Math 5601: Introduction to Numerical Analysis
Midterm project: Implicit numerical methods for the first order nonlinear ODEs

Dr. Xiaoming He*

Complete this project INDEPENDENTLY. Show all relevant work in detail to justify your conclusions. Partial credit depends upon the work you show. For each numerical experiment, all the .m files of your Matlab code should be electronically submitted to hex@mst.edu together with a .txt file which copies all the information in the Matlab command window when you run the code to obtain the numerical results.

Consider the initial value problem (IVP) for the first order ordinary differential equation (ODE):

$$y'(t) = f(t, y(t)),\ a \le t \le b,$$
$$y(a) = g_a,$$

where g_a is the initial value and $f(t, y)$ is a nonlinear function of y. The existence and uniqueness of the solution is guaranteed by the following theorem.

Theorem: Let $f(t, y)$ be a continuous function defined on

$$S = \{(t, y) : t, y \in \mathbb{R}, |t - a| < \gamma, |y| < \infty\}.$$

If f satisfies a Lipschitz condition in the y variable over S. Then $y'(t) = f(t, y(t))$ with $y(a) = g_a$ has a unique solution for $|t - a| < \gamma$.

In this project, we will study the implicit numerical methods for the above IVP.

1 Algorithm development and theory

Assume that we have a uniform partition of $[a, b]$ into J elements with mesh size h. Let $t_j = a + jh,\ j = 0, 1, \cdots, J$ denote the mesh nodes and y_j denote the numerical solution of $y(t_j)$. Then the initial condition implies: $y_0 = y(a) = g_a$.

1.1 Derivation of finite difference schemes

A straightforward discretization of $f(t, y(t))$ at t_j is $f(t_j, y_j)$. As for the discretization of $y'(t)$ at t_j, by using the following Taylor's expansions

$$\begin{aligned} y(x + h) &= y(x) + y'(x)h + \frac{1}{2}y''(x)h^2 + \frac{1}{6}y'''(x)h^3 + O\left(h^4\right), \\ y(x - h) &= y(x) - y'(x)h + \frac{1}{2}y''(x)h^2 - \frac{1}{6}y'''(x)h^3 + O\left(h^4\right), \end{aligned}$$

*Department of Mathematics and Statistics, Missouri University of Science and Technology, Rolla, MO 65409, hex@mst.edu

we obtain the following conclusions in the lecture slides of Chapter 6:

$$\begin{aligned} y'(t_j) &= \frac{y(t_{j+1}) - y(t_j)}{h} + O(h), \\ y'(t_j) &= \frac{y(t_j) - y(t_{j-1})}{h} + O(h), \\ y'(t_j) &= \frac{y(t_{j+1}) - y(t_{j-1})}{2h} + O\left(h^2\right). \end{aligned}$$

Hence we obtain the following difference schemes:

$$\begin{aligned} &\text{forward difference for } y'(t_j) \approx \frac{y_{j+1} - y_j}{h}, \\ &\text{backward difference for } y'(t_j) \approx \frac{y_j - y_{j-1}}{h}, \\ &\text{centered difference for } y'(t_j) \approx \frac{y_{j+1} - y_{j-1}}{2h}. \end{aligned}$$

By applying the forward difference, we derive the forward Euler method for the IVP as follows:

$$\begin{aligned} & y'(t) = f(t, y(t)) \\ \Rightarrow \quad & y'(t_j) = f(t_j, y(t_j)),\ j = 0, \cdots, J-1 \\ \Rightarrow \quad & \frac{y(t_{j+1}) - y(t_j)}{h} + O(h) = f(t_j, y(t_j)),\ j = 0, \cdots, J-1 \\ \Rightarrow \quad & \frac{y_{j+1} - y_j}{h} = f(t_j, y_j),\ j = 0, \cdots, J-1 \\ \Rightarrow \quad & y_{j+1} = y_j + h \cdot f(t_j, y_j),\ j = 0, \cdots, J-1, \\ & y_0 = y(a) = g_a. \end{aligned}$$

This is an explicit method since y_{j+1} can be explicitly expressed in term of the numerical results from the previous step. It is well known that the explicit methods for the IVP are usually easy to be implemented but conditionally stable. On the other hand, implicit methods are usually unconditionally stable even though they are more complicated in implementation. In this project, we will discuss several popular implicit methods. The most fundamental one is the backward Euler method:

$$\begin{aligned} & y'(t) = f(t, y(t)) \\ \Rightarrow \quad & y'(t_j) = f(t_j, y(t_j)),\ j = 1, \cdots, J \\ \Rightarrow \quad & \frac{y(t_j) - y(t_{j-1})}{h} + O(h) = f(t_j, y(t_j)),\ j = 1, \cdots, J \\ \Rightarrow \quad & \frac{y_j - y_{j-1}}{h} = f(t_j, y_j),\ j = 1, \cdots, J \\ \Rightarrow \quad & \frac{y_{j+1} - y_j}{h} = f(t_{j+1}, y_{j+1}),\ j = 0, \cdots, J-1 \\ \Rightarrow \quad & y_{j+1} = y_j + h \cdot f(t_{j+1}, y_{j+1}),\ j = 0, \cdots, J-1, \\ & y_0 = y(a) = g_a. \end{aligned}$$

From the dropped remainders in the above derivation, it is expected that both of the forward and backward Euler methods have first order accuracy (The rigorous proof we did in class is more complicated than this). There are many other implicit methods with higher accuracy order, which can be derived from the Taylor expansion in a similar way. In the following three problems, we will discuss three popular implicit schemes.

Problem #1 (10 points): By using Taylor expansion, derive the trapezoidal rule (Crank-Nicolson scheme if it's applied to PDEs) and find its accuracy orders by using the accuracy order of the dropped remainder in your derivation:

$$\begin{aligned} & \frac{y_{j+1} - y_j}{h} = \frac{f(t_{j+1}, y_{j+1}) + f(t_j, y_j)}{2},\ j = 0, \cdots, J-1 \\ \Rightarrow \quad & y_{j+1} = y_j + h \cdot \frac{f(t_{j+1}, y_{j+1}) + f(t_j, y_j)}{2},\ j = 0, \cdots, J-1. \end{aligned}$$

Problem #2 (10 points): By using Taylor expansion, derive the following two-step backward differentiation which has second order accuracy:

$$\frac{3y_{j+1} - 4y_j + y_{j-1}}{2h} = f(t_{j+1}, y_{j+1}),\ j = 1, \cdots, J-1$$
$$\Rightarrow \quad y_{j+1} = \frac{4}{3}y_j - \frac{1}{3}y_{j-1} + \frac{2h}{3}f(t_{j+1}, y_{j+1}),\ j = 1, \cdots, J-1.$$

Problem #3 (10 points): Find the coefficients α_i $(i = 1, 2, 3, 4)$ in the three-step backward differentiation

$$\frac{\alpha_1 y_{j+1} + \alpha_2 y_j + \alpha_3 y_{j-1} + \alpha_4 y_{j-2}}{h} = f(t_{j+1}, y_{j+1})\ (j = 2, \cdots, J-1)$$

which has third order accuracy. Show your derivation in details.

1.2 Numerical methods for nonlinear equations

It can be seen that all of the implicit schemes has a nonlinear equation to solve at each iteration step if the ODE is nonlinear, i.e., $f(t, y(t)$ is a nonlinear function of $y(t)$. Then we need to apply a proper numerical method we learned from Chapter 2 to solve this nonlinear equation. In the second part of this project, we will consider and compare the bisection method, fixed point method, Newton's method and secant method when we apply them to solve the nonlinear equation numerically. In this section, we will discuss some issues on the theory and algorithm development for the fixed point method and the Newton's method.

The fixed point method for solving $x = g(x)$ is defined as $x_{k+1} = g(x_k)$ $(k = 0, 1, 2, \cdots)$. It's very important since it provides a very general idea which covers many famous methods. Particularly, the Newton's method, which is the most popular method for nonlinear equations, is a special case of the fixed point method. In the lecture slides and homework of Chapter 2, we have discussed and practiced on the following fundamental theory for the convergence of the fixed point method.

Contraction Mapping Theorem in one variable: Suppose that g maps G into itself (i.e., if $x \in G$, then $g(x) \in G$) and g satisfies a Lipschitz condition with with $0 \le L < 1$(i.e., g is a contraction on G). Then there exists a unique fixed point $z \in G$ (i.e., $z = g(z)$), and the sequence $\{x_k\}$ determined by $x_0 \in G$ and $x_{k+1} = g(x_k)$ $(k = 0, 1, 2, \cdots)$ converges to z.

In the lecture slides of Chapter 2, we also show that the fixed point method is at least linearly convergent (first order convergence). It is possible to obtain a higher convergence order for some special choices of $g(x)$. Specifically, choose $g(x) = x - \frac{f(x)}{f'(x)}$ for the fixed point method. Then we obtain the Newton's method method for solving $f(x) = 0$ (or finding the root of $f(x)$):

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\ (k = 0, 1, \cdots).$$

In the lecture slides of Chapter 2, we show that the Newton's method is quadratically convergent (second order convergence) for a root of multiplicity 1 of $f(x)$ as follows.

Recall the following theorem from the lecture slides of Chapter 2.

Theorem: Assume that the iterations $x_{k+1} = g(x_k)$ converge to a fixed point z and $g \in C^q(G)$ where G contains z. Furthermore, assume that q is the first positive integer for which $g^{(q)}(z) \neq 0$ and if $q = 1$ then $g'(z) < 1$. Then the sequence $\{x_k\}$ converges to z with order q.

Suppose $f(z) = 0$ and $f'(z) \neq 0$. Then

$$g(x) = x - \frac{f(x)}{f'(x)}$$
$$\Rightarrow \quad g'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = 1 - \left\{1 - \frac{f(x)f''(x)}{[f'(x)]^2}\right\} = \frac{f(x)f''(x)}{[f'(x)]^2}$$
$$\Rightarrow \quad g'(z) = \frac{f(z)f''(z)}{[f'(z)]^2} = 0.$$

Hence the Newton's method is quadratically convergent.

In the above derivation, we assume $f(z) = 0$ and $f'(z) \neq 0$. That is, z is a root of multiplicity 1 of $f(x)$. Then a nature question shows up: what if z is a root of multiplicity $m \geq 2$ of $f(x)$? The following two problems will lead you to the answer.

Problem #4 (10 points): What's the convergence order of the Newton's method for a root of multiplicity $m \geq 2$ of $f(x)$? Justify your conclusion by using the above theory of the convergence order. (Hint: Assume $f(x) = (x-z)^m r(x)$.)

Problem #5 (10 points): Compared with second order convergence of the Newton's method for a root of multiplicity 1, what problem do you observe from your conclusion in problem #4? How can we resolve this problem? Propose your idea and justify it. (Hint: Modify the Newton's method by using m. The results of problem #4 and $g'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = 1 - \left\{1 - \frac{f(x)f''(x)}{[f'(x)]^2}\right\}$ provide the key information about how to do so.)

2 Computation

Now we turn to the computational issues of the backward Euler method

$$y_0 = y(a) = g_a, \quad y_{j+1} = y_j + h \cdot f(t_{j+1}, y_{j+1}),\ j = 0, \cdots, J-1.$$

At each iteration step, the goal is to obtain y_{j+1} based on the y_j from the previous step. The iteration is straightforward. But at the j-th iteration step, we need to solve a nonlinear equation since $f(t, y)$ is nonlinear for y. In our lecture of Chapter 2, we have discussed different iterative methods for solving nonlinear equations, including the bisection method, fixed point method, Newton's method and secant method. Therefore, theoretically we just need to assemble the iteration of the numerical methods for nonlinear equations into the iteration of the backward Euler method. However, all the needed information for those methods are directly given in the lecture. This kind of ideal situation rarely happens in practice. In the following we first discuss how to obtain the needed information and assemble these nonlinear equation solvers into the iteration of the backward Euler method. The techniques are straightforward here since this is a simple problem. For a more complicated problem or a more realistic situation, you will need more advanced techniques and more complicated ideas. But hopefully this procedure can give you a first look at how to apply a method you learn in a practical situation.

Problem #6 (10 points): The bisection method, Newton's method and secant method are developed to solve the equation $\tilde{f}(x) = 0$. The fixed point method is developed to solve the equation $\tilde{g}(x) = x$. Given the y_j, t_{j+1}, and h at the j-th step of the backward Euler method, what are the functions $\tilde{f}(x)$ and $\tilde{g}(x)$ for the equation

$$y_{j+1} = y_j + h \cdot f(t_{j+1}, y_{j+1})?$$

At each step of the backward Euler method, we know the step size h, the time t_{j+1} for the current step and y_j from the last step, from which we need to figure out all the information needed by the numerical methods for nonlinear equations. The tolerance and the maximum number of iteration steps can be chosen based on the accuracy requirement, h, $f(t, y)$, and our experience. But it is tricky to choose a proper initial point to start the iteration of numerical methods for nonlinear equations. For the fixed point method and the Newton's method, one easy way is to choose the initial point x_0 to be y_j in order to start their iterations. This often gives a fast convergence because y_{j+1} is usually close to y_j when the mesh size h is small.

Problem #7 (10 points): Consider the bisection method and the secant method to solve the nonlinear equation $x = y_j + h \cdot f(t_{j+1}, x)$ for x.

(a) Among the information you need to prepare before you can start the iteration of the bisection method, which one is difficult to obtain?

(b) Among the information you need to prepare before you can start the iteration of the secant method, which one is difficult to obtain?

Now we can complete the code of the backward Euler method based on the guided coding in class and the above preparation. The following problem is set up to obtain the numerical solutions and verify some theoretical conclusions.

Problem #8 (20 points): Use backward Euler method to solve the following IVP:

$$y'(t) = e^{2t}y^2,\ 0 \le t \le 1,$$
$$y(0) = 0.1.$$

(a) Program for the backward Euler method with the fixed point method and the Newton's method for the nonlinear equations. Use the functions $\tilde{f}(x)$ and $\tilde{g}(x)$ you find in problem #6. Set all the tolerances to be 10^{-6}, maximum number of iteration steps to be 100 and the initial point x_0 to be y_j for the iteration of both the fixed point method and the Newton's method.

(b) Use your code to solve the equation with $h = 1/4, 1/8, 1/16, 1/32, 1/64, 1/128$. What are the numerical solutions at the end time $t = 1$ from the backward Euler method with the fixed point method and the Newton's method for the nonlinear equations? List your results into the following table.

h	solutions with fixed point method	solutions with Newton's method
1/4		
1/8		
1/16		
1/32		
1/64		
1/128		

Table 1: The numerical solutions at the end time $t = 1$ from the backward Euler method with the fixed point method and the Newton's method for the nonlinear equations.

(c) Solve this IVP by hand to obtain the analytic solution. Compute the difference (error) between the numerical solutions and the value of the analytic solution at the end time $t = 1$. List your results into the following table. What do you observe? Explain your observation in term of the accuracy order of backward Euler method.

h	errors with fixed point method	errors with Newton's method
1/4		
1/8		
1/16		
1/32		
1/64		
1/128		

Table 2: The numerical errors at the end time $t = 1$ from the backward Euler method with the fixed point method and the Newton's method for the nonlinear equations.

(d) Consider the nonlinear equation in the last iteration step of the backward Euler method to obtain the numerical solution at the end time $t = 1$. When $h = 1/4$ and $h = 1/8$, what are the numbers of the iteration steps of the fixed point method and Newton's method for solving this nonlinear equation? Explain your observations in term of the convergence orders of the fixed point method and Newton's method.

(e) Consider the nonlinear equation in the last iteration step of the backward Euler method to obtain the numerical solution at the end time $t = 1$. For $h = 1/4, 1/8, 1/16, 1/32, 1/64, 1/128$, output the numbers of the iteration steps of the fixed point method and Newton's method. What do you observe? Explain the observation in term of our choice of the initial point x_0 for the fixed point method and Newton's method.

Now let's try some small variations of the above problem to see what could happen.

Problem #9 (10 points): Slightly modify the code of problem #8 to solve the following problem with backward Euler method to solve the following IVP:

(a)

$$y'(t) = e^{2t}y^2,\ 0 \le t \le 1,$$
$$y(0) = 0.2.$$

(b)

$$y'(t) = e^{4t}y^2,\ 0 \le t \le 1,$$
$$y(0) = 0.1.$$

What do you observe?

In the above numerical experiments, we set up different tests to numerically investigate the influence of the convergence order, the mesh size h, the initial condition, and the coefficient in the equation. You are encouraged to set up different cases to investigate the influence of other factors, such as the right hand side function $f(t, y)$, the problem domain $[a, b]$, the tolerance, the maximum number of iteration steps, and so on. From these numerical experiments, we can see that the convergence and accuracy of the numerical methods are affected by many different factors. Sometimes the convergence and accuracy are rather sensitive to some factors even for a simple problem like #8 and #9. Therefore, we DO need to pay attention to the influence of these factors when we apply numerical methods to solve real world problems.

3 Conclusions and future works

In general implicit methods are more popular than the explicit methods due to their unconditional stability. But the additional cost for solving the nonlinear equations is a key issue. For a nonlinear time-dependent partial differential equation, a system of nonlinear ODEs will be obtained from the finite element method, finite difference method or finite volume method. Then the cost for solving the system will increase dramatically, even to a formidable level. Therefore, you should learn more efficient numerical methods for solving nonlinear equations/systems in the literature. For example, in order to learn the Newton's method for system of nonlinear equations, Frechet derivative is a fundamental definition you need to learn first.

As you can see in the lecture, homework, and problems #1, #2, and #3 in this project, Taylor's expansion is a very convenient tool to derive different finite difference schemes for different purposes. Another popular tool is Lagrange polynomial approximation, which we also discussed in our lecture. Even though the schemes are the core of the development of the numerical methods for differential equations, there are many other important related issues in both practice and theory, such as the improvement of the existing methods discussed in problems #4 and #5, numerical experiments discussed in problems #6 and many others. You need to understand all components one by one and properly assemble them into a big system in order to develop efficient and robust numerical methods for differential equations. In your practical work, you may not need to go through the whole development procedure since some existing components are good enough for your work. But you still need to understand the whole system first before you start to do research on part of it. It is a very important capability in practice to identify the key components for your research work and resolve the corresponding problems in a way compatible with the rest of the system.

Once a numerical method is developed, numerical experiments are usually a necessary step to test if the method works well or not. The observations from different tests can provide critical information to justify or improve the numerical method. Of course, how to set up the numerical tests for different testing purposes to provide more useful and accurate information is an important practical issue, which you should gradually learn in your work.

On the other hand, the theory is also important to analyze how good or how bad the numerical methods are. Many theoretical conclusions, such as the convergence rate, stability, computational complexity and the theoretical requirements of different methods, are critical for us to choose a proper method for a specific problem. Moreover, as

you can see in this project, the theory is critical in the derivation (problems #1, #2, and #3 and improvement of the numerical methods (problems #4 and #5). Therefore, the conclusion is that both engineers and mathematicians need the theory throughout the whole procedure of algorithm development.

Even though a numerical method works well for your numerical tests and has theoretical guarantees, it is still possible for this method to fail in a realistic problem due to different reasons. For example, as you can see in my chemical engineering project I showed you in class, the real-world problems are often ill-conditioned and nonlinear, which will cause "weird" nonlinear systems. It is very difficult for the general numerical methods, such as the traditional Newton's method, to solve these systems. Hence it is important to learn different kinds of variations of the general methods or propose your own idea on the algorithm improvement for different specific purposes. The modified Newton's method in problem #5 is just one "old" variation of the traditional Newton's method. But what you did in problem #4 and #5 may help you better understand the basic procedure in practice: try the traditional methods first; if they fail or have some problems, find the reason; then find proper variations of the traditional methods or propose your own idea to modify the traditional methods correspondingly.

Hopefully, the fundamental knowledge/framwork and the learning technique you obtain from this introduction class of numerical analysis can help you with your future self-study for the numerical methods you need in your practical work.