**EECS 639 Intro. to Scientific Computing Project**

Pierce Lane and Harlan Williams

**Credit:**

Forward Euler's Method: Pierce Lane and Harlan Williams

Backward Euler's Method: Harlan Williams

Trapezoidal Method: Pierce Lane

RK4: Harlan Williams

Predictor-Corrector Method: Harlan Williams

Part 2 Testing: Pierce Lane

Part 3 Chemistry IVP: Pierce Lane

Part 3 Animation: Harlan Williams

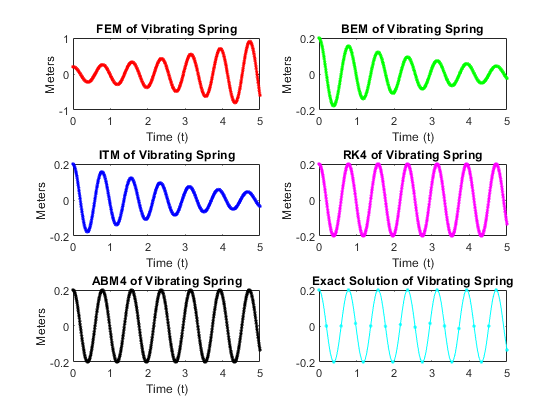
Graphs: Pierce Lane

Report: Pierce Lane and Harlan Williams

**Part B: Testing**

**Vibrating Spring:**

Here, we tested the vibrating spring problem from time *t = 0* to *t = 5*, using step size *h = 0.01*. We also plotted the exact solution given in the project description using symbolic notation in MATLAB for comparison.



**FEM** elapsed time is 0.002103 seconds.

**BEM** elapsed time is 0.004442 seconds.

**ITM** elapsed time is 0.005104 seconds.

**RK4** elapsed time is 0.003223 seconds.

**PC** elapsed time is 0.005076 seconds.

**Discussion of Methods:**

The Forward Euler Method (FEM) had the worst stability and accuracy of any of the solvers. That being said, each iteration was computationally cheap, being the fastest of all methods. It ballooned the displacement of the spring to nearly a meter, when the maximum value it should be able to achieve is 0.2 meters.

Compared to FEM, the Backward Euler Method (BEM) was better in terms of stability and accuracy, and on par with FEM in terms of efficiency. That being said, with a step size of *h = 0.01,* it was still unstable and introduced dampening into the system which caused the amplitude to decay to 0. The accuracy of BEM was greater than FEM, but still the second worst of all methods shown. On subsequent runs, the variance in timing for BEM was large, giving times ranging from the given 0.002351 seconds to 0.004010 seconds.

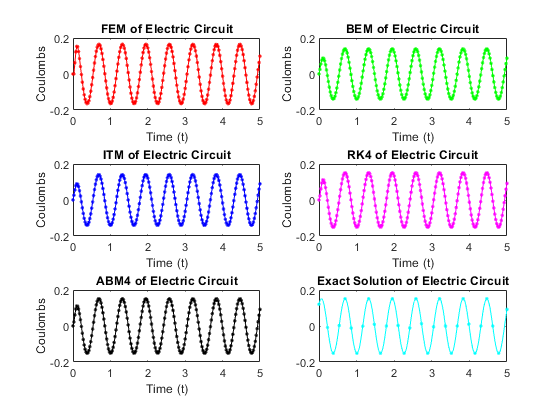
The Implicit Trapezoidal (ITM) method seems to be about as stable and accurate as BEM, while taking about 1.6 times longer to complete. This result is surprising because BEM is first-order accurate and ITM is second-order accurate, meaning that ITM should have a better accuracy than BEM. In this case, however, ITM is almost exactly as accurate as BEM.

The Fourth-Order Runge-Kutta (RK4) is the most stable and accurate solution so far. It is almost exactly the solution to the ODE, perfectly stable, and is the second fastest method, nearly tying with BEM.

The Adams-Bashforth predictor and Adams-Moulton corrector method (ABM4) was on par with RK4 in terms of stability and accuracy, but was notably slower by about 1.6 times.

**Electric Circuit:**

Here, we tested the electric circuit problem from time *t = 0* to *t = 5*, using step size *h = 0.025*. We also plotted the exact solution given in the project description using symbolic notation in MATLAB for comparison. Strangely, the exact solution doesn't have the same initial value at *t = 0*. The analysis ignores this and assumes that they are the same.

****

**FEM** elapsed time is 0.002086 seconds.

**BEM** elapsed time is 0.003951 seconds.

**ITM** elapsed time is 0.005460 seconds.

**RK4** elapsed time is 0.003168 seconds.

**PC** elapsed time is 0.004645 seconds.

**Discussion of Methods:**

FEM is the least accurate solution here, but is notably stable. Compared to the vibrating spring test case, even with a larger step size of *h = 0.025,* FEM doesn't run away from the exact solution. That being said, it's still one of the least accurate methods on display here. Also worth noting is its fast runtime, being the fastest method shown.

BEM is much more accurate than FEM, similar to the vibrating spring problem. However, this time, BEM is not only stable, but very accurate to the exact solution. Comparing it to the most accurate methods, RK4 and PC, it produces nearly the same solution curve in about the same amount of time as RK4.

ITM is, as in the vibrating spring problem, mostly the same as BEM. It took about 1.8 times longer to run, however, making it less favorable in this test case.

RK4 is, as in the vibrating spring problem, accurate and stable, even with the increased step size. It also has about the same runtime as BEM here, making the methods almost indistinguishable.

ABM4 is similar to RK4 and BEM in that they are stable and accurate solutions. However, ABM4 takes approximately 0.0015 seconds longer to run, making it less favorable than RK4 or BEM for this test case.

**Part C:** **Chemistry Application**

Here, we used the methods tested above to model the behavior of the famous Belousov-Zhabotinsky (BZ) reaction.

The BZ reaction can be described by the following system of first-order ODEs:

where *k0-5* are known rate constants, *A* and *B* are the concentrations of bromate and organic species, assumed to be constant, and *X, Y,* and *Z* are the concentrations of bromous acid, bromide, and cerium IV. *f* is an adjustable stoichiometric factor, set to be a constant 1 here. *A* is set to 0.06 M and *B* is set to 0.02 M.

However, the output of this system is very unstable, so we decided to use a scaled version that normalizes the concentrations *X, Y,* and *Z* to make the output easier to analyze. That ODE is the following:

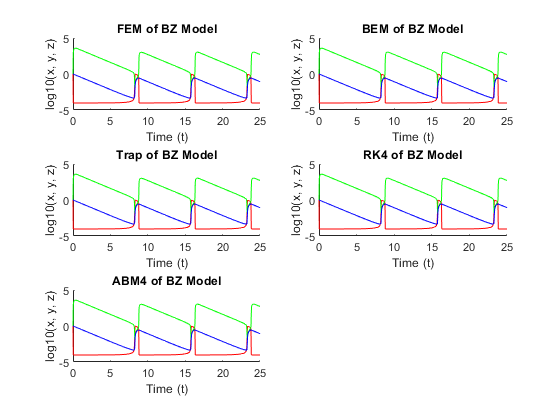
where ***x*** *= 2k4X/(k5A),* ***y*** *= k2Y/(k5A),* ***z*** *= k0k4BZ/(k5A)2,* ***τ*** *= k0Bt*

***ϵ*** *= k0B/(k5A),* ***ϵ′*** *= 2k0k4B/(k2k5A),* ***q*** *= 2k3k4/(k2k5)*

Notice *ϵ*, *ϵ'*, and *q* are all constants.

In practice, we can consider *x, y,* and *z* to be our output variables and *τ* to be our input variable. This means that we plot *x, y, z,* and *τ,* instead of plotting *X, Y, Z,* and *t*. Finally, plotting the log10 of all of the outputs makes for an easier to read graph, so that is done here.

Thus, we get to the output of these scaled equations when solved with each method. The initial values for *A* and *B* are 0.06 M and 0.02 M respectively, with starting *x, y,* and *z* of 1, 1, and 1 respectively. The time *t* goes from 0 to 25, with a step size of a tiny *h = 0.0000025.* This step size was experimentally found to be the largest step size while maintaining the accuracy of all methods. Without a time step this small, most of the methods immediately become unstable and diverge. Red represents *X,* blue represents *Y,* and green represents *Z*.



**FEM** elapsed time is 1.613260 seconds.

**BEM** elapsed time is 5.838480 seconds.

**ITM** elapsed time is 11.371833 seconds.

**RK4** elapsed time is 4.283857 seconds.

**ABM4** elapsed time is 11.967800 seconds.

**Discussion:**

The initial conditions (*A = 0.06 M, B = 0.02 M*, *x = y = z = 1*) resulted in oscillation. This is because after the system stabilized from its initial conditions, after a large amount of steps, around 8 seconds in simulation time, the system would very nearly return to a state it was already in, thus resulting in a period of oscillation.

FEM produced a surprisingly accurate and stable curve quickly. Even when the step size is doubled, FEM maintains stability. It's the fastest method by far, surpassing the second fastest by about 2.6 seconds. This comes as a surprise; in theory, FEM should have the smallest stability window compared to the other methods on display.

BEM is one of the methods that breaks down when step size is increased. This is likely due to the nearly vertical changes that *x, y,* and *z* experience every cycle breaking down the fixed point iteration being used to predict the next value. Otherwise, this is an accurate method that runs relatively quickly.

ITM also breaks down at higher step sizes, likely due to the same reasons as BEM. It also runs the second slowest with similar accuracy to every other method, making it less favorable than FEM, BEM, or RK4.

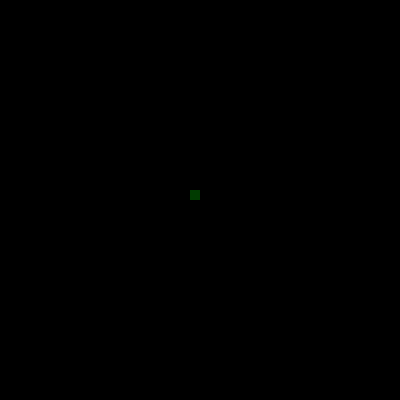
RK4 runs relatively quickly and, similar to FEM, maintains some stability at higher step sizes. RK4 runs about as fast as BEM but has significantly increased stability in this application, making it the second favorite method behind FEM. It's hard to say whether FEM or RK4 is more accurate to the exact solution because there is no exact solution to this system, but given the large time saved using FEM, it should be the preferred method.

ABM4 also breaks down quickly as step size is increased. Notably, it also takes nearly 12 seconds to run, being the slowest method displayed.

Seeing as the methods with the most instability were the implicit methods (BEM, ITM, and ABM4), there is a surprising conclusion to draw here: for some systems of ODEs, explicit methods are more stable than implicit methods. This contradicts the notion that implicit methods are more stable than explicit ones. Otherwise, we've learned that the BZ reaction is highly volatile to initial conditions and requires careful setup to get a parseable result.

**Animation:**

We created an animation of the Oregonator system by initializing an array of 40 by 40 pixels that each held information about the concentrations of the X, Y, and Z species. The pixel at (20, 20) was given the initial condition X = 0 M, Y = 0.26 M, and Z = 0 M, and then the system was simulated using the FEM method with a step size of . At every 10,000 steps, to simulate diffusion, a convolution was applied using the conv2 MATLAB function which spread out the concentrations of each species to neighboring cells with a 3 by 3 convolution kernel. The animation fails to model circular dispersion because the kernel does not accurately take into account Euclidean distance in the grid, however it still displays expected behavior with X initially dominating, and then Y, and then X rebounds momentarily. In the animation, the concentration of X corresponds to red, Y to blue, and Z to green, with intensity of color proportional to the concentration in a pixel.



**Sources:**

<http://www.scholarpedia.org/article/Oregonator>

<https://people.sc.fsu.edu/~jburkardt/m_src/oregonator_ode/oregonator_ode.html>