Doublet Lattice Theory Document

Graeme J. Kennedy *

1 Introduction

This document contains a brief outline of the DLM theory as it is implemented within the python/Fortran DLM contained within this repository. This implementation is based in part on the method of Albano and Rodden, 1969, and the extension by Rodden, Taylor and McIntosh, 1998.

The pressure potential equation can be written as follows:

$$\beta^2 p_{xx} + p_{yy} + p_{zz} - \frac{2U}{a^2} p_{xt} - \frac{1}{a^2} p_{tt} = 0$$

where p is both the pressure and the pressure potential. Here the free-stream is aligned with the x-coordinate axis. Solutions to the pressure potential equation can be written as follows:

$$p = \frac{1}{R} f(t - \tau)$$

where f is a general function and τ and R are defined as follows:

$$R = \sqrt{(x-\xi)^2 + \beta^2((y-\eta)^2 + (z-\zeta)^2)}$$
$$\tau = \frac{R - M(x-\xi)}{a\beta^2}$$

Now, specializing this general expression to harmonic behavior in time, we can arrive at the following expression:

$$p = \frac{1}{R} e^{\left[\frac{i\omega}{a\beta^2}(M(x-\xi)-R)\right]} e^{i\omega t}$$

$$\begin{aligned}\psi &= \frac{\partial}{\partial z}\left[\frac{1}{R} e^{\left[\frac{i\omega}{a\beta^2}(M(x-\xi)-R)\right]}\right] \\ &= \beta^2(z-\zeta)\left[\frac{1}{R^2} - \frac{1}{R^3}\right] e^{\left[\frac{i\omega}{a\beta^2}(M(x-\xi)-R)\right]}\end{aligned}$$

$$\phi = -\frac{1}{U} e^{-\frac{i\omega x}{U}} \int_{-\infty}^{x} e^{\frac{i\omega x}{U}} \psi \, dx$$

*Assistant Professor, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, email: graeme.kennedy@ae.gatech.edu, phone: 404-894-8911

$$\hat{w} = -\frac{1}{U}\frac{\partial \phi}{\partial z} = \frac{1}{U^2} e^{-\frac{i\omega x}{U}} \int_{-\infty}^{x} e^{\frac{i\omega x}{U}} \psi \, dx$$

$$\hat{w} = \frac{1}{4\pi\rho U^2} \int_S \Delta p e^{-\frac{i\omega(x-\xi)}{U}} \frac{\partial^2}{\partial z^2} \left[\int_{-\infty}^{x-\xi} \frac{1}{R} e^{i\omega\left(\frac{x}{U} - \frac{R-Mx}{a\beta^2}\right)} dx \right] dS$$

The doublet lattice method (DLM) is based on the following expression for the normal wash, $w(x,s)$ on a wing in an oscillating flow where the x-direction is parallel to the free-stream direction:

$$\hat{w}(x,s) = \frac{1}{8\pi} \int_S \Delta C_p(\xi,\eta) \, K(x-\xi, s-\eta) \, d\xi \, d\eta \tag{1}$$

where the kernel function K is given by the following expression:

$$K = e^{-i\omega x_0/U} \left[\frac{K_1 T_1}{r_1^2} + \frac{K_2 T_2}{r_1^4} \right]$$

where:

$$x_0 = x - \xi \qquad y_0 = y - \eta \qquad z_0 = z - \zeta \qquad r_1 = \sqrt{y_0^2 + z_0^2}$$

The terms T_1 and T_2 are the normal and transverse contributions to the normal wash defined as follows:

$$T_1 = \cos(\gamma_r - \gamma_s)$$
$$T_2 = (z_0 \cos\gamma_r - y_0 \sin\gamma_r)(z_0 \cos\gamma_r - y_0 \sin\gamma_r)$$

The terms K_1 and K_2 are contributions to the kernel function given as follows:

$$K_1 = I_1 + \frac{M r_1}{R} \frac{e^{-ik_1 u_1}}{\sqrt{1+u_1^2}}$$

$$K_2 = -3I_2 - i\frac{k_1 M^2 r_1^2}{R^2} \frac{e^{-ik_1 u_1}}{\sqrt{1+u_1^2}} - \frac{M r_1}{R} \left[(1+u_1^2)\frac{\beta^2 r_1^2}{R^2} + 2 + \frac{M r_1 u_1}{R} \right] \frac{e^{-ik_1 u_1}}{(1+u_1^2)^{3/2}}$$

where the following definitions are used:

$$\beta^2 = 1 - M^2 \qquad k_1 = \frac{\omega r_1}{U} \qquad R^2 = x_0^2 + \beta^2 r_1^2 \qquad u_1 = \frac{MR - x_0}{\beta^2 r_1}$$

Finally, I_1 and I_2 are integrals that are given as follows:

$$I_1 = \int_{u_1}^{\infty} \frac{e^{-ik_1 u}}{(1+u)^{3/2}} \, du$$
$$I_2 = \int_{u_1}^{\infty} \frac{e^{-ik_1 u}}{(1+u)^{5/2}} \, du \tag{2}$$

The doublet lattice method contains two essential components:

1. The integral expressions (2) are approximated using an approximation of the function $1 - u/\sqrt{1+u^2}$
2. The kernel function itself is approximated using a quartic expression and integrated across the panel length

2 Approximate I_1 and I_2 integrals

The I_1 and I_2 integrals can be approximated in a similar manner. In both cases, the integrals can be expressed as an integral in $1-u/\sqrt{1+u^2}$. This can be obtained by integrating by parts and shifting the expression for the integrand. For I_1, we can obtain the following expression:

$$\begin{aligned}
I_1 &= \int_{u_1}^{\infty} \frac{e^{-ik_1u}}{(1+u)^{3/2}}\,du \\
&= \frac{ue^{-ik_1u}}{\sqrt{1+u^2}}\bigg|_{u=u_1}^{\infty} + ik_1\int_{u_1}^{\infty}\frac{ue^{-ik_1u}}{\sqrt{1+u^2}}\,du \\
&= \left[1-\frac{u_1}{\sqrt{1+u_1^2}}\right]e^{-ik_1u_1} - ik_1\int_{u_1}^{\infty}\left[1-\frac{u}{\sqrt{1+u^2}}\right]e^{-ik_1u}\,du \\
&= \left[1-\frac{u_1}{\sqrt{1+u_1^2}}\right]e^{-ik_1u_1} - ik_1I_0
\end{aligned}$$

where the integral I_0 is defined as:

$$I_0 = \int_{u_1}^{\infty}\left[1-\frac{u}{\sqrt{1+u^2}}\right]e^{-ik_1u}\,du$$

The second integral can be expressed as follows:

$$I_2 = \int_{u_1}^{\infty}\frac{e^{-ik_1u}}{(1+u^2)^{5/2}}\,du$$

This integral can be simplified as follows:

$$3I_2 = \left[(2+ik_1u_1)\left(1-\frac{u_1}{\sqrt{1+u_1^2}}\right) - \frac{u_1}{(1+u_1^2)^{3/2}}\right]e^{-ik_1u_1} - ik_1I_0 + k_1^2J_0$$

The last integral J_0 is defined as follows:

$$J_0 = \int_{u_1}^{\infty} u\left[1-\frac{u}{\sqrt{1+u^2}}\right]du$$

The integrals I_0 and J_0 can be obtained based on the following approximation:

$$1-\frac{u}{\sqrt{1+u^2}} = \sum_{j=1}^{n} a_j e^{-p_ju}$$

where $n=12$, $p_j = b2^j$, $b=0.009054814793$, and the remaining coefficients are given in Table 1.

Coefficients: a_j
0.000319759140
-0.000055461471
0.002726074362
0.005749551566
0.031455895072
0.106031126212
0.406838011567
0.798112357155
-0.417749229098
0.077480713894
-0.012677284771
0.001787032960

Table 1: Coefficients for the 12-term approximation

Based on these approximations, the integral I_0 can be approximated as follows:

$$I_0 \approx \int_{u_1}^{\infty} \sum_{j=1}^{n} a_j e^{-p_j u - ik_1 u}\, du = \sum_{j=1}^{n} \frac{a_j e^{-(p_j+ik_1)u_1}}{p_j^2+k_1^2}(p_j - ik_1)$$

The integral J_0 can be approximated in a similar manner as follows:

$$J_0 \approx \int_{u_1}^{\infty} \sum_{j=1}^{n} a_j u e^{-(p_j+ik_1)u}\, du = \sum_{j=1}^{n} \frac{a_j e^{-(p_j+ik_1)u_1}((p_j+ik_1)u_1+1)(p_j-ik_1)^2}{(p_j^2+k_1^2)^2}$$

3 Steady-state horseshoe vortex code

The steady state components are calculated based on a horseshoe vortex formulation. In this approach, all that is required is a Pradtl-weighted distance the bound vortex start and end locations to the receiving point.

These distances are denoted as follows:

$$\mathbf{a} = \begin{bmatrix} (x_r - x_i)/\beta \\ y_r - y_i \\ z_r - z_i \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} (x_r - x_o)/\beta \\ y_r - y_o \\ z_r - z_o \end{bmatrix}$$

where (x_r, y_r, z_r) is the receiving point, (x_i, y_i, z_i) is the inboard vortex line point and (x_o, y_o, z_o) is the outboard vortex point. Note that $\beta = \sqrt{1-M^2}$.

The velocity induced by the bound vortex segment is given as follows:

$$\mathbf{v}_b = \frac{\Gamma}{4\pi} \frac{\mathbf{a} \times \mathbf{b}(A+B)}{AB(AB + \mathbf{a}\cdot\mathbf{b})}$$

where $A = \sqrt{\mathbf{a}\cdot\mathbf{a}}$ and $B = \sqrt{\mathbf{b}\cdot\mathbf{b}}$. Note that this formulation for $\mathbf{v}$ is singular only when $\mathbf{a}$ and $\mathbf{b}$ are co-linear and lie pointing towards one another such that $AB = -\mathbf{a}\cdot\mathbf{b}$. Based on the definitions of $\mathbf{a}$ and $\mathbf{b}$, this singularity will only occur when the receiving point lies between the inboard and outboard points.

To account for the inboard and outboard vorticies traveling from a point projected infinitely far downstream, we define the vectors $\mathbf{c}$ and $\mathbf{d}$ as follows:

$$\mathbf{c} = \begin{bmatrix} -\infty \\ y_r - y_i \\ z_r - z_i \end{bmatrix} \qquad \mathbf{d} = \begin{bmatrix} -\infty \\ y_r - y_o \\ z_r - z_o \end{bmatrix}$$

This yields the following expression for the contribution from the inboard vortex segment:

$$\begin{aligned} \mathbf{v} &= \lim_{C\to\infty} \frac{\Gamma}{4\pi} \frac{\mathbf{c}\times\mathbf{a}(A+C)}{AC(AC+\mathbf{c}\cdot\mathbf{a})} \\ &= \frac{\Gamma}{4\pi} \frac{-\mathbf{i}\times\mathbf{a}}{A(A-\mathbf{i}\cdot\mathbf{a})} \\ &= \frac{\Gamma}{4\pi} \frac{\mathbf{a}\times\mathbf{i}}{A(A-\mathbf{i}\cdot\mathbf{a})} \\ &= \frac{\Gamma}{4\pi} \frac{a_z\mathbf{j} - a_y\mathbf{k}}{A(A-a_x)} \end{aligned}$$

Note that we have used the fact that $\lim_{C\to\infty} \mathbf{c}/C = -\mathbf{i}$. Now, accounting for the remaining vortex from the outboard point yields the following:

$$\begin{aligned} \mathbf{v}_o &= \lim_{D\to\infty} \frac{\Gamma}{4\pi} \frac{\mathbf{b}\times\mathbf{d}(D+B)}{BD(BD+\mathbf{b}\cdot\mathbf{d})} \\ &= \frac{\Gamma}{4\pi} \frac{-\mathbf{b}\times\mathbf{i}}{B(B-\mathbf{i}\cdot\mathbf{b})} \\ &= \frac{\Gamma}{4\pi} \frac{-b_z\mathbf{j} + b_y\mathbf{k}}{B(B-b_x)} \end{aligned}$$

Where we have used the fact that $\lim_{D\to\infty} \mathbf{d}/D = -\mathbf{i}$.

4 Flutter analysis

Within the context of this work, we use a flutter analysis that takes the following form:

$$\left[p^2\mathbf{M}(\mathbf{x}) + \mathbf{K}(\mathbf{x}) - q_\infty \mathbf{A}(p)\right]\mathbf{u} = 0 \tag{3}$$

where $\mathbf{M}(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$ are the mass and stiffness matrices from the finite-element equations, which are functions of the design variables $\mathbf{x}$. Furthermore, q_∞ is the dynamic pressure and $\mathbf{A}(p) = \mathbf{T}^T\mathbf{A}_{\mathrm{IC}}(p)\mathbf{T}$, where $\mathbf{A}_{\mathrm{IC}}$ is the aerodynamic influence coefficient matrix, and $\mathbf{T}$ is the load and displacement transfer interpolation matrix. The eigenvalue p gives the frequency and damping of the motion.

The DLM method presented above only uses the imaginary component of the eigenvalue, $\Im\{p\}$, which is the frequency of the oscillation. In this section, we will present two methods that are used to compute approximate solutions of the flutter equation (3). Note that the influence coefficient matrix is complex and is a nonlinear function of p. Therefore, the flutter equation (3) is a generalized nonlinear eigenvalue problem with a solution given by the triplet $(p, \mathbf{v}, \mathbf{u})$, where p is the complex eigenvalue and $\mathbf{v}$ and $\mathbf{u}$ are the left and right eigenvectors, respectively. The triplet satisfies the following equations:

$$\left[p^2\mathbf{M}(\mathbf{x}) + \mathbf{K}(\mathbf{x}) - q_\infty \mathbf{A}(p)\right]\mathbf{u} = 0$$
$$\mathbf{v}^H\left[p^2\mathbf{M}(\mathbf{x}) + \mathbf{K}(\mathbf{x}) - q_\infty \mathbf{A}(p)\right] = 0$$

4.1 Exact flutter derivatives

The exact derivative of the eigenvalue, p, with respect to the design variables can be obtained by differentiating the governing equation of flutter (3) with respect to the design variables. For the k-th design variable, x_k, this yields the following expression:

$$\left[2p\mathbf{M}(\mathbf{x}) - q_\infty \frac{\partial \mathbf{A}}{\partial p}\right]\mathbf{u}\frac{\partial p}{\partial x_k} +$$
$$\left[p^2\frac{\partial \mathbf{M}}{\partial x_k} + \frac{\partial \mathbf{K}}{\partial x_k} - q_\infty \frac{\partial \mathbf{A}(p)}{\partial x_k}\right]\mathbf{u} +$$
$$\left[p^2\mathbf{M}(\mathbf{x}) + \mathbf{K}(\mathbf{x}) - q_\infty \mathbf{A}(p)\right]\frac{\partial \mathbf{u}}{\partial x_k} = 0.$$

Within this work, we fix the aerodynamic mesh and use only structural variables, therefore $\partial\mathbf{A}/\partial x_k = 0$. Pre-multiplying by the conjugate transpose of the left eigenvector eliminates the term with right eigenvector derivatives, and the eigenvalue derivative can be written as follows:

$$\frac{\mathrm{d}p}{\mathrm{d}x_k} = -\frac{\mathbf{v}^H\left[p^2\dfrac{\partial \mathbf{M}}{\partial x_k} + \dfrac{\partial \mathbf{K}}{\partial x_k}\right]\mathbf{u}}{\mathbf{v}^H\left[2p\mathbf{M} - q_\infty \dfrac{\partial \mathbf{A}}{\partial p}\right]\mathbf{u}}.$$

Unfortunately, this derivative can be difficult to evaluate since the exact left and right eigenvectors are not obtained as a byproduct of the solution procedure.

Instead of using the full eigenvalue problem (3), flutter analysis techniques often employ a reduced eigenproblem using a small number of natural frequencies. The reduced modes are the eigenvectors of the problem:

$$\left[\mathbf{K}(\mathbf{x}) - \omega_i^2\mathbf{M}(\mathbf{x})\right]\mathbf{u} = 0,$$

where ω_i is the natural frequency. The eigenvectors $\mathbf{u}$ for $i = 1, \ldots, r$ are collected in the matrix $\mathbf{Q}_r$. These eigenvectors are $\mathbf{M}$-orthonormal, such that $\mathbf{Q}_r^T\mathbf{M}\mathbf{Q}_r = \mathbf{I}_r$. The reduced eigenproblem can now be written as follows:

$$\left[\tilde{p}^2\mathbf{M}_r + \mathbf{K}_r - q_\infty \mathbf{A}_r(\tilde{p})\right]\mathbf{u}_r = 0,$$

with the solution $(\tilde{p}, \mathbf{u}_r, \mathbf{v}_r)$. The reduced matrices take the form:

$$\begin{aligned}\mathbf{M}_r &= \mathbf{Q}_r^T \mathbf{M} \mathbf{Q}_r = \mathbf{I}_r \in \mathbb{R}^{r\times r},\\ \mathbf{K}_r &= \mathbf{Q}_r^T \mathbf{K} \mathbf{Q}_r = \mathrm{diag}\{\omega_i^2\} \in \mathbb{R}^{r\times r},\\ \mathbf{A}_r(p) &= \mathbf{Q}_r^T \mathbf{A}(p) \mathbf{Q}_r \in \mathbb{C}^{r\times r}.\end{aligned}$$

Note that $\mathbf{A}_r$ has no sparsity structure and is a dense matrix in general.

The so-called frozen-mode approximation is to approximate the derivatives of the eigenvalues using the formula

$$\frac{\mathrm{d}p}{\mathrm{d}x_k} \approx -\frac{\mathbf{v}_r^H \left[\tilde{p}^2 \dfrac{\partial \mathbf{M}_r}{\partial x_k} + \dfrac{\partial \mathbf{K}_r}{\partial x_k}\right] \mathbf{u}_r}{\mathbf{v}_r^H \left[2\tilde{p}\mathbf{M} - q_\infty \dfrac{\partial \mathbf{A}_r}{\partial \tilde{p}}\right] \mathbf{u}_r}.$$

where the derivatives of the reduced matrices are approximated as follows:

$$\begin{aligned}\frac{\partial \mathbf{M}_r}{\partial x_k} &\approx \mathbf{Q}_r^T \frac{\partial \mathbf{M}}{\partial x_k} \mathbf{Q}_r\\ \frac{\partial \mathbf{K}_r}{\partial x_k} &\approx \mathbf{Q}_r^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{Q}_r\end{aligned}$$

The frozen-mode approximation can be interpreted in two equivalent ways: (1) that the derivative of the modes or eigenvectors, $\mathbf{Q}_r$, with respect to the design variables is zero, or (2) that the left and right eigenvectors of the full problem are well-approximated by

$$\mathbf{u} = \mathbf{Q}\mathbf{u}_r, \qquad \mathbf{v} = \mathbf{Q}\mathbf{v}_r.$$

Using the second interpretation, note that there is no bound on the error between $(p, \mathbf{u}, \mathbf{v})$ and $(\tilde{p}, \mathbf{Q}_r\mathbf{u}_r, \mathbf{Q}_r\mathbf{v}_r)$. Furthermore, even if $p - \tilde{p}$ is small, there is no guarantee that $\mathbf{Q}_r\mathbf{u}_r$ is then close to $\mathbf{u}$, or equivalently with the left eigenvector.

4.2 Solution methods for eigenvalue problems

In this work, we will make use of two types of generalized eigenvalue solution algorithms: a shifted Lanczos method, and a Jacobi–Davidson method. Both are briefly described in the following section.

4.2.1 Lanczos method

The Lanczos algorithm extracts eigenvalues for symmetric generalized eigenvalue problems. Here, we use this algorithm to solve for the natural frequencies of the structural problem without aerodynamic loads:

$$\mathbf{K}\mathbf{u} = \lambda \mathbf{M}\mathbf{u}.$$

Instead of solving this problem directly, we use a shift and invert strategy to zero-in on the desired spectrum to reduce the number of iterations required. This shift and invert technique produces the following eigenproblem that has the same eigenvectors but different eigenvalues:

$$\mathbf{M}(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{u} = \mu \mathbf{M}\mathbf{u},$$

where the transformed eigenvalue μ is related to the original eigenvalue λ through the relationship:

$$\mu = \frac{1}{\lambda - \sigma}.$$

When σ is chosen such that it lies close to the desired λ, the corresponding transformed eigenvalues, μ, become well separated, making the Lanczos algorithm more efficient.

The Lanczos algorithm uses an $\mathbf{M}$-orthonormal subspace, written as $\mathbf{V}_m \in \mathbb{R}^{n\times m}$, such that $\mathbf{V}_m^T\mathbf{M}\mathbf{V}_m = \mathbf{I}_m$. In exact arithmetic, this subspace can be formed directly from the Lanczos three-term recurrence. However, the resulting subspace loses orthogonality as the algorithm converges to an eigenvalue due to numerical truncation errors. Instead, we use an expensive, but effective, full-orthonormalization procedure (Gram–Schmidt) that enforces $\mathbf{M}$-orthonormality.

Lanczos method for computing eigenvalues/eigenvectors of $\mathbf{Ku} = \lambda\mathbf{Mu}$
Given: m, $\hat{\mathbf{v}}_1$, σ, ε_{tol}
Factor the matrix $(\mathbf{K} - \sigma\mathbf{M})$
Set $i = 1$
while $i \leq m$ **do**
  $\hat{\mathbf{v}}_{i+1} = (\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{v}_i$
  Set $j = 1$
  while $j \leq i$ **do** ▷ Full $\mathbf{M}$-orthonormalization
    $h_{ji} = \mathbf{v}_j^T\mathbf{M}\hat{\mathbf{v}}_{i+1}$
    $\hat{\mathbf{v}}_{i+1} \leftarrow \hat{\mathbf{v}}_{i+1} - h_{ji}\mathbf{v}_j$
    $j \leftarrow j + 1$
  end while
  $\alpha_i \leftarrow h_{ii}$
  $\beta_i = \sqrt{\hat{\mathbf{v}}_{i+1}^T\mathbf{M}\hat{\mathbf{v}}_{i+1}}$
  $\mathbf{v}_{i+1} = \hat{\mathbf{v}}_{i+1}/\beta_i$
  $\mathbf{T}_i = \text{tridiag}_k\{\beta_k, \alpha_k, \beta_{k-1}\}$ ▷ Solve the reduced eigenproblem
  Solve $\mathbf{T}_i\mathbf{y}_i = \theta\mathbf{y}_i$ for $(\theta, \mathbf{y}_i)$
  if $\beta_i\mathbf{y}_i^T\mathbf{e}_i < \varepsilon_{tol}$ **then** ▷ Test for convergence
    $\mathbf{u} = \mathbf{V}_i\mathbf{y}_i$
    $\lambda = \frac{1}{\theta} + \sigma$
    break
  end if
  $i \leftarrow i + 1$
end while

The Lanczos method can be easily extended to find multiple eigenpairs $(\lambda_i, \mathbf{u})$. A byproduct of the Lanczos method is the $\mathbf{M}$-orthonormal subspace. Instead of discarding this subspace, we use these vectors to enhance the flutter prediction and eigenvector computation.

4.2.2 Jacobi–Davidson method

The Lanczos method, and their Arnoldi cousins, are best-suited for regular eigenvalue problems, or generalized eigenproblems that can be transformed into regular eigenproblems as we did above. An alternative class of eigenvalue algorithm is the Jacobi–Davidson (JD) method that can be adapted for general nonlinear eigenproblems. JD methods, like Lanczos methods, build an orthonormal subspace using a series of vectors which are used to approximate the eigenvector. However, these vectors are built, not from a Krylov subspace, but instead from the corrections obtained from inexact Newton method. We briefly outline these methods below and describe an algorithm that builds from an $\mathbf{M}$-orthonormal subspace.

If we think of the eigenproblem as a system of nonlinear equations, with the normalization constraint appended, we arrive at the following system of equations:

$$\begin{aligned}\mathbf{F}(p)\mathbf{u} = \left[p^2\mathbf{M} + \mathbf{K} - q_\infty \mathbf{A}(p)\right]\mathbf{u} = 0,\\ \mathbf{u}^H\mathbf{M}\mathbf{u} = 1.\end{aligned}$$

Note that we have defined $\mathbf{F}(p)$ and we have used the Hermitian transpose since $\mathbf{u} \in \mathbb{C}^n$. Using Newton's method to find an update, $\mathbf{t}$ for the eigenvector and Δp for the eigenvalue produces the following system of equations:

$$\begin{bmatrix}\mathbf{F}(p) & \mathbf{F}'(p)\mathbf{u}\\ \mathbf{u}^H\mathbf{M} & 0\end{bmatrix}\begin{bmatrix}\mathbf{t}\\ \Delta p\end{bmatrix} = -\begin{bmatrix}\mathbf{F}(p)\mathbf{u}\\ 0\end{bmatrix}.$$

Note that the second equation implies that the update $\mathbf{t}$ will be $\mathbf{M}$-orthogonal such that $\mathbf{t} \perp \mathbf{M}\mathbf{u}$. The approximation $\mathbf{u}$ is constructed from the $\mathbf{M}$-orthonormal subspace basis, $\mathbf{V}_m$, as follows:

$$\mathbf{u} = \mathbf{V}_m\mathbf{y}_m.$$

The coefficients $\mathbf{y}_m$ can be obtained by solving the nonlinear eigenproblem:

$$\mathbf{V}_m^H\mathbf{F}(p)\mathbf{V}_m\mathbf{y}_m = 0,$$

which is a reduced eigenvalue problem. When $\mathbf{u}$ is in the subspace, $\mathbf{u} = \mathbf{V}_m\mathbf{y}_m$, the Newton system can be written in the following form:

$$\begin{aligned}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{u}^H}{\mathbf{w}^H\mathbf{u}}\right)\mathbf{F}(p)\left(\mathbf{I} - \frac{\mathbf{u}\mathbf{s}^H}{\mathbf{s}^H\mathbf{u}}\right)\mathbf{t} &= -\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{u}^H}{\mathbf{w}^H\mathbf{u}}\right)(\mathbf{F}(p)\mathbf{u} + \mathbf{w})\\ &= -\mathbf{F}(p)\mathbf{u}\end{aligned}$$

where $\mathbf{w} = \mathbf{F}'(p)\mathbf{u}$ and $\mathbf{s} = \mathbf{M}\mathbf{u}$.

Jacobi–Davidson method for compute eigenvalues/eigenvectors of $\mathbf{F}(p)\mathbf{u} = 0$
Given an initial i, $\mathbf{V}_i$, and m
while $i \leq m$ **do**
  Solve $\mathbf{V}_i^H\mathbf{F}(p)\mathbf{V}_i\mathbf{y}_i = 0$
  Compute $\mathbf{u} = \mathbf{V}_i\mathbf{y}_i$

if $||\mathbf{F}(p)\mathbf{u}|| \leq \varepsilon_{tol}$ **then**
break
end if
Compute $\mathbf{w} = \mathbf{F}'(p)\mathbf{u}$, and $\mathbf{s} = \mathbf{M}\mathbf{u}$
Approximately solve:

$$\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{u}^H}{\mathbf{w}^H\mathbf{u}}\right)\mathbf{F}(p)\left(\mathbf{I} - \frac{\mathbf{s}\mathbf{s}^H}{\mathbf{s}^H\mathbf{s}}\right)\mathbf{t} = -\mathbf{F}(p)\mathbf{u}$$

while $j \leq i$ **do** ▷ Full $\mathbf{M}$-orthonormalization
$h_{ji} = \mathbf{v}_j^H\mathbf{M}\mathbf{t}$
$\mathbf{t} \leftarrow \mathbf{t} - h_{ji}\mathbf{v}_j$
$j \leftarrow j + 1$
end while
$\mathbf{v}_{i+1} = \mathbf{t}/\sqrt{\mathbf{t}^H\mathbf{M}\mathbf{t}}$
end while

4.3 Flutter solution methods

In this section, we describe two flutter solution algorithms that we use to find solutions to reduced flutter problems. These $(p, \mathbf{u}, \mathbf{v})$ or their approximations.

4.3.1 Reduced determinant iteration

Hassig's method of determinant iteration can be used to solve the reduced nonlinear eigenvalue problem. This method is a secant method applied to the determinant equation:

$$\Delta(p) = \det \mathbf{Q}_r^T\mathbf{F}(p)\mathbf{Q}_r. \tag{4}$$

Note that the columns of $\mathbf{Q}_r \in \mathbb{R}^{n\times r}$ are the eigenvectors from the natural frequency eigenproblem. Given initial guesses p_1, and p_2, the method computes p_{k+2} as follows:

$$p_{k+2} = \frac{p_{k+1}\Delta(p_k) - p_k\Delta(p_{k+1})}{\Delta(p_k) - \Delta(p_{k+1})},$$

the iteration is continued until $|\Delta(p_{k+2})| \leq \varepsilon_{tol}$ for some specified tolerance.

4.3.2 Eigenproblem expansion

Another approach that we implement is to expand the reduced eigenproblem, $\mathbf{V}_m^H\mathbf{F}(p)\mathbf{V}_m = 0$, in a Taylor series expansion around an initial guess for p:

$$\begin{aligned}
\mathbf{V}_m^H\mathbf{F}(p+\Delta p)\mathbf{V}_m &= \mathbf{V}_m^H\left[\mathbf{F}_0 + p\mathbf{F}'(p) + \frac{1}{2}p^2\mathbf{F}''(p)\right]\mathbf{V}_m \\
&= \left[(p+\Delta p)^2\mathbf{I}_m + \mathbf{V}_m^H\mathbf{K}\mathbf{V}_m - q_\infty\mathbf{V}_m^H\left(\mathbf{A}(p) + \frac{\partial\mathbf{A}}{\partial p}\Delta p + \frac{1}{2}\frac{\partial^2\mathbf{A}}{\partial p^2}\Delta p^2\right)\mathbf{V}_m\right] \\
&= \mathbf{A}_0 + \Delta p\mathbf{B}_0 + \Delta p^2\mathbf{C}_0
\end{aligned}$$

This general eigenvalue problem can be solved by converting it into a dense general eigenvalue problem:

$$\begin{bmatrix} 0 & -\mathbf{I} \\ \mathbf{A}_0 & \mathbf{B}_0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_r \\ \mathbf{w}_r \end{bmatrix} + \Delta p \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{C}_0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_r \\ \mathbf{w}_r \end{bmatrix} = 0.$$

Note that this is a $2m \times 2m$ generalized eigenproblem that can be solved using LAPACK methods.

4.3.3 A hybrid Lanczos/Jacobi–Davidson flutter analysis method

Above we described both a Lanczos method and a Jacobi–Davidson method for constructing approximations to generalized eigenproblems using **M**-orthogonal subspaces. We use a combination of these methods when performing flutter analysis for design optimization. The flutter analysis proceeds in the following steps:

1. Using the Lanczos method, form an **M**-orthonormal basis to solve for the smallest r eigenvalues of the natural frequency problem, $\mathbf{Ku} = \lambda \mathbf{Mu}$.

2. Using the basis $\mathbf{V}_m$ obtained from the Lanczos method, execute the Jacobi–Davidson method until $||\mathbf{F}(p)\mathbf{u}|| \leq \varepsilon_{tol}$

3. Using the Jacobi–Davidson method, compute the left eigenvector, replacing the eigenproblem $\mathbf{F}(p)$, with the transposed eigenproblem $\mathbf{F}(p)^T$.

Note that using this procedure, we are able to bound the error for both the left and right eigenvectors.

One of the advantages of using the Lanczos method is that the generalized eigenvalue for the natural frequency problem is symmetric and therefore eigenvalues, eigenvectors and the subspace $\mathbf{V}_m \in \mathbb{R}^{n \times m}$ are real. Furthermore, the basis for this subspace will remain fixed, regardless of q_∞ and can be reused at different points within the subspace.

In the Jacobi–Davidson method, however, the subspace vectors are complex. While the subspace formed from subsequent iterations could be reused, this also expands the size of the subspace, increasing the computational cost of the method.

https://github.com/gjkennedy/dlm4py