Numerical Methods for Determining the Deflection of Beams

Corey Zhang
West Chester University of Pennsylvania
December 16, 2022

1 Intro

A part of the theory of the deflection curve is taught in many engineering courses. It is dedicated to the relations between the loads applied to beams and resulting deflections. This paper presents the derivation of the physical meaning of the general linear differential equation of the deflection curve and its general formulation and solution as a boundary value problem. Some algorithms are presented in which deflections are calculated for uniform beams, with continuous distributed load functions on simple supports.

2 Basic Theory

2.1 Differential Equation

The governing equation is derived from the Bernoulli-Euler theorem that the curvature is proportional to the bending moment.

$$\frac{M}{EL} = \frac{1}{r}$$

r = radius of curvature

M = bending moment

E = Young's modulus

M = moment of inertia of the beam cross=section with respect to the neutral axis.

The bending moment for uniform beams with simple supports are

Let $y(x)$ represent the vertical deflection at location x on beam, where $0 \leq x \leq L$, and Θ be the angle that is deflected and s to be arc length. For small dΘ, we have

$$ds = r \cdot d\Theta$$

or the curvature k,

$$k = \frac{1}{r} = \frac{d\Theta}{ds}$$

and if Θ is small,

$$ds = \frac{dx}{cos\Theta} \approx dx$$

Therefore, we get

$$EI\frac{d^2y}{dx^2} = -M$$

The bending moment is

$$M = \frac{q}{2}(Lx - x^2)$$

The moment for a uniformly loaded beam on immovable hinges at both ends is

$$M = \frac{q}{2}(Lx - x^2) - Hy$$

H is the unknown horizontal reaction and depends on the deflection pattern.

After we put the equation together, we get the final equation

$$EI\frac{d^2y}{dx^2} = -\frac{q}{2}(Lx - x^2) + Hy$$

So that

$$\frac{d^2y}{dx^2} = \frac{H}{EI}y - \frac{q}{2EI}(Lx - x^2)$$

[1]

2.2 Numerical Method

The numerical method for ODE approximates derivatives tends to the exact value when the step size is small. It breaks down a continuous equation into separated discrete cells. We will have a linear equation for each cell, then we end up solving a sparse linear algebra problem as $Ax = b$ Since A is a sparse matrix, to solve or approximate x, we need some methods as well.

1. Crout's Method

In linear algebra, Crout's method is an LU decomposition method that decomposes a large matrix into a lower triangular matrix(L), an upper triangular matrix (U). Let $A = LU$,which implies $Ax = (LU)x$. Since matrices multiplication is associative, we can get $Ax = L(Ux)$. Apparently, Ux is also a vector. Let $y = Ux$, we can solve y for $Ly = b$ first. Then solve x for $Ux = y$. Notice L and U are both triangular matices, it is easier to solve by substitutions.

Pros:

- It is a direct method
- It is done by only using$A$ once it's done, it can be used on any vector b

Cons

- It requires forward and backward substitution,
- It requires extra memory space.

2. Jacobi's Method

Jacobi method is an iterative algorithm for determining the solutions. It probably is the simplest iterative method for solving $Ax = b$ by giving a current approximation [2]

$$x^k = (x_1^k, x_2^k, \dots, x_n^k\ )$$

For x, then using the first equation and the current value of $(x_2^k, x_3^k, \dots, x_n^k\ )$ to find a new value x_1^{k+1}, reapeat the process to find new values for all x_i^{k+1}, we can get

$$x^{k+1} = (x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}\ )$$

This system can be written as

$$Dx^{k+1} + \ \ L + U\ \ x^k = b$$

Where D, L, U are diagona strict lower triangular and strict upper triangular parts of A, respectively, so that

$$x^{k+1} = D^{-1}[\ -L - U\ \ x^k + b]$$

Pros:

- Easy to understand
- Converging at an exponential rate
- Easy to be used in parallel computing*

Cons:

- Low numerical stability
- Inexact solution

*Parallel Computing

Parallel computing is the simultaneous usage of multiple computing resources to solve a computational problem. A problem is broken into a discrete series of instructions, they are distributed onto multiple processors to execute simultaneously to boost computing time.

3 Experiments

Recall the equation

$$\frac{d^2y}{dx^2} = \frac{H}{EI}y - \frac{q}{2EI}(Lx - x^2)$$

is to solve the deflection of a beam.

Giving initial value $L = 120\ in.$, $H = 1000\ lb$, $E = 3.0 \times 10^7 \frac{lb}{in.^2}$, $I = 625\ in.^4$ and $y\ \ 0\ \ = 0, y\ \ L\ \ = 0$.

We will use both Crout and Jacobi methods to estimate the deflection on each point along the beam. We are going to choose the step size 1 and set the tolerance of error of Jacobi to be $10^{-4}, 10^{-6}, 10^{-8}$, respectively. We can also compare the approximation with exact solution. Which is,

$$y(x) = -\frac{1}{20\left(1 + e^{(2\sqrt{3})/125}\right)}\Big(x^2 + e^{(2\sqrt{3})/125}\left(x^2 - 120\,x + 37500000\right) - 120\,x - 37500000\,e^{-(x-120)/(2500\sqrt{3})} - 37500000\,e^{x/(2500\sqrt{3})} + 37500000\Big)$$

There are the results we got,

tolerance $= 10^{-4}$

x	y(crout)	y(Jacobi)	y(exact)
20	0.007298	0.000001	0.007288
40	0.012343	0.000003	0.012514
60	0.014419	0.000005	0.014399
80	0.012532	0.000003	0.012514
100	0.007298	0.000001	0.007288
Running time	0.005 ms	0.027 ms	

tolerance $= 10^{-6}$

x	y(crout)	y(Jacobi)	y(exact)
20	0.007298	0.005253	0.007288
40	0.012343	0.009118	0.012514
60	0.014419	0.010328	0.014399
80	0.012532	0.009118	0.012514
100	0.007298	0.005253	0.007288
Running time	0.005 ms	0.055 ms	

tolerance = 10^{-8}

x	y(crout)	y(Jacobi)	y(exact)
20	0.007298	0.007278	0.007288
40	0.012343	0.012496	0.012514
60	0.014419	0.014378	0.014399
80	0.012532	0.012496	0.012514
100	0.007298	0.007278	0.007288
Running time	0.005 ms	0.1 ms	

Clearly the deflection gets the most value in the center of beam, which is very intuitive.

4 Conclusion

Apparently, the beam dented most in the center, which is not a supervise for me. We can see the Jacobi's method isn't quite accurate when the numbers we are trying to manipulate is quite small. When I set the tolerance to .0001, the result is obviously not right. However, the good sign is that result is not divergent. For coding purpose, the Jacobi Method is a balance of simplicity and powerful. The time efficiency for Jacobi method is bad for this specific model. But nevertheless, both methods get very good approximation.

5 Bibliography

[1] »Deflection of Beam,« University of Southern California, [Online]. Available: https://www-scf.usc.edu/~subhayad/deflection1.pdf.

[2] J. D. F. Richard L. Burden, Numerical Analysis, Boston: Brooks/Cole, 2010.