Using Algorithmic Differentiation for Frequency Sweep Analysis

Eder Medina

B.S. Candidate in Mechanical Engineering

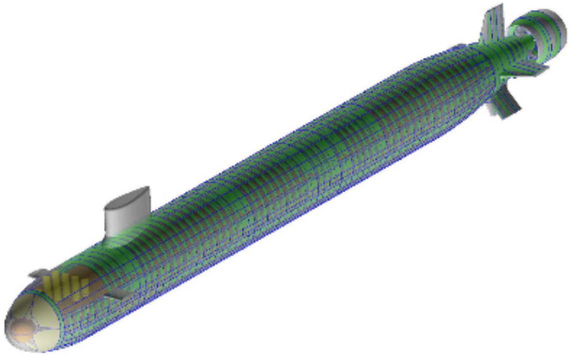
The University of Texas-Austin

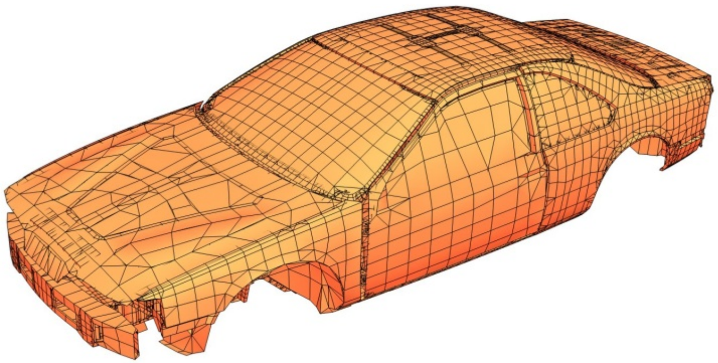
Estimated Graduation Date May 2016

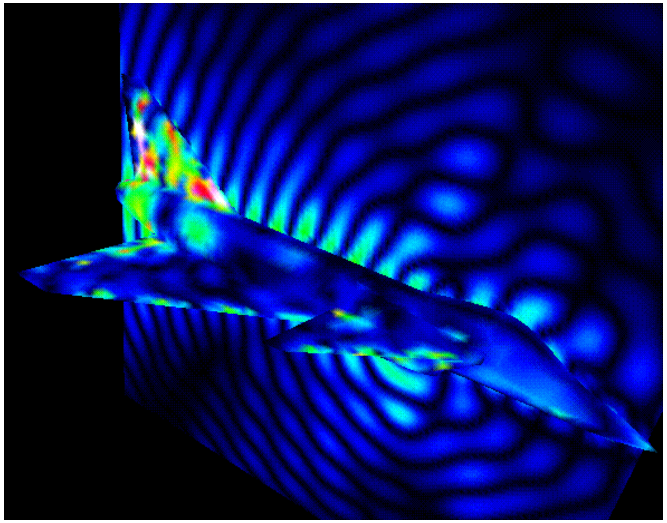
1

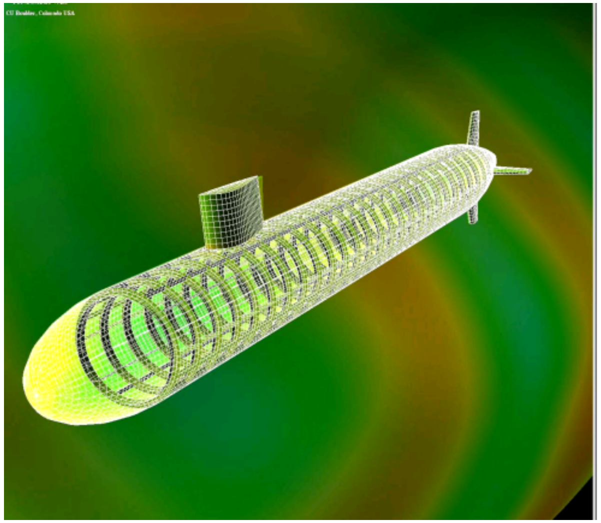
Frequency Sweep Importance

Structural Dynamics and Interior Helmholtz

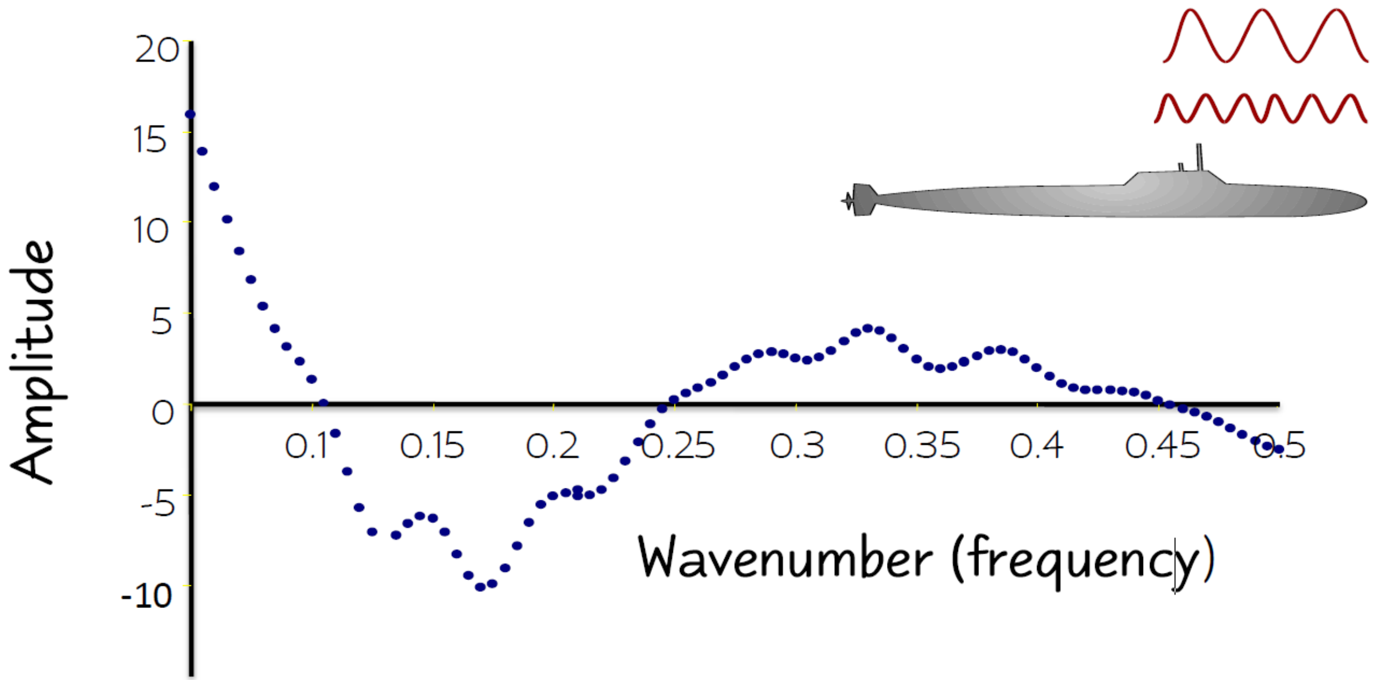








Frequency Sweeps



Solutions to Helmholtz Problem

$$[K]u(\omega) - \omega^2[M]u(\omega) = f(\omega)$$

where $[K]$ is the stiffness matrix and $[M]$ is the mass matrix

$$\frac{d}{d\omega}[(K - \omega^2 M)u(\omega) = f(\omega)]$$

If K and M matrix independent of ω

$$K\frac{du}{d\omega} - 2\omega M u - \omega^2 M\frac{du}{d\omega} = \frac{df}{d\omega}$$

$$K\frac{d^2u}{d\omega^2} - 2Mu - 4\omega M\frac{du}{d\omega} - \omega^2 M\frac{d^2u}{d\omega^2} = \frac{d^2f}{d\omega^2}$$

$$K\frac{d^3u}{d\omega^3} - 6M\frac{du}{d\omega} - 6\omega M\frac{d^2u}{d\omega^2} - \omega^2 M\frac{d^3u}{d\omega^3} = \frac{d^3f}{d\omega^3}$$

$$K\frac{d^4u}{d\omega^4} - 12M\frac{d^2u}{d\omega^2} - 8\omega M\frac{d^3u}{d\omega^3} - \omega^2 M\frac{d^4u}{d\omega^4} = \frac{d^4f}{d\omega^4}$$

$$K\frac{d^5u}{d\omega^5} - 20M\frac{d^3u}{d\omega^3} - 10\omega M\frac{d^4u}{d\omega^4} - \omega^2 M\frac{d^5u}{d\omega^5} = \frac{d^5f}{d\omega^5}$$

$$\vdots$$

$$K\frac{d^nu}{d\omega^n} - n(n-1)M\frac{d^{n-2}u}{d\omega^{n-2}} - 2n\omega M\frac{d^{n-1}u}{d\omega^{n-1}} - \omega^2 M\frac{d^nu}{d\omega^n} = \frac{d^nf}{d\omega^n}$$

Solutions to Helmholtz Problem

If K is dependent on ω then derivatives of the solution have form:

$$Ku(\omega) - \omega^2 M u(\omega) = f(\omega)$$
$$K\frac{du}{d\omega} - \omega^2 M\frac{du}{d\omega} = \frac{df}{d\omega} - 2\frac{dK}{dw}u + 2\omega M u$$
$$K\frac{d^2u}{d\omega^2} - \omega^2 M\frac{d^2u}{d\omega^2} = \frac{d^2f}{d\omega^2} - \frac{d^2K}{d\omega^2}u - 2\frac{dK}{d\omega}\frac{du}{d\omega} + 2Mu + 4\omega M\frac{du}{d\omega}$$
$$K\frac{d^3u}{d\omega^3} - \omega^2 M\frac{d^3u}{d\omega^3} = \frac{d^3f}{d\omega^3} - \frac{d^3K}{d\omega^3}u - 3\frac{d^2K}{d\omega^2}\frac{du}{d\omega} - 3\frac{dK}{d\omega}\frac{d^2u}{d\omega^2} + 6M\frac{du}{d\omega} + 6\omega M\frac{d^2u}{d\omega^2}$$
$$\vdots$$
$$K\frac{d^nu}{d\omega^n} - \omega^2 M\frac{d^nu}{d\omega^n} = \frac{d^nf}{d\omega^n} + \ldots$$

Using General Leibniz Rule- nth derivative can be written as:

$$(K - \omega^2 M)\frac{d^nu}{d\omega^n} = f^{(n)} - \sum_{k=1}^{n}\binom{n-k}{k}u^{(n-k)}(K - \omega^2 M)^{(k)}$$

Storing derivatives of K

Stiffness Matrix

$$K = a(w,u) = \int_\Omega \epsilon(w)^T D \epsilon(u)$$

By the Leibniz's Rule of differentiation under an integral

$$\frac{d}{d\omega}\int_\Omega \epsilon(w)^T D\epsilon(u)d\Omega = \int_\Omega \epsilon(w)^T \frac{dD}{d\omega}\epsilon(u)d\Omega$$

D is the constitutive matrix

Computation of the Derivatives of K

For Isotropic Material the constitutive matrix D has the following form:

$$\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

where the Lame parameter

$$\lambda = \frac{\mu(E-2\mu)}{3\mu-E}$$

where μ is the shear modulus and E is the modulus of elasticity.

Constitutive Matrix

$$\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \Rightarrow$$

$$\lambda \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \mu \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Derivatives can now be taken with respect to the scalar value

$$\frac{d^n \lambda}{d\omega^n}\begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\\1 & 1 & 1 & 0 & 0 & 0\\1 & 1 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \qquad \frac{d^n \mu}{d\omega^n}\begin{bmatrix}2 & 0 & 0 & 0 & 0 & 0\\0 & 2 & 0 & 0 & 0 & 0\\0 & 0 & 2 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 1 & 0\\0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

Rather than storing entire matrices we only store the values of the derivatives and the matrices

Acoustics of Rubber Material

Rubber Material is subject to partially inelastic wave scattering that is dependent on frequency. The material is assumed to be isotropic.

Subject to loss factor

$$\begin{aligned}
&\lambda = \tfrac{\mu(E-2\mu)}{3\mu-E}\\
&E^* = E_o(1+\eta_e i)\\
&\mu^* = \mu_o(1+\eta_\mu i)\\
&i^2 = -1\\
&E_o = E\omega + \alpha_1\\
&\mu_o = \mu\omega + \alpha_2\\
&\eta_e = \eta_1\omega + \alpha_3\\
&\eta_m u = \eta_2\omega + \alpha_4
\end{aligned}$$

*denotes complex modulus

Automatic Differentiation

- Also known as Algorithmic Differentiation or computational differentiation. Not symbolic or numerical differentiation. When implementing “forward” differentiation the software breaks down the problem into elementary operations and applies the Chain Rule to compute the derivatives.
- Several Software Packages out there to experiment with
  - ADOL-C -> fairly user friendly
  - Eigen -> has an unsupported library
  - Sacado-> trilinos product from Sandia National Labs already used in AERO-S

Automatic Differentiation

Forward Example:

$f(x1, x2)=2*x1^2+\cos(x2)$

Set up intermediate variables

v1=x1

v2=x2

v3=2*v1*v1

v4=cos(v2)

v5=v3+V4

f(x1,x2)=v5

Find the x1 partial derivative f

v1'=1

v2'=0

v3'=2*(v1'*v1+v1'*v1) <-CHAIN RULE

V4'=-sin(v2)*v2' <-CHAIN RULE

v5'=v3'+v4'

V4'=0 because -sin(v2)*v2'=0

f'(x1,x2)=2*(v1'*v1+v1'*v1)

plug in values of x1

Automatic Differentiation

```
adouble lam(adouble omega, double muo, double alpha, double etamu, double beta, double Eo, double alpha2, double etaE, double beta2){


        adouble a, b, c, d, lamTR, lamTI, gamma, zeta, lamTR1, lamTR2, lamTI1, lamTI2, gamma2, lamB, lamR, lamI;

        //setting up intermediate variables
        a= muo*omega+alpha;
        b= etamu*omega+beta;
        c= Eo*omega+alpha2;
        d= etaE*omega+beta2;

        //foiling the top out
        lamTR= a*c-2*a*a-a*b*c*d+2*a*a*b*b;
        lamTI= a*c*d-2*a*a*b+a*b*c-2*a*a*b;

        //brealing up the conjugate base for the imaginary part with intermediate variables
        gamma= 3*a-c;
        zeta= -3*a*b+c*d;

        //multiplying top by conjugate base but splitting it up into real and imaginary parts
        lamTR1= gamma*lamTR;
        lamTR2= -1*zeta*lamTI;
        lamTI1= lamTI*gamma;
        lamTI2= lamTR*zeta;

        //calculating new denominator
        gamma2= 3*a*b-c*d;
        lamB= gamma*gamma+gamma2*gamma2;

        //obtaining two functions that contain real and imaginary
        lamR= (lamTR1+lamTR2)/(lamB);
        lamI= (lamTI1+lamTI2)/(lamB);

        //return function lamR for real
        return lamI;
```

```
[eam3349@FRG-10 ADOL-C-2.4.1]$ ./parame
type the derivatives desired

155
what is the value of x at which you wish to compute the derivative at?
.2
to obtain value of the derivative at specified point multiply the nth taylor coefficent by n!
the 1st  taylor coefficient is -1.35139
the 2nd  taylor coefficient is -0.487158
the 3rd  taylor coefficient is 0.0454498
the 4th  taylor coefficient is -0.237613
the 5th  taylor coefficient is 1.07444
the 6th  taylor coefficient is -4.22385
the 7th  taylor coefficient is 13.8256
the 8th  taylor coefficient is -31.251
the 9th  taylor coefficient is -14.0946
the 10th  taylor coefficient is 736.118
etc
```

Future Work

Switching from ADOL-C to Sacado in order to not have to include another library. Learning how SACADO implements Automatic Differentiation is the most difficult part.

Acknowledgements

Radek Tezaur
Jari Toivanen

Resources

Hughes, T.J.R, "*The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*", Dover, Mineola, New York, 2000.

Thank you.
Questions?