Progress report

R Adhithya

Implementation details of Bloch boundary condition for a 2-D square plate

The weak form of the eigenvalue problem that is being solved is

$$a(\mathbf{u},\mathbf{v}) = \lambda b(\mathbf{u},\mathbf{v}) \quad \forall \in S \tag{1}$$

where, S is the problem domain.

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) &:= \int_\Omega \nabla_\mathbf{s}\mathbf{v}^*(\mathbf{x}) : \mathbf{C}(\mathbf{x}) : \nabla_\mathbf{s}\mathbf{u}(\mathbf{x}) d\mathbf{x} \\ b(\mathbf{u},\mathbf{v}) &:= \int_\Omega \rho(\mathbf{x})\mathbf{v}^*(\mathbf{x}).\mathbf{u}(\mathbf{x}) dx \end{aligned} \tag{2}$$

The boundary conditions for the problem are

$$\begin{aligned} \mathbf{u}(\mathbf{x}+\mathbf{h_i}) &= \mathbf{u}(\mathbf{x})\exp(i\mathbf{k}.\mathbf{h_i}) \\ \sigma(\mathbf{x}+\mathbf{h_i}).\mathbf{n}(\mathbf{x}+\mathbf{h_i}) &= \sigma(\mathbf{x}).\mathbf{n}(\mathbf{x})\exp(i\mathbf{k}.\mathbf{h_i}) \end{aligned} \tag{3}$$

To avoid using complex algebra, the variables are divided into real and imaginary components, u^{Re} and u^{Im}[1]. The equation is transformed into

$$\begin{aligned} u^{Re}(x+h_i) &= u^{Re}(x)cos(ka) - u^{Im}sin(ka) \\ u^{Im}(x+h_i) &= u^{Re}(x)sin(ka) + u^{Im}cos(ka) \end{aligned} \tag{4}$$

An example of a square plate discretized using four elements is taken to explain how the bloch boundary condition is implemented using constraint matrix. Equation (4) is expanded for the mesh shown in fig. 1.

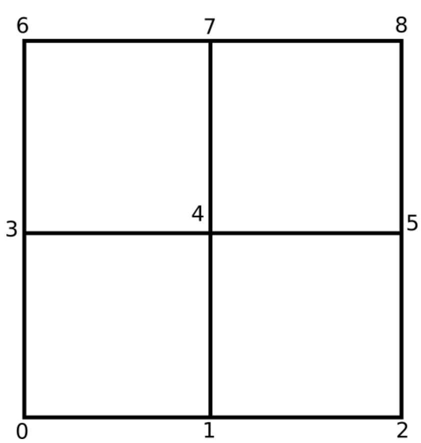

Fig. 1 Meshed square plate

$$
\begin{Bmatrix} \mathbf{u_0^{Re}} \\ \mathbf{u_1^{Re}} \\ \mathbf{u_2^{Re}} \\ \mathbf{u_3^{Re}} \\ \mathbf{u_4^{Re}} \\ \mathbf{u_5^{Re}} \\ \mathbf{u_6^{Re}} \\ \mathbf{u_7^{Re}} \\ \mathbf{u_8^{Re}} \\ \mathbf{u_0^{Im}} \\ \mathbf{u_1^{Im}} \\ \mathbf{u_2^{Im}} \\ \mathbf{u_3^{Im}} \\ \mathbf{u_4^{Im}} \\ \mathbf{u_5^{Im}} \\ \mathbf{u_6^{Im}} \\ \mathbf{u_7^{Im}} \\ \mathbf{u_8^{Im}} \end{Bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
cos(k_x * a) & 0 & 0 & 0 & -sin(k_x * a) & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & cos(k_x * a) & 0 & 0 & 0 & -sin(k_x * a) & 0 \\
cos(k_y * a) & 0 & 0 & 0 & -sin(k_y * a) & 0 & 0 & 0 \\
0 & cos(k_y * a) & 0 & 0 & 0 & -sin(k_x * a) & 0 & 0 \\
cos((k_x + k_y) * a) & 0 & 0 & 0 & -sin((k_x + k_y) * a) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
sin(k_x * a) & 0 & 0 & 0 & cos(k_x * a) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & sin(k_x * a) & 0 & 0 & 0 & cos(k_x * a) & 0 \\
sin(k_y * a) & 0 & 0 & 0 & cos(k_y * a) & 0 & 0 & 0 \\
0 & sin(k_y * a) & 0 & 0 & 0 & cos(k_x * a) & 0 & 0 \\
sin((k_x + k_y) * a) & 0 & 0 & 0 & cos((k_x + k_y) * a) & 0 & 0 & 0
\end{bmatrix} \begin{Bmatrix} \mathbf{u_0^{Re}} \\ \mathbf{u_1^{Re}} \\ \mathbf{u_3^{Re}} \\ \mathbf{u_4^{Re}} \\ \mathbf{u_0^{Im}} \\ \mathbf{u_1^{Im}} \\ \mathbf{u_3^{Im}} \\ \mathbf{u_4^{Im}} \end{Bmatrix} \tag{5}
$$

Note that **u** is a vector. This has to be expanded based on the number of dimensions of the problem. For the case of square plate each **u** will have two components.

The FEM implementation of eq. (2) is done by discretizing the domain into bilinear quadrilateral elements. This can be represented in the matrix form as,

$$
\left(\begin{bmatrix} [K] & 0 \\ 0 & [K] \end{bmatrix} - \omega^2 \begin{bmatrix} [M] & 0 \\ 0 & [M] \end{bmatrix} \right) \begin{Bmatrix} \mathbf{U^{Re}} \\ \mathbf{U^{Im}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F^{Re}} \\ \mathbf{F^{Im}} \end{Bmatrix} \tag{6}
$$

As shown in eq. (5), the **U** and **F** are replaced.

$$
\mathbf{Q}^T \left(\begin{bmatrix} [K] & 0 \\ 0 & [K] \end{bmatrix} - \omega^2 \begin{bmatrix} [M] & 0 \\ 0 & [M] \end{bmatrix} \right) \mathbf{Q} \begin{Bmatrix} \mathbf{\tilde{U}^{Re}} \\ \mathbf{\tilde{U}^{Im}} \end{Bmatrix} = \mathbf{Q}^T \begin{Bmatrix} \mathbf{F^{Re}} \\ \mathbf{F^{Im}} \end{Bmatrix} \mathbf{Q} \tag{7}
$$

The right hand side of the equation is taken as zero which automatically incorporates the neumann part of bloch boundary condition in eq. (3). The problem simplifies to,

$$
\mathbf{K_R \tilde{U}} = \omega^2 \mathbf{M_R} \tag{8}
$$

where,

$$
\begin{aligned} \mathbf{K_R} &= \mathbf{Q}^T \mathbf{KQ} \\ \mathbf{M_R} &= \mathbf{Q}^T \mathbf{MQ} \end{aligned} \tag{9}
$$

Validation

A square plate of 4 cm is taken. The square plate is assumed to be made of aluminium with the material parameters, $E = 7.31 * 10^{10} Pa$, $\nu = 0.325$, $\rho = 2770 kg/m^3$. The **k** is chosen parallel to the y axis, which corresponds to a normal incident wave. ($k_x = 0$)

The analytical result is

$$
\omega = c\sqrt{(n\pi/d)^2 + (k_y + m\pi/d)^2} \tag{10}
$$

where n and m are integers, and c is equal to the longitudinal wave velocity c_L or to the transverse wave velocity c_T.

$$
\begin{aligned}
c_L &= \sqrt{(\lambda + 2\mu)/\rho} \\
c_T &= \sqrt{\mu/\rho}
\end{aligned}
\tag{11}
$$

The dispersion curves obtained from FEM is compared with the analytical formula in eq. (10).

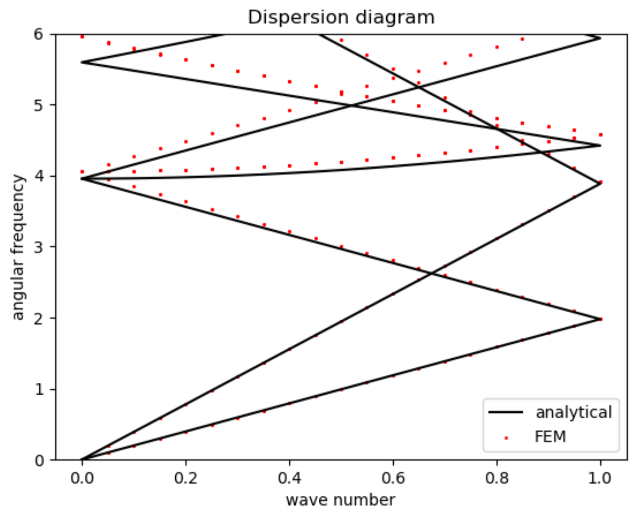

Fig. 2 Variation of reduced angular frequency $\tilde{\omega}$ with reduced wave number $\tilde{k}$

It is seen that there is excellent agreement between them when the $\tilde{\omega}$ is less than 4. It corresponds to the validity limit of the $\frac{\lambda}{4}$ criterion, which states that the largest length of each element in a given mesh has to be smaller than a quarter of the transverse wavelength in the material[2]. Very fine mesh is required to match the values of $\tilde{\omega}$ higher than 4.

Issues with MOOSE

The constraint matrix method as explained in the previous section can be implemented in MOOSE using the constraint system in MOOSE. The application of periodic boundary condition in MOOSE is done in a similar fashion but it's not implemented for bloch type boundary condition. In addition to this, there are issues with using periodic boundary condition and constraint system for eigenvalue problems which are still being looked at by the MOOSE team.

MOOSE does not have a separate stiffness and mass matrix. They are combined into a global jacobian matrix. The matrix can be printed using command line arguments but a way to accessing the global matrix was not found. To the best of my knowledge, the entries to the global matrix is written the adding elements at appropriate places and the rest of the execution is handled by the PetSc and SlepC solvers by using function calls. Even in the code for periodic boundary condition, there are lot of references to libMesh functions.

Additionally, writing the code for this problem would be efficient and faster compared to using MOOSE as the eigenvalue problem has to be solved for multiple values of wave number(k). In the approach mentioned here, the global stiffness and mass matrix have no dependency on the wave number or the lattice length. After the assembly, the constraint matrix is calculated to obtain the reduced mass and stiffness matrix for different values of k. In MOOSE, for each value of k, the entire assembly will be done and this will be inefficient.

Two approaches were tried

- Using functors suggested by the MOOSE team on github. This didnt' work. The code failed to compile.
- Modifying the entries of the jacobian matrix by manipulating the entries. This doesn't work as the eigenvalues are different when executed using the constraint matrix method and this way.

References

[1] Åberg, M., and Gudmundson, P., "The usage of standard finite element codes for computation of dispersion relations in materials with periodic microstructure," *The Journal of the Acoustical Society of America*, Vol. 102, No. 4, 1997, pp. 2007–2013.

[2] Langlet, P., Hladky-Hennion, A.-C., and Decarpigny, J.-N., "Analysis of the propagation of plane acoustic waves in passive periodic materials using the finite element method," *The Journal of the Acoustical Society of America*, Vol. 98, No. 5, 1995, pp. 2792–2800.