

AMATH 586 Homework 1

Cade Ballew #2120804

April 8, 2022

1 Problem 1

1.1 Part a

Using the Taylor series representation of the matrix exponential and assuming that we can differentiate termwise,

$$\frac{d}{dt}e^{tA} = \frac{d}{dt} \sum_{n=0}^{\infty} \frac{(tA)^n}{n!} = \sum_{n=0}^{\infty} A^n \frac{d}{dt} \frac{t^n}{n!} = \sum_{n=1}^{\infty} A^n \frac{t^{n-1}}{(n-1)!}$$

because the derivative of the $n = 0$ term is zero. Now, we reindex $n \rightarrow n - 1$ to get

$$\frac{d}{dt}e^{tA} = \sum_{n=0}^{\infty} A^{n+1} \frac{t^n}{n!} = A \sum_{n=0}^{\infty} \frac{(tA)^n}{n!} = Ae^{tA}.$$

Additionally,

$$\frac{d}{dt}e^{tA} = \sum_{n=0}^{\infty} A^{n+1} \frac{t^n}{n!} = \left(\sum_{n=0}^{\infty} \frac{(tA)^n}{n!} \right) A = e^{tA} A.$$

1.2 Part b

Now, consider $u(t) = e^{tA}\eta$ where η is constant with respect to t . By part a,

$$u'(t) = Ae^{tA}\eta = Au(t).$$

Additionally, we have from the lecture notes that $e^{0A} = I$, so

$$u(0) = e^{0A}\eta = I\eta = \eta.$$

Thus, $u(t)$ satisfies the IVP

$$\begin{cases} u'(t) = Au(t), \\ u(0) = \eta. \end{cases}$$

2 Problem 2

Consider the system

$$\begin{cases} u_1'(t) = u_1(t), \\ u_2'(t) = u_2(t). \end{cases}$$

Namely, we take $f(u(t)) = Iu(t) = u(t)$. Now, consider

$$u_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$v_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

and solve the systems

$$\begin{cases} u'(t) = f(u(t)), \\ u(0) = u_0, \end{cases} \quad \begin{cases} v'(t) = f(v(t)), \\ v(0) = v_0. \end{cases}$$

Then, because the identity matrix is diagonal, we can use (D.30) to compute

$$e^{tI} = \begin{pmatrix} e^t & 0 \\ 0 & e^t \end{pmatrix},$$

so problem 1 gives that

$$u(t) = e^{tI}u_0 = \begin{pmatrix} e^t & 0 \\ 0 & e^t \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ e^t \end{pmatrix},$$

and

$$v(t) = e^{tI}v_0 = \begin{pmatrix} e^t & 0 \\ 0 & e^t \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} e^t \\ 0 \end{pmatrix}.$$

Then, using the homogeneity of norms and the positivity of e^t ,

$$\|u(t) - v(t)\|_2 = \left\| \begin{pmatrix} 0 \\ e^t \end{pmatrix} - \begin{pmatrix} e^t \\ 0 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} -e^t \\ e^t \end{pmatrix} \right\|_2 = e^t \left\| \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\|_2 = e^t \|u_0 - v_0\|_2,$$

so we simply take $L = 1$ and

$$\|u(t) - v(t)\|_2 = \|u(0) - v(0)\|_2 e^{Lt}$$

holds. Note that $L = 1$ is a Lipschitz constant for f as

$$\|f(u) - f(v)\| = \|u - v\| = L\|u - v\|$$

for $L = 1$. However, if we instead consider

$$\begin{cases} u'(t) = -f(u(t)), \\ u(0) = u_0, \end{cases} \quad \begin{cases} v'(t) = -f(v(t)), \\ v(0) = v_0. \end{cases}$$

we now have

$$u(t) = e^{t(-I)}u_0 = \begin{pmatrix} e^{-t} & 0 \\ 0 & e^{-t} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ e^{-t} \end{pmatrix},$$

and

$$v(t) = e^{t(-I)}v_0 = \begin{pmatrix} e^{-t} & 0 \\ 0 & e^{-t} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} e^{-t} \\ 0 \end{pmatrix},$$

Then, we find that

$$\|u(t) - v(t)\|_2 = \left\| \begin{pmatrix} -e^{-t} \\ e^{-t} \end{pmatrix} \right\|_2 = e^{-t} \left\| \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\|_2 = e^{-t} \|u(0) - v(0)\|_2,$$

Thus, in order for

$$e^{-t} \|u(0) - v(0)\|_2 = \|u(t) - v(t)\|_2 = \|u(0) - v(0)\|_2 e^{Lt}$$

to hold, we need that $e^{(L+1)t} = 1$. However, in order for this to be true for $t \neq 0$, we need that $L = -1$ which implies that it is not a Lipschitz constant as a Lipschitz constant must be positive by definition. This means that this relation cannot hold no matter what the Lipschitz constant of $-f$ is.

3 Problem 3

Consider the IVP

$$\begin{cases} u_1'(t) = 2u_1(t), \\ u_2'(t) = 3u_1(t) - u_2(t), \end{cases}$$

with initial conditions specified at time $t = 0$. Denote these by η_1 and η_2 , respectively.

3.1 Part a

We first find that

$$\frac{u_1'(t)}{u_1(t)} = 2,$$

so a general solution is given by $u_1(x) = ce^{2t}$ which becomes

$$u_1(t) = \eta_1 e^{2t}$$

when we implement our boundary condition. Inserting this into our other equation,

$$u_2'(t) = 3\eta_1 e^{2t} - u_2(t).$$

Using (5.8),

$$u_2(t) = e^{-t}\eta_2 + \int_0^t e^{\tau-t} 3\eta_1 e^{2\tau} d\tau = e^{-t}\eta_2 + \eta_1 [e^{3\tau-t}]_0^t = \eta_1 (e^{2t} - e^{-t}) + \eta_2 e^{-t}.$$

3.2 Part b

Now, consider

$$u(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix},$$

$$A = \begin{pmatrix} 2 & 0 \\ 3 & -1 \end{pmatrix},$$

and

$$\eta = \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}.$$

Then, our IVP can be rewritten as

$$\begin{cases} u'(t) = Au(t), \\ u(0) = \eta, \end{cases}$$

so we can apply problem 1. To compute the matrix exponential using (D.30), note that¹

$$A = \begin{pmatrix} 2 & 0 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$$

and that

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Then,

$$e^{tA} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} e^{2t} & 0 \\ 0 & e^{-t} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} e^{2t} & 0 \\ e^{2t} - e^{-t} & e^{-t} \end{pmatrix}.$$

Then, problem 1 gives that

$$u(t) = e^{tA}\eta = \begin{pmatrix} e^{2t} & 0 \\ e^{2t} - e^{-t} & e^{-t} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = \begin{pmatrix} \eta_1 e^{2t} \\ \eta_1(e^{2t} - e^{-t}) + \eta_2 e^{-t} \end{pmatrix},$$

so as in part a,

$$u_1(t) = \eta_1 e^{2t},$$

and

$$u_2(t) = \eta_1(e^{2t} - e^{-t}) + \eta_2 e^{-t}.$$

¹Obtained with the help of Mathematica

4 Problem 4

Now, consider the IVP

$$\begin{cases} u_1'(t) = 2u_1(t), \\ u_2'(t) = 3u_1 + 2u_2(t) \end{cases}$$

with initial conditions specified at time $t = 0$. We solve this using the method of problem 3 part a by again denoting these by $u_1(0) = \eta_1$ and $u_2(0) = \eta_2$ and noting that we have the same conditions on u_1 as before, so

$$u_1(t) = \eta_1 e^{2t}.$$

Using (5.8),

$$u_2(t) = e^{2t}\eta_2 + \int_0^t e^{2(t-\tau)} 3\eta_1 e^{2\tau} d\tau = e^{2t}\eta_2 + 3\eta_1 e^{2t} [\tau]_0^t = 3\eta_1 t e^{2t} + \eta_2 e^{2t}.$$

5 Problem 5

Consider the Lotka–Volterra system

$$\begin{cases} u_1'(t) = \alpha u_1(t) - \beta u_1(t)u_2(t), \\ u_2'(t) = \delta u_1(t)u_2(t) - \gamma u_2(t) \end{cases}$$

for $\alpha = \delta = \gamma = \beta = 1$ and $u_1(0) = 5, u_2(0) = 0.8$. Using Julia, we approximate the solution with $k = 0.001$ for $t = 0, 0.001, \dots, 50$ using both forward and backward Euler timesteps. We apply forward Euler by defining

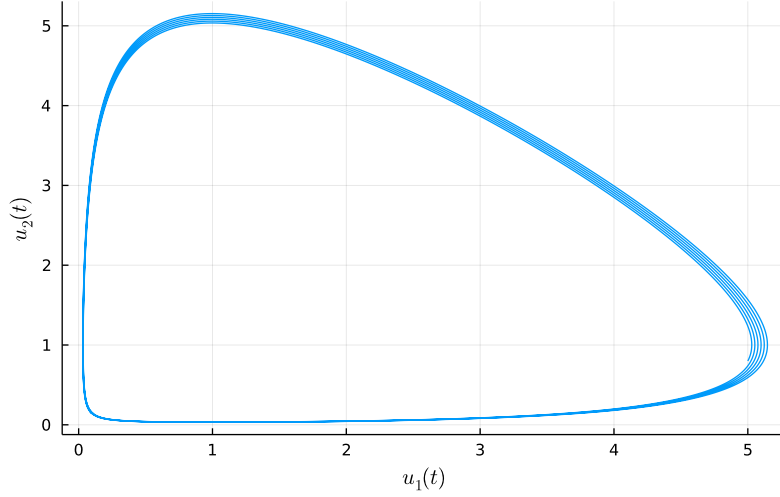
$$f(u) = \begin{pmatrix} \alpha u_1(t) - \beta u_1(t)u_2(t) \\ \delta u_1(t)u_2(t) - \gamma u_2(t) \end{pmatrix}$$

and taking the iteration

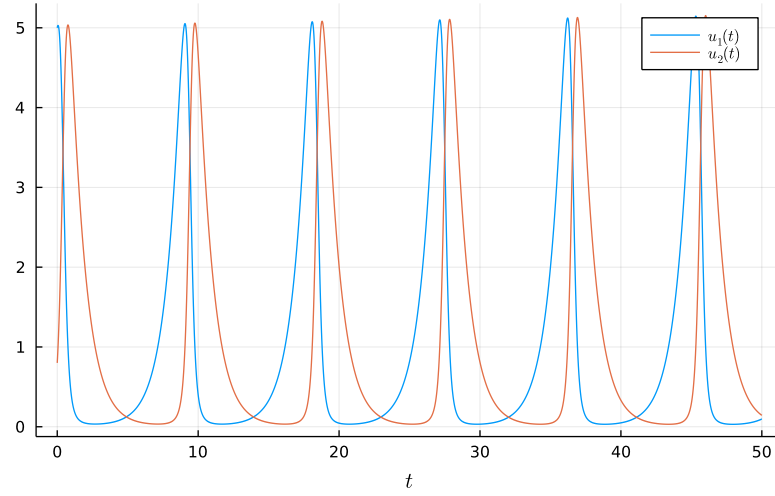
$$u^{n+1} = u^n + kf(u^n)$$

at each timestep. This produces the following solution plots.

Forward Euler with $k = 0.001$



Forward Euler with $k = 0.001$



Applying backward Euler requires the iteration

$$u^{n+1} = u^n + kf(u^{n+1})$$

which requires us to define

$$g(U) = u - u^n - kf(u)$$

which has Jacobian

$$D_u g(u) = \begin{pmatrix} 1 + k(\alpha - \beta u_2(t)) & -k\beta u_1(t) \\ k\delta u_2(t) & 1 + k(\delta u_1(t) - \gamma) \end{pmatrix}.$$

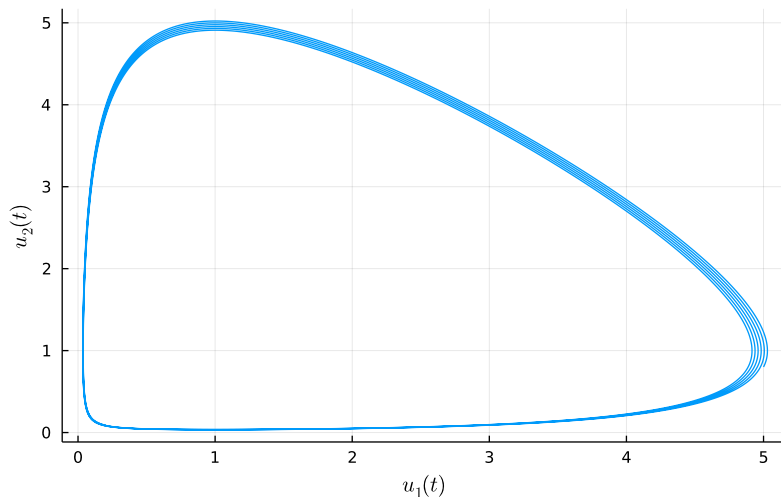
This allows us to approximate the solution to $g(u) = 0$ by taking Newton iterations

$$u_0^{n+1} = u^n$$

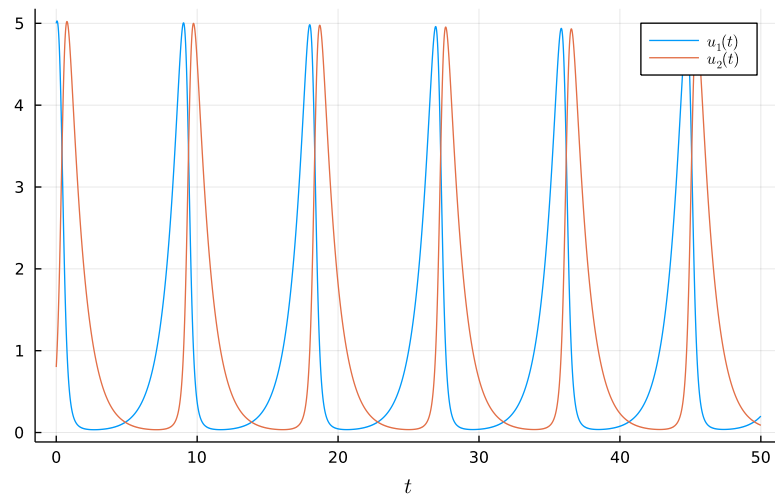
$$u_{k+1}^{n+1} = u_k^{n+1} - [D_u g(u_k^{n+1})]^{-1} g(u_k^{n+1}), \quad k = 0, 1, 2, \dots$$

Doing this yields the following solution plots.

Backward Euler with $k = 0.001$



Backward Euler with $k = 0.001$



The forward and backward plots look quite similar, but there is a subtle difference that one can observe in the (u_1, u_2) -plane; as time increases, the forward Euler solution appears to be spiraling outward while the backward Euler solution appears to be spiraling inward. What this suggests is that forward Euler is amplifying our solution while backward Euler is dampening it. This is consistent with the behavior we observed in class and suggests that neither method

will be accurate in long time intervals.
See Appendix A for the Julia code that produces these plots.

6 Problem 6

6.1 Part a

From (5.44), a 2-step LMM has form

$$\alpha_0 u^n + \alpha_1 u^{n+1} + \alpha_2 u^{n+2} = k (\beta_0 f(u^n) + \beta_1 f(u^{n+1}) + \beta_2 f(u^{n+2})).$$

From (5.45), we know that the 2-step Adams-Moulton method takes $\alpha_0 = 0$, $\alpha_1 = -1$, $\alpha_2 = 1$. Noting that these sum to zero, we can then compute the LTE using the formula on page 133 of the text.

$$\begin{aligned} \tau_{n+2} &= \left(\sum_{j=0}^2 (j\alpha_j - \beta_j) \right) u'(t_n) + k \left(\sum_{j=0}^2 \left(\frac{1}{2} j^2 \alpha_j - j\beta_j \right) \right) u''(t_n) \\ &\quad + k^2 \left(\sum_{j=0}^2 \left(\frac{1}{6} j^3 \alpha_j - \frac{1}{2} j^2 \beta_j \right) \right) u'''(t_n) + k^3 \left(\sum_{j=0}^2 \left(\frac{1}{24} j^4 \alpha_j - \frac{1}{6} j^3 \beta_j \right) \right) u^{(4)}(t_n) + O(k^4) \\ &= (-1 + 2 - \beta_0 - \beta_1 - \beta_2) u'(t_n) + k \left(\frac{-1}{2} + 2 - \beta_1 - 2\beta_2 \right) u''(t_n) \\ &\quad + k^2 \left(\frac{-1}{6} + \frac{4}{3} - \frac{1}{2} \beta_1 - 2\beta_2 \right) u'''(t_n) + k^3 \left(\frac{-1}{24} + \frac{2}{3} - \frac{1}{6} \beta_1 - \frac{4}{3} \beta_2 \right) u^{(4)}(t_n) + O(k^4). \end{aligned}$$

In order to achieve third order accuracy, we need the first three terms to each be zero. We can impose this condition as the following linear system.

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & \frac{1}{2} & 2 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3/2 \\ 7/6 \end{pmatrix}.$$

Solving this² yields solution

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} -1/12 \\ 2/3 \\ 5/12 \end{pmatrix}.$$

One can verify that this is not higher order by noticing that the k^3 term in the above LTE does not vanish in general for these values of β_j . These values of α_j and β_j yield the scheme

$$u^{n+2} = u^{n+1} + k \left(-\frac{1}{12} f(u^n) + \frac{2}{3} f(u^{n+1}) + \frac{5}{12} f(u^{n+2}) \right)$$

which matches the two-step implicit Adams-Moulton method found on page 132 of the text.

²Again with the help of Mathematica

6.2 Part b

Now, consider the three times $t_n = -k$, $t_{n+1} = 0$, and $t_{n+2} = k$ and polynomial

$$p(t) = A + B(t + k) + C(t + k)t$$

that we wish to interpolate through the three values $f(U^n)$, $f(U^{n+1})$ and $f(U^{n+2})$ at these points. Then, the relation

$$u(t_{n+2}) = u(t_{n+1}) + \int_{t_{n+1}}^{t_{n+2}} f(u(s)) ds.$$

can be applied to derive the approximation

$$u^{n+2} = u^{n+1} + \int_0^k p(t) dt.$$

We first determine the coefficients of $p(t)$ by setting

$$f(u^n) = p(-k) = A,$$

$$f(u^{n+1}) = p(0) = A + kB,$$

and

$$f(u^{n+2}) = p(k) = A + kB + 2k^2C.$$

Solving these sequentially yields

$$\begin{aligned} A &= f(u^n), \\ B &= \frac{f(u^{n+1}) - f(u^n)}{k}, \\ C &= \frac{f(u^{n+2}) - 2f(u^{n+1}) + f(u^n)}{2k^2}. \end{aligned}$$

Then,

$$\begin{aligned} \int_0^k p(t) dt &= \left[At + \frac{Bt^2}{2} + Btk + \frac{Ct^3}{3} + \frac{Ct^2k}{2} \right]_0^k = Ak + \frac{3B}{2}k^2 + \frac{5C}{6}k^3 \\ &= kf(u^n) + \frac{3k}{2}(f(u^{n+1}) - f(u^n)) + \frac{5k}{12}(f(u^{n+2}) - 2f(u^{n+1}) + f(u^n)) \\ &= k \left(\left(1 - \frac{3}{2} + \frac{5}{12} \right) f(u^n) + \left(\frac{3}{2} - \frac{5}{6} \right) f(u^{n+1}) + \frac{5}{12} f(u^{n+2}) \right) \\ &= k \left(-\frac{1}{12} f(u^n) + \frac{2}{3} f(u^{n+1}) + \frac{5}{12} f(u^{n+2}) \right). \end{aligned}$$

Thus, our approximation is given by

$$u^{n+2} = u^{n+1} + k \left(-\frac{1}{12} f(u^n) + \frac{2}{3} f(u^{n+1}) + \frac{5}{12} f(u^{n+2}) \right)$$

which is the same as what we derived before.

7 Appendix A

The following is displayed from Problem5.jl which is also in the Github repository.

```
using Plots, LaTeXStrings, Printf

 $\alpha, \beta, \gamma, \delta$  = 1.0, 1.0, 1.0, 1.0 #parameters
T = 50. #final time
k = 0.001 #step size

f = u-> [ $\alpha u[1] - \beta u[1] u[2]$ ,  $\delta u[1] u[2] - \gamma u[2]$ ] #function f such that  $u'(t) = f(u)$ 
u0 = [5., 0.8] #initial condition

# Forward Euler
n = convert{Int64, T/k} # Number of time steps, converted to Int64
Uf = zeros(2, n+1) # To save the solution values
Uf[:, 1] = u0
for i = 2:n+1
    Uf[:, i] = Uf[:, i-1] + k*f(Uf[:, i-1])
end
tvec = k*(0:n) #time vector

p1 = (plot(Uf[1, :], Uf[2, :], title = latexstring(@sprintf(
    "\mathrm{Forward-Euler-with}~k = %1.3f", k)), legend=false))
xlabel!(latexstring("u_1(t)"))
ylabel!(latexstring("u_2(t)"))
savefig(p1, "fe1.pdf")
display(p1)
p2 = plot(tvec, Uf[1, :], title = latexstring(@sprintf(
    "\mathrm{Forward-Euler-with}~k = %1.3f", k)), label=latexstring("u_1(t)"))
plot!(tvec, Uf[2, :], label=latexstring("u_2(t)"))
xlabel!(latexstring("t"))
savefig(p2, "fe2.pdf")
display(p2)

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

g = (U, Un) -> U - Un - k*f(U)
Dg = (U) -> [1.0+k*( $\alpha - \beta U[2]$ ) -k*\beta*U[1];
             k*\delta*U[2] 1.0+k*( $\delta * U[1] - \gamma$ )]

# Backward Euler
n = convert{Int64, T/k} # Number of time steps, converted to Int64
Ub = zeros(2, n+1) # To save the solution values
Ub[:, 1] = u0
for i = 2:n+1
    Unew = Ub[:, i-1] |> copy
    Newton!(Unew, u -> g(u, Ub[:, i-1]), Dg)
    Ub[:, i] = Unew
end
```

```

p3 = plot(Ub[1,:],Ub[2,:], title = latexstring(@sprintf(
    "\mathrm{Backward-Euler-with}~k = %1.3f",k)),legend=false)
xlabel!(latexstring("u_1(t)"))
ylabel!(latexstring("u_2(t)"))
savefig(p3, "be1.pdf")
display(p3)
p4 = plot(tvec,Ub[1,:], title = latexstring(@sprintf(
    "\mathrm{Backward-Euler-with}~k = %1.3f",k)),label=latexstring("u_1(t)"))
plot!(tvec,Ub[2:],label=latexstring("u_2(t)"))
xlabel!(latexstring("t"))
savefig(p4, "be2.pdf")
display(p4)

```