Beam Deflection in Hollow Profiles
Max Figura – 2024/03/07

1. Abstract

Mathematical modeling has a lot of utility in physical engineering. Here, we construct a set of linear approximations of a fourth-order differential equation to determine how a long beam might be displaced when a force is applied along its length. While the primary purpose of the project is to implement and test the compuational model, one condition that we wished to compare in our testing is the effect of the cross-section of a beam on its distortion. We specifically compared two kinds of hollow profiles - a square shape and a circular annulus - and found that when only under their own weight, they perform rather similarly.

2. Introduction

In engineering and materials science, it is frequently vital to consider how components of a construction will respond to different kinds of stress. Many factors can decide this, such as the size and shape of the component, the material used, how it is affixed to the larger structure, and the profile of force upon it. Here, we consider the relatively simple case of a beam, much longer than it is thick, with a force applied perpendicular along its edge. We can then consider the deflection of the beam with respect to the force as a fourth-order differential equation, the solution to which can be approximated computationally.

3. Mathematical Background

Consider a beam of length L with a relatively small cross-section, constant across the length, lying horizontal and suspended at discrete points. If $f(x)$ is the force applied at each point x along the beam, and $y(x)$ is the resulting vertical displacement, then the overall deflection is described the following equation:

$$EIy^{(4)}(x) = f(x) \qquad 0 \leq x \leq L$$

Here, E is the Young's modulus of the material used, expressed as a quantity of pressure, and I is the second moment of inertia, dependent on the cross-section of the beam and given as length to the fourth power. Because this is a fourth-order differential equation, we need to set four conditions to arrive at a single solution. This is where the manner in which the beam is affixed comes into play, as we can define the following:

- If a point x is *fixed*, it does not displace or rotate, hence $y(x) = 0$ and $y'(x) = 0$.
- If a point x is *pinned*, it does not displace but is allowed to rotate, and is thus assumed not to distort, so $y(x) = 0$ and $y''(x) = 0$.
- If either end of the beam is left *free*, such as at $x = L$, it may displace and rotate and is thus assumed not to distort, so $y''(L) = 0$, and analytically we also know $y'''(L) = 0$.

To keep it simple, we will only be considering the condition at either end of the beam and assuming the middle is unsupported; this gives us two boundary conditions at each end, for the necessary four.

To solve this problem computationally, we need to be able to consider x discretely. To do this, we utilise the generic Taylor series

$$f(x+h) = \sum_{k=0}^{\infty} \frac{h^k}{k!} f^{(k)}(x), \text{ or } f(x+h) = \sum_{k=0}^{n} \frac{h^k}{k!} f^{(k)}(x) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(\xi)$$

with ξ being some value between x and $x+h$ that represents the corresponding error for the chosen level of precision. In this manner we can divide the beam into n subintervals of length $h = \frac{L}{n}$ and index points along it as $x_i = i \times h$ for $0 \leq i \leq n$. Then we write $y_i = y(x_i)$ and $f_i = f(x_i)$. With this notation established, we can find a discrete approximiation of the fourth derivative of $y(x)$:

$$\begin{aligned}
&\frac{y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}}{h^4} \\
&= \frac{1}{h^4}\left(\sum_{k=0}^{\infty} \frac{(-2h)^k}{k!} y^{(k)}(x) - 4\sum_{k=0}^{\infty} \frac{(-h)^k}{k!} y^{(k)}(x) + 6y(x) - 4\sum_{k=0}^{\infty} \frac{h^k}{k!} y^{(k)}(x) + \sum_{k=0}^{\infty} \frac{(2h)^k}{k!} y^{(k)}(x)\right) \\
&= \frac{0y(x) + 0hy'(x) + 0h^2y''(x) + 0h^3y'''(x) + h^4y^{(4)}(x) + 0h^5y^{(5)}(x) + \frac{1}{6}h^6y^{(6)}(x) + \ldots}{h^4} \\
&= y^{(4)}(x) + \frac{1}{6}h^2y^{(6)}(\xi)
\end{aligned}$$

And so

$$y^{(4)}(x) = \frac{y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}}{h^4} - \frac{1}{6}h^2y^{(6)}(\xi)$$

And we can discretely approximate $y^{(4)}(x)$ with error $\mathcal{O}(h^2)$. With this we can write a system of linear equations with the bulk appearing as

$$y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2} = \frac{h^4}{EI} f_i \text{ for } 3 \leq i \leq n-3$$

and the coefficients at either end determined by how the beam is affixed.

For the initial implementation of this model, we are only considering f to be the force of gravity on the beam itself. Across the beam, this would be a constant value of $f = -S\rho g$, with S being the cross-sectional area, ρ the density of the material, and gravitational acceleration $g \approx 9.81\text{m/s}^2$. In all cases we will be using alumninium as the material, with $E = 69\text{GPa}$ and $\rho = 2700\text{kg/m}^3$. We will be considering two different hollow profiles: a square shape with outer side length 8.25cm and inner side length 4.25cm, and a circle (or annulus) with outer radius ~ 4.987cm and inner radius ~ 2.992cm. This should give us a constant cross-section area of $S = 50\text{cm}^2$, and by extension a constant mass, but with differing moments of inertia. Lastly, we will keep the left end of the beam fixed but compare when the right end is pinned or left free. For each combination, we want to examine the performance at multiple scales, so we will start with $n = 10$ and double after each to a maximum of $n = 1280$.

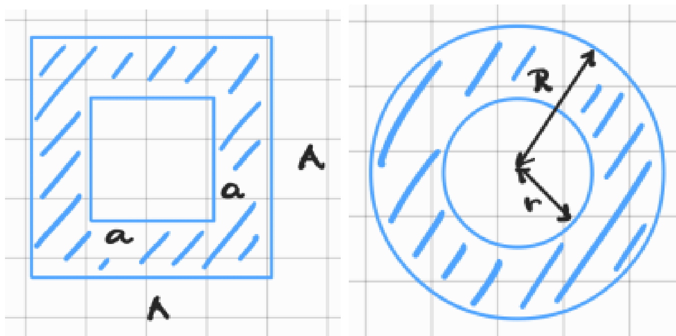


Figure 1. Sketches of the hollow profiles considered. Note that both have a single dimension each for inner and outer shapes. (Provided by Dr. Bělík)

4. Implementation

In finding a system of linear equations to describe this problem, we actually end up with what corresponds to a pentadiagonal matrix. This allows us to solve the system with an expedited form of Gaussian elimination; even though the stability of that algorithm is not necessarily guaranteed for the matrices we will be solving, they are derived in such a way that we can safely take this approach.

Using the GNU Octave language, I had created a function for solving pentadiagonal matrices given the values in each diagonal and the product vector. I tested its performance with a few known examples, including a randomly-generated problem and a large tridiagonal matrix and was satisfied that it works properly (Full code for `GaussElimPenta.m` and test logs can be found in Appendix A). I then constructed a program (Appendix B, modified from that provided by Dr. Bělík) that, given options for the cases we are here modeling, generates the corresponding system and solves it using that pentadiagonal solving algorithm.

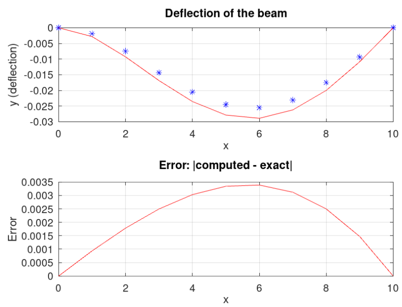
The analytical solutions to this problem take different form depending on how the beam is affixed, but they are known, and this fact allows us to check the valitidy of our model. The program accordingly generates plots of the approximated and actual displacements, and the error between the two.

5. Results

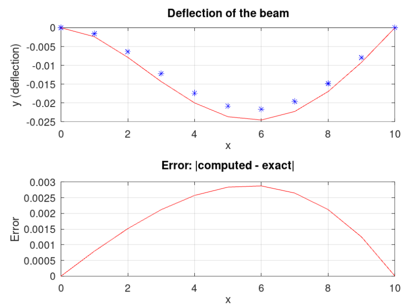
Our results from this experiment are quite promising. The scale of the graphs in Figure 2 exaggerate the amount by which the beam would deflect, but we see the kind of bend that we might expect in each case. In the free cases particularly, the largest displacement ends up being just under 70cm, which is several times the thickness of the beam and would be quite visible.

As we increase n for more precise results (Figure 3), the difference becomes less and less visually noticeable, but examining the error we can see a steady increase in accuracy. There aren't really any surprises here; error curve shrinks in scale but maintains the same general shape from one trial to the next. In the free cases the error is a little less intutive, with a low arc between 0 and 7 meters before spiking

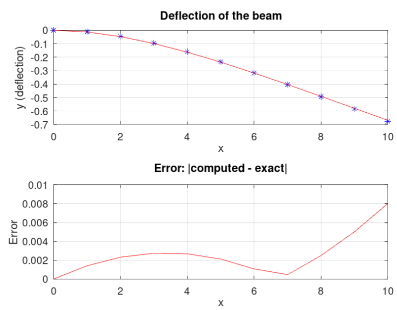
(A) Pin, Square



(B) Pin, Annulus



(C) Free, Square



(D) Free, Annulus

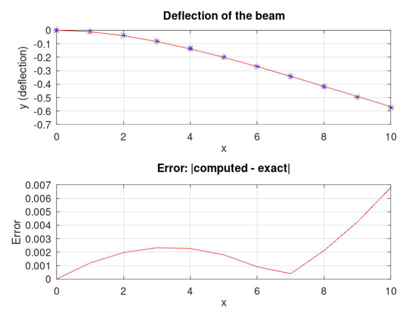


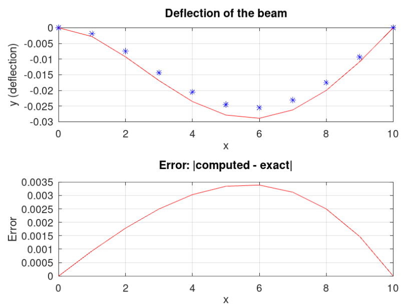
FIGURE 2. Plots showing each scheme for $n = 10$. Exact values are marked in blue stars, and absolute error is shown below each.

up at the right end, but this is consistent with the pinned cases if we consider that we are measuring absolute unsigned error. If we did not plot the absolute value of the error, we would see a similar single arc as to when pinned, only crossing $y = 0$ at a point where the approximation and exact curves happen to converge before switching sign. That meeting point does drift depending on the number of iterations (Figure 4), but it still might be worth investigating why that crossing occurs.

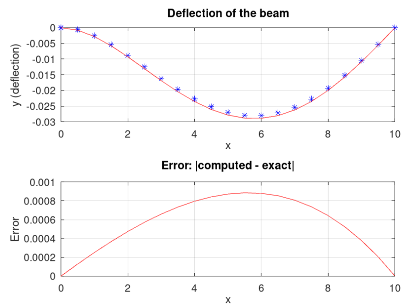
We can also look at the data in aggregate to assess the error of the model. In addition to the plots generated by each trial, the value of the maximum error was recorded into Table 1. This shows us a few trends present in the model's performance, including confirming that our error matches $\mathcal{O}(h^2)$ almost universally. The only apparent excepting to this is the last trial of the free cases, but otherwise any two consecutive trials differ by a factor of $\frac{1}{4}$ as we double n - and by extension, halve the value h.

Also of note is the performance differences between the cases. We consistently see a slightly lower error with the annulus compared to the square, but rather more consequential is the the trend of the free cases exhibiting around twice as much error as when pinned. But this is again not surprising. In the latter, we have explicitly defined the value of the displacement itself at either end, both providing two direct points of reference and presumably limiting the varience between. Meanwhile, the former has the same number of boundary conditions, but only one directly

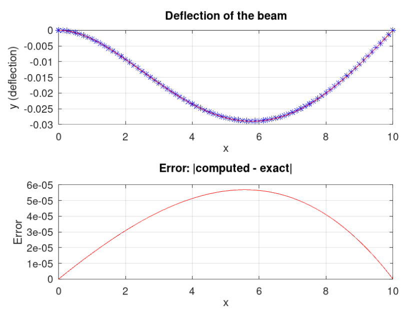
(A) $n = 10$



(B) $n = 20$



(C) $n = 80$



(D) $n = 1280$

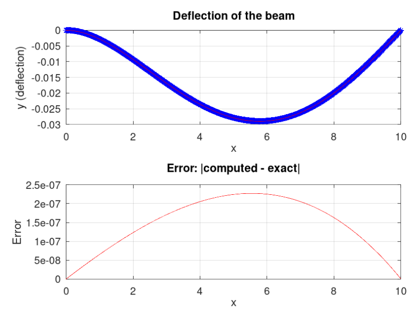
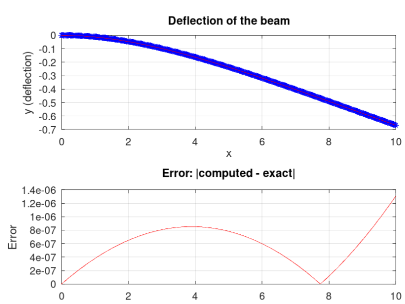


FIGURE 3. A selection of plots from the pinned square case.

(A) $n = 640$



(B) $n = 1280$

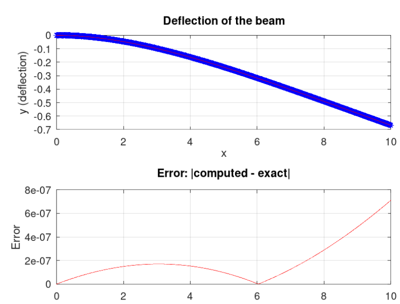


FIGURE 4. Selected plots from the free square cases, showing the approximation crosses with the exact curve at anywhere between 6 and 8 meters.

corresponds to the function we are trying to approximate, which leaves significantly more room for error.

Lastly, while the primary purpose of this exercise was to develop and test a model for beam deflection, it would be remiss not to reflect on the actual engineering implications of our results. The biggest take-away there is that the circular annulus

n	Pin, Square	Pin, Annulus	Free, Square	Free, Annulus
10	3.3874×10^{-3}	2.8752×10^{-3}	8.0228×10^{-3}	6.8098×10^{-3}
20	8.8541×10^{-4}	7.5154×10^{-4}	1.8386×10^{-3}	1.5606×10^{-3}
40	2.2540×10^{-4}	1.9132×10^{-4}	4.3875×10^{-4}	3.7241×10^{-4}
80	5.6839×10^{-5}	4.8245×10^{-5}	1.0708×10^{-4}	9.0886×10^{-5}
160	1.4271×10^{-5}	1.2113×10^{-5}	2.6440×10^{-5}	2.2442×10^{-5}
320	3.5751×10^{-6}	3.0346×10^{-6}	6.5394×10^{-6}	5.5506×10^{-6}
640	8.9515×10^{-7}	7.5980×10^{-7}	1.3107×10^{-6}	1.1125×10^{-6}
1280	2.2728×10^{-7}	1.9291×10^{-7}	7.1150×10^{-7}	6.0393×10^{-7}

Table 1. Maximum error in each trial

appears more structurally sound than the hollow square, with a lower maximum displacement in all cases. I would be inclined to generalise and suggest that an annulus profile performs better than any other simple hollow beam, as often seems to be the case in comparing circular to polygonal structures. But that also usually comes with the trade-off that circular components are harder to create and don't always hold up in larger constructions. Either way, the peak deflection was 2-3cm when pinned, which might be acceptable under the circumstances but still seems to be a concerningly large displacement for a beam only under its own weight.

6. Conclusion

All in all, our model appears quite successful and clearly establishes the expected quadratically diminishing error. We've also managed to draw some initial conclusions about the performance of two different hollow cross-sections, and that while the annulus may be stronger, the difference may be considered minor and practical considerations may be more consequential. Now that the initial model is constructed, there are several more experiments that could be explored, such as applying different forces, testing other materials or cross-sections, or affixing the beam differently. As a continuation of this study, I would be particularly interested in examining different scales of hollow profiles given their two-parameter nature. If the mass is still kept constant, does a larger but thinner annulus perform better or worse than the logical extreme of a tight, solid rod? And what other problems within this domain of materials stress testing could we model using a similar scheme? There is always much more to learn.

7. Sources

- "Project 1: Systems of Linear Equations: The Beam Problem", provided by Dr. Bělík
- "Beam Cross Sections.pdf", provided by Dr. Bělík
- Pseudocode for pentadiagonal system solver, adapted by Dr. Bělík from "Numerical Mathematics and Computing" by Ward Cheney and David Kincaid

Appendix A. GaussElimPenta.m Code and Testing

```
function x = GaussElimPenta(e, a, d, c, f, b)
    n = length(b);
    xmult = 0.0;
    for k = 1:(n-2)
        xmult = a(k)/d(k);
        d(k+1)-=xmult*c(k);
        c(k+1)-=xmult*f(k);
        b(k+1)-=xmult*b(k);
        xmult = e(k)/d(k);
        a(k+1)-=xmult*c(k);
        d(k+2)-=xmult*f(k);
        b(k+2)-=xmult*b(k);
    end
    xmult = a(n-1)/d(n-1);
    d(n)-=xmult*c(n-1);
    b(n)-=xmult*b(n-1);

    x = zeros([n,1]);
    x(n) = b(n)/d(n);
    x(n-1) = (b(n-1) - c(n-1)*x(n)) / d(n-1);
    for i = (n-2):-1:1
        x(i) = (b(i) - c(i)*x(i+1) - f(i)*x(i+2)) / d(i);
    end
endfunction
```

```
octave:1> d = [5, 5, 5, 5];
a = [1, 1, 1];
c = [-1, -1, -1];
e = [-1, -1];
f = [1, 1];
b = [5, 6, 4, 5];
x = GaussElimPenta(e, a, d, c, f, b)
x =

   1
   1
   1
   1

octave:8> d = 5*ones([50,1]);
a = ones([49,1]); c = ones([49,1]);
e = ones([48,1]); f = ones([48,1]);
b = [7; 8; 9*ones([46,1]); 8; 7];
GaussElimPenta(e, a, d, c, f, b)
ans =
```

```
   1.0000
   1.0000
   1.0000
   ...
   1.0000
   1.0000
   1.0000

octave:13> d = 10*ones([10,1]);
a = zeros([9,1]);
c = 2*ones([9,1]);
e = -1*ones([8,1]);
f = ones([8,1]);
b = [1; 8; -11; -1; -7; 18; -13; 24; 26; 18];
GaussElimPenta(e, a, d, c, f, b)
ans =

   0
   1
  -1
   0
  -1
   2
  -2
   2
   2
   2

octave:20> d = -4*ones([100,1]);
d(1)=d(100)=4;
a = ones([99,1]); c = ones([99,1]);
a(99)=c(1)=-1;
b = 40*ones([100,1]);
b(1)=b(100)=-20;
e = zeros([98,1]); f = zeros([98,1]);
GaussElimPenta(e, a, d, c, f, b)
ans =

   -9.2820
  -17.1281
  -19.2305
  -19.7938
  -19.9448
  -19.9852
  -19.9960
  ...
  -20.0000
```

```
-20.0000
...
-19.9960
-19.9852
-19.9448
-19.7938
-19.2305
-17.1281
 -9.2820
```

The first two cases were designed as a straightforward summation of the coefficients, so answer for both should be a vector of all '1's. The third uses constant values for each diagonal, but those values and the solution vector were generated randomly and then evaluated to find the product vector used as input. The last test is a tridiagonal system used in a previous exercise and, like seen here, the solution tended to -20 in the middle while peaking at either end.

Appendix B. Beam_Deflect.m

```
function Beam_Deflect(n, shap, attach)
    L = 10;         #10 meter length
    if shap==0          #Hollow square
        A = 8.25e-2;        #Outer size 8.25 cm
        a = 4.25e-2;        #Inner size 4.25 cm
        S = A^2-a^2;        #Cross-sectional area
        I = (A^4-a^4)/12;   #Second Moment of Inertia
    elseif shap==1          #Annulus
        R = 4.987e-2;       #Outer size 4.987 cm
        r = 2.992e-2;       #Inner size 2.992 cm
        S = pi*(R^2-r^2);   #Cross-sectional area
        I = pi*(R^4-r^4)/4; #Second Moment of Inertia
    end
    E = 6.9e10;     #Young's modulus - 69GPa
    rho = 2700;     #Mass density in kg/m^3
    g = 9.81;       #Gravitational acceleration
    w = S*rho*g;    #Force of gravity by length
    #n = 10;        #Precision determined by user
    h = L/n;
    N = n + 1;

    #"Output" vector
    f = -h^4/(E*I) * w * ones(N, 1);
    f(1) = f(2) = f(N-1) = f(N) = 0;
    if attach==2
        f(N) = 3*f(N-2)/2;  #Right end free
    end
    % Define the matrix of the system. Notation: d0 is the main diagonal;
    % dpn is superdiagonal n; dmn is subdiagonal n
```

```
d0 = 6*ones(1, N);
dp1 = dm1 = -4*ones(1, N-1);
dp2 = dm2 = ones(1, N-2);
#Left end fixed
d0(1) = 1.0; d0(2) = 4.0;
dp1(1) = 0.0; dp1(2) = -1.0;
dp2(1) = 0.0; dp2(2) = 0.0;
dm1(1) = 0.0;
dm2(1) = 0.0;
if attach==1    #Right end pinned
    d0(N) = 1.0; d0(N-1) = 5.0;
    dp1(N-1) = dm1(N-1) = 0.0;
    dp2(N-2) = dm2(N-2) = 0.0;
elseif attach==2    #Right end free
    d0(N) = 1.0; d0(N-1) = 5.0;
    dp1(N-1) = dm1(N-1) = -2.0;
else
    printf("Invalid attach option\n")
    return
end
#% Fixed right end
#d0(N) = 1.0; d0(N-1) = 4.0;
#dp1(N-1) = 0.0;
#dp2(N-2) = 0.0;
#dm1(N-1) = 0.0; dm1(N-2) = -1.0;
#dm2(N-2) = 0.0; dm2(N-3) = 0.0;

#Apply system solver
y = GaussElimPenta(dm2, dm1, d0, dp1, dp2, f);
#Show data
x = (0:h:L)';   #Transpose to be compatible with y
if attach==1
    y_exact = -S*rho*g/(48*E*I)*x.^2.*(3*L^2 - 5*L*x + 2*x.^2);
elseif attach==2
    y_exact = -S*rho*g/(24*E*I)*x.^2.*(6*L^2 - 4*L*x + x.^2);
end
#y_exact = -S*rho*g/(24*E*I)*x.^2.*(L - x).^2;
err = abs(y - y_exact);
printf("Maximum error: %.4e\n",max(err))
subplot(2,1,1);
plot(x, y, "b*", x, y_exact, "r")
title('Deflection of the beam');
xlabel('x'); ylabel('y (deflection)');
grid on;
subplot(2,1,2);
plot(x, err, "r")
title('Error: |computed - exact|');
xlabel('x'); ylabel('Error');
```

```
    grid on;
    print(sprintf("Beam_%d%d_%d.png",shap,attach,n));
endfunction
```