• Sensitivity analysis is performed through a form of automatic code differentiation.
• Analysis gives derivative output with respect to a specific input at the value of the input quantity.
• The whole code is viewed as a composition of elementary operations.
• Derivatives of each elementary operation are easy to compute.
• The derivative of the output is then obtained via the chain rule.
• The process is automated by overloading elementary operations (e.g., =, +, -, *, cos, sin, sum, etc.) to also compute derivatives.

Implementation in F90/95:
• Each real declaration is replaced by a new tangent type declaration.
• A pair of header files defines the tangent type as either real (default to original code) or tantype (to include the operator overloading).
• Tangent type variables are implemented as F90/95 structures:
  ```fortran
  integer, PARAMETER, PRIVATE :: double = kind(0.0d0)
  TYPE tangent_type
    real (double) :: v
    real (double) :: dv
  end TYPE tangent_type
  ```
• Intrinsic operations are overloaded so that, for example, \( x \cdot y \) includes the product rule:
  the variable is still the product \( x\%v \cdot y\%v \) while the derivative is \( x\%v \cdot y\%dv + y\%v \cdot x\%dv \).

Beam solution for vertical position as a function of time:
\[
y(x, t) = \frac{A}{\omega_1} \sin(\omega_1 t) W_1(x)\]
given an initial velocity \( v(x, 0) = AW_1(x) \)

Differentiate with respect to shear modulus and density:
\[
\frac{\partial y}{\partial \mu} = \frac{A}{2\mu\omega_1} (\omega_1 t \cos(\omega_1 t) - \sin(\omega_1 t)) W_1(x)
\]
\[
\frac{\partial y}{\partial \rho} = -\frac{A}{2\rho\omega_1} (\omega_1 t \cos(\omega_1 t) - \sin(\omega_1 t)) W_1(x)
\]

Derivatives have the same mode shape.