



Method Article

Customization of the angular spectrum method for calculating the acoustic piston field transmitted through a solid plate using MATLAB



M.M. Sæther

Department of Physics and Technology, University of Bergen, Bergen, Norway

ARTICLE INFO

Method name:

Angular spectrum method

Keywords:

Angular spectrum model for acoustic wave propagation
 Numerical integration involving an irregular oscillator
 Open-source MATLAB program

ABSTRACT

The angular spectrum (AS) model is customized to calculate the spatial acoustic pressure field generated by a piston source and transmitted through a steel plate immersed in water, for normal beam incidence. A MATLAB program is developed for this specific problem combining use of Gauss quadrature and a generalized Filon method. The program calculates the pressure wave number spectrum generated by the piston source and transforms the wave number spectrum into the spatial domain. Convergence analysis show that the MATLAB program is far more efficient than the more traditional approach of using the fast Fourier transform algorithm to transform the pressure wavenumber spectrum into the spatial domain. The MATLAB program is published here and free for others to use. The methods and MATLAB algorithms are obtained by

- Converting the original 2D AS model to a 1D model using cylindrical coordinates.
- Combining use of Gauss quadrature and a generalized Filon method for more accurate pressure calculations compared with use of the fast Fourier transform.
- Introducing adaptive numerical integration algorithms in MATLAB and error control parameters which are easy to use.

Specifications table

Subject Area:	Computer Science
More specific subject area:	Mathematical numeric modeling
Method name:	Angular spectrum method
Name and reference of original method:	M. J. Anderson, P. R. Martin, and C. M. Fortunko, Resonant transmission of a three-dimensional acoustic sound beam through a solid plate in air: Theory and measurement, <i>J. Acoust. Soc. Am.</i> 98 (5) (1995) 2628–2638, https://doi.org/10.1121/1.413229 .
Resource availability:	MATLAB code is included as supplementary material

Background

The angular spectrum (AS) approach has been used for decades to simulate and study the interaction of ultrasound and a fluid-immersed solid plate [1–8]. Anderson et al. [3] developed a model (ASM1) for beam transmission through such plates where a piston was used as the source. ASM1 is widely used but has some challenges related to high-accuracy pressure calculations [4]. The uniform grid used in the fast Fourier method typically used in the numerical calculations of ASM1 may give severe aliasing effects.

E-mail address: mathias.sather@uib.no

<https://doi.org/10.1016/j.mex.2023.102037>

Received 30 November 2022; Accepted 21 January 2023

Available online 24 January 2023

2215-0161/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>)

In the current work it is shown how these challenges may be resolved by customizing ASM1 into another model ASM2. A MATLAB code developed for the specific problem and free for others to use is presented here. The customized ASM2 has been used in [8] for high accuracy calculations of the transmitted pressure through a water-immersed steel plate at very large distances from the plate without the use of the MATLAB program presented here. As such, the current work also provides supplementary material to [8]. The MATLAB program is easy to use, has very few parameters for the user to specify and a detailed description of the program is given here. The program may be of use for others studying the interaction of ultrasound and solid plates. It is far more computationally efficient than implementations of ASM1 using the 2D fast Fourier transform algorithm, meaning that a higher accuracy is achieved using shorter computational time. The program provides accurate estimations of the pressure calculated at both long and short distances from the source. The ASM2 method with the current MATLAB implementation is especially well suited for studying near field effects which is proven to be present even at large distances from the source [8].

Method details

In the following, the customization of ASM1 into ASM2 is first described. Next, the MATLAB program for ASM2 is presented in detail, and finally some convergence tests are given comparing the use of ASM1 and ASM2.

Customization of the angular spectrum model

Anderson et al. [3] presented an AS model (ASM1) for beam transmission through a solid plate immersed in a fluid at normal beam incidence. The plate is assumed to be elastic, isotropic, homogeneous, and of infinite lateral extent. Using a time convention $e^{i\omega t}$ where $\omega = 2\pi f$ is the angular frequency and f is the frequency, the transmitted pressure, p_t is given as [3]:

$$p_t(x, y, z, f) = \frac{e^{i\omega t}}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\rho_f \omega}{h_{f,z}} HT e^{-ih_{f,z}(z-d)} e^{-i(h_{f,x}x+h_{f,y}y)} dh_{f,x} dh_{f,y}. \tag{1}$$

A cartesian coordinate system is used, with the front surface of the source located in the x-y plane at $z = 0$. ρ_f is the fluid density, and d is the plate thickness. $h_{f,x}$, $h_{f,y}$, and $h_{f,z}$ are the x, y, and z components of the wave vector in the fluid, respectively, $h_f = \omega/c_f$ is the acoustic wavenumber in the fluid, where c_f is the fluid sound velocity, and

$$h_{f,z} = \begin{cases} \sqrt{h_f^2 - h_{f,x}^2 - h_{f,y}^2}, & h_{f,x}^2 + h_{f,y}^2 \leq h_f^2, \\ -i\sqrt{h_{f,x}^2 + h_{f,y}^2 - h_f^2}, & h_{f,x}^2 + h_{f,y}^2 > h_f^2. \end{cases} \tag{2}$$

For $h_{f,x}^2 + h_{f,y}^2 > h_f^2$, $h_{f,z}$ becomes imaginary, representing evanescent (exponentially decaying) waves in the z direction. T is the plane wave transmission coefficient and H is the aperture function which for a baffled piston is given as [3]

$$H = 2v_0\pi a^2 \frac{J_1\left(a\sqrt{h_{f,x}^2 + h_{f,y}^2}\right)}{a\sqrt{h_{f,x}^2 + h_{f,y}^2}}. \tag{3}$$

a is the piston radius, v_0 is the particle velocity of the piston, and J_1 is the first order Bessel function of the first kind. Note that in [3] the factor “2” in Eq. (3) was incorrectly left out. It has been shown in e.g. [7] that this factor must be included.

Eq. (1) (ASM1) may be regarded as a 2D Fourier transform which is typically solved using the 2D fast Fourier transform (FFT2) routine. However, due to sharp gradients in the integrand of Eq. (1), the uniform sampling grid used in the FFT2 routine may potentially cause large aliasing effects [4]. In the following it is shown how the 2D integral in Eq. (1) can be transformed into a 1D integral resulting in a new model (ASM2), which can be solved using an adaptive numerical integration model.

By coordinate transformation from Cartesian to cylindrical coordinates (r, φ, z) , where $r = \sqrt{x^2 + y^2}$ is the radial range and φ is the polar angle (in the x-y plane), Eq. (1) may be written as

$$p_t(r, \varphi, z, f) = \frac{e^{i\omega t}}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} \frac{\rho_f \omega}{h_{f,z}} HT e^{-ih_{f,z}(z-d)} e^{-i\eta r \cos \varphi} \eta d\eta d\varphi. \tag{4}$$

η is the component of the acoustic wave vector in the plane of the plate (the r-direction), here referred to as the "horizontal wavenumber", and is expressed with $\eta^2 = h_{f,x}^2 + h_{f,y}^2$. By introducing the identity $J_0(\eta r) = \frac{1}{2\pi} \int_0^{2\pi} e^{-i\eta r \cos \varphi} d\varphi$ [8], and omitting the time dependent term $e^{i\omega t}$, Eq. (4) can be written as

$$p_t(r, z, f) = \frac{1}{2\pi} \int_0^{\infty} \frac{\rho_f \omega}{h_{f,z}} HT e^{-ih_{f,z}(z-d)} J_0(\eta r) \eta d\eta. \tag{5}$$

For the incident pressure on the plate, p_i , in absence of the plate,

$$p_i(r, z_0, f) = \frac{1}{2\pi} \int_0^{\infty} \frac{\rho_f \omega}{h_{f,z}} H e^{-ih_{f,z}z_0} J_0(\eta r) \eta d\eta. \tag{6}$$

z_0 is z-coordinate of the front surface of the plate. J_0 is the zeroth order Bessel function of the first kind. Eqs. (5) and (6) represent the customized AS model ASM2 used in [8].

```

1 - clear
2 - warning('') % Clear lastwarn. A MATLAB lastwarn warning on badly scaled polynomial fitting is expected when using Filon method.
3 % The warning is OK because it triggers MATLAB to execute Gauss method instead of Filon.
4
5 %% Input parameters
6 % f_vec = 400e3:10e3:500e3;
7 f_vec = 457.6e3;
8 z_vec = 100;
9 r_vec = 0;
10 plate_or_freefield = 'plate'; % input argument: 'plate', or 'freefield'
11 %% Input parameters numerical integration scheme
12 Err_tresh = 1e-1; % Absolute local error tolerance in the first and last Gauss adaptive method and the Filon-type adaptive method
13 Err_tresh_peak = 1; % Absolute local error tolerance in second Gauss method around the peak. Usually set higher than Err_tresh
14 eta_step_init = 1; % Initial starting step size
15 IntegrationBoundary_DeltaAbove_h_f = 200; % Set upper integration limit. Sets eta limit above h_f.
16 FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff = 0.2; % Typical value: 0.2. Defines when Filon method kicks in.
17 % For a value set in the range <0,10>, a eta period in the integrand is defined when Filon method kicks in. For a value set
18 % outside <0,10>, Filon method kicks in at the set value (for inf, the Gauss is always used, for 0, Filon is always used)
19
20 %% Loop over field
21 c_wa = FunctionsParent.c_f;
22 for ii = 1:length(r_vec)
23     r = r_vec(ii);
24     for jj = 1:length(z_vec)
25         z = z_vec(jj);
26         for kk = 1:length(f_vec)
27
28             f = f_vec(kk);
29             h_f = 2*pi*f/c_wa;
30             eta_end = h_f+IntegrationBoundary_DeltaAbove_h_f;
31
32             Integral_sum = 0;
33             eta_step = eta_step_init;
34             eta = 0;
35
36             if strcmp(plate_or_freefield,'plate')
37                 functions_r_z_f = FunctionsPlate(x,z,f);
38             else
39                 functions_r_z_f = FunctionsFluid(x,z,f);
40             end
41             gaussObj = Integral_Segment_Gauss(functions_r_z_f);
42             filonObj = Integral_Segment_Filon(functions_r_z_f);
43
44             eta_cutof_Shampine = sqrt( h_f^2 - ( 2*pi/z + sqrt(h_f^2-[0:1:h_f].^2) ).^2 ) - [0:1:h_f]; % find apparent period
45             indx_tmp = find(ismag(eta_cutof_Shampine) == 0,1); % find when imaginary part is zero
46             eta_cutoff = find(abs(eta_cutof_Shampine(indx_tmp:end))<FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff,1)+indx_tmp-1;
47             if isempty(eta_cutoff)
48                 eta_cutoff = eta_end;
49             end
50
51             if FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff >= 10 || FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff == 0
52                 eta_cutoff = FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff;
53             end
54             if eta_cutoff > eta_end
55                 eta_cutoff = eta_end;
56             end
57
58             %% Adaptive Gauss integration from eta in range (0,eta_cutoff)
59             while eta<eta_cutoff && eta<eta_end
60                 eta
61                 if eta+eta_step>eta_cutoff
62                     eta_step = eta_cutoff - eta;
63                 end
64                 gaussOrder4 = gaussObj.Gauss4(eta,eta_step);
65                 gaussOrder5 = gaussObj.Gauss5(eta,eta_step);
66                 if abs(gaussOrder4 - gaussOrder5) <= Err_tresh
67                     Integral_sum = Integral_sum + gaussOrder5;
68                     eta = eta+eta_step;
69                     eta_step = eta_step/2;
70                 else
71                     eta_step=eta_step/2;
72                 end
73             end
74         end
75     end
76 end

```

Fig. 1. Shows the codelines 1–73 for the script main script (i) ASM_Adapt_FilonGauss.m which calculates the pressure from Eqs. (5) and (6). Input parameters are needed in lines 7–9 and 12–16.

MATLAB implementation of model

The integrands in Eqs. (5) and (6) go to zero for $\eta > h_f + \Delta\eta_{cutoff}$, and the integrals can be calculated numerically. The integrands have an irregular oscillator, which is a classical problem in numerical integration [9,10]. To numerically solve the integrals in Eqs. (5) and (6), two different numerical integration techniques are used: Gauss and the generalized Filon method, which are complimentary to each other. The generalized Filon method is given for integrals on the form [10]:

$$I = \frac{1}{2\pi} \int_{\eta_1}^{\eta_2} f(\eta) e^{-ig(\eta)} d\eta. \quad (7)$$

Following [10], $f(\eta)$ and $g(\eta)$ are approximated by a second order polynomial, and the resulting integral is evaluated analytically. This method is suited for highly oscillatory integrands, which is not the case for the Gauss method. Visa versa, the Gauss method is suited for integrands without too much oscillations, which is not the case for the Filon method. The oscillation period is therefore tracked and the program chooses which of the Filon or Gauss method is to be used. The argument in the exponential in Eqs. (5) and (6) is nonlinear, but the instant oscillation period at a certain η_0 can be defined by a $\Delta\eta_0$ giving a 2π increase of the argument in the exponential, i.e.

$$ih_{f,z}(\eta_0 + \Delta\eta_0)z - ih_{f,z}(\eta_0 + \Delta\eta_0)z = 2\pi \quad (8)$$

```

74
75 %% Adaptive Filon-type integration from eta in range (eta_cutoff,lastwarn)
76 warning('')
77 while eta<eta_end
78     eta
79     if eta+eta_step>eta_end
80         eta_step = eta_end - eta;
81     end
82
83     Filon_segment_course = 1/2/pi*filonObj.DoIntegrationCoarse(eta, eta_step);
84     Filon_segment_fine = 1/2/pi*filonObj.DoIntegrationFine(eta, eta_step);
85
86     if ~isempty(lastwarn)
87         break
88     end
89     if abs(Filon_segment_course - Filon_segment_fine) <= Err_tresh
90         Integral_sum = Integral_sum + Filon_segment_fine;
91         eta = eta+eta_step;
92         eta_step = eta_step*2;
93     else
94         eta_step=eta_step/2;
95     end
96 end %eta
97
98 %% Adaptive Gauss integration around peak in integrand, i.e. eta in range (lastwarn,h_f+0.1)
99 while eta<h_f+0.1
100     eta
101     if eta+eta_step>h_f+0.1
102         eta_step = h_f+0.1 - eta;
103     end
104     gaussOrder4 = gaussObj.Gauss4(eta,eta_step);
105     gaussOrder5 = gaussObj.Gauss5(eta,eta_step);
106     if abs(gaussOrder4 - gaussOrder5) <= Err_tresh_peak
107         Integral_sum = Integral_sum + gaussOrder5;
108         eta = eta+eta_step;
109         eta_step = eta_step*2;
110     else
111         eta_step=eta_step/2;
112     end
113 end % eta
114
115 %% Adaptive Gauss integration in the evanescent region i.e. eta in range (h_f+0.1,eta_end)
116 while eta<eta_end
117     eta
118     if eta+eta_step>eta_end
119         eta_step = eta_end - eta;
120     end
121     gaussOrder4 = gaussObj.Gauss4(eta,eta_step);
122     gaussOrder5 = gaussObj.Gauss5(eta,eta_step);
123     if abs(gaussOrder4 - gaussOrder5) <= Err_tresh
124         Integral_sum = Integral_sum + gaussOrder5;
125         eta = eta+eta_step;
126         eta_step = eta_step*2;
127     else
128         eta_step=eta_step/2;
129     end
130 end % eta
131 Pressure_freq_spectrum(ii,jj,kk) = Integral_sum;
132
133 end % f
134 end % z
135 end % r
136 %% Extract pressure magnitude and phase
137 r_index = 1;
138 z_index = 1;
139 f_index = 1;]
140 P_level = 20*log10(abs(Pressure_freq_spectrum(r_index,z_index,f_index)))
141 P_phase = angle(Pressure_freq_spectrum(r_index,z_index,f_index))

```

Fig. 2. Shows the codelines 74–141 for the script main script (i) ASM_Adapt_FilonGauss.m which calculates the pressure from Eqs. (5) and (6).

By setting for example $\Delta\eta_0 = 0.2$ rad/m, η_0 for which the oscillation period is 0.2 rad/m is obtained, and η_0 is the value for which the Filon method is used instead of the Gauss method. The adaptive Gauss method is based on the 4th and 5th order Gauss quadrature rules. In the adaptive Filon method, Eq. (7) is calculated using first 3 and then 5 data points, producing a rough and a finer estimate of Eq. (7), and thus an error estimate.

In the MATLAB program, the Gauss method is used from $\eta = 0$ and up to $\eta = \eta_0$. From $\eta = \eta_0$ and up to approximately $\eta = h_f$ the Filon method is used. For $\eta > h_f$ the argument is real giving no oscillation, and the Gauss method is used again. A practical way is used to determine the η for which the program switch from the Filon method and back to the Gauss method again. For η close to h_f , there is a sharp gradient in the integrand, resulting in a warning from MATALB when trying to fit the polynomial in the Filon method. This warning is used to trigger the program to switch back to the Gauss method.

The MATLAB program consists of six files in total: (i): The main script. (ii) and (iii): Class definition files where the functions needed for the integration problems in Eqs. (5) and (6) are defined separately. (iv) and (v): Class definition files where the integration procedure using the Gauss and generalized Filon methods are defined, respectively. (vi): The parent class definition file where the input parameters such as densities and wave velocities are defined and passed on to the class definitions (ii) and (iii).

- (i) ASM_Adapt_FilonGauss.m
- (ii) FunctionsFluid.m
- (iii) FunctionsPlate.m
- (iv) Integral_Segment_Filon.m
- (v) Integral_Segment_Gauss.m
- (vi) FunctionsParent.m

```

1 classdef FunctionsParent
2
3
4     properties (Constant)
5         c_f = 1485; % Fluid velocity
6         rho_f = 1000; % Fluid density
7         a = 10.55e-3; % Piston radius
8         v0 = 1; % piston particle velocity
9         rho_p = 8000; % Plate density
10        c_pl = 5780; % Compressional wave velocity plate
11        c_ps = 3130; % Shear wave velocity plate
12        d = 6.05e-3; % Plate thickness
13        z0 = 270e-3; % Plate distance from piston source
14    end
15
16
17     properties (Access=public)
18         Filon_Integrate;
19     % Hardcoded Filon_Integrate;
20     % Filon_Integrate = @(A,B,C,K,L,M,eta_sym)exp(M.*-1i-L.*eta_sym.*1i-K.*eta_sym.^2.*1i).*(B.*5.0e-1i)/K-A.*1.0./K.^2.*L.*2.5e-1i)+ ...
21     % (A.*eta_sym.*exp(M.*-1i-L.*eta_sym.*1i-K.*eta_sym.^2.*1i).*5.0e-1i)/K+1.0./K.^2.*sqrt(pi).*exp(M.*-1i+(L.^2.*2.5e-1i)/K).* ...
22     % erf1((L.*5.0e-1i+K.*eta_sym.*1i).*1.0./sqrt(K.*-1i)).*1.0./sqrt(K.*-1i).*(A.*L.^2.*1i+C.*K.^2.*4.0i+A.*K.^2.0-B.*K.*L.*2.0i).^1.25e-1i;
23
24    end
25
26     methods
27         function obj = FunctionsParent() % constructor
28             syms A B C K L M eta_sym
29             integrand = (A*eta_sym^2+B*eta_sym+C) * exp(-1i*(K*eta_sym^2+L*eta_sym+M));
30             int_expression_sym = int(integrand,eta_sym);
31             obj.Filon_Integrate = matlabFunction(int_expression_sym,'Vars',[A B C K L M eta_sym]);
32         end
33     end
34 end
35
36 end
    
```

Fig. 3. Shows the for the codelines for class definition file (vi) FunctionsParent.m. Input parameters are defined in lines 5–13.

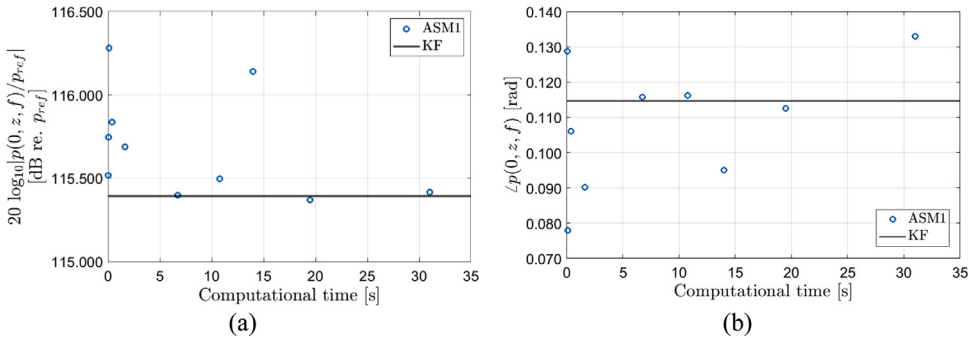


Fig. 4. Comparison of calculations of the freefield axial pressure, $p(0, z, f)$, at $z = 270$ mm and $f = 457.6$ kHz using the traditional ASM1 model, Eq. (1) and the analytical KF model [11]. Wavenumber grid spacings ranging from 16 to 0.5 rad/m is used for the datapoints which corresponds to the different computational times. (a) Magnitude, (b) phase.

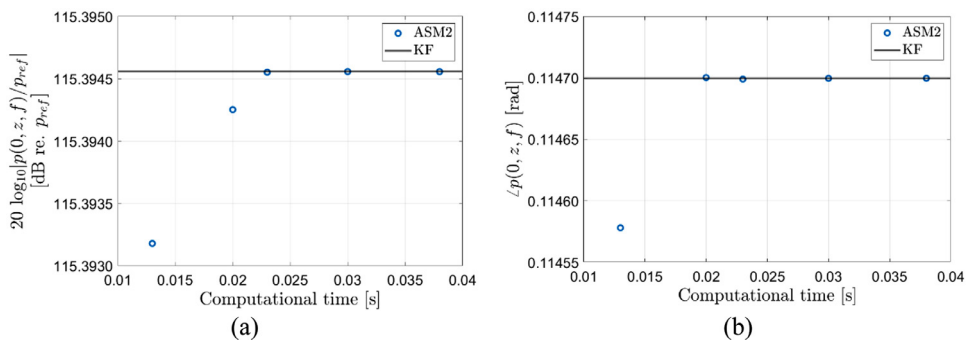


Fig. 5. Comparison of calculations of the freefield axial pressure, $p(0, z, f)$, at $z = 270$ mm and $f = 457.6$ kHz using the current ASM2 model, Eq. (6) and the analytical KF model [11]. The computation time depends on the factor “Err_tresh” which is set to 100, 10, 1, 0.1, and 0.01 for the simulations presented here. (a) Magnitude, (b) phase.

Eqs. (5) and (6) are solved by executing the MATLAB main script (i). To run the program, define the nine input parameters in (i) (lines 7–9 and 12–16 in Fig. 1) and nine input Constant properties in (vi) (lines 5–13 in Fig. 3), then run (i). MATLAB version 2021a has been used in the current work.

The codelines of the main script is presented in Figs. 1 and 2. “f_vec” (line 7) is the frequency vector. “z_vec” (line 8) and “r_vec” (line 9) are vectors defining the z and lateral distances, respectively. “plate_or_freefield” (line 10) must be set to “plate” if the transmitted pressure through a plate is to be calculated. Other values calculates the piston pressure freefield without the plate

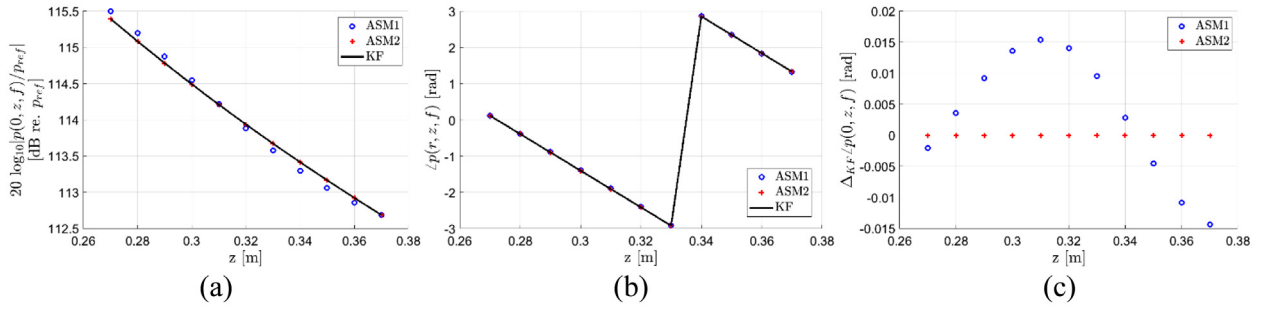


Fig. 6. Comparison of calculations of the freefield axial pressure, $p(0, z, f)$, for z in the range 270–370 mm and $f = 457.6$ kHz using ASM1, ASM2, and the analytical KF model [11]. In ASM1, the wavenumber grid spacing is 0.6 rad/m. In ASM2, “Err_tresh” is set to 0.1. (a) Magnitude, (b) phase.

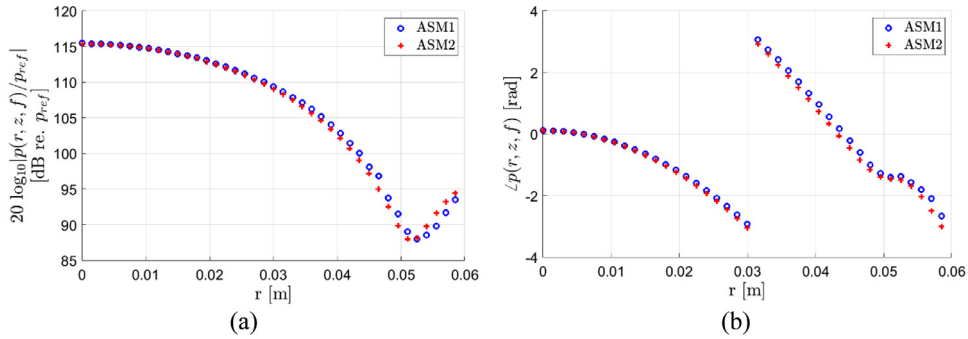


Fig. 7. Comparison of calculations of the freefield pressure distribution at $z = 270$ mm for r in the range 0–60 mm and $f = 457.6$ kHz using ASM1 and ASM2. In ASM1, the wavenumber grid spacing is 0.6 rad/m. In ASM2, “Err_tresh” is set to 0.1. (a) Magnitude, (b) phase.

present. “Err_tresh” (line 12) gives the maximum allowed difference between the 4th and 5th order Gauss quadrature rules. It is also the maximum allowed difference between the 3 and 5 point Filon methods. “Err_tresh_peak” (line 13) specifically sets the maximum allowed difference between the 4th and 5th order Gauss quadrature rules for η close to h_f . “eta_step_init” (line 14) is the initial η step. “IntegrationBoundary_DeltaAbove_h_f” (line 15) equals to $\Delta\eta_{cutoff}$ and defines the upper boundary for the integration. “FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff” (line 16) determines when the Filon method is to be used and equals $\Delta\eta_0$ when defined between 0 and 10. “FilonMethodStarts_OscPeriodCutOff_or_EtaCutoff” equals to η_0 when the parameter is set to 0 or higher than 10. When set to 0, the Filon method is used over the interval from 0 and close up to h_f .

In the script, the final calculated pressure $p(r, z, f)$ in 131 (Fig. 2) is obtained by looping over the vectors specifying r , z , and f , respectively. Inside the frequency loop starting at line 26 there are four while loops. In the first, the Gauss method is used (Fig. 1). In the second, the Filon method is used (Fig. 2). In the third and fourth (Fig. 2) the Gauss method is used again. The input parameters of the piston source, fluid and the solid plate are defined in FunctionsParent.m shown in Fig. 3 (lines 5–13).

Accuracy and computational time

In Fig. 4 the axial freefield pressure at $z = 270$ mm and $f = 457.6$ kHz is calculated using the traditional ASM1 method, Eq. (1), with the FFT2 algorithm. The shortest computational time corresponds to a wavenumber grid spacing ($dh_{f,x}$) of 16 rad/m, while the longest corresponds to a grid spacing 0.5 rad/m. The maximum wavenumber in the FFT2 is set to 2136 rad/m for all datapoints (in Figs. 4, 6, and 7) which corresponds to a spatial grid spacing of 1.5 mm (i.e. $0.46 h_f$). The solid line is the correct value calculated from the analytical model provided by Kinser et al. [11]. In part (a), the magnitude is shown and in part (b) the phase is shown. Neither the phase nor the magnitude are fully converged.

In Fig. 5 the axial freefield pressure at $z = 270$ mm and $f = 457.6$ kHz is calculated using the ASM2 method, Eq. (6), with the Gauss/Filon MATLAB program. The integration limit in Eq. (6) is set to 2136 rad/m. The computational times corresponds to “Err_tresh” settings of 100, 10, 1, 0.1, and 0.01. For simplicity, “Err_tresh_peak” is set equal to “Err_tresh”. The solid line is the correct value calculated from the analytical KF model [11]. In part (a), the magnitude is shown and in part (b) the phase is shown. For this example, the accuracy is far better and the computational time is far shorter compared with using the ASM1 method shown in Fig. 4.

In Fig. 6, the axial freefield pressure for z in the range 270–370 mm and $f = 457.6$ kHz is calculated using ASM1, ASM2, and the analytical KF model. In ASM1, the wavenumber grid spacing is 0.6 rad/m and the computational time is approximately 50 s. In ASM2, “Err_tresh” is set to 0.1, and the computational time for the datapoints is approximately 1.5 s. For the magnitude in part (a), the maximum deviation between the KF model and ASM1 is approximately 0.1 dB, and 0.000001 dB between the KF model and ASM2. No difference is visually observed in the phase in part (b). In part (c), Δ_{KF} on the ordinate axis refers to the deviation from the

analytical KF model. As can be observed in part (c), there is no visually observable deviation to the ASM2, and a maximum deviation to ASM1 of approximately 0.015 rad/m.

In Fig. 7, the freefield pressure distribution at $z = 270$ mm for r in the range 0–60 mm and $f = 457.6$ kHz is calculated using ASM1 and ASM2. In ASM1, the wavenumber grid spacing is 0.6 rad/m and the computational time is approximately 20 s. In ASM2, “Err_tresh” is set to 0.1. As no analytical model exists for the pressure distribution for a piston generated nearfield, ASM2 is only compared to ASM1. For ASM2 in Fig. 7, it is assumed that the error is about the same as observed in Figs. 5 and 6. In Fig. 7, the maximum deviation between ASM1 and ASM2 is about 2 dB for the magnitude (a) and about 0.4 rad/m for the phase (b).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The MATLAB code is attached.

Acknowledgments

The author wishes to acknowledge Per Lunde, Magne Vestrheim, and Simen Midtbø, University of Bergen, for stimulating discussions and valuable input during the work described.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.mex.2023.102037](https://doi.org/10.1016/j.mex.2023.102037).

References

- [1] R.K. Johnson, A.J. Devaney, Transducer model for plate thickness measurement, in: Proceedings of the IEEE International Ultrasonics Symposium, 1982.
- [2] N.A. Whitaker, H.A. Haus, Backward wave effects in acoustic scattering measurements in 1983, in: Proceedings of the Ultrasonics Symposium, 1983.
- [3] M.J. Anderson, P.R. Martin, C.M. Fortunko, Resonant transmission of a three-dimensional acoustic sound beam through a solid plate in air: theory and measurement, *J. Acoust. Soc. Am.* 98 (5) (1995) 2628–2638.
- [4] X. Zeng, R.J. McGough, Evaluation of the angular spectrum approach for simulations of near-field pressures, *J. Acoust. Soc. Am.* 123 (1) (2008) 68–76.
- [5] M. Aanes, Interaction of Piezoelectric Transducer Excited Ultrasonic Pulsed Beams with A Fluid-Embedded Viscoelastic Plate. Finite Element Modeling, Angular Spectrum Modeling and Measurements, University of Bergen, Bergen, Norway, 2014.
- [6] G. Waag, L. Hoff, P. Norli, Air-coupled ultrasonic through-transmission thickness measurements of steel plates, *Ultrasonics* 56 (2015) 332–339.
- [7] S.H. Midtbø, Beam Diffraction Effects in Guided-Wave Transmission of Fluid Embedded Elastic Plate. Influence of Receiver Distance and Finite Aperture, University of Bergen, Bergen, Norway, 2018.
- [8] M.M. Sæther, S.H. Midtbø, P. Lunde, Interaction of three-dimensional acoustic beam with fluid-loaded solid plate: Axial near- to far-field transmission at normal beam incidence, *Ultrasonics* 125 (2022) 106795.
- [9] G. Evans, J. Webster, A comparison of some methods for the evaluation of highly oscillatory integrals, *J. Comput. Appl. Math.* 112 (1-2) (1999) 55–69.
- [10] L. Shampine, Integrating oscillatory functions in MATLAB, *Electron. Trans. Numer. Anal.* 39 (2012) 403–413.
- [11] L.E. Kinsler, et al., *Fundamentals of Acoustics*, 4th ed., Wiley, New York, 2000.