Gaussian Beams and Transducer Modeling
**Learning Objectives**

- Characteristics of Gaussian Beams
- Propagation Laws
- Transmission/Reflection Laws
- Multi-Gaussian Beam Models for Ultrasonic Transducers
- MATLAB Examples
Fundamental Waves as Building Blocks

Plane wave:

$$p = P_0 \exp\left[ ik_p z - i\omega t \right]$$

Spherical wave:

$$p = \frac{P_0}{R} \exp\left[ ik_p R - i\omega t \right]$$
Gaussian beam

\[ p = \frac{P_0}{(z-z_0) + i \omega z_c} \exp\left[ ik_p z - i \omega t \right] \exp \left[ \frac{ik_p \rho^2}{2R(z)} - \frac{\rho^2}{(w(z))^2} \right] \]

beam width

\[ w(z) = w_0 \sqrt{\frac{(z-z_0)^2 + (z_c)^2}{(z_c)^2}} \]

\[ z_c = \pi w_0^2 / \lambda_p \]

beam waist

confocal distance
\[ p = \frac{P_0}{(z-z_0) + iz_c} \exp \left( ik_p z - i\omega t \right) \exp \left[ \frac{ik_p \rho^2}{2R(z)} - \frac{\rho^2}{(w(z))^2} \right] \]

wave front curvature

\[ R(z) = \frac{(z-z_0)^2 + (z_c)^2}{(z-z_0)} \]

\begin{align*}
R(z) &= \infty \quad z = z_0 \\
R(z) &= z \quad z \gg z_0, z_c
\end{align*}
Gaussian beam

\[ p = \frac{P_0}{(z - z_0) + iz_c} \exp\left[ ik_p z - i\omega t \right] \exp \left[ \frac{ik_p \rho^2}{2R(z)} - \frac{\rho^2}{(w(z))^2} \right] \]

Gaussian profile

wave front curvature

spherical wave (paraxial approximation)

\[ p = \frac{P_0}{z} \exp\left[ ik_p z \right] \exp \left[ \frac{ik_p \rho^2}{2z} \right] \]

can get from Gaussian beam by letting

\[ w_0 = z_0 = 0 \]
Paraxial Approximation

plane wave \[ p = A \exp \left( ik_p x_1 \sin \theta + ik_p x_3 \cos \theta - i \omega t \right) \]

write as \[ p = P(x_1, x_3, \omega) \exp \left( ik_p x_3 - i \omega t \right) \] quasi-plane wave traveling in the \( x_3 \) -direction

where \[ P(x_1, x_3, \omega) = A \exp \left[ ik_p x_1 \sin \theta + ik_p x_3 \left( 1 - \cos \theta \right) \right] \]
Paraxial Approximation

\[ P(x_1, x_3, \omega) = A \exp \left[ i k_p x_1 \sin \theta + i k_p x_3 (1 - \cos \theta) \right] \]

\[ \frac{\partial^2 P}{\partial x_1^2} = -k_p^2 P \sin^2 \theta \approx -k_p^2 P \theta^2 \]

\[ 2ik_p \frac{\partial P}{\partial x_3} = -2k_p^2 P (1 - \cos \theta) \approx k_p^2 P \theta^2 \]

\[ \frac{\partial^2 P}{\partial x_3^2} = -k_p^2 P (1 - \cos \theta)^2 \approx -\frac{k_p^2 P \theta^4}{4} \]

if \( \theta < 0.5 \text{ rad (about 30°)} \)
Paraxial Approximation

wave equation (cylindrical coordinates) with axial symmetry

\[
\frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial p}{\partial \rho} \right) + \frac{\partial^2 p}{\partial z^2} - \frac{1}{c_p^2} \frac{\partial^2 p}{\partial t^2} = 0
\]

NOTE: \( \rho \) here is a radius, not the density

Let \( p = P(\rho, z, \omega) \exp(ik_p z - i\omega t) \)

and assume

\[
\left| \frac{\partial^2 P}{\partial z^2} \right| \ll \left| 2ik_p \frac{\partial P}{\partial z} \right|, \left| \frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial P}{\partial \rho} \right) \right|
\]

then

\[
\frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial p}{\partial \rho} \right) + 2ik_p \frac{\partial p}{\partial z} = 0
\]

paraxial wave equation
Paraxial Wave Equation

\[
\frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial P}{\partial \rho} \right) + 2ik_p \frac{\partial P}{\partial z} = 0 \quad p = P(\rho, z, \omega) \exp(ik_p z - i\omega t)
\]

Solutions

\[
P = P_0 \quad \text{plane wave}
\]

\[
P = \frac{P_0}{z} \exp \left[ ik_p \frac{\rho^2}{2z} \right] \quad \text{spherical wave}
\]

\[
P = \frac{P_0}{(z - z_0) + iz_c} \exp \left[ \frac{ik_p \rho^2}{2R(z)} - \frac{\rho^2}{\left( w(z) \right)^2} \right] \quad \text{Gaussian beam}
\]
Complex form of the Gaussian beam commonly used

\[ p = p_0 \frac{q(0)}{q(z)} \exp[ik_p z] \exp \left[ \frac{ik_p \rho^2}{2q(z)} \right] \]

\[ q(z) = q(0) + z \quad q(0) = -\left( z_0 + \frac{i\pi w_0^2}{\lambda_p} \right) \]
Gaussian beam amplitudes obey the plane wave relationship

\[ p = \rho_m c_{pm} v_z \]

\[ \rho \rightarrow \text{density, not radius} \]

\[ p = p_0 \frac{q(0)}{q(z)} \exp[i k_{p1} z] \exp \left[ \frac{i k_{p1} \rho^2}{2q(z)} \right] \]

\[ k_{p1} = \omega / c_{p1} \]

\[ v_z = v_0 \frac{q(0)}{q(z)} \exp[i k_{p1} z] \exp \left[ \frac{i k_{p1} \rho^2}{2q(z)} \right] \]

\[ = \frac{p_0}{\rho_1 c_{p1}} \frac{q(0)}{q(z)} \exp[i k_{p1} z] \exp \left[ \frac{i k_{p1} \rho^2}{2q(z)} \right] \]
Reflection and Transmission at a Spherical Interface of radius $R_I$

$$p_i = \tilde{P}_i(z) \exp\left[ ik_{p_1}z + i\omega t_0 \right] \exp\left[ \frac{ik_{p_1}\rho^2}{2q_i(z)} \right]$$

$$p_r = \tilde{P}_r(z_2) \exp\left[ ik_{p_1}z_2 + i\omega t_0 \right] \exp\left[ \frac{ik_{p_1}\rho^2}{2q_r(z_2)} \right] \quad z_2 = -z$$

$$p_t = \tilde{P}_t(z) \exp\left[ ik_{p_2}z + i\omega t_0 \right] \exp\left[ \frac{ik_{p_2}\rho^2}{2q_t(z)} \right]$$
Amplitudes are related through plane wave reflection and transmission coefficients

\[
\tilde{P}_r(0) = \frac{\rho_2 c_{p_2} - \rho_1 c_{p_1}}{\rho_1 c_{p_1} + \rho_2 c_{p_2}} \tilde{P}_i(0) = R_p \tilde{P}_i(0)
\]

\[
\tilde{P}_2(0) = \frac{2 \rho_2 c_{p_2}}{\rho_1 c_{p_1} + \rho_2 c_{p_2}} \tilde{P}_i(0) = T_p \tilde{P}_i(0)
\]

Phase terms are related by

transmitted beam

\[
\frac{1}{q_i(0)} = \left(\frac{c_{p_2}}{c_{p_1}} - 1\right) \frac{1}{R_0} + \frac{c_{p_2}}{c_{p_1}} \frac{1}{q_i(0)}
\]

reflected beam

\[
\frac{1}{q_r(0)} = \frac{2}{R_0} + \frac{1}{q_i(0)}
\]

radius of interface
Transmission/reflection laws in terms of beam widths and wave front curvatures

\[ w_t(0) = w_r(0) = w_i(0) \]

\[ \frac{1}{R_t(0)} = \left( \frac{c_{p2}}{c_{p1}} - 1 \right) \frac{1}{R_0} + \frac{c_{p2}}{c_{p1}} \frac{1}{R_i(0)} \]

\[ \frac{1}{R_r(0)} = \frac{2}{R_0} + \frac{1}{R_i(0)} \]
Defocusing interface \( c_{p2} > c_{p1} \)

Focusing interface \( c_{p2} > c_{p1} \)
\[ q_1(z_1) = q_1(0) + z_1 \quad p(z_1, \omega) = \tilde{P}(0) \frac{q_1(0)}{q_1(z_1)} \exp(ik_{p_1}z_1) \exp \left[ \frac{ik_{p_1}}{2} \frac{\rho^2}{q_1(z_1)} \right] \]

\[ p(z_2, \omega) = \tilde{P}(0) \frac{q_2(Q_1)}{q_2(z_2)} T_{12} \frac{q_1(0)}{q_1(Q_1)} \exp(ik_{p_1}z_1 + ik_{p_2}z_2) \exp \left[ \frac{ik_{p_1}}{2} \frac{\rho^2}{q_2(z_2)} \right] \]

\[ q_2(z_2) = q_2(Q_1) + z_2 \quad q_2(Q_1) = \frac{q_1(Q_1)}{\left( \frac{c_{p_2}}{c_{p_1}} - 1 \right) \frac{q_1(Q_1)}{R_0} + \frac{c_{p_2}}{c_{p_1}}} \]
Gaussian beam transmitted through a planar interface

\[ p(z_2, \omega) = \tilde{P}(0) \frac{q_1(0)}{q_1(0) + \tilde{z}} \exp\left(ik_{p1}z_1 + ik_{p2}z_2\right) \exp \left(ik_{p1} \frac{\rho^2}{2q_1(0) + \tilde{z}}\right) \]

\[ \tilde{z} = z_1 + \frac{c_{p2}^2}{c_{p1}} z_2 \]
Gaussian beams in multiple media

\( p(z_{M+1}, \omega) = \tilde{P}(0) \frac{q_{M+1}(0)}{q_{M+1}(z_{M+1})} \prod_{m=1}^{M} \tilde{T}_{m}^{m} \frac{q_{m}(0)}{q_{m}(z_{m})} \)

\[ \cdot \exp \left[ i \sum_{m=1}^{M+1} k_{pm} z_{m} \right] \exp \left[ \frac{ik_{pM+1}}{2} \frac{\rho^2}{q_{M+1}(z_{M+1})} \right] \]
Propagation Laws

\[ q_m(z_m) = q_m(0) + z_m \]

Transmission Laws

\[ q_{m+1}(Q_m) = \frac{q_m(Q_m)}{\left( \frac{c_{p,m+1}}{c_{p,m}} - 1 \right) q_m(Q_m) + \frac{c_{p,m+1}}{c_{p,m}}} \]

Reflection Laws

\[ q_{m+1}(Q_m) = \frac{q_m(Q_m)}{2 \left( \frac{Q_m}{R_0}_m \right) + 1} \]
ABCD Matrix Forms of the Laws

\[ q_f = \frac{Aq_i + B}{Cq_i + D} \]

Propagation Laws

\[
\begin{bmatrix}
A^d & B^d \\
C^d & D^d
\end{bmatrix}
= \begin{bmatrix} 1 & z_m \\ 0 & 1 \end{bmatrix}
\]

Transmission Laws

\[
\begin{bmatrix}
A^t & B^t \\
C^t & D^t
\end{bmatrix}
= \begin{bmatrix} 1 & 0 \\ \\
\left(\frac{c_{p_{m+1}}}{c_{p_m}} - 1\right) & \frac{c_{p_{m+1}}}{c_{p_m}} \\
\left(\frac{R_0}{R_0}_m \right) & \frac{R_0}{R_0}_m \end{bmatrix}
\]

Reflection Laws

\[
\begin{bmatrix}
A^r & B^r \\
C^r & D^r
\end{bmatrix}
= \begin{bmatrix} 1 & 0 \\ \\
\left(\frac{R_0}{R_0}_m \right) & \frac{R_0}{R_0}_m \end{bmatrix}
\]
Gaussian beams in multiple media

\[ p(z_{M+1}, \omega) = \mathcal{T} \tilde{P}(0) \frac{q_1(0)}{A^G q_1(0) + B^G} \exp[i\omega t_0] \exp \left( \frac{ik_p}{2} \frac{\rho^2}{A^G q_1(0) + B^G} \right) \]

\[ \mathcal{T} = \prod_{m=1}^{M} \tilde{T}_{m, m+1} \]

\[ t_0 = \sum_{m=1}^{M+1} z_m / c_{pm} \]

transmission or reflection coefficients

\[
\begin{bmatrix}
A^G & B^G \\
C^G & D^G
\end{bmatrix} = \begin{bmatrix}
A_{M+1}^d & B_{M+1}^d \\
C_{M+1}^d & D_{M+1}^d
\end{bmatrix} \begin{bmatrix}
A_M^t & B_M^t \\
C_M^t & D_M^t
\end{bmatrix} \cdots \begin{bmatrix}
A_1^t & B_1^t \\
C_1^t & D_1^t
\end{bmatrix} \begin{bmatrix}
A_1^d & B_1^d \\
C_1^d & D_1^d
\end{bmatrix}
\]
Gaussian beam after propagating through $M+1$ media

$$p(z_{M+1}, \omega) = \mathcal{T} \tilde{P}(0) \frac{q_1(0)}{A^G q_1(0) + B^G} \exp[i\omega t_0] \exp \left[ \frac{ik_{pM+1}}{2} \rho^2 \frac{1}{A^G q_1(0) + B^G} \right] \frac{1}{C^G q_1(0) + D^G}$$

Gaussian beam in single medium

$$p(z_1, \omega) = \tilde{P}(0) \frac{q_1(0)}{A^d q_1(0) + B^d} \exp[ik_{p1}z_1] \exp \left[ \frac{ik_{p1}}{2} \frac{\rho^2}{A^d q_1(0) + B^d} \frac{1}{C^d q_1(0) + D^d} \right]$$

$$= \tilde{P}(0) \frac{q_1(0)}{q_1(0) + z_1} \exp[ik_{p1}z_1] \exp \left[ \frac{ik_{p1}}{2} \frac{\rho^2}{q_1(0) + z_1} \right]$$

$$\begin{bmatrix} A^d_1 & B^d_1 \\ C^d_1 & D^d_1 \end{bmatrix} = \begin{bmatrix} 1 & z_1 \\ 0 & 1 \end{bmatrix}$$
Multi-Gaussian Beam Model

circular piston transducer radiating into a fluid

\[ v_z = \begin{cases} 
  v_0 & \rho \leq a \\
  0 & \text{otherwise} 
\end{cases} \]

add ten Gaussians

\[ v_z = v_0 \sum_{n=1}^{10} A_n \exp\left(-B_n \rho^2 / a^2\right) \]
add ten Gaussians

\[ p = \sum_{n=1}^{10} \frac{\tilde{P}_n(0)}{q_1(0) + z_1^n} \exp[ik_p z] \exp\left[ \frac{ik_p \rho^2}{2q_1(0) + z_1^n} \right] \]

\[ \tilde{P}_n(0) = \rho c_p v_0 A_n \]

\[ \left[ q_1(0) \right]_n = \frac{-ik_p a^2}{B_n} \]
function [a, b] = gauss_c
% gauss_c returns the coefficients developed by Wen and Breazeale to model a piston transducer
a = zeros(10,1);
b = zeros(10,1);
a(1) = 11.428 + 0.95175*i;
a(2) = 0.06002 - 0.08013*i;
a(3) = -4.2743 - 8.5562*i;
a(4) = 1.6576 + 2.7015*i;
a(5) = -5.0418 + 3.2488*i;
a(6) = 1.1227 - 0.68854*i;
a(7) = -1.0106 - 0.26955*i;
a(8) = -2.5974 + 3.2202*i;
a(9) = -0.14840 - 0.31193*i;
a(10) = -0.20850 - 0.23851*i;
b(1) = 4.0697 + 0.22726*i;
b(2) = 1.1531 - 20.933*i;
b(3) = 4.4608 + 5.1268*i;
b(4) = 4.3521 + 14.997*i;
b(5) = 4.5443 + 10.003*i;
b(6) = 3.8478 + 20.078*i;
b(7) = 2.5280 - 10.310*i;
b(8) = 3.3197 - 4.8008*i;
b(9) = 1.9002 - 15.820*i;
b(10) = 2.6340 + 25.009*i;
Multi-Gaussian beam model of a piston transducer in multiple media

\[ p(z_{M+1}, \omega) = \sum_{n=1}^{10} \mathcal{T} \tilde{P}_n(0) \frac{\left[ q_1(0) \right]_n}{A^G \left[ q_1(0) \right]_n + B^G} \exp[i\omega t_0] \exp \left[ \frac{ik_{p,M+1}}{2} \left( \frac{\rho^2}{C^G \left[ q_1(0) \right]_n + D^G} \right) \right] \]
multi-Gaussian beam model for a spherically focused circular transducer

\[ v_z = v_0 \sum_{n=1}^{10} A_n \exp \left( -B_n \rho^2 / a^2 \right) \exp \left( -ik \rho^2 / 2F \right) \]

\[ = v_0 \sum_{n=1}^{10} A_n \exp \left( -\tilde{B}_n \rho^2 / a^2 \right) \]

\[ \tilde{B}_n = B_n + \frac{ika^2}{2F} \]
MATLAB multi-Gaussian Beam Model

\[ p(x, \omega) / p_0 = p(x, \omega) / \rho_1 c_{p1} v_0 \]

The time delay factor \( \exp \left( i k_{p1} z_1 + i k_{p2} z_2 \right) \) is not included in this model.

% MGB_script
% This script calculates the normalized
% pressure, \( p/p_0 \), due to a planar piston transducer radiating at normal
% incidence through a spherically curved interface.
% This is a multi-Gaussian beam model that uses the 10 coefficients
% of Wen and Breazeale to calculate the wave field.

clear
MATLAB multi- Gaussian Beam Model

% get input parameters
f = 5; % frequency (MHz)
a = 6.35; % transducer radius (mm)
Fl = inf; % transducer focal length (mm)
z1 = 0; % path length in medium 1 (mm)
z2 = linspace(6, 125, 200); % path length in medium 2 (mm)
r = 0.0; % distance from ray axis (mm)
R0 = inf; % interface radius of curvature (mm)
d1 = 1.0; % density, medium 1 (gm/cm^3)
d2 = 1.0; % density, medium 2 (gm/cm^3)
c1 = 1480.; % wave speed, medium 1 (m/sec)
c2 = 1480.; % wave speed, medium 2 (m/sec)

These parameters are for a ½ inch diameter, 5 MHz planar piston transducer radiating into water (no interface). The on-axis pressure values from 6 to 125 mm are being sought.
MATLAB multi- Gaussian Beam Model

% get Wen and Breazeale coefficients (10)
[A, B] = gauss_c;

% transmission coefficient (base on pressure ratio)
T = (2*c2*d2)/(c1*d1+c2*d2);

h = 1/R0; % interface curvature

zr = eps*(f == 0) + 1000*pi*(a^2)*f./c1; % "Rayleigh" distance, ka^2/2
k1 = 2*pi*1000*f./c1; % wave number in medium 1
MATLAB multi- Gaussian Beam Model

%initialize pressure to zero
p = 0;

%multi-Gaussian beam model
for  j = 1:10  % form up multi-Gaussian beam model with
% 10 Wen and Breazeale coefficients

    b =B(j) + i*zr./Fl; % modify coefficients for focused probe

    q = z1 - i*zr./b;
    K = q.*(1 -(c1/c2));
    M = (1 +K.*h);
    ZR = q./M;
    m = 1./(ZR +(c2/c1).*z2);
    t1 = A(j)./(1 + (i.*b./zr).*z1);
    t2 = t1.*T.*ZR.*m;
    p = p + t2.*exp(i.*(k1./2).*m.*(r.^2));

end
% plot magnitude of on-axis pressure
plot(z2, abs(p))
xlabel('z-axis')
ylabel('|p|/|p_0|')
Cross-axis plot at \( z = 130 \) mm (near last maximum)

% get input parameters
f = 5;                  % frequency (MHz)
a = 6.35;               % transducer radius (mm)
Fl = inf;               % transducer focal length (mm)
z1 = 0;                 % path length in medium 1 (mm)
z2 = 130;               % path length in medium 2 (mm)
r = linspace(-10, 10, 200); % distance from ray axis (mm)
R0 = inf;               % interface radius of curvature (mm)
d1 = 1.0;               % density, medium 1 (gm/cm^3)
d2 = 1.0;               % density, medium 2 (gm/cm^3)
c1 = 1480.;             % wave speed, medium 1 (m/sec)
c2 = 1480.;             % wave speed, medium 2 (m/sec)
% plot magnitude of on-axis pressure
plot(r, abs(p))
xlabel('r-axis')
ylabel('|p|/|p_0|')

>> MGB_script
Cross-axis plot at $z = 65$ mm (near last on-axis null)

% get input parameters
f = 5; % frequency (MHz)
a = 6.35; % transducer radius (mm)
Fl = inf; % transducer focal length (mm)
$z_1 = 0$; % path length in medium 1 (mm)
$z_2 = 65$; % path length in medium 2 (mm)
r = linspace(-10, 10, 200); % distance from ray axis (mm)
R0 = inf; % interface radius of curvature (mm)
d1 = 1.0; % density, medium 1 (gm/cm^3)
d2 = 1.0; % density, medium 2 (gm/cm^3)
c1 = 1480.; % wave speed, medium 1 (m/sec)
c2 = 1480.; % wave speed, medium 2 (m/sec)
>> MGB_script
Now, generate an image of the entire beam

% get input parameters
f = 5; % frequency (MHz)
a = 6.35; % transducer radius (mm)
Fl = inf; % transducer focal length (mm)
z1 = 0; % path length in medium 1 (mm)
z2t = linspace(0, 200, 500);
rt = linspace(-20, 20, 200);
[z2, r] = meshgrid(z2t, rt);
R0 = inf; % interface radius of curvature (mm)
d1 = 1.0; % density, medium 1 (gm/cm^3)
d2 = 1.0; % density, medium 2 (gm/cm^3)
c1 = 1480.; % wave speed, medium 1 (m/sec)
c2 = 1480.; % wave speed, medium 2 (m/sec)
%plot magnitude of on-axis pressure
image(z2t, rt, abs(p)*50)
xlabel('z-axis')
ylabel('r-axis')

Note: unequal scales

>> MGB_script
We can also examine a 10 MHz, 76.2 mm focal length focused transducer

% get input parameters
f = 10;       % frequency (MHz)
a = 6.35;     % transducer radius (mm)
F1 = 76.2;    % transducer focal length (mm)
z1 = 0;       % path length in medium 1 (mm)
z2 = linspace(0,200,500); % path length in medium 2 (mm)
r=0;         % distance from ray axis (mm)
R0 = inf;     % interface radius of curvature (mm)
d1 = 1.0;     % density, medium 1 (gm/cm^3)
d2 = 1.0;     % density, medium 2 (gm/cm^3)
c1 = 1480.;   % wave speed, medium 1 (m/sec)
c2 = 1480.;   % wave speed, medium 2 (m/sec)
%plot magnitude of on-axis pressure
plot(z2, abs(p))
xlabel('z-axis')
ylabel('|p/po|')

>> MGB_script
Now, plot an image of this focused transducer wave field

% get input parameters
f = 10;         % frequency (MHz)
a = 6.35;       % transducer radius (mm)
Ff = 76.2;      % transducer focal length (mm)
z1 = 0;         % path length in medium 1 (mm)
z2t = linspace(0,300,500);  
rt=linspace(-20,20,200);  
[z2, r]=meshgrid(z2t, rt);
R0 = inf;       % interface radius of curvature (mm)
d1 = 1.0;       % density, medium 1 (gm/cm^3)
d2 =1.0;        % density, medium 2 (gm/cm^3)
c1 = 1480.;     % wave speed, medium 1 (m/sec)
c2 = 1480.;     % wave speed, medium 2 (m/sec)
%plot magnitude of on-axis pressure
image(z2t, rt, abs(p)*25)
xlabel('z-axis')
ylabel('r-axis')

Note: unequal scales

>> MGB_script