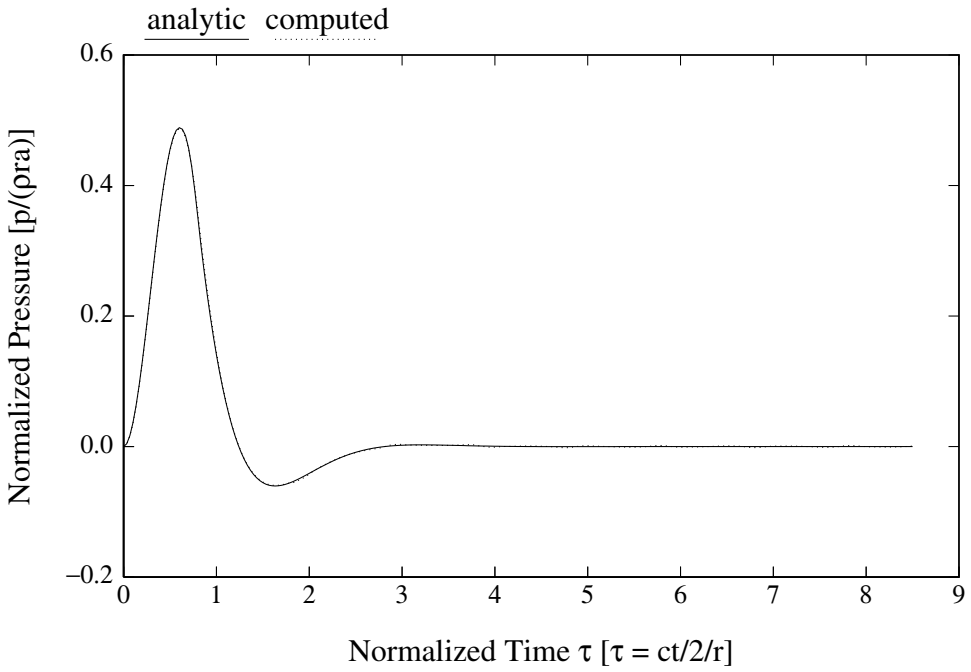

Figure 7: Surface pressure due to pulsed oscillating sphere

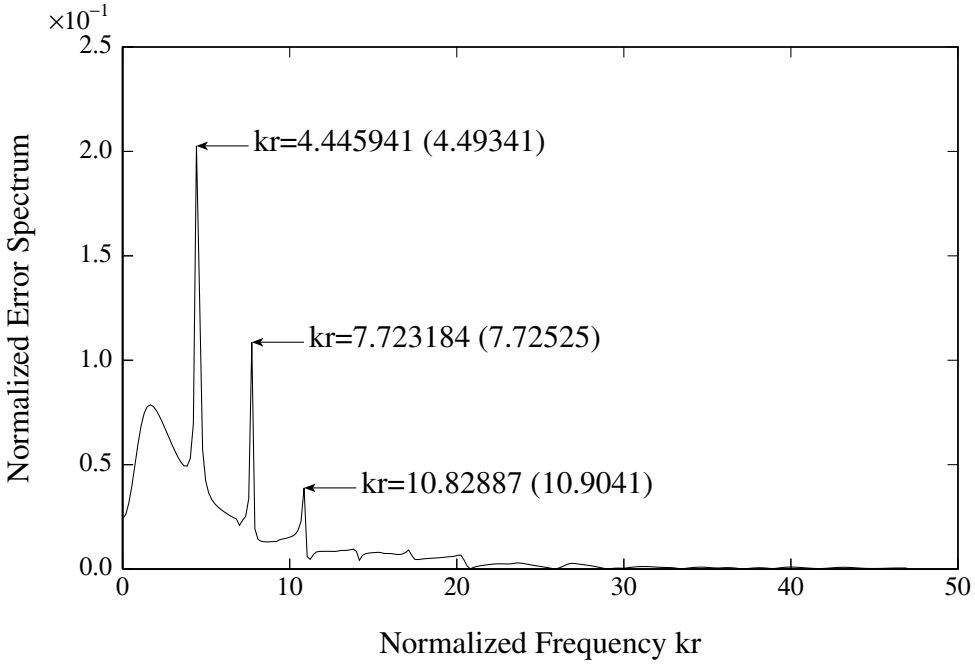

Figure 8: Error spectrum for pulsed oscillating sphere

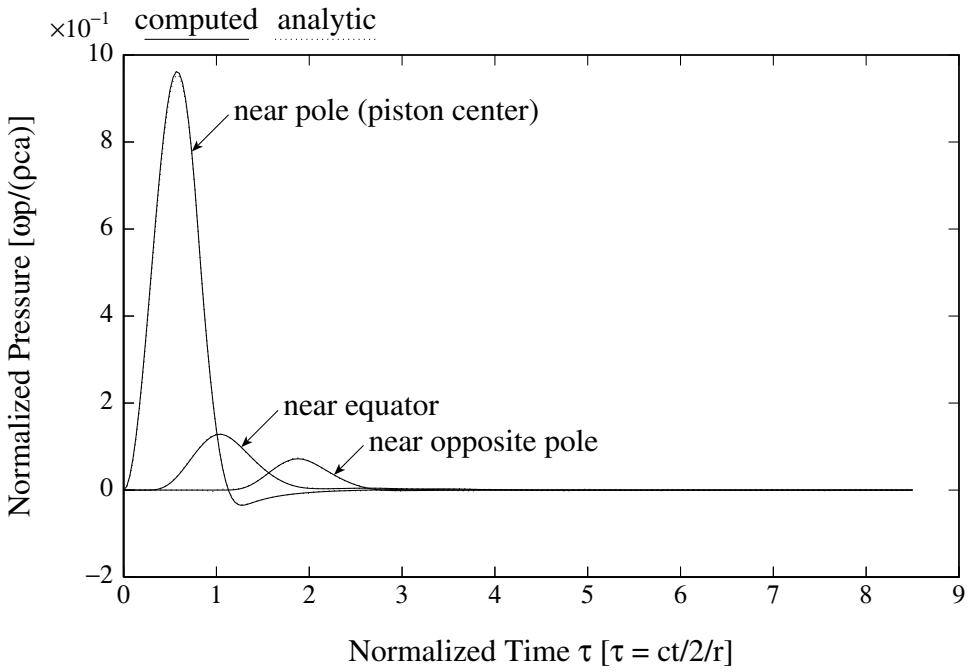

Figure 9: Surface pressure due to pulsed piston on a sphere

Appendix: Differentiation of an Empirical Function Containing Noise

Suppose we have empirical values of a function $f(t)$ at a certain number of equally spaced arguments with increment δ. Since the empirical values contain some noise, the usual difference approximations for the derivative can be greatly in error. To minimize the error we approximate the derivative of f at a sampled time t_0 by the derivative at $t = t_0$ of a quadratic function that is a least-square approximation to f in the vicinity of t_0. In this least-square approximation we use the values of f at t_0 and at m sampled times on each side of t_0. Let

$$g(t) = \alpha(t - t_0)^2 + \beta(t - t_0) + \gamma \tag{32}$$

be the quadratic function that we will use to approximate f. We will choose α, β, γ so as to minimize the error

$$e = \sum_{n=-m}^{m} \big[g(t_0+n\delta) - f(t_0+n\delta)\big]^2 = \sum_{n=-m}^{m} \big[\alpha n^2\delta^2 + \beta n\delta + \gamma - f(t_0+n\delta)\big]^2. \tag{33}$$

At the minimum the partial derivatives of e with respect to α, β, and γ must be zero. In particular,

$$\frac{\partial e}{\partial \beta} = 2 \sum_{n=-m}^{m} \big[\alpha n^2\delta^2 + \beta n\delta + \gamma - f(t_0 + n\delta)\big] n\delta = 0 \tag{34}$$

or

$$\alpha\delta^2 \sum_{n=-m}^{m} n^3 + \beta\delta \sum_{n=-m}^{m} n^2 + \gamma \sum_{n=-m}^{m} n = \sum_{n=-m}^{m} n\, f(t_0 + n\delta). \tag{35}$$

Since the symmetric sums of odd powers of n in equation (35) are zero, we have

$$\beta\delta \sum_{n=-m}^{m} n^2 = \sum_{n=-m}^{m} n\, f(t_0 + n\delta). \tag{36}$$

Moreover, it follows from equation (32) that

$$g'(t_0) = \beta. \tag{37}$$

Combining equations (36) and (37), we get

$$g'(t_0) = \frac{\sum_{n=-m}^{m} n\, f(t_0 + n\delta)}{\delta \sum_{n=-m}^{m} n^2} = \frac{\sum_{n=-m}^{m} n\, f(t_0 + n\delta)}{2\delta \sum_{n=1}^{m} n^2}. \tag{38}$$

To derive an expression for $\sum_{n=1}^{m} n^2$, we define $S_k(m)$ by

$$S_k(m) = \sum_{n=1}^{m} n^k \tag{39}$$

It follows from the relation $(k+1)^2 - k^2 = 2k + 1$ that

$$\begin{aligned}
1^2 - 0^2 &= 2 \cdot 0 + 1 \\
2^2 - 1^2 &= 2 \cdot 1 + 1 \\
3^2 - 2^2 &= 2 \cdot 2 + 1 \\
&\vdots \\
(m+1)^2 - m^2 &= 2 \cdot m + 1.
\end{aligned}$$

Adding these equations, we get

$$(m+1)^2 = 2S_1(m) + m + 1$$

or

$$S_1(m) = \frac{(m+1)^2 - m - 1}{2} = \frac{m(m+1)}{2}. \tag{40}$$

Similarly, it follows from the relation $(k+1)^3 - k^3 = 3k^2 + 3k + 1$ that

$$\begin{aligned}
1^3 - 0^3 &= 3 \cdot 0^2 + 3 \cdot 0 + 1 \\
2^3 - 1^3 &= 3 \cdot 1^2 + 3 \cdot 1 + 1 \\
3^3 - 2^3 &= 3 \cdot 2^2 + 3 \cdot 2 + 1 \\
&\vdots \\
(m+1)^3 - m^3 &= 3 \cdot m^2 + 3 \cdot m + 1.
\end{aligned}$$

Adding these equations, we get

$$(m+1)^3 = 3S_2(m) + 3S_1(m) + (m+1)$$

or

$$\begin{aligned} S_2(m) = \sum_{n=1}^{m} n^2 &= \big[(m+1)^3 - 3m(m+1)/2 - (m+1)\big]/3 \\ &= (m^3 + 3m^2/2 + m/2)/3 \\ &= m(m+1)(2m+1)/6. \end{aligned} \tag{41}$$

Combining equations (38) and (41), we obtain

$$g'(t_0) = \frac{1}{\delta} \sum_{n=-m}^{m} \frac{3n}{m(m+1)(2m+1)} f(n\delta). \tag{42}$$

We use $g'(t_0)$ as the approximate derivative of f at $t = t_0$. For $m = 1$ this reduces to the standard central difference approximation. For higher values of m there is some smoothing involved.