

AMATH 586 Homework 3

Cade Ballew #2120804

May 6, 2022

1 Problem 1

Consider a consistent and zero-stable LMM

$$\sum_{j=0}^r \alpha_j U^{n+j} = k \sum_{j=0}^r \beta_j f(U^{n+j}).$$

and define the characteristic polynomial $\pi(\xi; z) = \rho(\xi) - z\sigma(\xi)$. From (5.48), we have that a LMM is consistent iff

$$\sum_{j=0}^r \alpha_j = 0, \quad \sum_{j=0}^r j\alpha_j = \sum_{j=0}^r \beta_j.$$

We can then see from the definition that

$$\pi(1; 0) = \rho(1) = \sum_{j=0}^r \alpha_j = 0$$

if the LMM is consistent.

If our method is both consistent and zero-stable, then we know from consistency that $\rho(1) = 0$. However, zero-stability tells us that any roots of ρ with modulus 1 must have multiplicity 1. Thus, since ρ is a polynomial, we must be able to write

$$\rho(\xi) = (\xi - 1)g(\xi)$$

where g is a polynomial such that $g(1) \neq 0$. Then,

$$\rho'(\xi) = g(\xi) + (\xi - 1)g'(\xi),$$

so

$$\rho'(1) = g(1) \neq 0.$$

Now, we assume zero-stability and consistency and suppose $\xi = 1 + \eta(z)$ for z near zero so that $\pi(\xi; z) = \pi(1 + \eta(z); z) = 0$. Then, taking the derivative with respect to z of both sides of this equation,

$$\frac{\partial \pi}{\partial z} = \rho'(1 + \eta(z))\eta'(z) - \sigma(1 + \eta(z)) - z\sigma'(1 + \eta(z))\eta'(z) = 0.$$

This lets us find that

$$\eta'(z) = \frac{\sigma(1 + \eta(z))}{\rho'(1 + \eta(z)) - z\sigma'(1 + \eta(z))}.$$

Taking $\eta(0) = 0$,

$$\eta'(0) = \frac{\sigma(1)}{\rho'(1)}.$$

Now, note that

$$\rho'(\xi) = \sum_{j=1}^r j\alpha_j \xi^{j-1},$$

so consistency gives that

$$\rho'(1) = \sum_{j=1}^r j\alpha_j = \sum_{j=0}^r \beta_j = \sigma(1).$$

Thus, we find that $\eta'(0) = 1$. We can see this implies that there must be an interval $(0, \epsilon]$ for some small $\epsilon > 0$ that does not lie in the absolute stability region S by letting $\epsilon > 0$ be small. Then, the mean value theorem gives that

$$\eta(\epsilon) - \eta(0) = \eta'(\zeta)(\epsilon - 0) = \eta'(\zeta)\epsilon$$

for some $\zeta \in (0, \epsilon)$. If ϵ is sufficiently small, then continuity gives that $\eta'(\zeta) \approx \eta'(0) = 1$ and $\eta'(\zeta) > 0$. Then,

$$\xi = 1 + \eta(\epsilon) = 1 + \eta'(\zeta)\epsilon > 1$$

meaning that the root condition is violated and ϵ (or any nonnegative values less than it) cannot fall in the absolute stability region. We could also observe this by Taylor expanding η as

$$\eta(z) = \eta(0) + \eta'(0)z + o(z) = z + o(z)$$

where $z > 0$ is sufficiently close to 0 and using the reverse triangle inequality to write

$$|\xi| = |1 + z + o(z)| \geq |1 + z| - |o(z)| = 1 + z - |o(z)| > 1$$

as $z \gg o(z)$.

2 Problem 2

Consider the system of ODEs

$$\begin{bmatrix} u'_1(t) \\ u'_2(t) \\ u'_3(t) \\ u'_4(t) \end{bmatrix} = \begin{bmatrix} u_4(t) - \mu u_3(t) + \lambda g(u_1(t) + u_2(t)) \\ \mu u_3(t) - u_4(t) + \lambda g(u_1(t) + u_2(t)) \\ -\sigma u_4(t) \\ \sigma u_3(t) \end{bmatrix}, \quad g(u) = u(1 - u)^2, \quad \lambda > 0, \quad \sigma, \mu \in \mathbb{R}.$$

To solve for $u_3(t), u_4(t)$ in terms of $u_3(0) = \eta_3, u_4(0) = \eta_4$, we note that

$$\begin{pmatrix} u'_3(t) \\ u'_4(t) \end{pmatrix} = \begin{pmatrix} -\sigma u_4(t) \\ \sigma u_3(t) \end{pmatrix} = \begin{pmatrix} 0 & -\sigma \\ \sigma & 0 \end{pmatrix} \begin{pmatrix} u_3(t) \\ u_4(t) \end{pmatrix}$$

and define

$$A = \begin{pmatrix} 0 & -\sigma \\ \sigma & 0 \end{pmatrix}.$$

We compute the matrix exponential by diagonalizing

$$A = \begin{pmatrix} -i & i \\ 1 & 1 \end{pmatrix} \begin{pmatrix} -i\sigma & 0 \\ 0 & i\sigma \end{pmatrix} \begin{pmatrix} i/2 & 1/2 \\ -i/2 & 1/2 \end{pmatrix},$$

so

$$\begin{aligned} e^{tA} &= \begin{pmatrix} -i & i \\ 1 & 1 \end{pmatrix} \begin{pmatrix} e^{-i\sigma t} & 0 \\ 0 & e^{i\sigma t} \end{pmatrix} \begin{pmatrix} i/2 & 1/2 \\ -i/2 & 1/2 \end{pmatrix} = \begin{pmatrix} -ie^{-i\sigma t} & ie^{i\sigma t} \\ e^{-i\sigma t} & e^{i\sigma t} \end{pmatrix} \begin{pmatrix} i/2 & 1/2 \\ -i/2 & 1/2 \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2}(e^{-i\sigma t} + e^{i\sigma t}) & \frac{i}{2}(e^{i\sigma t} - e^{-i\sigma t}) \\ \frac{i}{2}(e^{-i\sigma t} - e^{i\sigma t}) & \frac{1}{2}(e^{-i\sigma t} + e^{i\sigma t}) \end{pmatrix} = \begin{pmatrix} \cos(\sigma t) & -\sin(\sigma t) \\ \sin(\sigma t) & \cos(\sigma t) \end{pmatrix}. \end{aligned}$$

Thus,

$$\begin{pmatrix} u_3(t) \\ u_4(t) \end{pmatrix} = e^{tA} \eta = \begin{pmatrix} \eta_3 \cos(\sigma t) - \eta_4 \sin(\sigma t) \\ \eta_3 \sin(\sigma t) + \eta_4 \cos(\sigma t) \end{pmatrix}.$$

Now, we define $w(t) = u_1(t) - u_2(t)$ and subtract the first two equations of our original system to find that

$$\begin{aligned} w'(t) &= u'_1(t) - u'_2(t) = (u_4(t) - \mu u_3(t) + \lambda g(u_1(t) + u_2(t))) - (\mu u_3(t) - u_4(t) + \lambda g(u_1(t) + u_2(t))) \\ &= 2(u_4(t) - \mu u_3(t)) = 2(\eta_3 \sin(\sigma t) + \eta_4 \cos(\sigma t) - \mu \eta_3 \cos(\sigma t) + \mu \eta_4 \sin(\sigma t)). \end{aligned}$$

We solve this ODE by integrating both sides to get

$$w(t) = \frac{2}{\sigma}(-\eta_3 \cos(\sigma t) + \eta_4 \sin(\sigma t) - \mu \eta_3 \sin(\sigma t) - \mu \eta_4 \cos(\sigma t)) + C.$$

For any fixed choice of $u_1(0) = \eta_1, u_2(0) = \eta_2, u_3(0) = \eta_3, u_4(0) = \eta_4$, we have the boundary condition $w(0) = \eta_1 - \eta_2$ which gives that

$$\eta_1 - \eta_2 = \frac{2}{\sigma}(-\eta_3 - \mu \eta_4) + C,$$

so

$$w(t) = \frac{2}{\sigma}(-\eta_3 \cos(\sigma t) + \eta_4 \sin(\sigma t) - \mu \eta_3 \sin(\sigma t) - \mu \eta_4 \cos(\sigma t) + \eta_3 + \mu \eta_4) + \eta_1 - \eta_2.$$

This is clearly bounded as $t \rightarrow \infty$ as by the triangle inequality

$$\begin{aligned} |w(t)| &\leq \frac{2}{|\sigma|}(|\eta_3 \cos(\sigma t)| + |\eta_4 \sin(\sigma t)| + |\mu \eta_3 \sin(\sigma t)| + |\mu \eta_4 \cos(\sigma t)| + |\eta_3| + |\mu \eta_4|) + |\eta_1| + |\eta_2| \\ &\leq \frac{2}{|\sigma|}(|\eta_3| + |\eta_4| + |\mu \eta_3| + |\mu \eta_4| + |\eta_3| + |\mu \eta_4|) + |\eta_1| + |\eta_2|, \end{aligned}$$

so $|w(t)|$ is bounded by initial conditions that are fixed (finite) and constant $\frac{2}{|\sigma|}$. Now, consider $v(t) = u_1(t) + u_2(t)$. We add the first two equations of our system to find that

$$\begin{aligned} v'(t) &= u'_1(t) + u'_2(t) = (u_4(t) - \mu u_3(t) + \lambda g(u_1(t) + u_2(t))) + (\mu u_3(t) - u_4(t) + \lambda g(u_1(t) + u_2(t))) \\ &= 2\lambda g(u_1(t) + u_2(t)) = 2\lambda v(t)(1 - v(t))^2. \end{aligned}$$

Note that $v(0) = u_1(0) + u_2(0) = \eta_1 + \eta_2$, so if $0 < \eta_1 + \eta_2 < 1$, $0 < v(0) < 1$. This implies that $v'(0) > 0$ as $\lambda > 0$, so $v(t)$ will increase initially and continue to do so until $v(t) = 1$ as $2\lambda v(t)(1 - v(t))^2 > 0$ for $\lambda \in (0, 1)$. Once $v(t) = 1$, $2\lambda v(t)(1 - v(t))^2 = 0$, so $v'(t) = 0$ for all t after $v(t) = 1$ is first achieved (This is an absorbing state). Thus, $|v(t)| \leq 1$ as $t \rightarrow \infty$ meaning that $v(t)$ is bounded. Since $v(t), w(t)$ are both bounded as $t \rightarrow \infty$, both $u_1(t)$ and $u_2(t)$ are bounded as well as

$$|u_1(t)| = |\frac{1}{2}(w(t) + v(t))| \leq \frac{1}{2}(|w(t)| + |v(t)|)$$

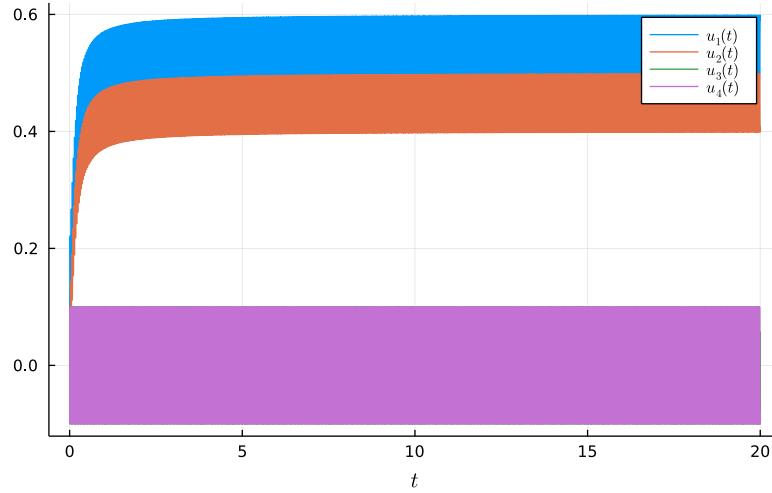
and

$$|u_2(t)| = |\frac{1}{2}(w(t) - v(t))| \leq \frac{1}{2}(|w(t)| + |v(t)|).$$

3 Problem 3

Now, we consider the system of ODEs from problem 2 and take $u_1(0) = 0.1$, $u_2(0) = 0.1$, $u_3(0) = 0$, $u_4(0) = 0.1$ and $\lambda = 10$, $\sigma = 200$, $\mu = 100$. Using Julia, we first solve this problem with trapezoid with $k = 0.0001$ which we treat as the true solution. This is displayed in the following plot which has a final time of $T = 20$.

"True" solution given by trapezoid

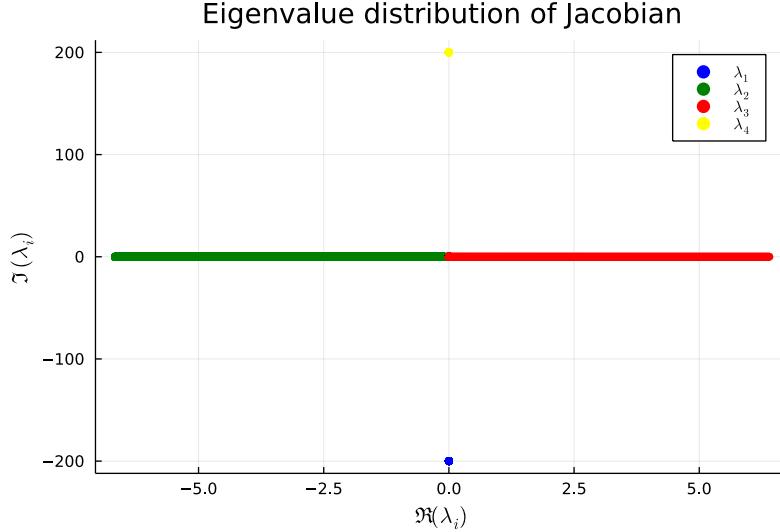


Note that the apparent shaded regions are due to heavy oscillation and is why

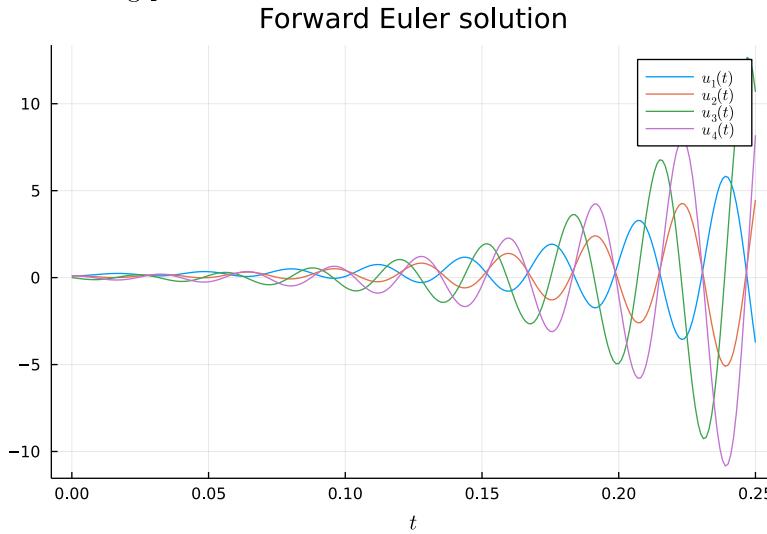
$u_4(t)$ appears superimposed over $u_3(t)$. The Jacobian of our system is given by

$$J_u = \begin{pmatrix} \lambda g'(u_1 + u_2) & \lambda g'(u_1 + u_2) & -\mu & 1 \\ \lambda g'(u_1 + u_2) & \lambda g'(u_1 + u_2) & \mu & -1 \\ 0 & 0 & 0 & -\sigma \\ 0 & 0 & \sigma & 0 \end{pmatrix}.$$

The following is a plot of the eigenvalues sorted by imaginary part of the Jacobian at each timestep of this solution

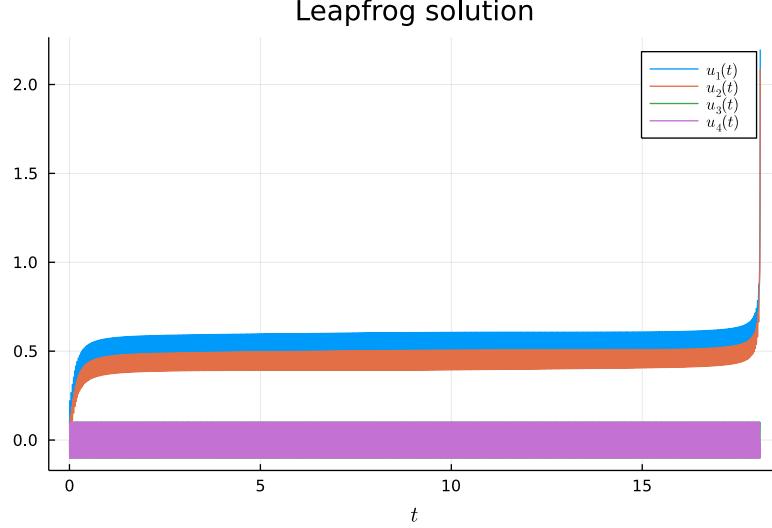


Now, we take $k = 0.001$ and solve the system with forward Euler which yields the following plot.



Notice how instability begins almost immediately and our values are an order of magnitude removed from the true values by $t = 0.25$.

Now, we again take $k = 0.001$ and solve the system with leapfrog which yields the following plot.



This time, our solution does become unstable but only around $t = 18$ which is significantly better than forward Euler.

The accuracy of these methods can be explained by how well their regions of stability capture the eigenvalues of our Jacobian. Since forward Euler's region of stability is a disk of unit radius centered at -1 , it is only able to capture the eigenvalues on the imaginary axis with a sufficiently small timestep. Leapfrog also captures the eigenvalues poorly as its stability region is entirely on the imaginary axis from $-i$ to i , so we again need to choose a sufficiently small timestep to ensure stability and the eigenvalues on the real axis are only captured when they are zero. On the other hand, trapezoid's stability region is the left half-plane including the imaginary axis, so the eigenvalues on the imaginary axis are always captured (regardless of stepsize) along with whichever real eigenvalue is nonpositive, meaning that 3 out of 4 eigenvalues are always captured. Thus, we can choose our stepsize to maximize accuracy rather than having to choose one which ensures stability.

See Appendix A for the Julia code utilized in this problem.

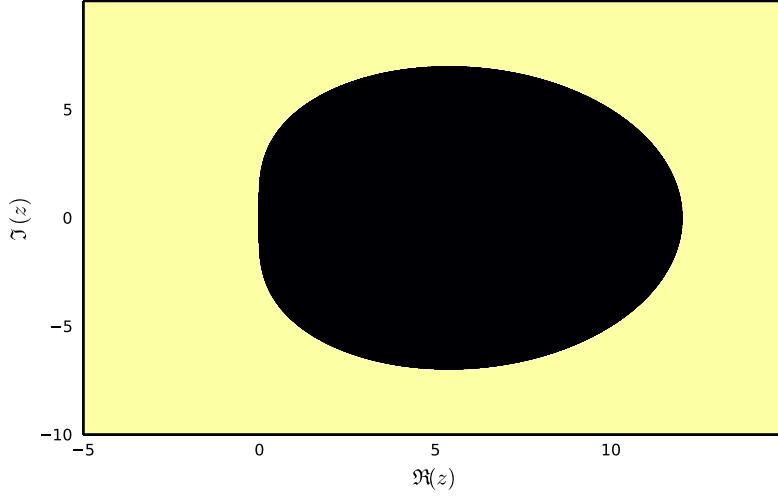
4 Problem 4

Using Julia, we plot the absolute stability region for the TR-BDF2 method (8.6) by recalling from homework 2 that

$$R(z) = \frac{12 + 5z}{(3 - z)(4 - z)}.$$

We obtain the following plot in which the yellow denotes the stability region.

Stability region for TR-BDF2



See Appendix A for the relevant code.

To show that this method is A-stable, let $z = iy$ where $y \in \mathbb{R}$. Then,

$$|R(z)|^2 = \frac{|12 + 5iy|^2}{|(3 - iy)(4 - iy)|^2} = \frac{|12 + 5iy|^2}{|(12 + y^2) - 7iy|^2} = \frac{144 + 25y^2}{144 + 24y^2 + y^4 + 49y^2}.$$

Thus, $|R(iy)| \leq 1$ iff $144 + 25y^2 \leq 144 + 73y^2 + y^4$ which holds iff $y^4 + 48y^2 = y^2(y^2 + 48) \geq 0$. This clearly holds for any $y \in \mathbb{R}$, so we must have that $|R(z)| \leq 1$ on the imaginary axis. Now, we note that R is analytic in the left halfplane (as its only singularities are at $z = 3, 4$) and that $R(z) \rightarrow 0$ as $z \rightarrow \infty$. We apply the maximum modulus principle (for rational functions) which gives that the modulus of R in the closure of the left halfplane must be attained on its boundary (the imaginary axis and infinity). Since $R(z) \rightarrow 0$ as $z \rightarrow \infty$, we must have that $|R(z)|$ attains its maximum in this region on the imaginary axis, so it must actually hold that $|R(z)| \leq 1$ for all z such that $\Re z \leq 0$. Thus, the method is A-stable.

As we have already claimed¹,

$$\lim_{z \rightarrow \infty} |R(z)| = 0,$$

so our method is also L-stable.

5 Problem 5

Let $g(x) = 0$ represent a system of s nonlinear equations in s unknowns, so $x \in \mathbb{R}^s$ and $g : \mathbb{R}^s \rightarrow \mathbb{R}^s$. A vector $\bar{x} \in \mathbb{R}^s$ is a *fixed point* of $g(x)$ if

$$\bar{x} = g(\bar{x}). \quad (1)$$

¹The denominator has a higher degree than the numerator.

One way to attempt to compute \bar{x} is with *fixed point iteration*: from some starting guess x^0 , compute

$$x^{j+1} = g(x^j) \quad (2)$$

for $j = 0, 1, \dots$

5.1 Part a

Assume that there exists a norm $\|\cdot\|$ such that $g(x)$ is Lipschitz continuous with constant $L < 1$ in a neighborhood of \bar{x} . We first need to demonstrate that a fixed point iteration that starts in this neighborhood remains in it. Namely, we assume that

$$\|x - \bar{x}\| < \epsilon$$

where $\epsilon > 0$ is the radius of the neighborhood and use the Lipschitz continuity to find that

$$\|g(x) - \bar{x}\| = \|g(x) - g(\bar{x})\| \leq L\|x - \bar{x}\| = L\epsilon < \epsilon.$$

This means that the fixed point iteration applied to a point in the neighborhood produces a point that is also in the neighborhood. Applying this property inductively yields that all iterations of the fixed point iteration are within this neighborhood if we assume that the starting value x^0 is within it. With this, we can apply the Lipschitz property at each iteration. Namely, subtracting (1) from (2),

$$\begin{aligned} \|x^{j+1} - \bar{x}\| &= \|g(x^j) - g(\bar{x})\| \leq L\|x^j - \bar{x}\| = \|g(x^{j-1}) - g(\bar{x})\| \\ &\leq L^2\|x^{j-1} - \bar{x}\| \leq \dots \leq L^{j+1}\|x^0 - \bar{x}\| \end{aligned}$$

for $j = 0, 1, \dots$. Thus,

$$\lim_{j \rightarrow \infty} \|x^j - \bar{x}\| = \lim_{j \rightarrow \infty} L^j \|x^0 - \bar{x}\| = 0$$

since $L < 1$. Thus, it must hold that

$$\lim_{j \rightarrow \infty} x^j = \bar{x},$$

so the fixed point iteration converges.

5.2 Part b

Suppose $g(x)$ is differentiable and let $D_x g(x)$ be the $s \times s$ Jacobian matrix. Consider some x^* which is within a neighborhood of \bar{x} required for Lipschitz continuity and Taylor expansion. Then,

$$g(x^*) = g(\bar{x}) + D_x g(\bar{x})(x^* - \bar{x}) + o(\|x^* - \bar{x}\|),$$

so

$$D_x g(\bar{x})(x^* - \bar{x}) = g(x^*) - g(\bar{x}) + o(\|x^* - \bar{x}\|),$$

and

$$\|D_x g(\bar{x})(x^* - \bar{x})\| \leq \|g(x^*) - g(\bar{x})\| + o(\|x^* - \bar{x}\|) \leq L\|x^* - \bar{x}\| + o(\|x^* - \bar{x}\|).$$

Then,

$$\frac{\|D_x g(\bar{x})(x^* - \bar{x})\|}{\|x^* - \bar{x}\|} \leq L + o(1).$$

Since this must hold for all x^* in a certain neighborhood, it must hold that

$$\max_{\|x^* - \bar{x}\| = \delta} \frac{\|D_x g(\bar{x})(x^* - \bar{x})\|}{\|x^* - \bar{x}\|} \leq L + o(1).$$

where $\delta > 0$ is less than the radius of our neighborhood. However, we can simply let $y = (x^* - \bar{x})/\delta$ to get that

$$\max_{\|y\| = 1} \|D_x g(\bar{x})y\| \leq L + o(1).$$

The LHS is the definition of the operator norm, so taking the limit of both sides as $\delta \rightarrow 0$, we find that

$$\|D_x g(\bar{x})\| \leq L < 1.$$

It also holds that for any operator norm,

$$\rho(A) \leq \|A\|$$

where A is a matrix and ρ is its spectral radius, so we must also have that

$$\rho(D_x g(\bar{x})) < 1.$$

5.3 Part c

Consider a predictor-corrector method consisting of forward Euler as the predictor and backward Euler as the corrector, and suppose we make N correction iterations, i.e., we set

$$\begin{aligned} \hat{U}^0 &= U^n + kf(U^n) \\ \text{for } j &= 0, 1, \dots, N-1 \\ \hat{U}^{j+1} &= U^n + kf(\hat{U}^j) \\ \text{end} \\ U^{n+1} &= \hat{U}^N. \end{aligned}$$

To find the stability region for this method, we consider the method applied for $f(U) = \lambda U$. Using induction, we prove that

$$\hat{U}^N = \left(\sum_{j=0}^{N+1} z^j \right) U^n$$

where $z = k\lambda$. As a base case where $N = 0$, we find that

$$\hat{U}^0 = U^n + kf(U^n) = U^n + k\lambda U^n = (1 + z)U^n.$$

Now, say that the formula holds for $N = \ell - 1$. Then,

$$\hat{U}^\ell = U^n + kf(\hat{U}^{\ell-1}) = U^n + k\lambda \left(\sum_{j=0}^{\ell} z^j \right) U^n = \left(1 + \sum_{j=0}^{\ell} z^{j+1} \right) U^n = \left(\sum_{j=0}^{\ell+1} z^j \right) U^n$$

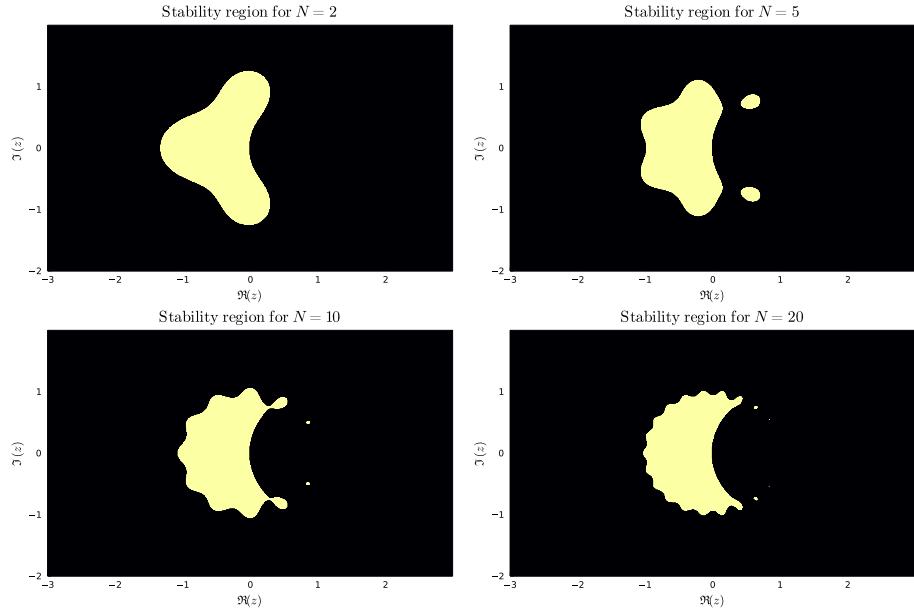
after reindexing. Thus, the formula holds for all $N \in \mathbb{N}$ by induction. Since $U^{n+1} = \hat{U}^N$,

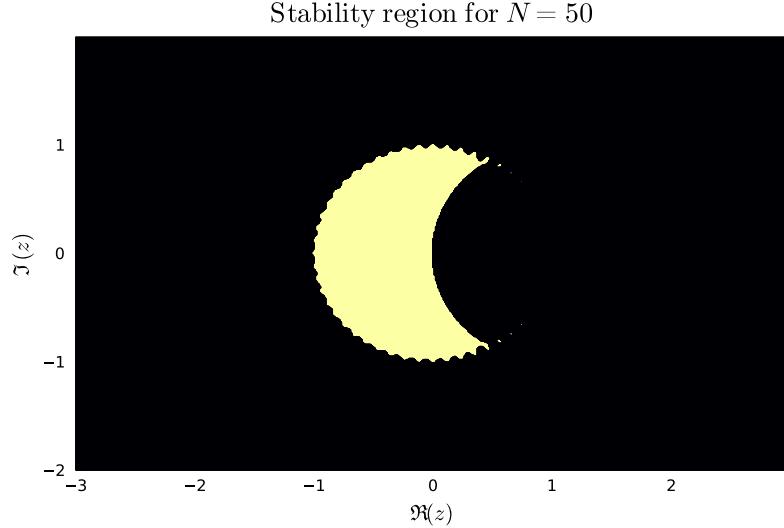
$$U^{n+1} = \left(\sum_{j=0}^{N+1} z^j \right) U^n,$$

so

$$R(z) = \sum_{j=0}^{N+1} z^j.$$

Using Julia, we plot the stability region corresponding to $|R(z)| \leq 1$ for $N = 2, 5, 10, 20, 50$ and obtain the following plots.





5.4 Part d

Consider $g(U) = U^n + kf(U)$ as a fixed point iteration². Then, its Jacobian is given by

$$D_u g(u) = kf'(u)$$

where $f'(u)$ denotes the Jacobian of f . If we let λ_{\max} denote the eigenvalue of $f'(u)$ with maximum modulus, then

$$\rho(D_u g(u)) = \rho(kf'(u)) = |k|\rho(f'(u)) = |k||\lambda_{\max}| = |k\lambda_{\max}|.$$

If for some eigenvalue λ of $f'(u)$ $|k\lambda| \geq 1$, then $|k\lambda_{\max}| \geq 1$ meaning that $\rho(D_u g(u)) \geq 1$. Then, the contrapositive of part b implies that the condition of part a cannot hold, meaning that the fixed point iteration may not hold. Thus, we can only expect convergence when $|k\lambda| < 1$.

5.5 Part e

From part d, we expect that the stability region only includes z such that $|z| < 1$. This suggests that the stability region should be the intersection of the stability region for backwards Euler with this disk which does appear to be what our region is converging to as $N \rightarrow \infty$.

²Note that this is the iteration being used in part c.

6 Problem 6

Consider the matrix $M_r = I - rT$ where T is the $m \times m$ matrix.

$$T = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}$$

and $r \geq 0$. Because T is a tridiagonal symmetric Toeplitz matrix, its eigenvalues are given by (C.22) as

$$\lambda_j = -2 + 2 \cos(j\pi h)$$

for $j = 1, \dots, m$ where $h = \frac{1}{m+1}$. Then, the eigenvalues of M_r are given by

$$1 - r\lambda_j = 1 - r(-2 + 2 \cos(j\pi h)) = (1 + 2r) - 2r \cos\left(\frac{j\pi}{m+1}\right)$$

for $j = 1, \dots, m$. Since $r \geq 0$ and $\cos\left(\frac{j\pi}{m+1}\right) \leq 1$,

$$1 - r\lambda_j \geq (1 + 2r) - 2r = 1 > 0,$$

so it must hold that all eigenvalues of M_r are positive for any $r \geq 0$. This means that M_r is invertible for any $r \geq 0$, so the largest value of c such that M_r is invertible for all $r \in [0, c)$ is $c = \infty$.

7 Appendix A

The following Julia code is used for Problem 3.

```
using Plots, Printf, LaTeXStrings, LinearAlgebra
λ, σ, μ = 10., 200., 100. #constants

import Base.isless #define complex ordering
function isless(a::ComplexF64,b::ComplexF64)
    return imag(a) < imag(b)
end

gfunc = u -> u*(1-u)^2
gprime = u -> 3u^2-4u+1 #derivative of g
f = u -> [u[4]-μ*u[3]+λ*gfunc(u[1]+u[2]),
            μ*u[3]-u[4]+λ*gfunc(u[1]+u[2]), -σ*u[4], σ*u[3]]

η = [0.1, 0.1, 0., 0.1]

k, T = 0.0001, 20.

#define Jacobian
Df = u -> [λ*gprime(u[1]+u[2]) λ*gprime(u[1]+u[2]) -μ 1.,
              λ*gprime(u[1]+u[2]) λ*gprime(u[1]+u[2]) μ 1.;
              0. 0. 0. -σ];
```

```

0. 0. σ 0.]
```

```

#Newton's method from demo
function Newton(x,g,Dg; tol = 1e-13, nmax = 100)
    for j = 1:nmax
        idk = Dg(x)
        dka = g(x)
        step = Dg(x)\g(x)
        x -= step
        if maximum(abs.(step)) < tol
            break
        end
        if j == nmax
            println("Newton's method did not terminate")
        end
    end
    x
end

#Trapezoid
g_tr = (u,Un,t,tn) -> u - Un - (k/2)*(f(u)+f(Un))
Dg_tr = u -> I - (k/2)*Df(u)
n = convert(Int64,ceil(T/k))
U = zeros(n+1,length(η)) # To save the solution values
U[1,:] = η
t = zeros(n+1)
t[1] = 0.
max_iter = 10
for i = 2:n+1
    t[i] = t[i-1] + k
    U[i,:] = Newton(U[i-1,:],u -> g_tr(u,U[i-1,:],t[i],t[i-1]), Dg_tr;
                      tol = k^3/10)
end

p1 = plot(t, U,label = [L"u_1(t)" L"u_2(t)" L"u_3(t)" L"u_4(t)"])
xlabel!(L"t")
title!("True\ solution given by trapezoid")
display(p1)
savefig(p1,"trap.pdf")

λs = map( t -> eigvals(Df(U[convert(Int64,round(t/k+1)),:])), t)
λs = [ sort(i) for i in λs]
λ1 = [i[1] for i in λs]
λ2 = [i[2] for i in λs]
λ3 = [i[3] for i in λs]
λ4 = [i[4] for i in λs]

p2 = scatter(λ1 |> real, λ1 |> imag, markercolor = :blue, markerstrokewidth=0,
            label = L"\lambda_1")
scatter!(λ2 |> real, λ2 |> imag, markercolor = :green, markerstrokewidth=0,
         label = L"\lambda_2")
scatter!(λ3 |> real, λ3 |> imag, markercolor = :red, markerstrokewidth=0,
         label = L"\lambda_3")
scatter!(λ4 |> real, λ4 |> imag, markercolor = :yellow, markerstrokewidth=0,
         label = L"\lambda_4")
xlabel!(L"\Re(\lambda_i)")
ylabel!(L"\Im(\lambda_i)")
title!("Eigenvalue distribution of Jacobian")
display(p2)
savefig(p2,"eigvals.pdf")

#Forward Euler
k, T = 0.001, 0.25
n = convert(Int64,ceil(T/k))
U_fe = zeros(n+1,length(η)) # To save the solution values
U_fe[1,:] = η
t = zeros(n+1) # To save times
t[1] = 0.

```

```

for i = 2:n+1
    U_fe[i,:] = U_fe[i-1,:] + k*f(U_fe[i-1,:])
    t[i] = t[i-1] + k
end
p3 = plot(t, U_fe,label = [L"u_1(t)" L"u_2(t)" L"u_3(t)" L"u_4(t)"])
xlabel!(L"t")
title!("Forward Euler solution")
display(p3)
savefig(p3,"fe.pdf")

#Leapfrogf
k, T = 0.001, 18.09
n = convert(Int64,ceil(T/k))
U_lf=zeros(n+1,length(η)) # To save the solution values
U_lf[1,:] = η
t = zeros(n+1) # To save times
t[1] = 0.
#step of Forward Euler
U_lf[2,:] = U_lf[1,:] + k*f(U_lf[1,:])
t[2] = k
for i = 3:n+1
    U_lf[i,:] = U_lf[i-2,:] + (2*k)*f(U_lf[i-1,:]) #Leapfrog
    t[i] = t[i-1] + k
end
p4 = plot(t, U_lf,label = [L"u_1(t)" L"u_2(t)" L"u_3(t)" L"u_4(t)"])
xlabel!(L"t")
title!("Leapfrog solution")
display(p4)
savefig(p4,"lf.pdf")

```

The following Julia code is used for Problem 4.

```

using Plots, LaTeXStrings

R = z -> (12+5z)/((4-z)*(3-z))
xrange = [-5,15]; yrange = [-10,10]
p = contourf(xrange[1]:0.01(1+rand())/10:xrange[2],
              yrange[1]:0.01(1+rand())/10:yrange[2], (x,y)-> sign(-(abs(R(x+1im*y)))+1),
              colorbar=false)
xlabel!(L"\Re(z)")
ylabel!(L"\Im(z)")
title!("Stability region for TR-BDF2")
display(p)
savefig(p,"trbdf.pdf")

```

The following Julia code is used for Problem 5.

```

using Plots, LaTeXStrings

#build stability polynomial
function R_N(z,N)
    eval = 0
    for j = 0:N+1
        eval += z^j
    end
    eval
end

xrange = [-3,3]; yrange = [-2,2]
for N = [2 5 10 20 50]
    p = contourf(xrange[1]:0.01(1+rand())/10:xrange[2],
                  yrange[1]:0.01(1+rand())/10:yrange[2],

```

```

(x,y)-> sign(-(abs(R_N(x+1im*y,N))+1), colorbar=false)
xlabel!(L"\text{Re}(z)")
ylabel!(L"\text{Im}(z)")
title!(texstring(@sprintf("\mathrm{Stability~region~for}~N=%i", N)))
display(p)
savefig(p, @sprintf("N%i.pdf", N))
end

```