Sensitivity of Cantilevered Beam Displacement

This is a simple example of direct differentiation to obtain exact response sensitivities. Not even a structural analysis is involved; consider the linear elastic structural analysis problem of a cantilevered beam with length L subjected to a point load F at the free end. For a solid rectangular cross-section of width b and height h, made of a material with modulus of elasticity, E, the maximum displacement of the beam is:

$$u = \frac{F\,L^3}{3\,E\left(\frac{b\,h^3}{12}\right)};$$

For the sake of the subsequent sensitivity calculations, a relatively short “two-by-eight” dimension lumber beam is considered, bending about its weak axis in order to avoid questions about lateral torsional buckling. The input values, considered to be means because the value of the parameters may vary, are specified in units of N and mm:

```
μF = 100;
μL = 2000;
μE = 9500;
μb = 184;
μh = 38;
means = {F -> μF, L -> μL, E -> μE, b -> μb, h -> μh};
```

With those values, the first-order approximation of the mean displacement is:

```
uMean = u /. means // N
```

which yields: `33.3624`

Exact first-order response derivatives

Because we have an analytical expression for the displacement, exact response sensitivities are readily calculated in Mathematica:

```
dudF = D[u, F];
dudL = D[u, L];
dudE = D[u, E];
dudb = D[u, b];
dudh = D[u, h];
```

That means the “gradient vector” is:

```
∇u = {dudF, dudL, dudE, dudb, dudh};
∇u // MatrixForm
```

which yields: $\begin{pmatrix} \frac{4 L^3}{b h^3 E} \\ \frac{12 F L^2}{b h^3 E} \\ -\frac{4 F L^3}{b h^3 E^2} \\ -\frac{4 F L^3}{b^2 h^3 E} \\ -\frac{12 F L^3}{b h^4 E} \end{pmatrix}$

Importance ranking of parameters by predicted displacement change

Even if we evaluate the gradient vector at the mean values, we cannot use the components of that vector to judge which parameter is most influential on the structural response. That is because the units of the components of the gradient vector differ. In order to rank the parameters according to importance it is necessary to normalize the components of the gradient vector. To that end, variability is first assigned to the parameters in terms of coefficients of variation:

```
δF = 0.2;
δL = 0.01;
δE = 0.1;
δb = 0.01;
δh = 0.01;
stdvs = {σF -> δF F, σL -> δL L, σE -> δE E, σb -> δb b, σh -> δh h};
```

Next, each component of the gradient vector is multiplied by the corresponding standard deviation. The components of the resulting vector $\Delta u_i = (\partial u / \partial x_i)\, \sigma_i$ have the same unit, and can be compared. This is described on Page 39 of Professor Der Kiureghian’s 2022 book on Structural and System

Reliability, and I published a paper in 2023 in Structural Safety to amend this approach with the consideration of correlation. In the present case, no correlation is specified and the importance vector is:

```
uChange = Times[∇u,  {σF, σL, σE, σb, σh}] /. stdvs /. means;
uChange // MatrixForm
```

which yields: $\begin{pmatrix} 6.67249 \\ 1.00087 \\ -3.33624 \\ -0.333624 \\ -1.00087 \end{pmatrix}$

That importance ranking shows that the first parameter is most important, and that it is a load parameter. The ranking also shows that the third parameter is second-most important, and the minus sign suggests that it is a resistance parameter. This information makes sense, because those two parameters are *F* and *E*, respectively. The components of this importance vector $\Delta u_i = (\partial u / \partial x_i)\,\sigma_i$ is essentially the predicted change in the displacement response for individual single-standard-deviation changes in the parameter values. From that perspective, the total change in the response is obtained by summing up those changes, Δu_i, giving the total predicted change in the displacement:

```
Total[Abs[uChange]]
```

which yields: 12.3441

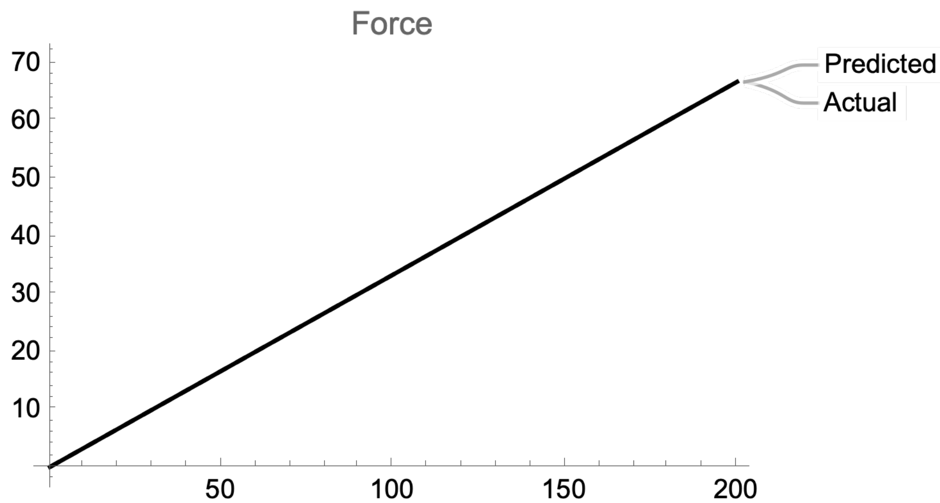
Actual vs. predicted change in displacement

There are two problems with the previous prediction of total displacement change. One issue is that, for all parameters except *F*, the response, *u*, is a nonlinear function of the input parameters. To understand this nonlinearity, consider the difference between the prediction $\mu_u + \frac{\partial u}{\partial x}\,\sigma_x$ and the actual displacement $u(\mu_x + \sigma_x)$ caused by actual parameter perturbations:

```
range = 5;
```

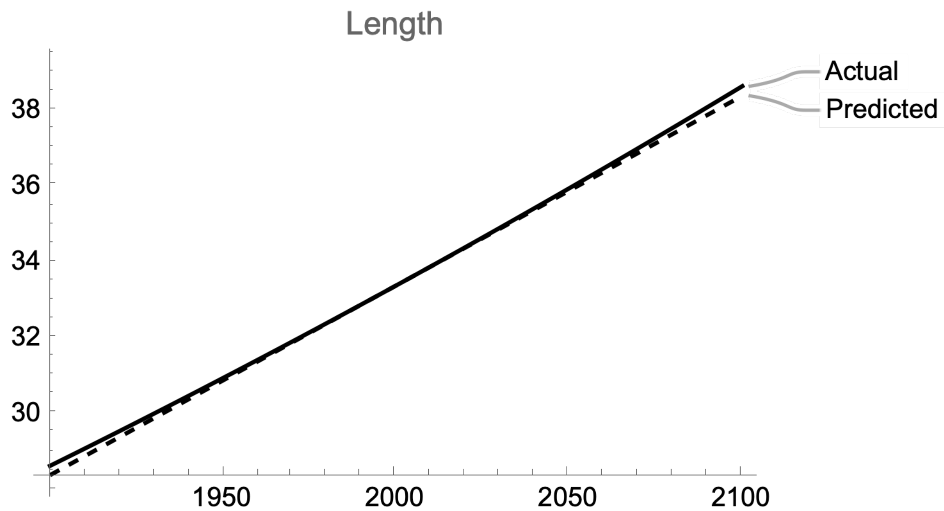
```
uofx = u /. {F -> x} /. means;
uPredicted = uMean + (dudF (x - F)) /. means;
lower = (F - range σF) /. stdvs /. means;
upper = (F + range σF) /. stdvs /. means;
Plot[{uofx, uPredicted}, {x, lower, upper}, PlotLabel -> "Force",
 PlotLabels -> {"Actual", "Predicted"},
 PlotStyle -> {{Black}, {Dashed, Black}}]
```

which yields:



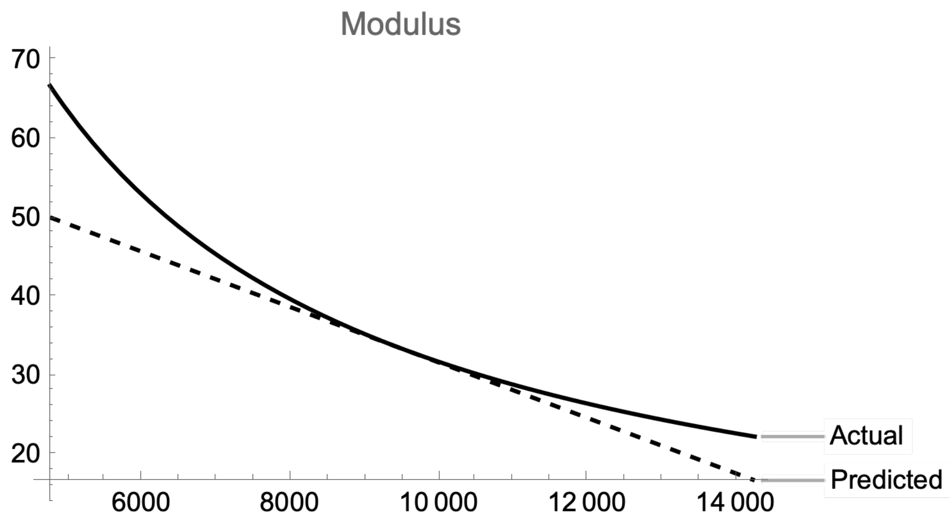
```
uofx = u /. {L -> x} /. means;
uPredicted = uMean + (dudL (x - L)) /. means;
lower = (L - range σL) /. stdvs /. means;
upper = (L + range σL) /. stdvs /. means;
Plot[{uofx, uPredicted}, {x, lower, upper}, PlotLabel -> "Length",
 PlotLabels -> {"Actual", "Predicted"},
 PlotStyle -> {{Black}, {Dashed, Black}}]
```

which yields:



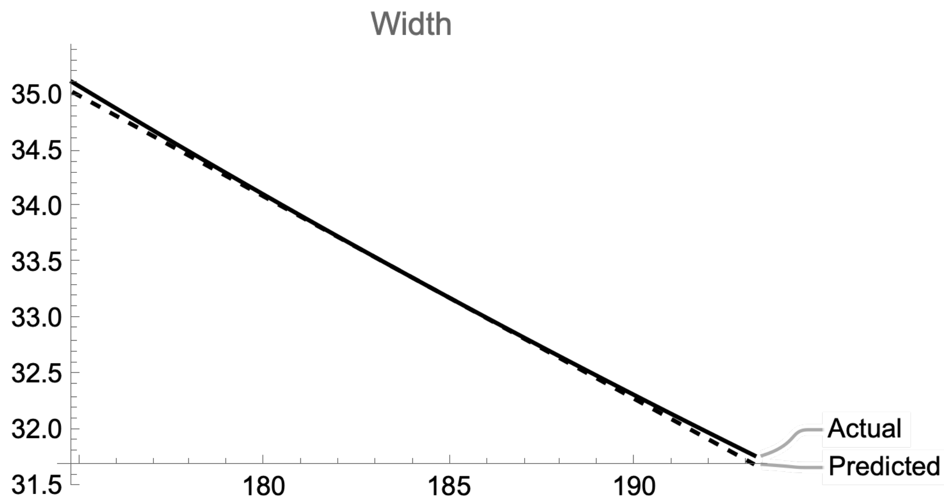
```
uofx = u /. {E -> x} /. means;
uPredicted = uMean + (dudE (x - E)) /. means;
lower = (E - range σE) /. stdvs /. means;
upper = (E + range σE) /. stdvs /. means;
Plot[{uofx, uPredicted}, {x, lower, upper}, PlotLabel -> "Modulus",
 PlotLabels -> {"Actual", "Predicted"},
 PlotStyle -> {{Black}, {Dashed, Black}}]
```

which yields:



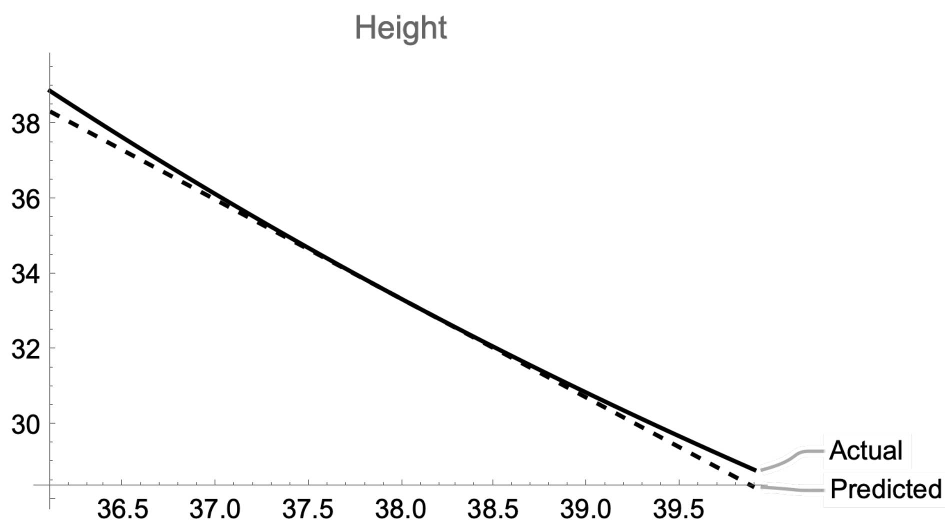
```
uofx = u /. {b -> x} /. means;
uPredicted = uMean + (dudb (x - b)) /. means;
lower = (b - range σb) /. stdvs /. means;
upper = (b + range σb) /. stdvs /. means;
Plot[{uofx, uPredicted}, {x, lower, upper}, PlotLabel -> "Width",
 PlotLabels -> {"Actual", "Predicted"},
 PlotStyle -> {{Black}, {Dashed, Black}}]
```

which yields:



```
uofx = u /. {h -> x} /. means;
uPredicted = uMean + (dudh (x - h)) /. means;
lower = (h - range σh) /. stdvs /. means;
upper = (h + range σh) /. stdvs /. means;
Plot[{uofx, uPredicted}, {x, lower, upper}, PlotLabel -> "Height",
 PlotLabels -> {"Actual", "Predicted"},
 PlotStyle -> {{Black}, {Dashed, Black}}]
```

which yields:



That nonlinearity is included in the following, where the change in displacement response due to actual and individual parameter changes are calculated using the formula $u(\mu_x + \sigma_x) - u(\mu_x)$:

```
i = 1;
uofx = u /. {F -> x} /. means;
uChangeActual =
  {(uofx /. x -> (F + (Sign[dudF] σF) /. stdvs /. means // N)) - uMean};
uChangeActual[[i++]]
```

which yields: `6.67249`

```
uofx = u /. {L -> x} /. means;
AppendTo[uChangeActual,
  (uofx /. x -> (L + (Sign[dudL] σL) /. stdvs /. means // N)) - uMean];
uChangeActual[[i++]]
```

which yields: `1.01092`

```
uofx = u /. {E -> x} /. means;
AppendTo[uChangeActual,
  (uofx /. x -> (E + (Sign[dudE] σE) /. stdvs /. means // N)) - uMean];
uChangeActual[[i++]]
```

which yields: `3.70694`

```
uofx = u /. {b -> x} /. means;
AppendTo[uChangeActual,
  (uofx /. x -> (b + (Sign[dudb] σb) /. stdvs /. means // N)) - uMean];
uChangeActual[[i++]]
```

which yields: `0.336994`

```
uofx = u /. {h -> x} /. means;
AppendTo[uChangeActual,
  (uofx /. x -> (h + (Sign[dudh] σh) /. stdvs /. means // N)) - uMean];
uChangeActual[[i++]]
```

which yields: `1.02123`

Summing up those changes, the total change in the displacement is slightly larger than before:

```
Total[uChangeActual]
```

which yields: `12.7486`

In fact, here is a side-by-side comparison of predicted vs. actual changes in u for individual parameter changes:

```
Abs[uChange]
```

which yields: `{6.67249, 1.00087, 3.33624, 0.333624, 1.00087}`

```
uChangeActual
```

which yields: `{6.67249, 1.01092, 3.70694, 0.336994, 1.02123}`

Accounting for interaction

For a summative model, the change in the displacement when two or more parameters are perturbed simultaneously is accurately predicted by summing the individual changes quantified earlier in this document. However, in the face of interaction between parameters, which is a feature of the present model for *u*, the change in the displacement when two or more parameters are perturbed simultaneously require further attention. Consider the simultaneous change in *E* and *h*. The actual total change by summation of individual contributions is:

```
uChangeActual[[3]] + uChangeActual[[5]]
```

which yields: 4.72817

In contrast, the change in the displacement when both those parameters are perturbed simultaneously is:

```
((u /. {F -> μF, L -> μL, E -> (μE + Sign[dudE] σE), b -> μb,
      h -> (μh + Sign[dudh] σh)}) - uMean) /. stdvs /. means
```

which yields: 4.84164

That increase is caused by interaction, i.e., the multiplicative nature of the considered displacement model. When all parameters are varied simultaneously, accounting for interaction yields the following total change in displacement:

```
uChangeActualInteraction =
 ((u /. {F -> (μF + Sign[dudF] σF), L -> (μL + Sign[dudL] σL),
       E -> (μE + Sign[dudE] σE), b -> (μb + Sign[dudb] σb),
       h -> (μh + Sign[dudh] σh)}) - uMean) /. stdvs /. means
```

which yields: 14.3487

That result is compared with the two previously presented results, the first without nonlinearity and interaction, the second without interaction:

```
Total[Abs[uChange]]
```

which yields: 12.3441

```
Total[uChangeActual]
```

which yields: 12.7486

Put simply, interaction matters, and it is NOT captured by individual first-order derivatives.

Exact second-order response derivatives

In order to include interaction in the direct differentiation approach, second-order response derivatives are considered:

```
dudFF = D[dudF, F];
dudFL = D[dudF, L];
dudFE = D[dudF, E];
dudFb = D[dudF, b];
dudFh = D[dudF, h];

dudLF = D[dudL, F];
dudLL = D[dudL, L];
dudLE = D[dudL, E];
dudLb = D[dudL, b];
dudLh = D[dudL, h];

dudEF = D[dudE, F];
dudEL = D[dudE, L];
dudEE = D[dudE, E];
dudEb = D[dudE, b];
dudEh = D[dudE, h];

dudbF = D[dudb, F];
dudbL = D[dudb, L];
dudbE = D[dudb, E];
dudbb = D[dudb, b];
dudbh = D[dudb, h];

dudhF = D[dudh, F];
dudhL = D[dudh, L];
dudhE = D[dudh, E];
dudhb = D[dudh, b];
dudhh = D[dudh, h];
```

Similar to the collection of first-order derivatives in the gradient vector, the second-order derivatives are collected in the Hessian matrix:

```
H = {{dudFF, dudFL, dudFE, dudFb, dudFh},
   {dudLF, dudLL, dudLE, dudLb, dudLh},
   {dudEF, dudEL, dudEE, dudEb, dudEh},
   {dudbF, dudbL, dudbE, dudbb, dudbh},
   {dudhF, dudhL, dudhE, dudhb, dudhh}};
H // MatrixForm
```

which yields:

$$\begin{pmatrix} 0 & \frac{12 L^2}{b h^3 E} & -\frac{4 L^3}{b h^3 E^2} & -\frac{4 L^3}{b^2 h^3 E} & -\frac{12 L^3}{b h^4 E} \\ \frac{12 L^2}{b h^3 E} & \frac{24 F L}{b h^3 E} & -\frac{12 F L^2}{b h^3 E^2} & -\frac{12 F L^2}{b^2 h^3 E} & -\frac{36 F L^2}{b h^4 E} \\ -\frac{4 L^3}{b h^3 E^2} & -\frac{12 F L^2}{b h^3 E^2} & \frac{8 F L^3}{b h^3 E^3} & \frac{4 F L^3}{b^2 h^3 E^2} & \frac{12 F L^3}{b h^4 E^2} \\ -\frac{4 L^3}{b^2 h^3 E} & -\frac{12 F L^2}{b^2 h^3 E} & \frac{4 F L^3}{b^2 h^3 E^2} & \frac{8 F L^3}{b^3 h^3 E} & \frac{12 F L^3}{b^2 h^4 E} \\ -\frac{12 L^3}{b h^4 E} & -\frac{36 F L^2}{b h^4 E} & \frac{12 F L^3}{b h^4 E^2} & \frac{12 F L^3}{b^2 h^4 E} & \frac{48 F L^3}{b h^5 E} \end{pmatrix}$$

New predictions

Having access to second-order derivatives, new individual sensitivity-based predictions of the change in the displacement u are possible. Here are the results accounting for curvature, but still *not* interaction:

```
i = 1;
uChangeSecond = {(Abs[dudF] σF + 1/2 dudFF σF^2) /. stdvs /. means // N};
uChangeSecond[[i++]]
```

which yields: 6.67249

```
AppendTo[uChangeSecond,
  (Abs[dudL] σL + 1/2 dudLL σL^2) /. stdvs /. means // N];
uChangeSecond[[i++]]
```

which yields: 1.01088

```
AppendTo[uChangeSecond,
   (Abs[dudE] σE + 1/2 dudEE σE^2) /. stdvs /. means // N];
uChangeSecond[[i++]]
```

which yields: `3.66987`

```
AppendTo[uChangeSecond,
   (Abs[dudb] σb + 1/2 dudbb σb^2) /. stdvs /. means // N];
uChangeSecond[[i++]]
```

which yields: `0.336961`

```
AppendTo[uChangeSecond,
   (Abs[dudh] σh + 1/2 dudhh σh^2) /. stdvs /. means // N];
uChangeSecond[[i++]]
```

which yields: `1.02089`

Notice that all diagonal components of the Hessian matrix are positive. These results show that the predicted change in the displacement is starting to align closely with the actual change.

```
uChangeSecond
```

which yields: `{6.67249, 1.01088, 3.66987, 0.336961, 1.02089}`

Compare that to previously obtained results:

```
Abs[uChange]
```

which yields: `{6.67249, 1.00087, 3.33624, 0.333624, 1.00087}`

```
uChangeActual
```

which yields: `{6.67249, 1.01092, 3.70694, 0.336994, 1.02123}`

The observation is made that the results based on second-order response sensitivities are now closely aligned with the response changes when the parameters are actually perturbed. That is reflected in the new second-order-based sum of response changes, still observing that the effect of interaction is missing:

```
Total[uChangeSecond]
```

which yields: 12.7111

```
Total[uChangeActual]
```

which yields: 12.7486

```
uChangeActualInteraction
```

which yields: 14.3487

Predictions with interaction

In order to account for interaction in the sensitivity-based predictions, it is necessary to consider the off-diagonal components of the Hessian matrix. That said, it is still the first-order derivatives that govern the direction of the imagined parameter perturbations:

```
Fchange = Sign[dudF] σF;
Lchange = Sign[dudL] σL;
Echange = Sign[dudE] σE;
bchange = Sign[dudb] σb;
hchange = Sign[dudh] σh;
vectorOfParameterChanges =
  {Fchange, Lchange, Echange, bchange, hchange};
```

The sensitivity-based prediction of the displacement change is now:

```
(∇u.vectorOfParameterChanges +
    1/2 vectorOfParameterChanges.H.vectorOfParameterChanges) /. stdvs /.
  means // N
```

which yields: 14.129

That is indeed closer to the actual response change if all parameters were simultaneously perturbed:

```
uChangeActualInteraction
```

which yields: `14.3487`